MOBILE AD-HOC NETWORKS: PROTOCOL DESIGN

Edited by **Xin Wang**

INTECHWEB.ORG

Mobile Ad-Hoc Networks: Protocol Design
Edited by Xin Wang

Published by InTech
Janeza Trdine 9, 51000 Rijeka, Croatia

Statements and opinions expressed in the chapters are these of the individual contributors and not necessarily those of the editors or publisher. No responsibility is accepted for the accuracy of information contained in the published articles. The publisher assumes no responsibility for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained in the book.

Publishing Process Manager Iva Lipovic
Technical Editor Teodora Smiljanic
Cover Designer Martina Sirotic
Image Copyright Andreas Guskos, 2010. Used under license from Shutterstock.com

First published January, 2011
Printed in India

A free online edition of this book is available at www.intechopen.com
Additional hard copies can be obtained from orders@intechweb.org

Mobile Ad-Hoc Networks: Protocol Design, Edited by Xin Wang
p. cm.
ISBN 978-953-307-402-3

Contents

Preface IX

Part 1 Quality of Service and Video Commucation in Ad Hoc Networks 1

Chapter 1 QoE Enhancement of Audio-Video IP Transmission in Cross-Layer Designed Ad Hoc Networks 3
Toshiro Nunome and Shuji Tasaka

Chapter 2 Quality of Service (QoS) Provisioning in Mobile Ad-Hoc Networks (MANETs) 23
Masoumeh Karimi

Chapter 3 Video Communications Over Wireless Ad-Hoc Networks Using Source Coding Diversity and Multiple Paths 39
Yiting Liao and Jerry D. Gibson

Chapter 4 Available Bandwidth Estimation and Prediction in Ad hoc Networks 61
Haitao Zhao, Jibo Wei, Shan Wang and Yong Xi

Chapter 5 Mathematic Models for Quality of Service Purposes in Ad Hoc Networks 85
Khalil Amine

Chapter 6 Towards Reliable Mobile Ad Hoc Networks 99
Ricardo Lent and Javier Barria

Chapter 7 ADHOCTCP: Improving TCP Performance in Ad Hoc Networks 121
Seyed Mohsen Mirhosseini and Fatemeh Torgheh

Part 2 Cross Layer Design in Ad Hoc Networks 139

Chapter 8 Cross–Layer Design in Wireless Ad Hoc Networks with Multiple Antennas 141
Ehsan Soleimani-Nasab, Mehrdad Ardebilipour and Mahdi Kashiha

Chapter 9 **Performance Modeling of MAC and Multipath Routing Interactions in Multi-hop Wireless Networks 167**
Xin Wang, J.J. Garcia-Luna-Aceves, Hamid R. Sadjadpour

Chapter 10 **A Bandwidth Reservation QoS Routing Protocol for Mobile Ad Hoc Networks 185**
Wen-Hwa Liao

Chapter 11 **Link Quality Aware Robust Routing for Mobile Multihop Ad Hoc Networks 201**
Sangman Moh, Moonsoo Kang, and Ilyong Chung

Chapter 12 **A Location Prediction Based Routing Protocol and its Extensions for Multicast and Multi-path Routing in Mobile Ad hoc Networks 217**
Natarajan Meghanathan

Chapter 13 **An Adaptive Broadcasting Scheme in Mobile Ad Hoc Networks 243**
Dimitrios Liarokapis and Ali Shahrabi

Chapter 14 **Predictive RSS with Fuzzy Logic based Vertical Handoff Decision Scheme for Seamless Ubiquitous Access 261**
Sunisa Kunarak and Raungrong Suleesathira

Chapter 15 **Energy Issues and Energy aware Routing in Wireless Ad-hoc Networks 281**
Marco Fotino and Floriano De Rango

Part 3 Routing in Ad Hoc Networks 297

Chapter 16 **Routing in Mobile Ad Hoc Networks 299**
Fenglien Lee

Chapter 17 **Fault-Tolerant Routing in Mobile *Ad Hoc* Networks 323**
B. John Oommen and Luis Rueda

Chapter 18 **LLD: Loop-free Link Metrics for Proactive Link-State Routing in Wireless Ad Hoc Networks 345**
Takuya Yoshihiro

Chapter 19 **Stability Oriented Routing in Mobile Ad-Hoc Networks Based on Simple Automatons 363**
Miklós Molnár and Raymond Marie

Chapter 20 **Capacity, Bandwidth, and Available Bandwidth in Wireless Ad Hoc Networks: Definitions and Estimations 391**
Marco A. Alzate, Néstor M. Peña and Miguel A. Labrador

Chapter 21 **QoS Routing Solutions for Mobile Ad Hoc Network 417**
Jiwa Abdullah

Chapter 22 **A Novel Secure Routing Protocol for MANETs 455**
Zhongwei Zhang

Part 4 Other Topics 467

Chapter 23 **Security and Dynamic Encryption System in Mobile Ad-Hoc Network 469**
Peter H. Yu and Udo W. Pooch

Chapter 24 **Security of Access in Hostile Environments Based on the History of Nodes in Ad Hoc Networks 491**
Saud Rugeish Alotaibi

Chapter 25 **Trust Establishment in Mobile Ad Hoc Networks: Direct Trust Distribution-Performance and Simulation 513**
Dawoud D.S., Richard L. Gordon,
Ashraph Suliman and Kasmir Raja S.V.

Chapter 26 **Data Delivery in Delay Tolerant Networks: A Survey 565**
Shyam Kapadia, Bhaskar Krishnamachari and Lin Zhang

Chapter 27 **Broadcasting in Moblie Ad Hoc Networks 579**
Sangwoo Lee and Chaewoo Lee

Chapter 28 **Energy Efficient Resource Allocation in Cognitive Radio Wireless Ad Hoc Networks 595**
Song Gao, Lijun Qian, and D.R. Vaman

Chapter 29 **Theory and Applications of Ad Hoc Networks 615**
Takuo Nakashima

Chapter 30 **The Dimensioning of Non-Token-Bucket Parameters for Efficient and Reliable QoS Routing Decisions in Bluetooth Ad Hoc Network 639**
Halabi Hasbullah and Mahamod Ismail

Preface

Mobile Ad hoc Networks (MANETs) are a fundamental element of pervasive networks, where user can communicate anywhere, any time and on-the-fly. MANETs introduce a new communication paradigm, which does not require a fixed infrastructure - they rely on wireless terminals for routing and transport services. This edited volume covers the most advanced research and development in MANET. It seeks to provide an opportunity for readers to explore the emerging fields about MANET.

It includes four parts in total. Part 1 discusses the quality of service and video communication in MANET. Part 2 introduces some novel approaches in cross-layer protocol design. Part 3 focuses on the routing protocols. Some interesting topics about security, power consumption, capacity, etc. are discussed in Part 4.

Acknowledgements
The editors are particularly grateful to the authors who present their work in this book. They would also like to express their sincere thanks to all the reviewers, who helped to maintain the high quality of this book. We hope that the readers will share our excitement to present this volume on ad-hoc networks and will find it useful.

Prof. Xin Wang
University of California, Santa Cruz,
USA

Part 1

Quality of Service and Video Commucation in Ad Hoc Networks

QoE Enhancement of Audio-Video IP Transmission in Cross-Layer Designed Ad Hoc Networks

Toshiro Nunome and Shuji Tasaka
Nagoya Institute of Technology
Japan

1. Introduction

The *QoE (Quality of Experience)*, which is perceptual quality for the users, is the most important *QoS (Quality of Service)* among those at all levels since the users are the ultimate recipients of the services. Even in *mobile ad hoc networks (MANET)*, provision of high QoE is one of the most important issues.

Some applications of ad hoc networks require the ability to support real-time multimedia streaming such as live audio and video over the networks. Therefore, the realization of this type of service with high quality is highly demanded; nevertheless, it is very difficult to achieve high quality in ad hoc networks.

The cross-layer design architecture (Srivastava & Motani, 2005) is expected as an approach to high quality communication in ad hoc networks. The architecture exploits interaction among more than two layers. Although the layered architecture in IP-based networks has some advantages such as reduction of network design complexity, it is not well suited to wireless networks. This is because the nature of the wireless medium makes it difficult to decouple the layers.

There are many studies on the cross-layer design architecture for multimedia streaming. The number of hops maintained by the routing protocol is used for selecting the video coding rate to the network capacity (Gharavi & Ban, 2004), (Zhao et al., 2006). If there are many hops from the sender to the receiver, the approach reduces the coding rate at the sender. It is a cross-layer design between the network and application layers. Abd El Al et al. (2006) have proposed an error recovery mechanism for real-time video streaming that combines FEC and multipath retransmission. This scheme determines strength of the error correction code and a quantization parameter for video encoding according to the number of hops. Frias et al. (2005) exploit the multipath routing protocol for scheduling prioritized video streams and best effort traffic. They schedule the traffic on the basis of the number of multiple routes. Nunome & Tasaka (2005) have proposed the *MultiPath streaming scheme with Media Synchronization control (MPMS)*. It treats audio and video as two separate transport streams and sends the two streams to different routes if multipath routes are available. Furthermore, in order to remedy the temporal structure of the media streams disturbed by the multipath transmission, *media synchronization control* is employed; it is application-level QoS control.

While the above approaches refer to cross-layering between the network and application layers, Setton et al. (2005) have explored a new framework for cross-layer design that incorporates adaptation across all layers of the protocol stack: application, transport protocols, resource allocation, and link layer techniques. It should be noted that all of the previous studies mentioned above do not evaluate the QoE of transmitted multimedia streams. Furthermore, these studies except for (Nunome & Tasaka, 2005) consider video only and do not assess its temporal quality.

The routing protocol is an essential component in ad hoc networks. The *link quality-based routing* is one of the most promising approaches to establishment of routes with high quality and high throughput. It has been studied as *QoS routing* (Zhang & Mouftah, 2005) and *multirate aware routing* (Lin et al., 2003), (Seok et al., 2003). It can avoid using links with low data rates by taking account of link quality such as signal strength and link utilization level for route selection; this implies a cross-layer design among the network and lower layers.

The aim of this chapter is to achieve high QoE of audio and video streams transmitted over ad hoc networks. The cross-layer design with media synchronization control and the link quality-based routing can be one of the most effective solutions for this purpose.

In this chapter, we assess application-level QoS and QoE of audio-video streaming with media synchronization control and link quality-based routing protocols in a wireless ad hoc network. We adopt three link quality-based routing protocols: *OLSR-SS (Signal Strength)* (Itaya et al., 2005), *AODV-SS* (Budke et al., 2006), and *LQHR (Link Quality-Based Hybrid Routing)* (Nakaoka et al., 2006). OLSR-SS is a modified version of OLSR (Clausen & Jacquet, 2003), which is a proactive routing protocol. AODV-SS is a reactive protocol based on AODV (Perkins et al., 2003). LQHR is a hybrid protocol, which is a combination of proactive and reactive routing protocols. We clarify advantages and disadvantages of the three types in audio-video streaming with media synchronization control.

The quality of the audio-video stream can fluctuate largely in ad hoc networks, and then it is difficult to assess the QoE. That is, the assessment method is one of the important research issues. We employ a continuous time assessment method of QoE in audio-video transmission proposed in (Ito et al., 2005); it utilizes the *method of successive categories* (Tasaka & Ito, 2003), which is a psychometric method, continuously in time.

The rest of this chapter is organized as follows. Section 2 explains link quality-based routing protocols for ad hoc networks. We introduce the continuous time assessment method of QoE in Section 3. Section 4 illustrates a methodology for the QoS/QoE assessment, including the network configuration, simulation method, QoS parameters, and QoE assessment method. The QoS assessment results are presented and discussed in Section 5. Section 6 discusses the result of QoE assessment.

2. Link quality-based routing

A variety of studies on link quality-based routing protocols have been reported. As in traditional hop-based routing protocols, they can be classified into three categories: proactive, reactive, and hybrid. We then give an overview of the three types of protocols.

2.1 Proactive routing protocol

The proactive routing protocol periodically exchanges the routing information between nodes. The protocol performs well for fixed or low mobility networks.

Itaya et al. (2005) have proposed two techniques of multi-rate aware routing for improving the stability of communication. The first technique is employment of a threshold for *signal strength (SS)* of received routing packets. It is used to avoid routing packets via unreliable neighbors with poor radio links. The second technique is *synchronous update (SU)* of routing tables. It is used to avoid loops due to mismatch in timing of route updates. The techniques can be implemented as modifications to conventional routing protocols. They have implemented these techniques into OLSR. Although the first technique can be applied to reactive routing protocols, they have implemented nothing in (Itaya et al., 2005).

As the proactive routing protocol for the comparison in this chapter, we employ the scheme proposed in (Itaya et al., 2005) with a little modification. The threshold for signal strength is kept constant for simplicity; in this chapter, we denote the threshold by T_h. Furthermore, we assume that the time synchronization among the nodes is performed completely, because the simulation environment can get the global time synchronization automatically. We refer to the scheme as *OLSR-SS*, although it is called *OLSR-SS-SU* in (Itaya et al., 2005).

2.2 Reactive routing protocol

The reactive routing protocol discovers routing paths when the source wants to send data; that is, it works on demand. It is appropriate for the use in highly mobile networks.

For example, Fan (2004) proposes high throughput reactive routing in multi-rate ad hoc networks. He modifies the AODV protocol in order to select suitable links with high data rates. In the scheme, the routing cost is calculated on the basis of MAC delay, which is equal to total delay of RTS/CTS/DATA/ACK communication. However, the scheme needs the information on the transmission speed of each link; that is, it is not a pure reactive scheme.

On the other hand, Budke et al. (2006) evaluate the QoS extensions for supporting real-time multiplayer game applications in IEEE 802.11 mobile ad hoc networks. They select AODV and add signal strength monitoring for *Route Request (RREQ)* packets. That is, the scheme can be regarded as a reactive version of the scheme proposed in (Itaya et al., 2005); thus, we refer to the scheme as *AODV-SS*.

In this chapter, as the reactive routing protocol for the comparison, we specify AODV-SS as follows. When an intermediate node receives RREQ, it decides whether the packet should be forwarded or not by received signal strength. If the received signal strength at the intermediate node is lower than the threshold T_h, which is the same as that in OLSR-SS, the node drops the packet.

2.3 Hybrid routing protocol

The hybrid routing protocol is a combination of proactive and reactive routing protocols.

Nakaoka et al. (2006) propose LQHR. In LQHR, each node maintains routing information produced by an existing proactive routing protocol and measures link quality between the neighboring nodes. When a source node makes a communication request which needs high quality links, it selects a route to the destination node by referring to the link quality on an on-demand basis.

LQHR takes account of link quality representing both reliability and the link utilization level of each node. We revise the LQHR algorithm in order to overcome difficulties related to networks with many route selections.

LQHR consists of two modules:

- Quality Measurement (QM) Module
  The QM module produces and maintains routing information by means of a proactive routing protocol; for example, OLSR is employed in (Nakaoka et al., 2006). It also periodically measures the link quality between adjacent nodes. The link quality is represented as a vector whose components are some quality parameters.
- Route Selection (RS) Module
  The RS module selects a route to the destination node by referring to the link quality, which is measured by the QM module, on an on-demand basis when a communication request is made at a node.

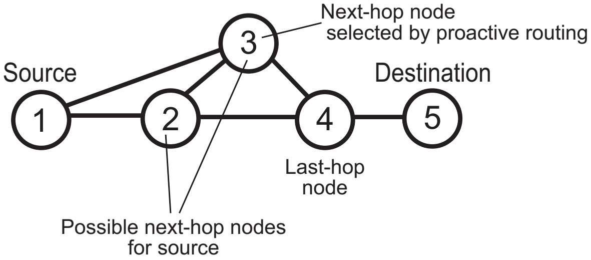


Fig. 1. Example of route discovery in LQHR.

On having a communication request, the source node sends a *Route Quality Request (RQReq)* message to each of the *possible next-hop nodes*. The possible next-hop node is a candidate of the next-hop node on the route to the destination. For example, in Fig. 1, we assume that node 1 is the source node and that node 5 is the destination node. Then, nodes 2 and 3 are the possible next-hop nodes for node 1.

The nodes receiving the RQReq message refer to the destination address and then forward it to each of their own possible next-hop nodes. The RQReq message is forwarded up to *last-hop nodes*. The last-hop node means the single-hop neighbor node to the destination. In Fig. 1, node 4 is the last-hop node to node 5.

Once the RQReq message reaches the last-hop node, it forwards back a *Route Quality Response (RQRsp)* message, via the series of the possible next-hop nodes the RQReq message has gone through, finally to the source node; thus a route from the source to the destination is selected. The RQRsp messages are chosen and discarded on the way to the source node on the basis of the link quality of each forwarding node.

In this chapter, we impose two restrictions on the algorithm of LQHR in order to overcome problems related to networks with many route candidates; many RQReq and RQRsp packets are generated, and then the effectiveness of the route discovery mechanism may degrade. One restriction is for the possible next-hop nodes, and the other is for the last-hop nodes.

At first, the revised algorithm restricts the possible next-hop nodes. The original LQHR algorithm sends RQReq packets to all the possible next-hop nodes. However, if there are many possible next-hop nodes, this is not a good strategy because the node will generate many RQReq packets, which cause congestion. Thus, the revised algorithm sends RQReq packets to only r_1 nodes which has higher link quality than other nodes. In this chapter, we set the value of r_1 to 5.

In addition, we also employ the following condition for the possible next-hop nodes. When link quality between two nodes is very high at each node, the two nodes may be geometrically close to each other. If the routing algorithm selects such links, the route will have a large number of hops. Thus, a node does not send RQReq packets to a possible next-hop node in which the link between the two nodes is one of the best r_2 links. In this chapter, we set the value of r_2 to 3.

Next, the revised algorithm also restricts the last-hop nodes. In some topology, there are a large number of last-hop nodes. However, it may not be true that all the candidates of last-hop nodes have good quality links to the destination node. Thus, as the last-hop node, the algorithm permits only nodes with the link to the destination whose quality is larger than the threshold T_h.

3. Continuous time assessment of QoE

In this chapter, we employ the method of continuous time QoE assessment in (Ito et al., 2005). This section describes the method, which utilizes the method of successive categories continuously in time.

3.1 Method of successive categories

For a start, we introduce four types of measurement scales. With the psychometric methods, the human subjectivity can be represented by a measurement scale. We can define four types of the measurement scales according to the mathematical operations that can be performed legitimately on the numbers obtained by the measurement; from lower to higher levels, we have *nominal*, *ordinal*, *interval*, and *ratio* scales (Guilford, 1954). Since almost all the statistical procedures can be applied to the interval scale and the ratio scale, it is desirable to represent the QoE by an interval scale or a ratio scale. With the psychometric methods used in (Tasaka & Ito, 2003), we can represent QoE by an interval scale.

In the method of successive categories, a subjective score is measured by the *rating scale method* (Guilford, 1954), in which subjects classify each stimulus into one of a certain number of categories. Here, a stimulus means an object for evaluation, such as audio and video. Each category has a predefined number. For example, "excellent" is assigned 5, "good" 4, "fair" 3, "poor" 2, and "bad" 1. However, in the strict sense, we cannot use the assigned number for assessing the QoE since the assigned number is an ordinal scale.

In order to obtain an interval scale as the QoE metric, we first measure the frequency of each category with which the stimulus was placed in the category by the rating-scale method. With the *law of categorical judgment* (Tasaka & Ito, 2003), we can translate the frequency obtained by the rating-scale method into an interval scale. We can apply almost all the operations to the scale.

3.2 The Law of Categorical Judgment

The law of categorical judgment makes the following assumptions. Let the number of the categories be m +1. When stimulus j ($j = 1, \ldots, n$) is presented to an assessor, a psychological value designated by s_j occurs on an interval scale in him/her. For the $m + 1$ categories, their *boundaries* have values on the interval scale. We denote the upper boundary of category g ($g = 1, \ldots, m + 1$) by c_g and define $c_0 \triangleq -\infty$ and $c_{m+1} \triangleq +\infty$. The assessor sorts n stimuli into the $m + 1$ categories ($n > m + 1$) by comparing s_j with c_g. If $c_{g-1} < s_j \le c_g$, then stimulus j is classified into category g. The categories can be arranged in a rank order, in the sense that each stimulus in category g is judged to have a psychological value which is "less than" the one for any stimulus in category g + 1. This statement holds for all values of g from 1 to m. The variable c_g is normally distributed with mean t_g and standard deviation d_g. Also, the variable s_j is normally distributed with mean R_j and standard deviation σ_j. Then, we can consider R_j as an interval scale.

Since the law of categorical judgment is a suite of assumptions, we must test goodness of fit between the obtained interval scale and the measurement result. Mosteller proposed a method of testing the goodness of fit for a scale calculated with Thurstone's law (Mosteller, 1951). The method can be applied to a scale obtained by the law of categorical judgment. In this chapter, we use Mosteller's method to test the goodness of fit.

3.3 Continuous time QoE assessment with the method of successive categories

We utilizes the method of successive categories continuously in time. The audio-video stream for evaluation is partitioned into many *fragments* each with time length Δ. For example, a stream with total length T is divided into T/Δ or $T/(\Delta + 1)$ fragments. We regard each fragment as a stimulus and utilize the method of successive categories for all stimuli (fragments). That is, assessors classify the current fragment into one of the categories every Δ. Then, we apply the law of categorical judgment to the result for all fragments.

Since the assessor only has to judge which one of the categories is the most appropriate for the stimulus every Δ, the method imposes little burden on the assessor. Moreover, by setting the number of categories to 3 or 5, the assessors can continuously enter their judgment in an input device by their fingers without directing their attention to the device. In addition to this, by utilizing the law of categorical judgment, we can obtain values of QoE metric in the form of the interval scale.

4. Methodology of QoS/QoE assessment

We assess the application-level QoS and the QoE of audio-video streaming in ad hoc networks with the three schemes of link quality-based routing: LQHR, OLSR-SS, and AODV-SS. For this purpose, we performed computer simulation with *ns-2 (network simulator version 2)*.

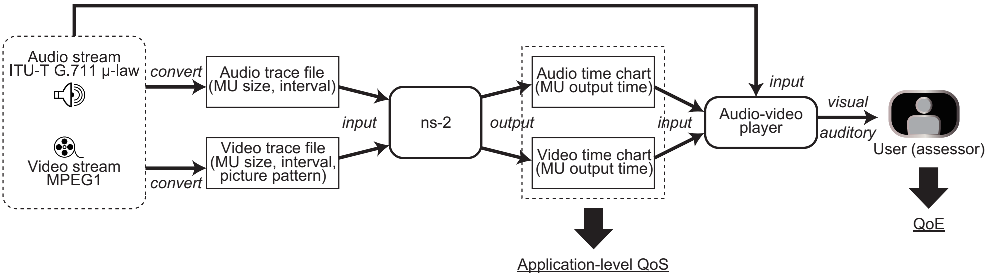


Fig. 2. Schematic diagram of QoS/QoE assessment.

Figure 2 shows the schematic diagram of the QoS/QoE assessment. We refer to the transmission unit at the application-level as a *Media Unit (MU)*; we define a video frame as a video MU and a constant number of audio samples as an audio MU. From the practical audio and video streams, we get traffic trace files for the simulation. The files include each MU size and inter-MU time. In addition, the file for video also includes the picture type of each video MU. In the simulation, we take into consideration the capturing and encoding delay time before the transmission procedure in order to emulate the audio-video streaming inputted real-time. With the traffic trace files and a simulation scenario, ns-2 outputs time charts in which the output timing of each MU is described. We can achieve application-level QoS parameter values by the time charts. Furthermore, for the QoE assessment, the audio-video player plays the practical audio-video stream with the output timing obtained from the time charts.

4.1 Network configuration

In this chapter, we consider a simple mesh topology network to assess the characteristics of the three schemes of link quality-based routing with media synchronization control in ad hoc networks. The network consists of 24 nodes as shown in Fig. 3. Each node has an omni-directional antenna. We employ the shadowing model (Rappaport, 1996) as the propagation model in the simulation. In the model, received signal strength at the receiver is determined by the following equation:

$$\left[\frac{P_r(d)}{P_r(d_0)}\right]_{\text{dB}} = -10\beta\log\left(\frac{d}{d_0}\right) + X_{\text{dB}} \tag{1}$$

If $P_r(d)$ exceeds the threshold of received signal strength, the packet can be received. Here, β means path loss exponent and is set to 2 in the simulation. d_0 is close-in distance and is set to 1.0. X_{dB} shows a Gaussian random variable; the average and the standard deviation are set to 0 and 4.0, respectively. These are default values in ns-2. The model does not consider propagation errors or fading.

In the simulation, we assume seven patterns of the mesh topology by changing the distance between two vertically or horizontally adjacent nodes; we refer to the distance as the *inter-node distance*.

In mesh topology networks, there are many available routes; therefore, the networks are suitable for the assessment of the behavior of routing schemes. However, it should be noted that as a next step of this study, we need assessment in more practical topology networks like those with many mobile nodes.

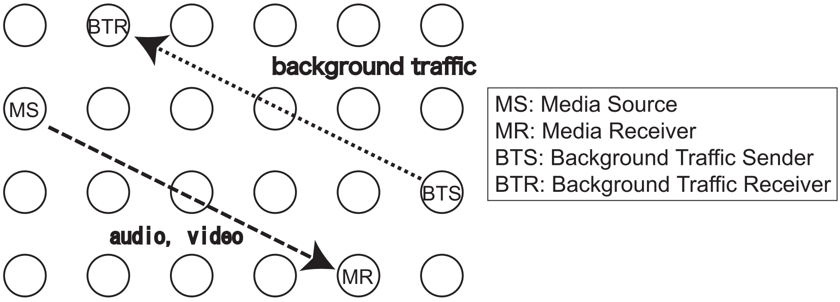


Fig. 3. Network configuration.

We formulate a detailed simulation model which is based on the *distributed coordination function (DCF)* of the IEEE 802.11b. The transmission speed is automatically changed from 2 Mb/s to 11 Mb/s by means of the rate adaptation mechanism. In this chapter, we employ

ARF (Automatic Rate Fallback) (Kamerman & Monteban, 1997). The transmission speed is controlled for each link, and broadcast frames are transmitted at 2 Mb/s. The maximum number of trials of frame retransmission is set to four. The RTS/CTS mechanism is not used in the simulation, because it has been reported that the conventional RTS/CTS mechanism does not work well in ad hoc networks (Xu et al., 2002), (Ray et al., 2003).

Because the received signal strength changes dynamically in the shadowing model, the communication range of each node fluctuates in time and is determined by the transmission speed. In the simulation, a node can receive a packet with probability 0.95 when the distance between the node and the sender is 34.54 m at 11 Mb/s, 48.36 m at 5.5 Mb/s, and 62.17 m at 2 Mb/s. These values are calculated by the *threshold* program, which is included in ns-2.

4.2 Method of simulation

In Fig. 3, we assume *MS (Media Source)* as the audio and video sources. MS transmits the media streams to *MR (Media Receiver)* with RTP/UDP. We use an audio stream of ITU-T G.711 μ-law and an MPEG1 video stream, which have been prepared by encoding a part of Japanese news program. Table 1 shows the specifications of the audio and video.

In the simulation, we take the media capturing and encoding delay time into consideration. The capture duration of an audio MU equals the inter-MU time, which is 40 ms in this chapter, and the encoding time is negligible; therefore, we set the capturing and encoding delay time of each audio MU to 40 ms. On the other hand, the capture duration of a video MU is just a moment. However, it takes much time to encode a video frame. Furthermore, in MPEG, the captured frame is buffered in the frame buffer for its predictive coding. Thus, in this chapter, we set the capturing and encoding delay time of each video MU to 74 ms; each MU leaves the source the capturing and encoding delay time after its timestamp. This value includes capturing, buffering and encoding delay for a picture. We assume that the encoding delay is 7.3 ms, which is approximately the same as that of JPEG video in (Tasaka et al., 2000). We also consider that the buffering delay is the same as the frame interval, 66.7 ms.

item	audio	video
coding scheme	ITU-T G.711 μ-law	MPEG1 GOP IPPPP
image size [pixels]	–	320 × 240
original average MU size [bytes]	320	2708
original average MU rate [MU/s]	25.0	15.0
original average inter-MU time [ms]	40.0	66.7
original average bit rate [kb/s]	64.0	325.0
measurement time [s]	120.0	

Table 1. Specifications of the audio and video.

We exert media synchronization control with the enhanced VTR algorithm (Tasaka et al., 2000). The parameter values in the enhanced VTR algorithm are set to the same as those in (Nunome & Tasaka, 2004). That is, we set the *initial buffering time* $J_{\max}$ (Tasaka et al., 2000) and the *maximum allowable delay* Δ_{al} (Tasaka et al., 2000) to 100 ms and 300 ms, respectively.

In the simulation, if MR cannot receive a picture, the succeeding P-pictures are discarded until the next I-picture appears for preserving spatial quality of the video stream; that is, the spatial quality does not degrade over the network.

Each simulation runs for 145 seconds. The source starts to generate audio and video streams at time 21 from the beginning of the simulation. In LQHR, the route is requested one second before starting audio and video streams; that is, the source generates an RQReq packet to the destination at time 20. In addition, LQHR periodically renews the route every five seconds after sending the first RQReq. For a fair comparison, AODV-SS also searches the route one second before starting to generate the streams by transmitting a dummy packet.

In this chapter, LQHR employs the received signal strength as a link quality instead of Signal-to-Noise Ratio (SNR). This is because the simulation by the original ns-2 cannot consider the strength of background noise and therefore cannot calculate SNR. The threshold value for signal strength T_h is set to −62.7 dBm, which is the threshold for acceptable signal strength at 11 Mb/s in the simulation, for all the three schemes.

The decision mechanism of the optimal T_h value is out of scope in this chapter, because we focus on basic characteristics of the three schemes. However, for example, a method for optimizing the threshold value discussed in (Itaya et al., 2005) can be used in the three schemes.

BTS (Background Traffic Sender) and *BTR (Background Traffic Receiver)* are used to handle an independent interference traffic flow for the media streams. We also employ the same routing scheme as that for the media transmission. BTS generates fixed-size IP datagrams of 1500 bytes each at exponentially distributed intervals and then sends to BTR. BTS starts to generate the traffic at time 20. The amount of the interference traffic is adjusted by changing the average of the interval. We refer to the average amount of the interference traffic as the *average load*. We set the average load to 100 kb/s in the simulation.

The route for audio-video transmission and that for background traffic are established autonomously and individually. Thus, the two routes are not always in parallel and can intersect each other. Furthermore, owing to the characteristics of the wireless radio, even if the two routes do not cross, they can affect each other.

4.3 Application-level and lower-level QoS parameters

In order to assess the application-level QoS of the media streams, we need to examine the intra-stream and inter-stream synchronization quality.

For the quality assessment of intra-stream synchronization for audio or video, we evaluate the *coefficient of variation of output interval*, which is defined as the ratio of the standard deviation of the MU output interval (i.e., the presentation time interval of two MUs at the destination) of a stream to its average; this represents the smoothness of the output stream.

For the inter-stream synchronization quality, we calculate the *mean square error*, which is defined as the average square of the difference between the output time of each video MU and its *derived output time*. The derived output time of each video MU is defined as the output time of the corresponding audio MU plus the difference between the timestamps of the two MUs.

As a measure of transfer efficiency, we assess the *average MU rate*, which is the output rate of MUs. Here, the discarded MUs are not included into the output MUs.

The *average MU delay*, which is the average of *MU delay*, is a key measure for live media. The MU delay is defined as the time interval from the moment an MU is generated until the instant the MU is output.

In addition, we also assess the behavior of the three routing schemes. For this purpose, we employ the *percentage of the number of hops*, the *percentage of selected transmission speed*, and the *number of control packets for routing*. The percentage of the number of hops shows the relative frequency of the number of hops from the source to the destination. The percentage of selected transmission speed represents the relative frequency of the transmission speed for all the links. These parameters show characteristics of the selected routes.

The number of control packets for routing means the total number of the routing packets, such as route request packets, route reply packets, and topology information packets. It shows the routing overhead.

4.4 QoE assessment

In this chapter, we assess QoE of the audio-video stream transferred with the three schemes by a subjective experiment. It was conducted as follows.

We made *stimuli* for subjective assessment by actually outputting the audio and video MUs with the output timing obtained from the simulation. Each stimulus lasts 120 seconds.

We put the stimuli in a random order and presented them to 30 assessors, using a laptop PC with headphones. The laptop PC is equipped with a 12-inch XGA (1024 × 768 pixels) LCD display. The assessors are male and female. They were in their twenties and non-experts in the sense that they were not directly concerned with audio and video quality as a part of their normal work.

A subjective score is measured by the rating-scale method. We adopted the following five categories of impairment: "imperceptible" assigned integer 5, "perceptible, but not annoying" 4, "slightly annoying" 3, "annoying" 2, "very annoying" 1. The integer value is regarded as a subjective score.

In audio-video streaming in ad hoc networks, its quality can fluctuate quite largely. In the rating-scale method, each assessor is supposed to give a subjective score for a stimulus. However, it is difficult for the assessors to give the average of the perceived quality at the end of each stimulus because of the temporal fluctuation. Thus, we asked the assessors to give a score for each fragment of a stimulus as stated below.

While a stimulus is presented to each assessor, he/she classifies every instantaneous quality into one of the five categories of impairment according to his/her subjective assessment. The assessor inputs a score by the laptop PC's keyboard whenever his/her classification changes from a score that had been input immediately before. The input score is kept until the assessor changes it to another; it is sampled every one second. The sampled value is assumed as a subjective score for the fragment for the one second.

In this chapter, we utilize the method of successive categories in order to obtain an interval scale as the QoE metric. We first measure the frequency of each category with which the fragment of the stimulus was placed in the category by the rating-scale method. With the law of categorical judgment, we can translate the frequency obtained by the rating-scale method into an interval scale. We then perform Mosteller's test, which tests the goodness of fit between the obtained interval scale and the measurement result. The interval scale of which we have confirmed the goodness of fit is referred to as the *psychological scale*.

The assessors assessed stimuli for the three routing schemes. For each routing scheme, there were four stimuli, which correspond to the inter-node distances of 20 m, 25 m, 30 m, and 35 m. It took about 40 minutes for an assessor to finish all assessment which includes the presentation of the original audio-video stream, a stimulus for practice, and 3×4=12 stimuli.

5. Assessment results of application-level and lower-level QoS

In this section, we first show the application-level QoS of the three schemes. We then present the statistics of the behavior of the routing schemes.

Each symbol in the figures to be shown represents the average of 30 measured values which were obtained by changing the random seed for generating the interference traffic. We also show 95 % confidence intervals of the measured values in the figures. However, when the interval is smaller than the size of the corresponding symbol representing the simulation result, we do not show it in the figures.

5.1 Application-level QoS of audio and video streams

In this section, we also evaluate the application-level QoS with original AODV and that with original OLSR.

Figure 4 depicts the coefficient of variation of output interval for audio as a function of the inter-node distance. Figure 5 plots the coefficient for video versus the inter-node distance.

We see in Fig. 4 that when the inter-node distance is shorter than 30 m, the coefficient of variation of output interval for LQHR is the smallest among the three link quality-based schemes. In Fig. 5, we also find that for most of the inter-node distances smaller than 30 m, the coefficient for LQHR is the smallest. This is because LQHR can select appropriate routes owing to the combination of the two routing strategies: periodical acquisition of link quality and on-demand route discovery.

On the other hand, we notice in Figs. 4 and 5 that when the inter-node distance is equal to or longer than 30 m, the coefficient of variation with LQHR suddenly becomes large. The reason is as follows. The implementation of LQHR in this chapter is an enhanced version for networks with many nodes. In the enhancement, we optimize the algorithm for comparatively dense networks by means of a heuristic approach. The enhanced algorithm restricts the selection of the highest quality links for the route; those links often have very short distances to the receivers. If those links are used, there are huge number of hops, or the

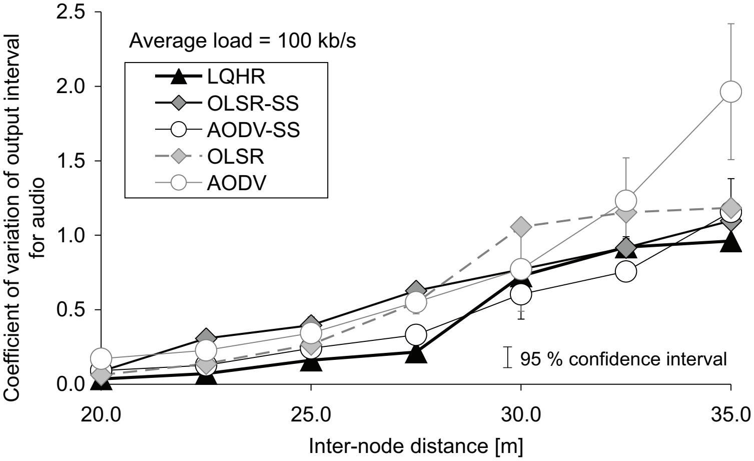


Fig. 4. Coefficient of variation of output interval for audio.

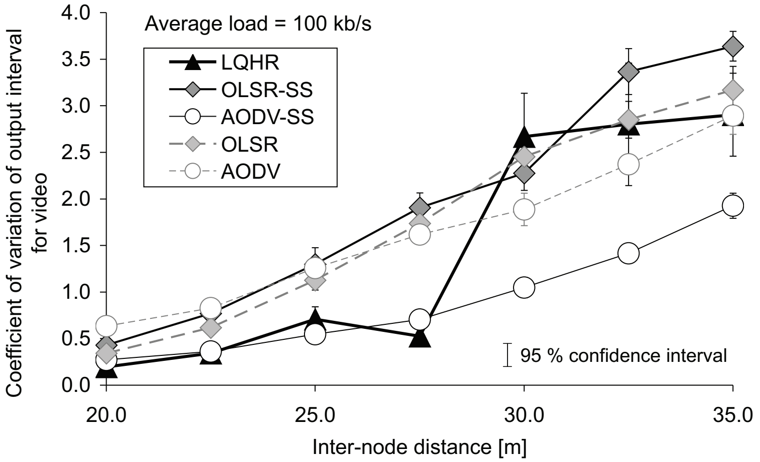


Fig. 5. Coefficient of variation of output interval for video.

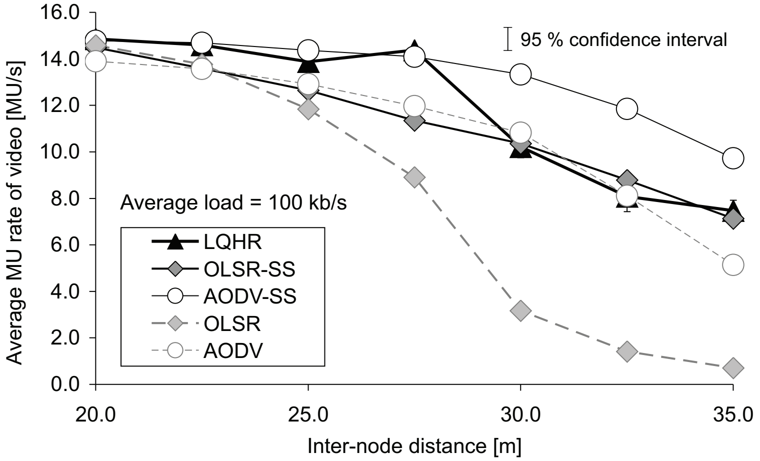


Fig. 6. Average MU rate of video.

RQReq packets cannot reach the destination. The mechanism can avoid the situations. However, when the network becomes sparse, the limitation cannot work well. This is because links with excessive quality do not exist in the sparse networks, and then the limitation may remove adequate links from the candidates. Therefore, the performance of LQHR suddenly decreases in those networks.

In Figs. 4 and 5, we also find that for almost all the inter-node distances, OLSR-SS has approximately the same or larger coefficients than the other link quality-based schemes. OLSR-SS renews its routing information periodically, and the periodical update is done on a distributed basis. Thus, the output timing of the media streams is disturbed owing to mismatch of the routing information.

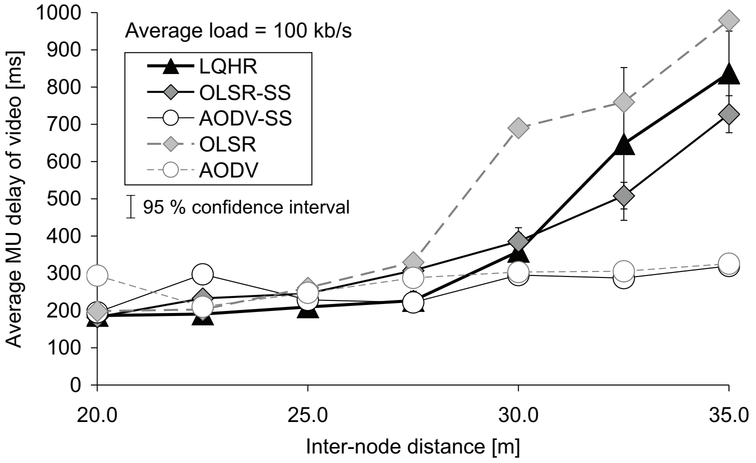


Fig. 7. Average MU delay of video.

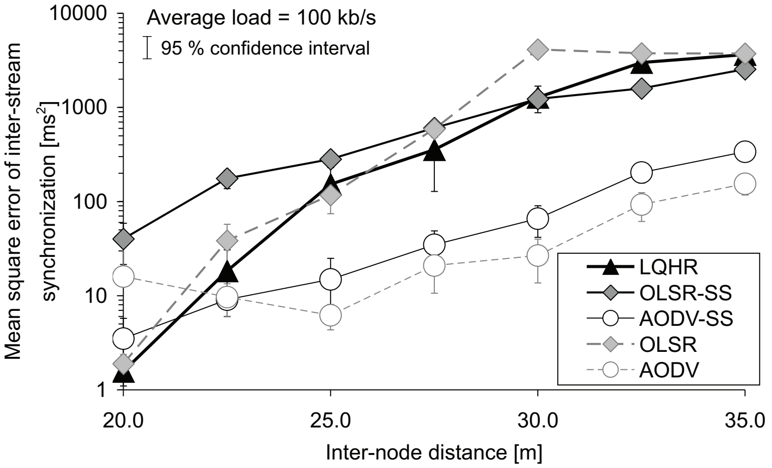


Fig. 8. Mean square error of inter-stream synchronization.

In Fig. 5, we notice that when the inter-node distance is equal to 30 m or longer, the coefficient for video with AODV-SS is the smallest among the three link quality-based schemes. This is due to the higher average MU rate described below.

Figure 6 displays the average MU rate of video versus the inter-node distance. In this figure, we see that AODV-SS has approximately the same or higher MU rate of video than the other schemes. This is because AODV-SS can avoid congestion by dynamical update of the route. However, in AODV-SS, the source starts to find the route when it initiates the generation of audio and video streams; although in the simulation, for a fair comparison, the source starts to find the route one second before. Furthermore, AODV-SS employs a mechanism of incremental route search (Perkins et al., 2003). Therefore, at the start of audio-video streaming, AODV-SS loses some packets. On the other hand, the hybrid approach (namely, LQHR) can transmit packets by using a proactively selected route even if the route is not found immediately.

Figure 7 displays the average MU delay of video. Since the relationship of the average MU delay of audio between the schemes is similar to that in Fig. 7, we do not show it here.

In Fig. 7, we find that for the inter-node distances equal to 30 m or longer, the MU delay with AODV-SS is the smallest among the three link quality-based schemes. This is because AODV-SS immediately stops using routes with unstable links because of its reactive property. AODV-SS renews the route whenever it notices route disconnection, which is detected as the excess of the MAC retry limit. In the unstable route, congestion is caused by the retransmission delay at the MAC layer; the node cannot send further packets and then the queue becomes full. The scheme can avoid congestion because it can stop to use the unstable route immediately.

On the other hand, the proactive approach and the hybrid one, namely, OLSR-SS and LQHR, continue to use the selected route during the routing update interval, which is set to five seconds in the simulation, and then congestion occurs.

In Figs. 4 through 7, we can observe that the application-level QoS with the threshold for received signal strength (namely, AODV-SS and OLSR-SS) is better than that without the threshold (namely, original AODV and original OLSR, respectively). Therefore, the link quality-based routing protocols are effective in the improvement of the application-level QoS of the audio-video streaming.

Figure 8 plots the mean square error of inter-stream synchronization versus the inter-node distance. In this figure, we can confirm that in the whole range of the inter-node distance considered here, the mean square errors of inter-stream synchronization for all the schemes are smaller than 6400 ms^2 (= 80^2 ms^2), which is a threshold of high inter-stream synchronization quality reported by Steinmetz (Steinmetz, 1996).

5.2 Statistics of the behavior of routing schemes

Table 2 shows the average number of disconnections of the audio-video route in AODV-SS. The disconnected route must be renewed, and then the number of route disconnections means the frequency of route updates.

When the route is updated every five seconds in OLSR-SS and LQHR, the number of route updates during the audio-video transmission in a simulation run is 120/5 = 24. We find in Tab. 2 that the frequency of route updates in AODV-SS is more than OLSR-SS or LQHR when the inter-node distance is equal to or longer than 25 m.

inter-node distance [m]	number of disconnections
20	10.20
22.5	15.67
25	30.20
27.5	42.63
30	72.27
32.5	133.13
35	211.40

Table 2. Average number of disconnections of audio-video route in AODV-SS.

Figure 9 depicts the percentage of the number of hops in the audio-video route. The percentage of selected transmission speed for the audio-video stream is shown in Fig. 10.

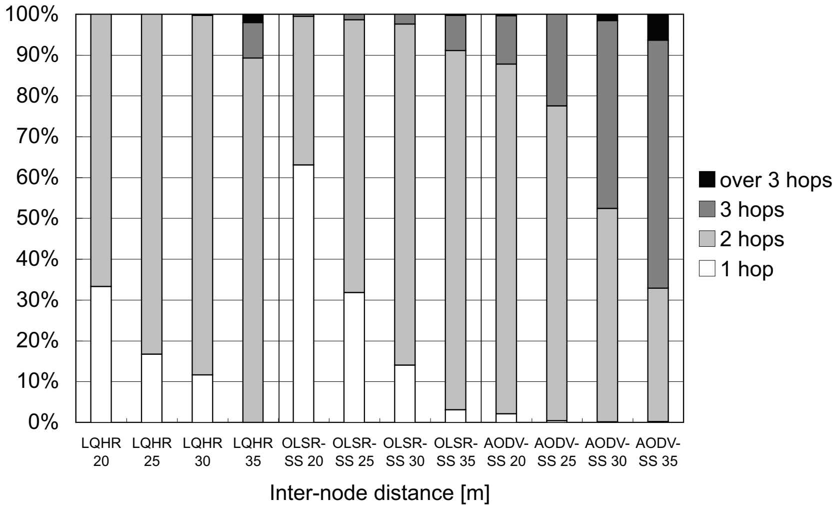


Fig. 9. The percentage of the number of hops in audio-video route.

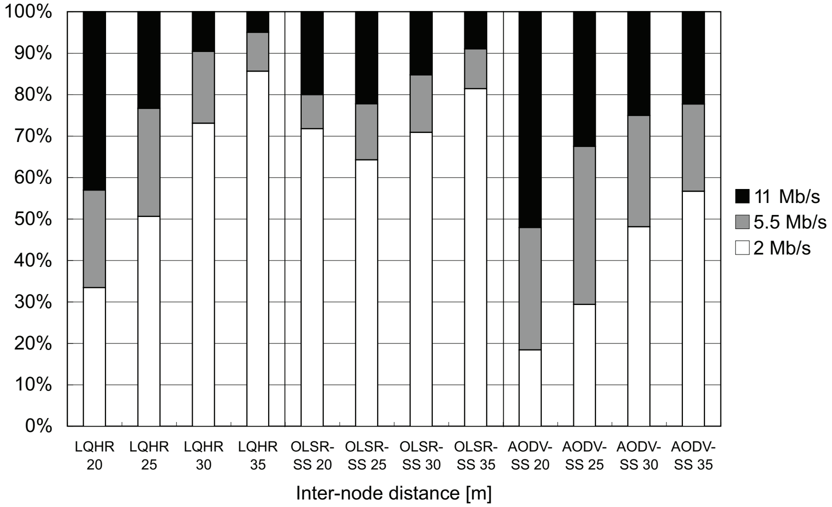


Fig. 10. The percentage of selected transmission speed for audio-video stream.

We notice in Fig. 9 that AODV-SS selects more hops than LQHR and OLSR-SS. This is because AODV-SS dynamically discovers routes in a purely on-demand way.

In Figs. 9 and 10, we can observe that the selected transmission speed is closely related to the number of hops; AODV-SS selects higher transmission speeds than the other schemes. In addition, LQHR may not select routes with higher speed links compared to AODV-SS. This is because LQHR is not optimized well; as discussed earlier, the protocol may not select appropriate links especially in the sparse networks. We need to modify the mechanism more efficiently.

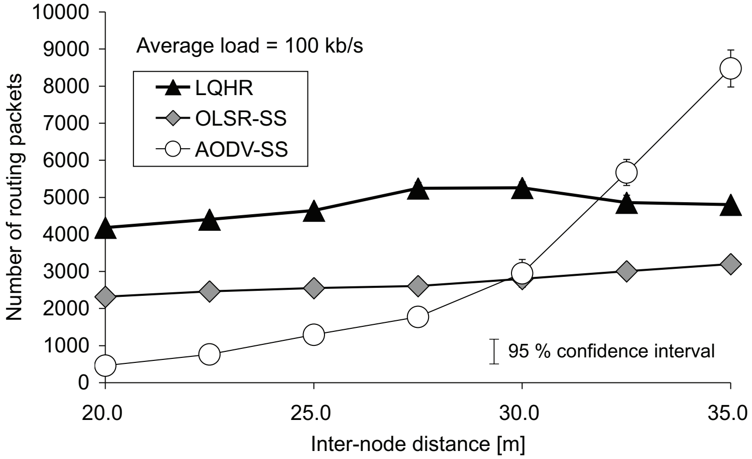


Fig. 11. Number of control packets for routing.

Figure 11 shows the number of routing packets during a simulation run. We can observe in this figure that for the inter-node distances equal to 30 m or shorter, the number of routing packets with LQHR is the largest among the three schemes. This is because LQHR adds a mechanism of on-demand route searching to the link-state routing mechanism in the original OLSR.

In Fig. 11, we also find that when the inter-node distance is equal to or longer than 32.5 m, the number of routing packets in AODV-SS is the largest. This is because it is hard to discover stable routes in AODV-SS when the distance between the nodes becomes longer. On the other hand, the routing overhead of OLSR-SS is hardly affected by the inter-node distance owing to the periodical transmission of the control packets.

From the above observation, we find that AODV-SS basically achieves high performance particularly when the inter-node distance is long. On the other hand, LQHR can achieve high QoS in networks with short inter-node distances, although it has a room for improvement. OLSR-SS has smaller routing overhead than the other schemes in networks with long inter-node distances.

6. QoE assessment result

In this section, we show the result of QoE assessment of the three schemes: AODV-SS, OLSR-SS, and LQHR.

6.1 Calculation for all the inter-node distances

We first calculate the psychological scale for all the inter-node distances employed in the assessment. We processed the result in the period of time 30 through 120. As a result of the Mosteller's test, we found that the null hypothesis that obtained interval scale fits the observed data can be rejected at significance level 0.01. This is because the obtained scale does not fit well for all the schemes for the inter-node distance 20 m, OLSR-SS for the inter-node distance 25 m, and AODV-SS for the inter-node distance 30 m.

We checked the fragments which give large errors of Mosteller's test. As a result, by removing about 27 % of the fragments, we saw that the hypothesis cannot be rejected. Figure 12 depicts the psychological scale versus the elapsed time for the inter-node distance 20 m.

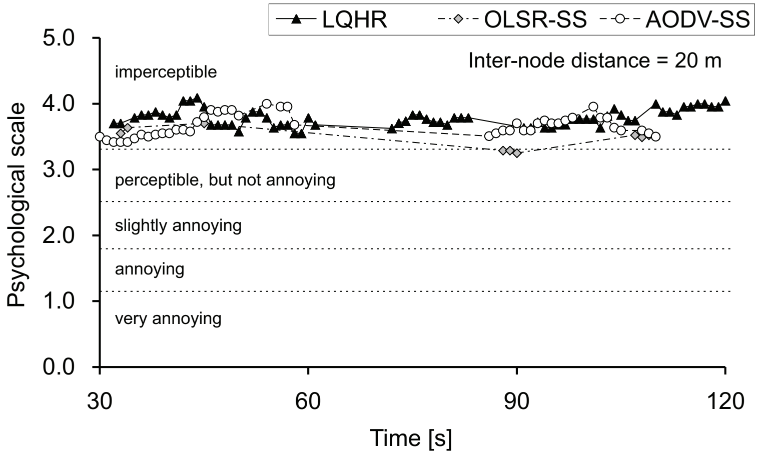


Fig. 12. Psychological scale for inter-node distance 20 m.

Note that we can select any origin of an interval scale. In this chapter, for convenience, we regard the minimum value of the psychological scale for the inter-node distance 35 m as the origin for all the inter-node distances.

Horizontal dotted lines in Fig. 12 show boundaries between the categories. Note that the lower bound of category 1 is −∞, and the upper bound of category 5 is ∞.

In Fig. 12, the removed fragments are not shown; there are a lot of removed fragments especially for OLSR-SS.

6.2 Calculation for each inter-node distance

Because the observed data can be categorized by the inter-node distances, we individually calculate the psychological scale for each inter-node distance.

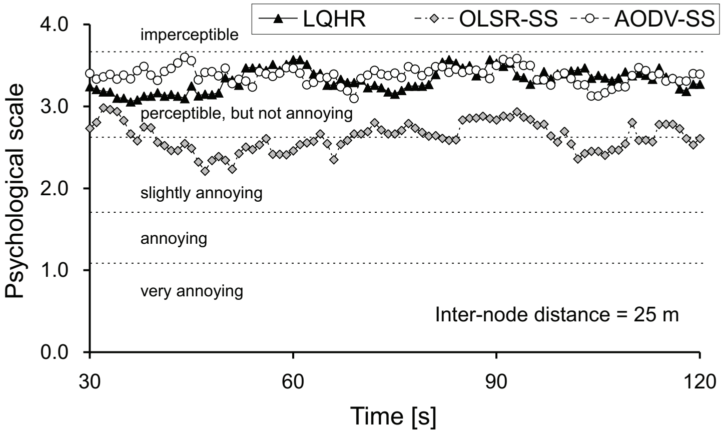


Fig. 13. Psychological scale for inter-node distance 25 m.

For the inter-node distance 20 m, we could not obtain the psychological scale. This is because the output quality of audio-video does not largely degrade for all the schemes, and then no assessor classified the stimuli into category 1, "very annoying". It can be observed in Fig. 12 that all the schemes have high output quality; almost all the fragments are categorized as category 5, "imperceptible".

On the other hand, for the inter-node distance 25 m, as a result of the Mosteller's test, we found that the null hypothesis cannot be rejected at significance level 0.01. Therefore, we consider that the obtained interval scale for this inter-node distance is appropriate for the QoE metric. Figure 13 plots the psychological scale versus the elapsed time for the inter-node distance 25 m.

For the inter-node distance 30 m, by removing about 8 % of the fragments, we found that the hypothesis cannot be rejected. Figure 14 plots the psychological scale. In the case of this inter-node distance, the quality severely changes from seed to seed, i.e, from assessor to assessor. Thus, it is more difficult for the case than the others to fit the interval scale to the obtained score.

For the inter-node distance 35 m, we saw that the hypothesis cannot be rejected by removing about 5 % of the fragments. Figure 15 indicates the psychological scale.

Comparing Fig. 15 to Figs. 13 and 14, we find that the ratio of the width of category 4, "perceptible, but not annoying" to that of category 3, "slightly annoying" for the inter-node distance 35 m is smaller than that for the inter-node distance 25 m or 30 m. This is because there are few fragments which have high quality for the inter-node distance 35 m, and then assessors did not classify the stimuli into high categories.

We notice in Figs. 13 through 15 that AODV-SS achieves higher QoE than OLSR-SS for all the inter-node distances. We also see in these figures that LQHR has approximately the same QoE as AODV-SS for inter-node distance equal to 25 m; however, when the inter-node distance is 35 m, the QoE of LQHR is almost the same as that of OLSR-SS. This is because LQHR can achieve appropriate routes in short inter-node distances, while LQHR is not optimized well for long inter-node distances.

7. Conclusions

In this chapter, we assessed the application-level QoS and QoE of audio-video streaming in a cross-layer designed wireless ad hoc network with media synchronization control at the

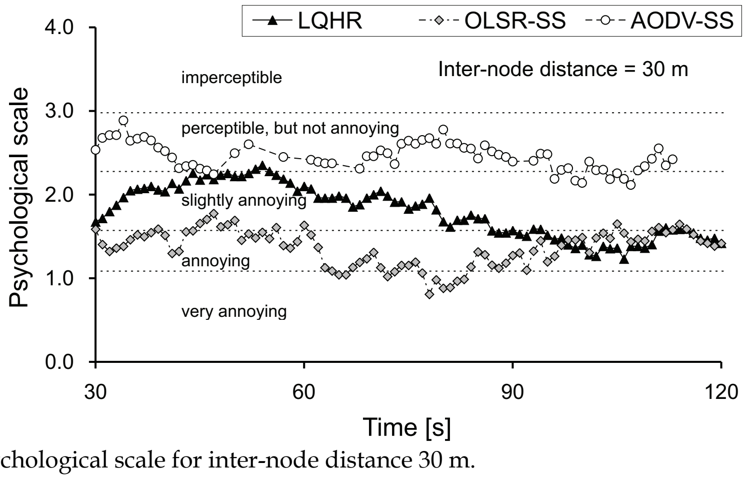


Fig. 14. Psychological scale for inter-node distance 30 m.

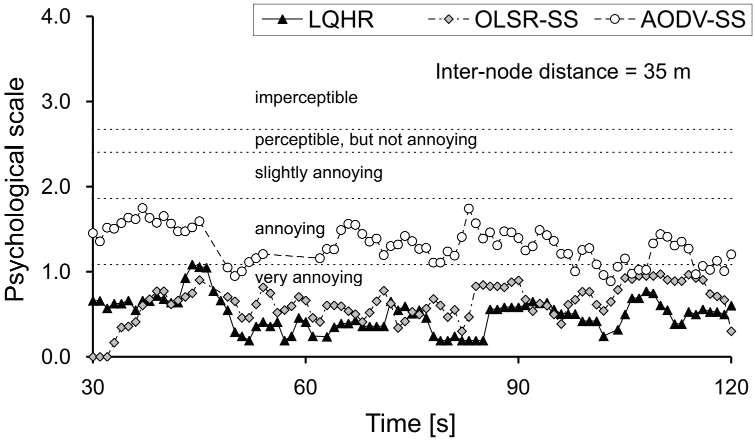


Fig. 15. Psychological scale for inter-node distance 35 m.

application-level and link quality-based routing protocols at the network-level. As a result, we found that AODV-SS, which is a reactive scheme, can achieve better application-level QoS and QoE than the other schemes in networks with long inter-node distances. However, it takes long time to search route when the source has no route.

When the inter-node distance is short, LQHR can achieve high QoE/QoS because of the combination of the proactive link quality acquisition and the reactive route discovery. However, LQHR is not optimized well and has a room for improvement. Thus, as a next step of our research, the modification of the LQHR protocol is necessary.

While this chapter does not assume QoS control mechanism in the MAC layer, IEEE 802.11e has been expected for QoS provision. Romdhani & Bonnet (2005) present a cross-layer routing protocol which is based on the cooperation between the AODV routing protocol and the IEEE 802.11e EDCA MAC protocol. We have a plan to investigate the efficiency of the IEEE 802.11e in the cross-layer design architecture for audio-video streaming.

In addition, we must assess QoE of the three schemes in the practical propagation model of the wireless channel.

8. References

Abd El Al, A., Saadawi, T.& Lee, M. (2006). A cross-layer optimized error recovery mechanism for real-time video in ad-hoc networks, *Proc. IEEE ICPADS 2006*.

Budke, D., Farkas, K., Plattner, B., Wellnitz, O. & Wolf, L. (2006). Real-time multiplayer game support using QoS mechanisms in mobile ad hoc networks, *Proc. IEEE/IFIP WONS 2006*.

Clausen, T. & Jacquet, P. (2003). RFC 3626, *Optimized link state routing protocol (OLSR)*.

Fan, Z. (2004). High throughput reactive routing in multi-rate ad hoc networks, *Electronics Letters* 40(25).

Frias, V. C., Delgado, G. D., Igartua, M. A., Delgado, J. A. & Diaz, J. M. (2005). QoS provision for video-streaming applications over ad hoc networks, *Proc. EUROCON 2005*, pp. 640–643.

Gharavi, H. & Ban, K. (2004). Dynamic adjustment packet control for video communications over ad-hoc networks, *Conf. Rec. IEEE ICC 2004*.

Guilford, J. P. (1954). *Psychometric methods*, McGraw-Hill.

Itaya, S., Hasegawa, J., Hasegawa, A. & Davis, P. (2005). Achieving stable operation of ad hoc wireless networks with neighbor pre-selection and synchronous route updates, *Proc. IEEE LCN 2005*, pp. 697–702.

Ito, Y., Tasaka, S. & Ito, R. (2005). Continuous time assessment and mapping of user-level QoS in audio-video transmission over IP networks, *Proc. IASTED Intl. Conf. COMMUNICATIONS AND COMPUTER NETWORKS*, pp. 230–237.

Kamerman, A. & Monteban, L. (1997). WaveLAN-II: A high-performance wireless LAN for the unlicensed band, *Bell Labs Technical Journal*, pp. 118–133.

Lin, X. H., Kwok, Y. K. & Lau, V. K. N. (2003). On channel-adaptive routing in an IEEE 802.11b based ad hoc wireless network, *Conf. Rec. IEEE GLOBECOM 2003*.

Mosteller, F. (1951). Remarks on the method of paired comparisons: III a test of significance for paired comparisons when equal standard deviations and equal correlations are assumed, *Psychometrika* 16(2): 207–218.

Nakaoka, K., Oba, M. & Tasaka, S. (2006). LQHR: A link quality-based hybrid routing protocol for wireless ad hoc networks, *Proc. IEEE PIMRC 2006*.

Nunome, T. & Tasaka, S. (2004). An application-level QoS comparison of single-stream and multi-stream approaches in a wireless ad hoc network, *Proc. IEEE ICCCN 2004*, pp. 25–30.

Nunome, T. & Tasaka, S. (2005). An audio-video multipath streaming scheme with media synchronization control: application-level QoS assessment in a wireless ad hoc network, *IEICE Trans. on Commun.* E88-B(9): 3623–3634.

Perkins, C., Royer, E. B. & Das, S. (2003). RFC 3561, *Ad hoc on-demand distance vector (AODV) routing*.

Rappaport, T. S. (1996). *Wireless communications, principles and practice*, Prentice Hall.

Ray, S., Carruthers, J. B. & Starobinski, D. (2003). RTS/CTS-induced congestion in ad hoc wireless LANs, *Proc. IEEE WCNC 2003*, pp. 1516–1521.

Romdhani, L. & Bonnet, C. (2005). A cross-layer on-demand routing protocol for delay-sensitive applications, *Proc. IEEE PIMRC 2005*.

Seok, Y., Park, J. & Choi, Y. (2003). Multi-rate aware routing protocol for mobile ad hoc networks, *Proc. IEEE VTC 2003-Spring*.

Setton, E., Yoo, T., Zhu, X., Goldsmith, A. & Girod, B. (2005). Cross-layer design of ad hoc networks for real-time video streaming, *IEEE Wireless Commun.* 12(4): 59–64.

Srivastava, V. & Motani, M. (2005). Cross-layer design: A survey and the road ahead, *IEEE Commun. Mag.* 43(12): 112–119.

Steinmetz, R. (1996). Human perception of jitter and media synchronization, *IEEE J. Sel. Areas in Commun.* 14(1): 61–72.

Tasaka, S. & Ito, Y. (2003). Psychometric analysis of the mutually compensatory property of multimedia QoS, *Conf. Rec. IEEE ICC 2003*, pp. 1880–1886.

Tasaka, S., Nunome, T. & Ishibashi, Y. (2000). Live media synchronization quality of a retransmission-based error recovery scheme, *Conf. Rec. IEEE ICC 2000*, pp. 1535–1541.

Xu, K., Gerla, M. & Bae, S. (2002). How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks?, *Conf. Rec. IEEE GLOBECOM 2002*, pp. 72–76.

Zhang, B. & Mouftah, H. T. (2005). QoS routing for wireless ad hoc networks: problems, algorithms, and protocols, *IEEE Commun. Mag.* 43(10): 110–117.

Zhao, Z., Long, S. & Shu, Y. (2006). Cross-layer adaptive rate control for video transport over wireless ad hoc networks, *Proc. Canadian Conf. on Elec. and Compt. Eng.*, pp. 1558–1561.

2

Quality of Service (QoS) Provisioning in Mobile Ad-Hoc Networks (MANETs)

Masoumeh Karimi
Technological University of American (TUA)
USA

1. Introduction

Next generation of wireless communication systems are engineered to service independent mobile users. These autonomous mobile users (nodes) are connected through wireless links to build a live and on-the-fly network called a Mobile Ad-hoc Network (MANET). The nodes involved in this system should collaborate among themselves and can function as both hosts and routers. They work together only based on the mutual agreement, without knowing about the network topology around them. Hence, maintaining appropriate Quality of Service (QoS) for MANETs is a complex task due to the dynamic behavior of the network topology.

Commonly, QoS for a network is measured in terms of the guaranteed amount of data which a network transfers from one place to another during a certain time. The QoS is identified as a set of measurable pre-specified service requirements; such as delay, bandwidth, probability of packet loss, and delay variance (gitter). Therefore, a network needs to meet such requirements for the end users to satisfy a particular application while transporting a packet stream from a source to its destination. The traffic types in ad-hoc networks are quite different from other infrastructures and the widespread use of wireless technologies in MANETs make the QoS approaches more complicated.

The application of MANETs was first proposed for military battlefield and disaster recovery. MANETs are mainly used when we require a quick deployment of a cooperative and distributed computing network, such as wireless sensor networks and integrated cellular networks. Accordingly, such networks are demanding to have special features; i.e., autonomous architecture, distributed operation, multi-hop routing, reconfigurable topology, fluctuating link capacity, and light weight terminals. Thus, several interesting issues can be technically involved when designing MANETs; such as security, routing, reliability, inter-networking, and power consumption due to the shared nature of the wireless medium, node mobility, and battery limitations. Therefore, providing suitable QoS for delivery of real-time communications in MANETs is more challenging than the ones in the fixed networks.

This chapter attempts to provide the reader with a basic understanding of the needs and techniques utilized for the MANETs in today's telecommunication networks by emphasizing on the scalability issues, routing protocols, security administrations, and energy management strategies. Also, a special attention is paid on the fundamental problems that will occur when trying to provide the QoS. The structure of this chapter is organized as follows. First, we discuss and analyze the dynamic nature of the mobile ad-hoc networks. Then, we identify different constrains and technical challenges which may happen while providing the required QoS. After that, we address the related works and also review several QoS frameworks for the MANETs that have been proposed in this area so far. Finally, we investigate some open research issues and give some directions for the future research works.

2. Mobile Ad-hoc networks

A mobile ad-hoc network is an independent system of mobile nodes connected by wireless links forming a short, live, on-the-fly network (as shown in Figure 1) even when access to the Internet is unavailable. Nodes in MANETs generally operate on low power battery devices (Roche et al., 2002). These nodes can function both as hosts and as routers. As a host, nodes function as a source and destination in the network and as a router, nodes act as intermediate bridges between the source and the destination giving store-and-forward services to all the neighbouring nodes in the network. Easy deployment, speed of development, and decreased dependency on the infrastructure are the main reasons to use ad-hoc network.

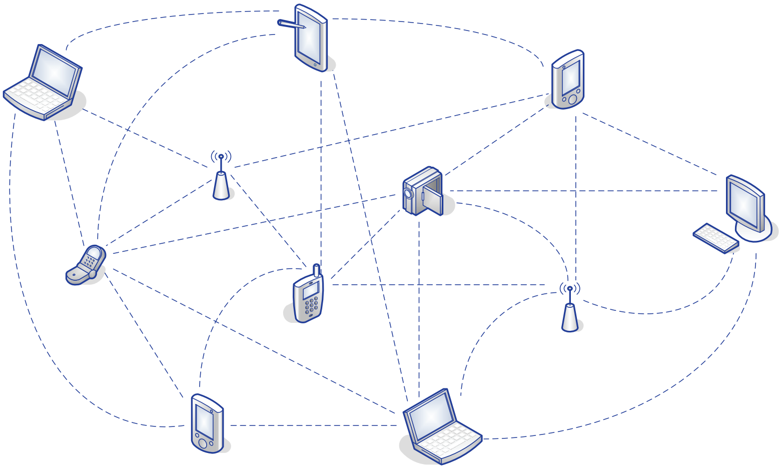


Fig. 1. A Mobile Ad-hoc Network (MANET)

In the past researches, mobile ad-hoc networks are seen as a part of the Internet, with IP-centric layered architecture. This architecture has two main advantages: it simplifies the interconnection to the Internet, and guarantees the independence from heterogeneous wireless technologies. The layered paradigm, which has significantly simplified the Internet design and led to the robust scalable protocols, can result in poor performances when applied to mobile ad-hoc networks.

Wireless networks characterize the new computer prototype. They are presenting to their user a permanent access to the network without depending on their physical location. In recent years with the decrease in the costs of mobile devices and the increase in their capacity, a new idea which is called ad-hoc network has been emerged. Through this technology, communication is made immediately and directly between person to person, person to machine or machine to person and they use wireless interface to send packet data without having fixed infrastructure like access point in a wireless local area network or base station in a cellular wireless network. Thus, due to the lack of infrastructure, they can be used quickly anywhere and anytime. MANETs have different features such as autonomous terminal, distributed operation, multi-hop routing, dynamic network topology, fluctuating link capacity, and light weight terminals. In MANETs, each mobile terminal is an autonomous node as shown in Figure 2, since the nodes can serve as routers and hosts; they can forward packets on behalf of the other nodes and run user applications.

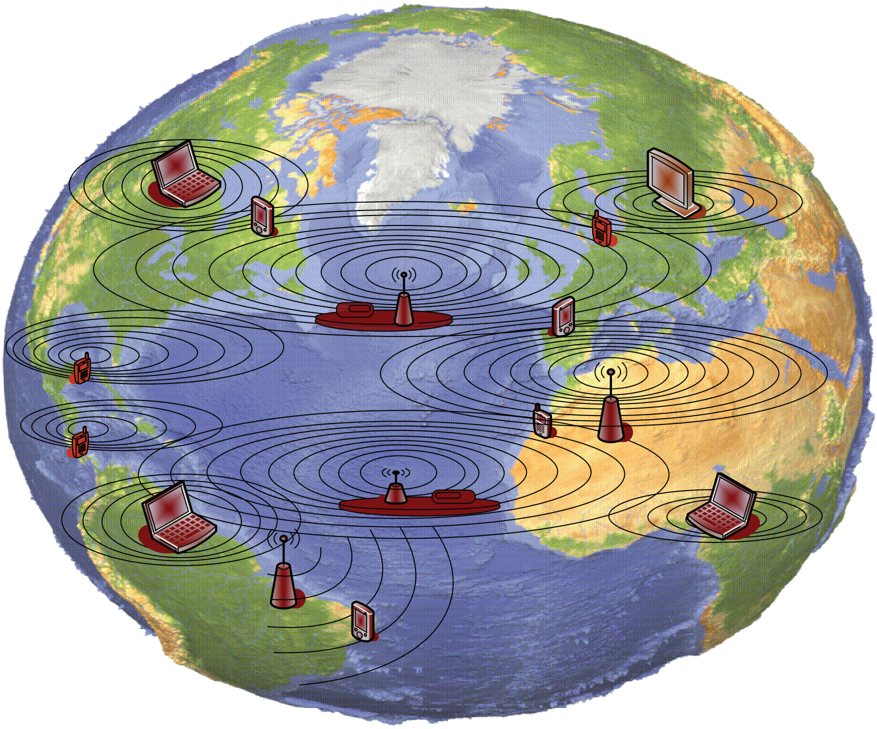


Fig. 2. Autonomous nodes in MANETs

As in distributed operations, there is no background network for the central control of the network operations, the control and management of the network is distributed among the terminals. The nodes involved in a MANET should collaborate amongst themselves and each node acts as a relay when needed to implement functions. In multi-hop routing, basic types of ad-hoc routing algorithms can be single-hop and multi-hop. In multi-hopping, nodes cooperate to relay traffic on behalf of one another to reach remote stations. This technique has increased network capacity, since the spatial domain could be reused for concurrent but physically separated multi-hop sessions (Kumar Sarkar et al., 2008).

In fluctuating link capacity, the nature of high bit-error rates of wireless connection might be more profound in a MANET. The channel over each terminal is subject to noise, fading, and interference and has less bandwidth than a wired network. In light weight terminals, the MANET nodes are mobile devices with less Central Processing Unit (CPU), small memory size, and low power storage (Murthy & Manoj, 2004).

The traffic types in ad-hoc networks are quite different from an infrastructure wireless network, including peer to peer, remote to remote, and dynamic traffic (Tan et al. 2005). In peer to peer communication between two nodes that are within one hop, the flow of the traffic is usually constant. In remote to remote communication between two nodes that are beyond a single hop, and a stable route exist between the two nodes; the traffic is similar to the standard network traffic and several nodes staying within communication range of each other in a single area or possibly moving as a group. In dynamic traffic, the problem occurs when nodes are mobile and moving around, thus, routes must be reconstructed. This causes poor network activity and connectivity in short bursts (Mirhahhak et al., 2000).

MANETs are used when we require quick deployment of a network, collaborative and distributed computing, wireless mesh networks, wireless sensor networks, and integrated cellular and ad-hoc wireless networks. When designing mobile ad-ho c networks, several interesting and difficult problems can arise (such as routing, security and reliability, quality of service, internetworking and power consumption) due to the shared nature of the wireless medium, limited transmission power of wireless devices, node mobility, and battery limitations. Figure 3 illustrates the major issues that affect performance and design of mobile ad-hoc networks.

Generally, QoS for a network is measured in terms of guaranteed amount of data which a network transfers from one place to another during a certain time (Vidhyasanker et al., 2006). There are several service models in wired networks. The two QoS models are the Integrated Services (IntServ) (Braden et al., 1994) and the Differentiated Service (DiffServ) models (Blake et al., 1998). Both of these models require accurate link state such as available bandwidth, packet loss rate, delay, and topology information.

The time-varying low-capacity resources of the network make maintaining the accurate routing information very difficult. The IntServ model provides QoS on a flow basis. It means IntServ architecture allows sources to communicate their QoS requirements to routers and destinations on the data path by means of a signaling protocol. The DiffServ model overcomes the difficulty in implementing and deploying IntServ model and Resource Reservation Protocol (RSVP) (Zhang et al., 1993) in the Internet. The RSVP is used for reserving the resources along the route. In DiffServ model, flows are aggregated into a limited number of service classes. This solves the scalability problem in the IntServ model, but it does not guarantee services on per-hop basis. This problem makes DiffServ model difficult to use in the Internet, and will be a weakness for MANETs.

Quality of Service providing a set of service requirements to the flows while routing them through the network (Crawley et al., 1998). The widespread use of wireless technologies has increased QoS for multimedia applications in wireless networks and traditional internet QoS protocols like RSVP (Braden et al., 1994) can not be used for wireless environment due to the error-prone nature of wireless links and the high mobility of mobile devices in MANETs. Therefore, providing QoS in MANETs is more challenging than in fixed and wireless networks.

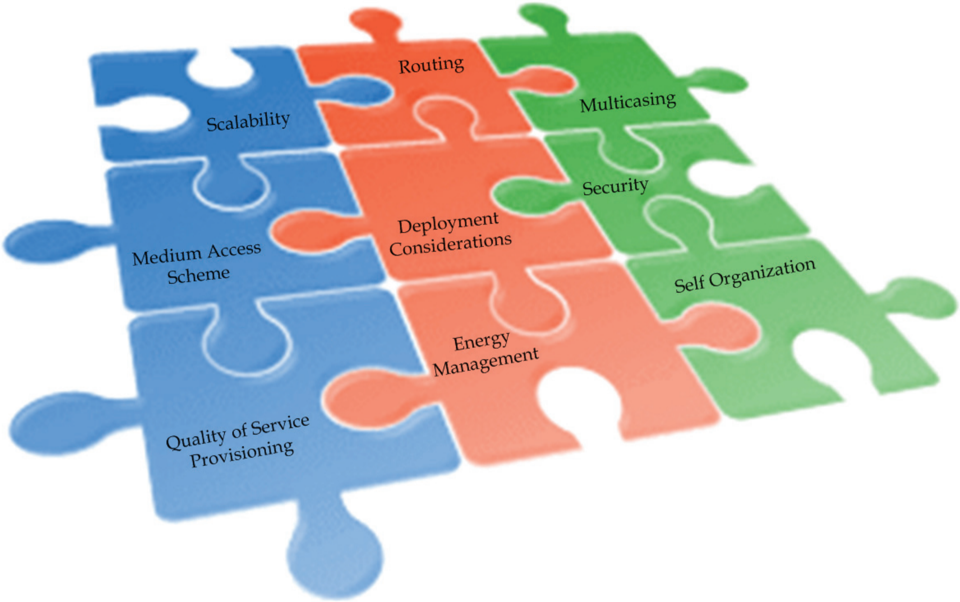


Fig. 3. The major issues that affect the performance and design of mobile ad-hoc networks.

3. Related works

A number of research have been conducted on required QoS in internet and traditional wireless networks, but current results are not appropriate for MANETs and still quality of service for MANETs is an open problem. Suitable QoS for delivery of real-time communications such as audio/ video creates a number of different technical challenges. In this section, we review several QoS frameworks for MANETs that have been proposed in this area. A framework for QoS is described as a complete system that offers essential services to each user or application. In (Xiao et al., 2000), a flexible QoS model for mobile ad-hoc networks (FQMM) is presented, which is a hybrid service model and based on IntServ and Diffserv model.

FQMM combines the reservation procedure for high priority traffic with service differentiation for low-priority traffic. Thus, FQMM provides the ideal QoS for per flow and overcomes the scalability problem by classifying the low-priority traffic into service classes. This protocol addresses the basic problem appeared by QoS frameworks (Murthy & Manoj, 2004). But it can not solve other problems such as, decision upon traffic classification, allotment of per flow or aggregated service for the given flow, amount of traffic belonging to per flow service, and scheduling or forwarding of the traffic by the intermediate nodes.

Reference (Luo et al., 2004) describes a packet scheduling approach for QoS provisioning in multihop wireless networks. Besides the minimum throughput and delay bounds for each flow, the scheduling disciplines seek to achieve fair and maximum allocation of the shared wireless channel bandwidth. The coordination of the adaptation between the different layers of the network in order to solve the problems introduced by scarce and dynamic network resources is described in (Bharghavan et al., 1998).

Mobiware effort has investigated the concept of QoS ranges, adaptively, and other mechanisms for providing QoS in wireless environment (Angin et al., 1998). More recently, the INSIGNIA protocol combines the idea of QoS ranges with lightweight signaling carried in the data packet headers as an approach to providing QoS in a mobile ad hoc network (Mirhahhak et al., 2000). This IP-based quality of service framework is designed to be lightweight and highly responsive to changes in the network. Adaptive services support applications that require only a minimum quantitative QoS guarantee (minimum bandwidth) called base quality of service (Lee et al., 2000). INSIGNIA is an in-band signaling protocol, integrated with an ad-hoc routing protocol. An in-band signaling system supports fast flow reservation, restoration, and end-to-end adaptation based on the inherent flexibility, robustness and scalability found in IP networks. This soft state reservation scheme used in this framework guarantees that resources are quickly released at the time of path reconfiguration.

Network feedback based on link and acceptable throughput measurements were made to support higher layer and soft quality of service. However, these schemes do not consider the inherent characteristics (changing network topology, limited resource availability, and error-prone shared radio channel) of MANETs and drawbacks of integrated services and differentiated services (Guimar et al., 2004). Therefore, for supporting a combination of real-time (voice or video) and non-real-time services (data or FTP), an accurate model has to be designed to investigate its applicability within the MANETs.

4. Identifying problems and solutions

In general, the application of MANETs was first proposed for military battlefield and disaster recovery. However, as a result of evolution in multimedia technology and the commercial interest of companies, quality of service in mobile ad-hoc networks has become an area of interest. Because of various requirements of different applications, the services required and the QoS parameters will change for each application. Therefore, quality of service is identified as a set of measurable pre-specified service requirements such as delay, bandwidth, probability of packet loss, and delay variance (jitter) which a network needs to make them available for the end users while transporting a packet stream from a source to its destination.

Real time applications need mechanisms that guarantee restricted delay and delay jitter. For instance, the most important delays that affect the end to end delay in packet delivery from one node to another node are: the queuing delay at the source and intermediate nodes, the processing time at the intermediate nodes, the transmission delay, and the propagation duration over multiple hops from the source node to the destination node (Kurose & Ross, 2007).

Generally in wired networks, QoS parameters are characterized by the requirements of multimedia traffic. But in ad-hoc networks QoS requires new constraints due to highly dynamic network topology and traffic load conditions, time-variant QoS parameters like throughput, latency, low communication bandwidth, limited processing and power capacity than wire-based network.

Moreover, QoS in ad-hoc networks relates not only to the available resources in the network but also to the mobility speed of these resources. This is because mobility of nodes in ad-hoc networks may cause link failures and broken paths. In order to continue a communication therefore, it requires finding a new path. However, delay will occur for establishing a new path, also some of the packets may get lost (Grossglauser & Tse, 2002). Figure 4 depicts some challenges when proving the QoS in MANTEs.

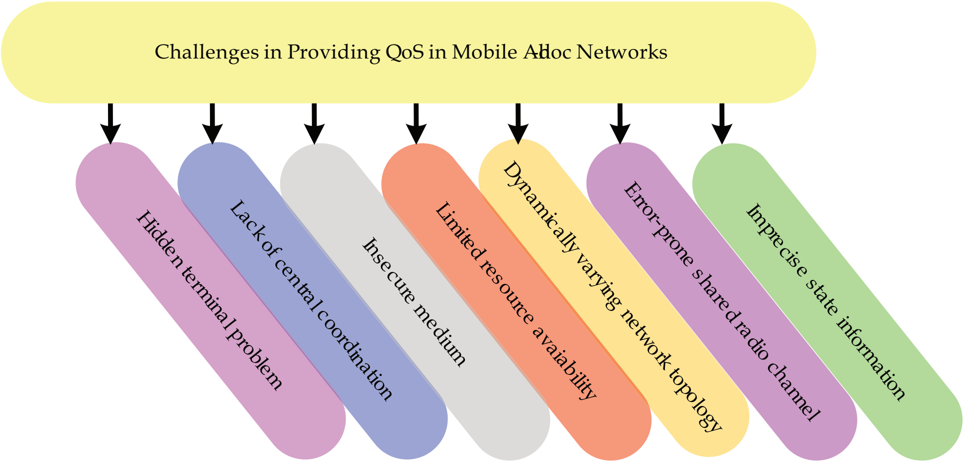

Challenges in Providing QoS in Mobile Ad-Hoc Networks
Hidden terminal problem
Lack of central coordination
Insecure medium
Limited resource avaiability
Dynamically varying network topology
Error-prone shared radio channel
Imprecise state information

Fig. 4. Some challenges when proving the QoS in MANTEs (Murthy & Manoj, 2004).

Error-prone shared radio channel is another issue for providing QoS as the radio channel in a broadcast medium, thus, during propagation through the wireless environment the radio waves go through several impairments (e.g. attenuation, multipath propagation, and interference) from other wireless devices working in the surrounding area (Rappaport, 2002).

In mobile ad-hoc networks, mobile computation devices are usually battery powered. A limited energy budget constraints the computation and communication capacity of each device. Energy resources and computation workloads have different distributions within the network. The main reasons for energy management in ad-hoc networks are limited energy reserve, difficulties in replacing the batteries, lack of central coordination, constrains on the battery source, and selection of optimal transmission power (Murthy & Manoj, 2004), (Kumar Sarkar et al., 2008).

The battery Life, bandwidth, and buffer space are the important resources in each network. Usually, the transmitter power consumes the most energy in the node and it is essential to conserve the available energy in MANETs either by low-power design of hardware (Lahiri et al., 2002) or special power control mechanisms (Agarwalet al, 2001), (Wattenhofer et al, 2001), (Cartigny et al., 2003).

The hidden terminal problem is inherent mobile ad-hoc networks (Sekido et al. 2005). This may happen when packets originating from two or more sender nodes which are not within the direct transmission range of each other (Figure 5), crash at a general receiver node. Thus, it requires the retransmission of the packets that may not be adequate for flows.

Security issue is an important factor in providing QoS in mobile ad-hoc networks. Communications in wireless environment are not secure due to the broadcasting behaviour of this type of network (Carvalho, 2008). Generally, MANETs have fewer resources than fixed networks and they are more influenced by the resource constraints of the nodes. Therefore, it is hard for these networks to support different applications with appropriate QoS requirements (Wu & Harms, 2001).

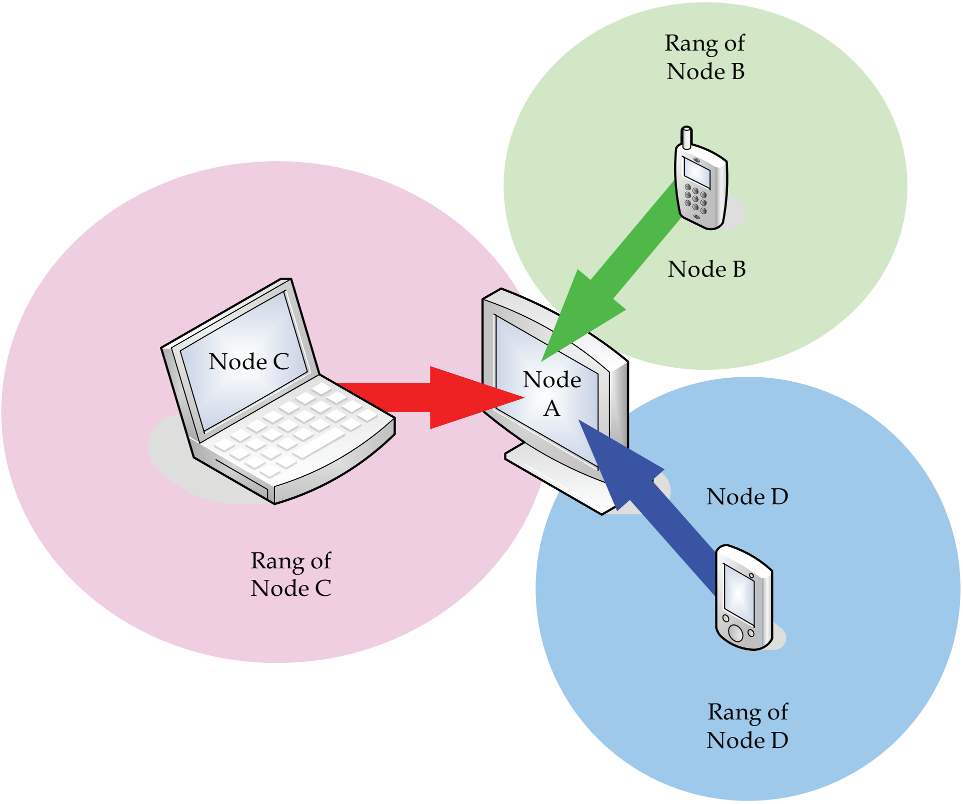


Fig. 5. Hidden terminal problem: when Node C is transmitting to Node A, one or more nodes (here Node B & D) are concurrently transmitting to Node A.

The four main goals of cryptography for any networks are Confidentiality, Integrity, Availability, and Non-repudiation, as demonstrated in Figure 6. The major issues to provide security are as follows: shared radio broadcast channel, unsecured operational environment,

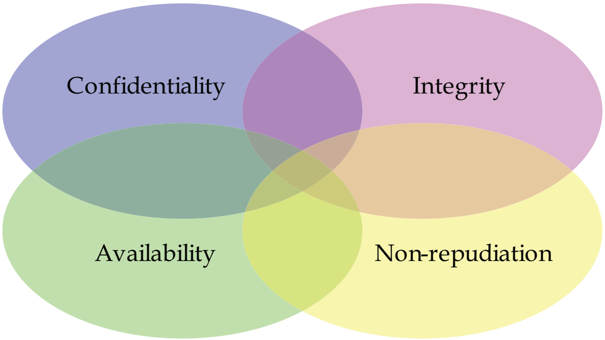


Fig. 6. The four fundamental requirements for a secure network.

lack of central authority, lack of association, limited resource availability, and physical vulnerability. Accordingly, the requirements of a secure routing protocol for MANETs are: detection of malicious nodes, guarantee of correct route discovery, confidentiality of network topology, and stability against attacks (Murthy & Manoj, 2004). Figure 7 displays the security issues in each TCP/IP layer (Yang et al., 2004), (Comer, 2005).

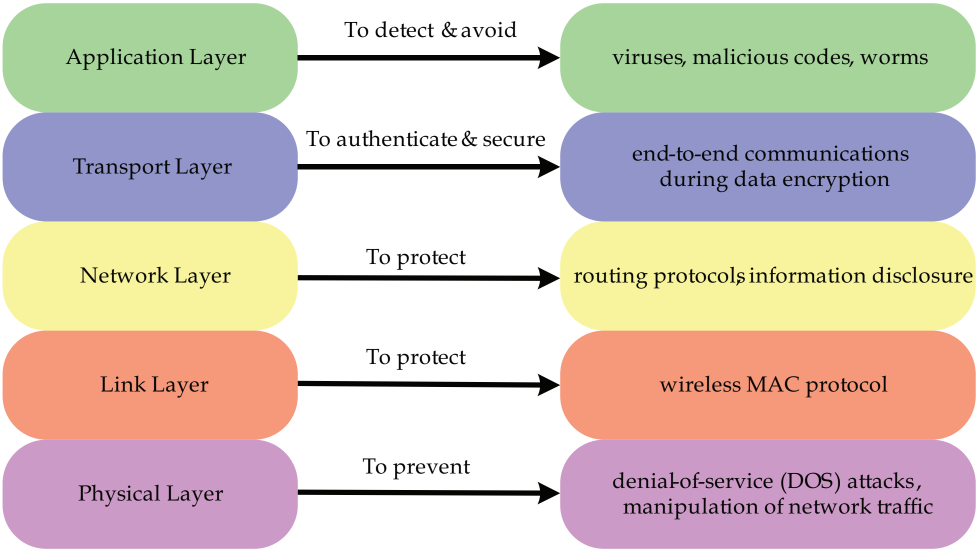

Application Layer
To detect & avoid
viruses, malicious codes, worms
Transport Layer
To authenticate & secure
end-to-end communications during data encryption
Network Layer
To protect
routing protocols, information disclosure
Link Layer
To protect
wireless MAC protocol
Physical Layer
To prevent
denial-of-service (DOS) attacks, manipulation of network traffic

Fig. 7. The security issues: challenges and corresponding layers (Yang et al., 2004).

The other important problems in MANETs when providing QoS are routing, maintenance and variable resource problems (Lin & Liu, 1999).

1. Routing problem: It explains how to find a loop-free from the source to the destination in the network that can be able to support a requested level of QoS. Route selection strategies can be based on the power aware, level of the signal strength, link stability, and the shortest path.
2. Maintenance problem: It describes how to make sure that, when network topology changes, new routes that can support existing QoS obligations are available, or can be quickly found.
3. Variable resource problem: It addresses how to react to changes in available resources, either as the result of a route change, or as the result of changes in link characteristics within a given route.

As we mentioned earlier a mobile ad-hoc network (MANET) may include a group of mobile nodes with a wireless communications device and a controller, in which they operate in accordance with a multi-layer protocol hierarchy (Tanenbaum, 2003). The QoS solutions can be classified based on the QoS approaches or based on the layer at which they operate in the network protocol stack. Generally, the QoS approaches can be classified based on the interaction between the routing protocol and the QoS provisioning mechanism, and the interaction between the network and the Medium Access Control (MAC) layers, or based on the routing information update mechanism (Murthy & Manoj, 2004).

Based on the interaction between the routing protocol and the QoS provisioning mechanism, QoS approaches can be divided into Coupled and Decoupled (Shah & Nahrstedt, 2002).

1. In Coupled QoS the routing protocol and the QoS provisioning mechanism directly interact with each other for delivering the required QoS.
2. In Decoupled QoS the QoS provisioning mechanism does not depend on any specific routing protocol with the intention of having required QoS.

In addition, QoS approaches can be categorized as independent and dependent based on the interaction between the routing protocol and the MAC protocol (Murthy & Manoj, 2004). In independent QoS, the network layer is not dependent on the MAC layer QoS provisioning. While for dependent QoS, it requires the MAC layer to support the routing protocol for QoS provisioning.

Figure 8 illustrates some ad-hoc network routing protocols, each one established for a particular purpose (http://wiki.uni.lu/secan-lab/).

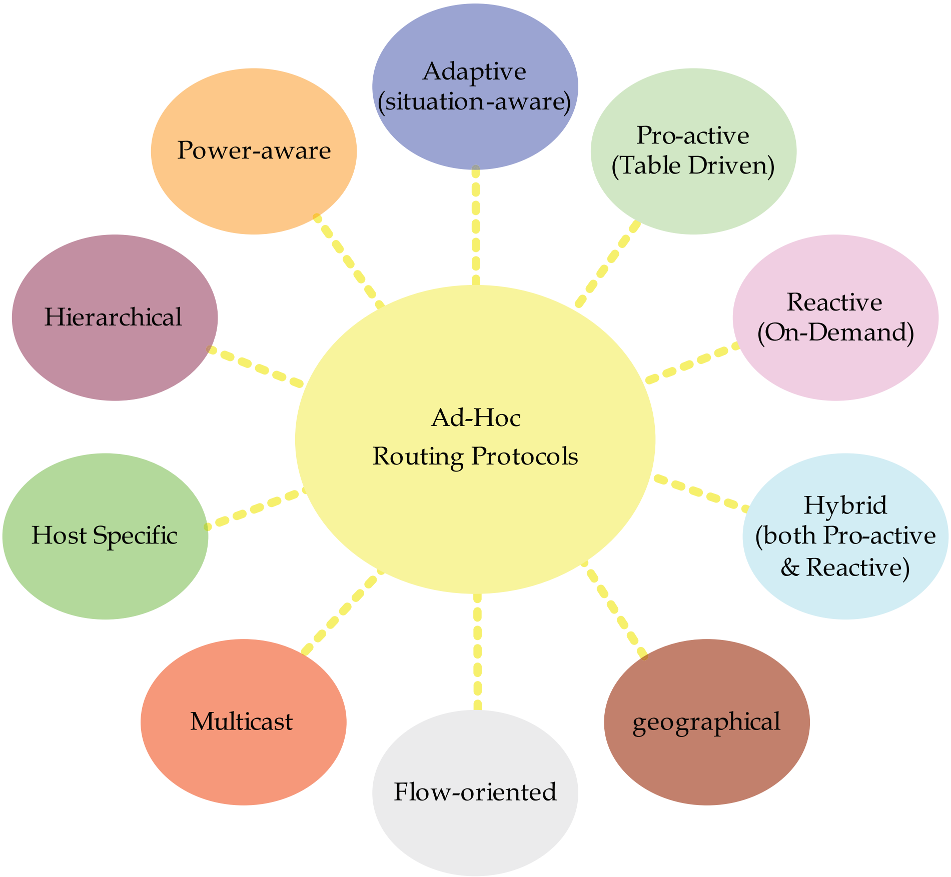


Fig. 8. Some ad-hoc routing protocols (algorithms)

As can be seen, routing strategies can be also categorized as adaptive routing and not-Adaptive routing. For Ad-Hoc-Networks, only adaptive strategies are useful (http://www.cs.uiuc.edu). Accordingly, the route selection strategies are characterized as follows: power-aware routing, signal strength, link stability, shortest path, link-state routing, and distance-vector routing.

Furthermore, QoS approaches based on the routing information update mechanism can be classified as table-driven, on-demand, and hybrid (Mbarushimana & Shahrabi, 2007).

1. Table-driven (Pro-Active), each node in the network holds a routing table which can support the forwarding packets. The routing tables are called periodically or event-driven and it will be only updated if any change happens in the network. The main disadvantages of table-driven QoS are bandwidth consumption in transmitting routing tables and also saving the table of the routes that are not used in future (Reddy ET AL., 2006).
2. On-demand (Reactive), there is no any routing table at nodes; thus, the source node has to discover the route by flooding the network with route request packet. In this technique, routes are calculated when they are needed. The main disadvantages of the on-demand approach are delay when the source node trying to find a route and also excessive flooding can be led to the network clogging (Chen et al., 2002).
3. Hybrid (Pro-Active/ Reactive), which integrates attributes of the two above approaches. The disadvantage of the hybrid technique depends on the number of active nodes in the network (Pandey et al., 2006).

A list of some ad-hoc routing protocols is given in Figure 9 (http://wiki.uni.lu/secan-lab), such as Destination Sequenced Distance Vector (DSDV) Routing (Perkins, 2001), Wireless Routing Protocol (WRP) (Murthy & Aceves, 1995), Hieracical State Routing (HSR) (Iwata et al., 1999), Ad-hoc On demand Distance Vector (AODV) Routing (Royer et al. 2000), Dynamic Source Routing (DSR) (Johnson & Maltz, 1996), Temporally Ordered Routing Algorithm (TORA) (Park & Corson, 1997), Zone Routing Protocol (ZRP) (Haas, 1997), Hazy Sighted Link State (HSLS) Routing Protocol (Santivanez & Ramanathan, 2001), Scalable Source Routing (SSR) (Fuhrmann et al., 2006).

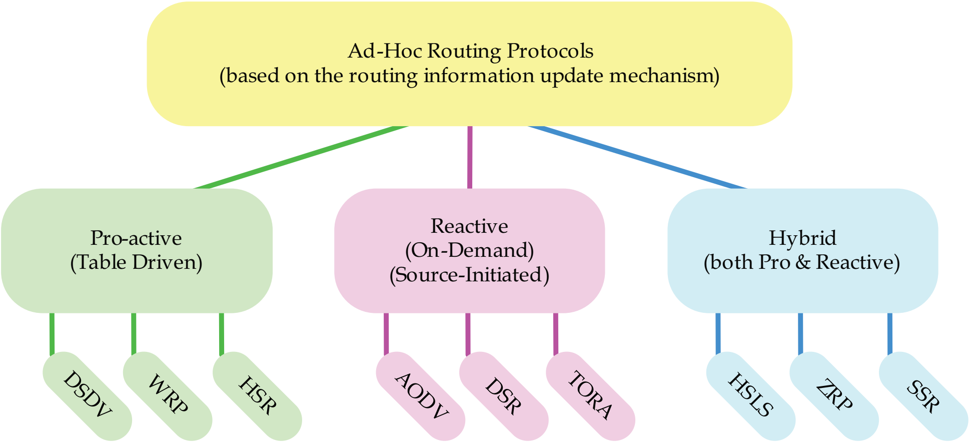


Fig. 9. Some ad-hoc routing protocols (based on the routing information update mechanism)

Finally, some major open issues and challenges in MANETs (Taneja & Patel, 2007) are depicted in Figure 10.

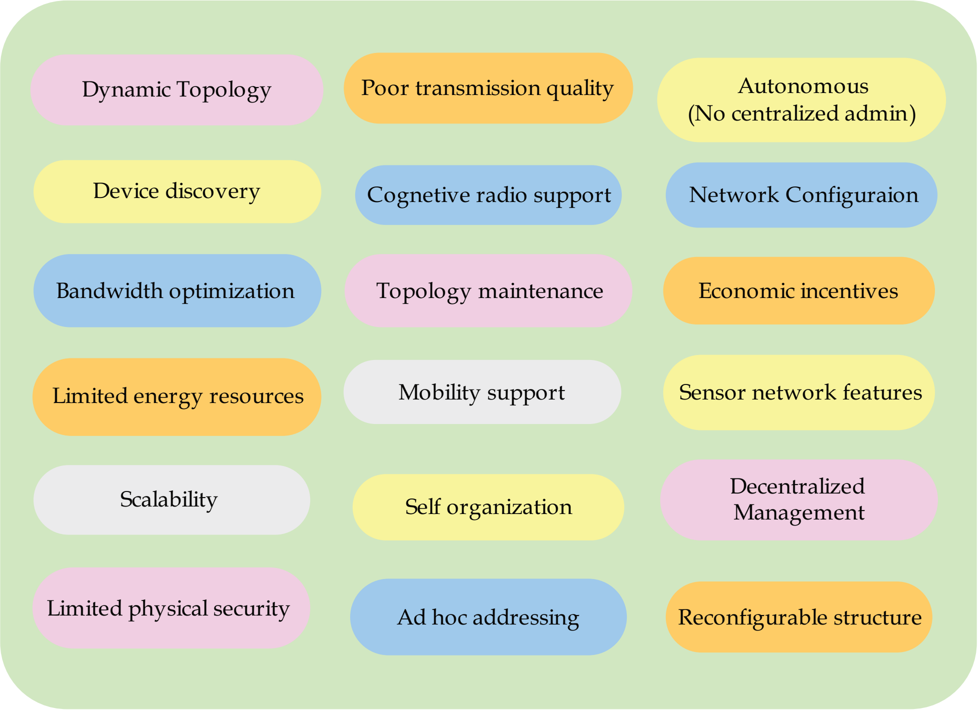


Fig. 10. Some major open issues in MANETs.

5. Conclusions

Multi-hop mobile radio network, also called mobile ad-hoc network is created by a set of mobile nodes on a shared wireless channel. This network is adaptable to the highly dynamic topology resulted from the mobility of network nodes and changing propagation conditions. MANETs are expected to have a significant place in the development of wireless communication systems. Such networks are attractive because they can be rapidly deployed anywhere and anytime without the existence of fixed base stations and system administrators. Hence, mobile ad-hoc networks must be able to provide the required quality of service for the delivery of real-time communications such as audio and video that poses a number of different technical challenges and new definitions.

Many ideas regarding QoS inherited from the wire-based networks can be used for MANETs if we consider various constraints due to the dynamic nature, bandwidth restriction, the limited processing, and capabilities of mobile nodes. Thus, for providing efficient quality of service in mobile ad-hoc networks, there is a solid need to create new architectures and services for routine network controls.

6. Future works

The development of mobile ad-hoc networks provides great chances in various areas including academic, defence, disaster recovery, industrial environments, and healthcare. Nevertheless, there are many challenges that require to be addressed as well. These challenges needs to develop efficient routing procedures, mechanisms for reducing power consumption and extending the battery life, mechanisms for efficient use of limited bandwidth and communication capacity, new algorithms for information security, and making smaller but more powerful mobile devices.

7. References

Agarwal, S.; Katz, R.H.; Krishnamurthy, S.V. & Dao, S.K. (2001). Distributed power control in ad-hoc wireless networks, *Proceedings of 12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, October 2001, pp. F-59-F-66 Vol. 2, San Diego, CA , USA, ISBN: 0-7803-7244-1.

Angin, O.; Campbell, A.; Kounavis, M. & Liao, R. (1998). The Mobiware Tollkit: Programmable Support for Adaptive Mobile Computing, IEEE Personal Communications Magazine, Special Issue on Adapting to Network and Client Variability, Vol. 5, No.4, August 1998, Sponsored by: IEEE Communications Society pp. 32-44, ISSN: 1070-9916.

Bharghavan, V.; Lee, K.; Lu, S.; Ha, S.; Li, J. R. & Dwyer, D. (1998). Timely Adaptive Resource Management Architecture, IEEE Personal Communication Magazine, Vol. 5, No.4, August 1998, Sponsored by: IEEE Communications Society pp. 20-31, ISSN: 1070-9916.

Blake, S.; Black, D.; Carlson, M.; Davies, E.; Wang, Z. & Weiss, W. (1998). Architecture for Differentiated Services, IETF RFC 2475, 1998, RFC Editor, USA.

Braden, R.; Clark, D. & Shenker, S. (1994). Integrated Services in the Internet Architecture: an overview, *IETF RFC 1633*, 1994, RFC Editor, USA.

Cartigny, J.; Simplot, D. & Stojmenovic, I. (2003). Localized minimum-energy broadcasting in ad-hoc networks, *proceedings of IEEE Computer and Communications conference (INFOCOM)*, pp. 2210 – 2217, Vol.3, April 2003, San Francisco, USA, ISBN: 0-7803-7752-4.

Carvalho, M. (2008). Security in Mobile Ad Hoc Networks, *IEEE Security and Privacy Magazine*, Vol. 6, No. 2, pp. 72-75, April 2008, ISSN: 1540-7993.

Chen, Y.; Tseng, Y.; Sheu, J. & Kuo, P. (2002). On-Demand, Link State, Multipath QoS Routing in a Wireless Mobile Ad-hoc Network, *Proceedings of European Wireless*, pp.135-141, February 2002.

Comer, D.E. (2005). *Internetworking with TCP/IP: Principles, Protocols and Architecture*, Pearson Prentice Hall, 2005, ISBN 0-13-187671-6.

Crawley, E.; Nair, R.; Rajagopalan, B. & Sandick, H. (1998). A Framework for QoS Based Routing in the Internet, *RFC 2386*, August 1998, RFC Editor, USA.

Grossglauser, M. & Tse, D. N. C. (2002). Mobility Increases the Capacity of Ad-hoc Wireless Networks, *IEEE/ACM Transactions on Networking (TON)*, Vol. 10, No.4, August 2002, pp. 477-486, ISSN:1063-6692, IEEE Press Piscataway, NJ, USA.

Guimar, R.; Morillo, J.; Cerd, L.; Barcel, J. & Garc, J. (2004). Quality of service for mobile Ad-hoc Networks: An Overview, *Technical Report UPC-DAC-2004-24*, Polytechnic University of Catalonia, June 2004.

Haas, Z. J. (1997). A New Routing Protocol For The Reconfigurable Wireless Networks, *Proceedings of 6th IEEE International Conference on Universal Personal Communications (IEEE ICUPC)*, pp. 562-566, October, 1997, San Diego, California, USA.

http://www.cs.uiuc.edu/class/sp07/cs525/slides_022007.pdf

http://wiki.uni.lu/secan-lab

Johnson, D. B. & Maltz, D. A. (1996), *Dynamic Source Routing in Ad Hoc Wireless Networks*, Mobile Computing, Vol. 353, Chapter 5, pp. 153-181, Kluwer Academic Publishers, 1996.

Iwata, A.; Chiang , C. C.; Pei, G.; Gerla, M. & Chen, T. W. (1999). Scalable Routing Strategies for Ad hoc Wireless Networks, *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, pp.1369-1379, August 1999.

Kumar Sarkar, S.; Basavaraju, T.G. & Puttamadappa, C. (2008) *Ad-hoc Mobile Wireless Networks Principles, Protocols, and Applications*, Auerbach Publications, Taylor & Francis Group, ISBN-13:-4200-6221-2, USA.

Kuros, J. & Ross, K. (2007). *Computer networking: a top-down approach*, Pearson Addison Wesley, 4th edition, ISBN-13: 978-0-321-49770-3, USA.

Lahiri, K.; Raghunathan, A.; Dey, S. & Panigrahi, D. (2002). Battery-Driven System Design: A New Frontier in Low-Power Design, *Proceedings of ASP-DAC/VLSI Design 2002*, pp. 261-267, ISBN: 0-7695-1441-3, Bangalore , India.

Lee, S. B.; Gahng-Seop, A.; Zhang, X. & Campbell, A. T. (2000). INSIGNIA: An IP-based Quality of Service Framework for Mobile Ad-hoc Networks, *Journal of Parallel and Distributed Computing*, Vol. 60, No.4, April 2000, pp. 374-406. ISSN:0743-7315, Academic Press, Inc. Orlando, FL, USA.

Lin, C. R. & Liu, J. (1999). QoS Routing in Ad-hoc Wireless Networks, *IEEE Journal on Selected Areas in Communications*, Vol. 17, No.8, August 1999, pp.1426-1438, ISSN: 0733-8716.

Luo, H.; Lu, S.; Bharghavan, V.; Cheng, 1. & Zhong, G. (2004). A Packet Scheduling Approach to QoS Support in Multi-hop Wireless Networks, *Mobile Networks and Applications*, Vol. 9, No. 3, June 2004, pp. 193-206. ISSN: 1383-469X, Kluwer Academic Publishers, Hingham, MA, USA.

Mbarushimana, C. & Shahrabi, A. (2007). Comparative Study of Reactive and Proactive Routing Protocols Performance in Mobile Ad Hoc Networks, *Proceedings of 21st Conference on Advanced Information Networking and Applications Workshops (AINAW)*, Vol. 2, pp. 679 - 684 , ISBN:0-7695-2847-3, May 2007, IEEE Computer Society Washington, DC, USA.

Mirhahhak, M.; Schult, N. & Thomson, D. (2000). Dynamic Quality-of-Service for Mobile Ad-hoc Networks, *Proceedings of the 1st ACM International Symposium on Mobile Ad-hoc Networking & Computing*, pp. 137–138, ISBN:0-7803-6534-8, Boston, Massachusetts, 2000, IEEE Press Piscataway, NJ, USA.

Murthy, S. & Aceves, J. J. G. (1995). A Routing Protocol for Packet Radio Networks, *ACM International Conference on Mobile Computing and Networking (MOBICOM)*, pp. 86-95, November 1995, Berkeley, California, USA.

Murthy C. S. R. & Manoj, B. S. (2004). *Ad-hoc Wireless Networks Architectures and Protocols*, Prentice Hall, ISBN-13: 9780131470231, Upper Saddle River, NJ 07458.

Pandey, A.; Nasir Ahmed, Md.; Kumar, N. & Gupta, P. (2006). A Hybrid Routing Scheme for Mobile Ad-hoc Networks with Mobile Backbones, *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, Vol. 4297 / 2006, pp.411-423, DOI: 10.1007/11945918_41.

Park, V. D. & Corson , M. S. (1997). A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks, *proceedings of IEEE Conference on Computer Communications (INFOCOM)*, April 1997, Vol. 3, pp. 1405-1413, Kobe, Japan.

Perkins, C. E. (2001). Ad Hoc Networking, Chapter 3, pp. 53-74, Addison-Wesley, ISBN: 0-201-30976-9.

Perkins, C. E. & Bhagwat, P. (1994). Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, *ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM)*, pp. 234-244, August 1994, London, UK.

Fuhrmann,T.; Di,P.; Kutzner,K. & Cramer, C. (2006). Pushing Chord into the Underlay: Scalable Routing for Hybrid MANETs, Universität Karlsruhe (TH), Fakultät für Informatik, *Technical Report 2006-12*, June 2006.

Rappaport, T. S. (2002). *Wireless Communications: Principles and Practice*, Prentice Hall, 2nd edition, ISBN-13: 978-0130422323, USA.

Reddy, T. B.; Karthigeyan, I.; Manoj B. S. & Siva Ram Murthy C. (2006). Quality of Service Provisioning in Ad-hoc Wireless Networks: a Survey of Issues and Solutions, *Ad Hoc Networks*, Vol. 4, No. 1., January 2006, pp. 83-124, ISSN:1570-8705, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherland.

Roche, A. ; Westphall, C. B. & Graf von Mecklenburg, P.A.C. K-S. (2002). Quality of Service for Ad-hoc Wireless Network, *Proceedings of 22nd International Conference of the Chilean (SCCC'02)*, Computer Science Society, pp. l00-l05, ISBN: 0-7695-1867-2, Copiapo, Atacama, Chile, November 2002.

Royer, E. M.; Perkins, C. E. & Das, S. R. (2000). Ad hoc On-Demand Distance Vector (AODV) Routing, on the Internet (2003): http://www.ietf.org/rfc/rfc3561.txt

Santivanez C. & Ramanathan, R. (2001). Hazy Sighted Link State Routing Protocol (Hsls), *Bbn Technical Memorandum*, No. 1301, 31 August 2001.

Sekido, M.; Takata, M.; Bandai, M. & Watanabe, T. (2005). A directional hidden terminal problem in ad hoc network MAC protocols with smart antennas and its solutions, *proceedings of IEEE Global Telecommunications Conference (GLOBECOM)*, December 2005, pp. 2583, ISBN: 0-7803-9414-3, St. Louis, MO.

Shah, S. H. & Nahrstedt, K. (2002). Predictive Location-Based QoS Routing in Mobile Ad-hoc Networks, *Proceedings of IEEE International Conference on Communications* (ICC'02), pp. l 022-1027 (Vol. 2), ISBN: 0-7803-7400-2, May 2002.

Stallings, W. (2004). Wireless Communications & Networks (2nd Edition), Prentice Hall, November 22, 2004, ISBN-10: 0131918354, ISBN-13: 978-0131918351.

Tan, Y. Y. E.; McLaughlin, S. & Laurenson, D. L. (2005). Quality of Service (QoS) Framework for Multi-rate Wireless Ad-hoc Network (MWAN), *Proceedings of The 2005 International Workshop on Wireless Ad-hoc Networks (IWWAN '05)*, London, UK, 2005.

Taneja, K. & Patel, R. B. (2007). Mobile Ad hoc Networks: Challenges and Future, *Proceedings of National Conference on Challenges & Opportunities in Information Technology (COIT)*, RIMT-IET, March 2007, Mandi Gobindgarh.

Tanenbaum, A. S. (2003). *Computer Networks*, Prentice Hall, ISBN: 0-13-066102-3, Upper Saddle River, NJ, USA.

Vidhyasanker, V.; Manoj, B.S. & Siva Ram Murthy, C. (2006). Slot Allocation Schemes for Delay-Sensitive Traffic Support in Asynchronous Wireless Mesh Networks, *The International Journal of Computer and Telecommunications Networking*, Vol. 50, No. 15, October 2006, pp. 2595-2613, ISSN:1389-1286.

Wattenhofer, R.; Li, L. ; Bahl, P. & Wang, Y. M. (2001). Distributed Topology Control for Power-Efficient Operation in Multi-Hop Wireless Ad Hoc Networks, *Proceedings of IEEE INFOCOM 2001*, pp. 1388-1397, Vol. 3, April 2001, Anchorage, AK , USA, ISBN: 0-7803-7016-3.

Wu, K. & Harms, J. (2001). QoS Support in Mobile Ad- hoc Networks, *Crossing Boundaries, an Interdisciplinary Journal*, Vol. 1, No.1, 2001.

Xiao, H.; Seah, W.G.; Lo, A. & Chua, K.C. (2000). A Flexible Quality of Service Model for Mobile Ad-hoc Networks (FQMM), *Proceedings of IEEE Vehicular Technology Conference (VTC 2000-Spring)*, Vol. 1, No.4, pp.397-413. ISBN: 0-7803-5718-3, Tokyo, Japan, May 2000.

Yang, H.; Luo, H.; Ye, F.; Lu, S. & Zhang, L. (2004). Security In Mobile Ad Hoc Networks: Challenges And Solutions, *IEEE Wireless Communications Magazine,* Sponsored by IEEE Communications Society, Vol. 11, No. 1, February 2004, pp. 38-47, ISSN: 1536-1284.

Zhang, L.; Deering, S. & Estrin, D. (1993). RSVP: A New Resource Reservation Protocol, *IEEE Network Magazine*, Vol. 7, No.5, September l993, pp. 8-18, ISSN: 0890-8044.

3

Video Communications Over Wireless Ad-Hoc Networks Using Source Coding Diversity and Multiple Paths

Yiting Liao and Jerry D. Gibson
University of California, Santa Barbara
USA

1. Introduction

Providing reliable video communications over wireless ad-hoc networks is becoming increasingly important as these networks become widely employed in military, homeland defense security, and disaster recovery applications. However, wireless ad-hoc networks have a dynamically changing topology that can cause failures of links and nodes, thus resulting in path loss. Additionally, video communications over wireless ad-hoc networks can suffer from noise and fading effects in the channel. Therefore, it is important to provide error resilience for reliable video communications over such an error-prone network.

A number of solutions have been proposed for this problem, including source coding diversity and multipath routing. Source coding diversity methods such as multiple description coding (MDC) have proven to be effective for robust video communications, especially when combined with network path diversity (Gogate et al., 2002; Mao et al., 2003; Apostolopoulos & Trott, 2004). We investigate new MDC methods combined with path diversity to enhance the error resilience of video communications over wireless ad hoc networks. The basic idea of MDC is to encode the video sequence into several descriptions for transmission over multiple paths. Each description can be independently decoded and combined with the other descriptions to provide an acceptable video quality. When more descriptions are received for reconstruction, higher video quality can be achieved. As long as all descriptions are not lost simultaneously, somewhat acceptable quality can be maintained. In order to reduce the likelihood of simultaneous loss of descriptions, different descriptions are transmitted through different paths. This is referred as MDC with path diversity, which reduces the possibility of simultaneous loss of different descriptions and enables load balancing in networks.

Many MDC algorithms have been proposed (Goyal, 2001) and they can be divided into three categories: subsampling algorithms in the temporal (Apostolopoulos, 2001), spatial (Franchi et al., 2005) or frequency domain (Reibman et al., 2001), multiple description quantization algorithms (Vaishampayan, 1993; Dumitrescu & Wu, 2009), and multiple description transform coding (Wang et al., 2001). Wang, Reibman & Lin (2005) provides a good review for MDC algorithms.

Since subsampling methods are easy to implement and compatible with different video standards, they have been the most commonly investigated MDC algorithms. These methods generally work in the spatial, temporal, or frequency domain to generate multiple descriptions, and any corresponding correlation is used to recover a lost description. One of the most popular MDC methods is multiple state video coding (MSVC) (Apostolopoulos, 2001), which temporally downsamples the video sequence and uses the correlation between adjacent frames in two descriptions to recover from frame loss. More details about MSVC are discussed in Section 2.

We use a rate-distortion optimized mode selection framework to estimate the end-to-end distortion for MSVC by considering the network conditions and multiple state recovery. The estimated end-to-end distortion is used to select the optimal coding mode to reduce error propagation due to packet losses. At the decoder, we investigate a refined error concealment method that uses correlation between different descriptions to better reconstruct the corrupted frames. This method provides better concealment for intra macroblocks (MBs) by using temporal correlation between adjacent intra frames and improves inter MB concealment by using additional reference frames for motion-compensated concealment. We present the performance of these methods over a wireless ad-hoc network with random and burst losses. The results show that the proposed method achieves improvements in objective video quality for a wide range of different burst and random packet loss rates. In addition, we use a multiuser perceptual video quality indicator to capture the distribution of distorted frames across all video frames and multiple channel uses. When combined with MSVC and path diversity techniques, our proposed methods provide better perceptual video experience for multiple network uses.

2. Multiple description video coding with path diversity

MDC is an effective approach to enhance the error resilience of video transmission over lossy networks. The general idea is to encode the video sequence into several descriptions with equal importance. Each description can be decoded independently or combined with other descriptions for reconstruction. In general, the reconstructed video achieves better video quality when more descriptions are received.

Among the many proposed MDC algorithms (Wang, Reibman & Lin, 2005), multiple state video coding (MSVC) proposed by Apostolopoulos (2001) is a very popular method since it is easy to implement and compatible with different video standards. In MSVC, the system includes a multiple state video encoder/decoder and a path diversity transmission system as shown in Fig. 1.

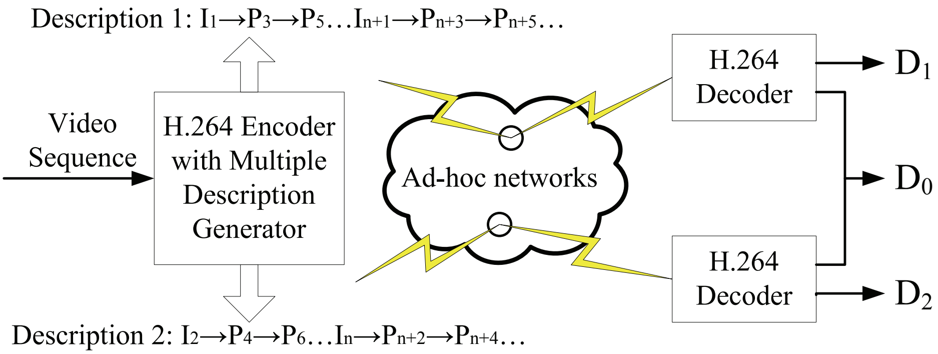

Description 1: I1→P3→P5…In+1→Pn+3→Pn+5…
Video Sequence
H.264 Encoder with Multiple Description Generator
Ad-hoc networks
H.264 Decoder
H.264 Decoder
D1
D0
D2
Description 2: I2→P4→P6…In→Pn+2→Pn+4…

Fig. 1. MSVC system architecture

At the encoder, the video sequence is first temporally down-sampled into two sub-sequences, i.e. odd frames in the original sequence are extracted as one sub-sequence and even frames as the other. The two sub-sequences are encoded separately using a H.264 video encoder (Wiegand et al., 2003) and transmitted over the networks in two different paths. At the decoder, they are decoded and interleaved to get the reconstructed video sequence.

When one description experiences packet loss, the information in the other description can be used to improve the recovery of the corrupted video segment. This is referred as multiple state recovery (Apostolopoulos, 2001). In Liao & Gibson (2008), the performance of MSVC is further improved by applying refined error concealment methods on a MB basis. For MSVC, the even and odd frames are transmitted in different paths, so burst losses in one description can be well concealed by frames in the other description and cause less damage to the reconstructed video than for single description coding (SDC), where burst losses may result in the loss of consecutive frames. However, random packet loss may cost error propagation in both descriptions and we try to alleviate the error propagation by applying optimal mode selection for MSVC.

3. MSVC with refined error concealment

When video is transmitted over wireless networks, a typical maximum transfer unit (MTU) size is around 100 bytes (Wenger, 2003), which means each frame consists of more than one packet. Therefore, a packet loss only causes some MB losses in a frame. In Apostolopoulos (2001), it is assumed that every packet loss leads to one entire frame loss and the state recovery methods introduced are on a frame basis. Therefore, we propose the refined error concealment methods on a MB basis to enhance the reconstructed video quality for MSVC (Liao & Gibson, 2008). We refer to the approach as multiple state video coding with refined error concealment (MSVC_REC).

3.1 Related work

Error concealment techniques, which have been well developed for decades (Wang & Zhu, 1998), make use of the spatial and temporal correlation between video pixel values to recover a corrupted video stream with random channel errors.

Some error concealment techniques explore the spatial redundancy of video sequences for spatial domain or transform domain reconstruction. Aign & Fazel (1995) proposed to interpolate lost pixel values from the boundary pixels of the four neighboring MBs. In addition, Kwok & Sun (1993), Zeng & Liu (1999), Zhang et al. (2004), Kung et al. (2006), Hyun et al. (2008), Luo et al. (2009) used different algorithms to detect the edges within the lost MBs and directionally interpolate the lost pixels along the edges. A more complex approach called coarse-to-fine block replenishment (CFBR) (Belfiore et al., 2002) performed the interpolation by first recovering the smooth large-scale patterns, then the large-scale structures, and finally the local edges in the lost MB. In addition to reconstruction in the spatial domain, a number of papers address the transform coefficient recovery problem by interpolating the lost coefficient from corresponding coefficients in the neighboring MBs (Hemami & Meng, 1995; Lee et al., 2004), performing optimization based on a smoothness constraint (Wang et al., 1993; Park et al., 1997), using the fuzzy logic approach to recover the high-frequency components (Lee et al., 1995), or using an iterative procedure called "projections onto convex sets" (POCS) (Sun & Kwok, 1995).

Other concealment techniques exploit temporal redundancy to estimate the lost motion information and replace the lost MB with the motion-compensated MB from one of the previous frames. Numerous approaches have been studied to recover the lost motion vectors (MVs). Haskell & Messerschmitt (1992) discussed the use of zero MV, the MV of the co-located MB in the previous frame, and the average or median MV of spatially adjacent MBs for the lost MB. The boundary matching algorithm (BMA) (Lam et al., 1993) is proposed to select the best MV among a set of candidate MVs. Zhang, Arnold & Frater (2000), Tsekeridou & Pitas (2000), Kung et al. (2006), Wu et al. (2008) presented different block matching techniques that estimate the MV based on the set of surrounding MBs of the lost MB. Salama et al. (2000) modeled the motion field as a Markov random field (MRF) and found the maximum a posteriori (MAP) estimate of the lost MV given its neighboring MVs. This method is further improved by using an adaptive Huber function in an MRF model (Zhou et al., 2005).

More recently, hybrid algorithms have been proposed to obtain better recovery. They are effective but generally introduce more complexity. Shirani et al. (2000) first obtained initial estimates of the missing MB by motion compensation or spatial interpolation and then used a MAP estimator to refine the initial estimates. Atzori et al. (2001) proposed a concealment method which replaces the lost MB using BMA and applies a mesh-based warping (MBW) to reduce the artifacts. In Chen et al. (2008), the lost MV is first estimated by a spatio-temporal BMA algorithm, and a partial differential equation (PDE) based algorithm is used to refine the reconstruction.

These error concealment techniques can be exploited to fill in lost data, however, the effectiveness of traditional methods is constrained by the fact that information available across descriptions are not exploited. Therefore, some studies propose error concealment methods targeted for different MDC methods to better utilize the information available in descriptions. Lee & Altunbasak (2002) adopted a MAP estimation approach to conceal the corrupted description in multiple description transform coding (Wang et al., 2001) and Wang, Canagarajah, Agrafiotis & Bull (2005) proposed error concealment method for a three-loop slice group MDC approach (Wang, Canagarajah & Bull, 2005). In Apostolopoulos (2001), Lu et al. (2005), and Ma et al. (2008), different concealment methods have been proposed to recover the lost frame in MSVC. However, these recovery approaches are designed to recover the loss of an entire frame, while a video bitstream transmitted over wireless networks may suffer random packet loss that causes only some MB losses. In the next two sub-sections, we introduce the MB-based error concealment methods for intra and inter MBs in MSVC respectively.

3.2 Refined intra MB concealment for MSVC

In H.264, the lost MB in an intra frame is concealed spatially based on weighted pixel interpolation (Lim et al., 2005). As shown in Fig 2, each pixel in the lost MB is estimated from the weighted sum of the boundary pixels in the adjacent MBs, where the weight is the inverse distance between the pixel to be concealed and the boundary pixel.

In other words, the lost pixel can be calculated by

$$Y(x,y) = \frac{\sum_{i=1}^{4} Y_i(16-d_i)}{\sum_{i=1}^{4}(16-d_i)} \tag{1}$$

where d_i is the distance between the concealed pixel and the boundary pixel in the neighboring MB, and Y_i is the boundary pixel value as shown in Fig. 2.

Only the correctly received neighboring MBs are used for the concealment unless less than two correctly received MBs are available. In that case, the neighboring concealed MBs are also used for the interpolation. For SDC, each group of picture (GOP) only contains one intra

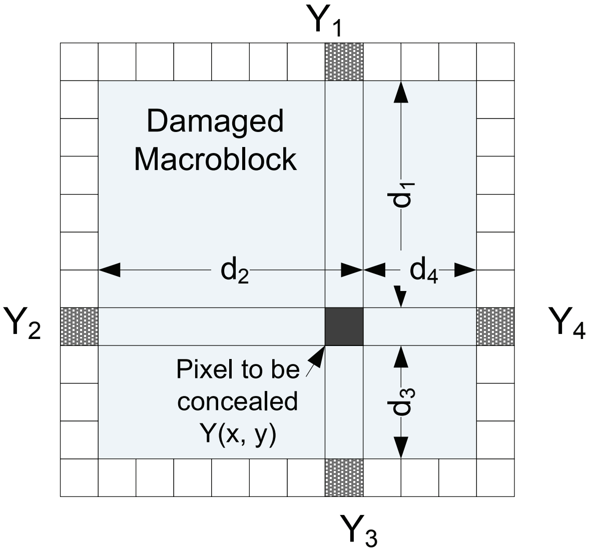


Fig. 2. Intra MB Concealment in H.264

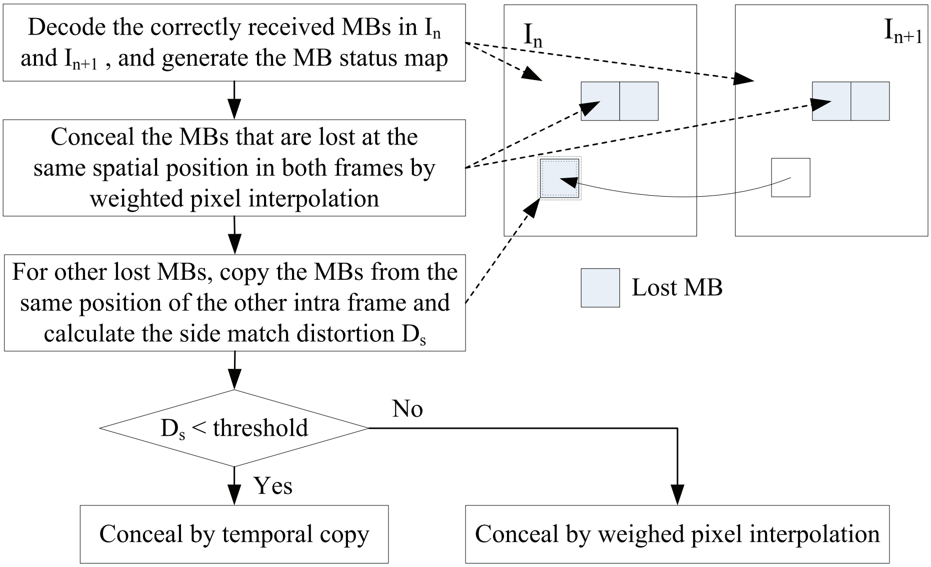

Decode the correctly received MBs in I_n and I_n+1 , and generate the MB status map
Conceal the MBs that are lost at the same spatial position in both frames by weighted pixel interpolation
For other lost MBs, copy the MBs from the same position of the other intra frame and calculate the side match distortion D_s
D_s < threshold
No
Yes
Conceal by temporal copy
Conceal by weighed pixel interpolation
I_n
I_n+1
Lost MB

Fig. 3. Error concealment in intra frame for MSVC

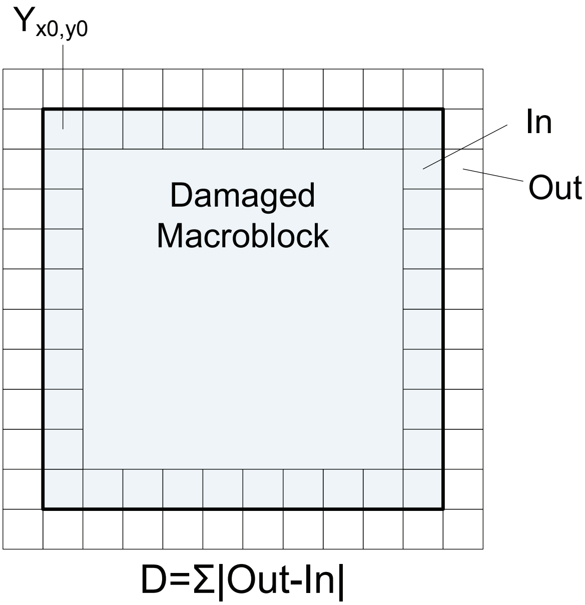


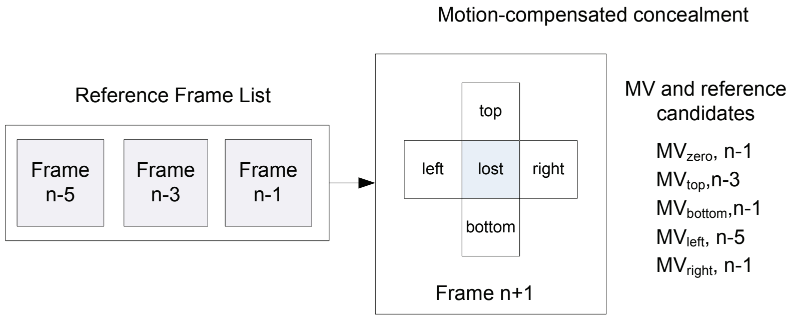
Fig. 4. Side Match Distortion

frame. In order to stop the error propagation from the previous GOP, the lost MBs in an intra frame in SDC is only concealed spatially. For MSVC, each description has an intra frame in every GOP and the two intra frames are consecutive as shown in Fig. 1. Therefore, we can apply both temporal and spatial concealment for the lost MBs in the two consecutive intra frames for MSVC.

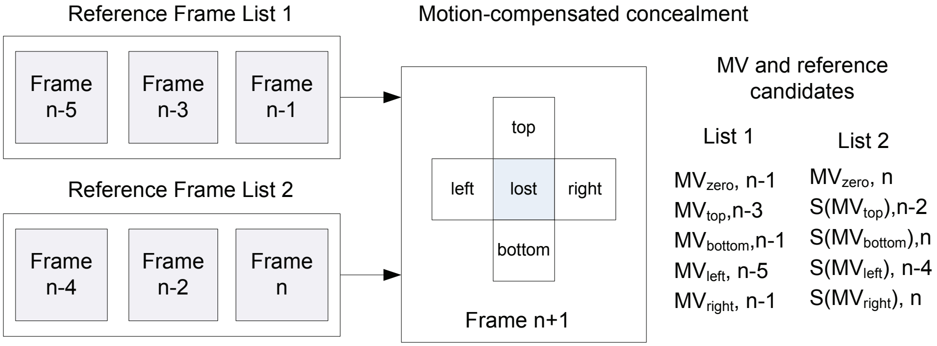
The process to conceal lost MBs in the two consecutive intra frames is shown in Fig. 3. First, the correctly received MBs in the two intra frames are decoded. Then for the MBs that are lost at the same spatial position in both intra frames, the weighted pixel interpolation method shown in Fig. 2 is applied for concealment. For other lost MBs, we copy the MBs in the corresponding position in the other intra frame and calculate the side match distortion (Lam et al., 1993) based on the correctly received neighbor MBs. As shown in Fig. 4, the side match distortion D_{SM} is the sum of absolute luminance differences between the concealed MB and neighboring MBs at the boundary,

$$\begin{aligned} D_{SM} &= \sum_{i=0}^{15} |Y_{x_0+i,y_0} - Y_{x_0+i,y_0-1}| + \sum_{i=0}^{15} |Y_{x_0,y_0+i} - Y_{x_0-1,y_0+i}| \\ &\quad + \sum_{i=0}^{15} |Y_{x_0+i,y_0+15} - Y_{x_0+i,y_0+16}| + \sum_{i=0}^{15} |Y_{x_0+15,y_0+i} - Y_{x_0+16,y_0+i}| \end{aligned} \tag{2}$$

We compare the side match distortion to a pre-defined threshold. If the side match distortion is smaller than the threshold, the temporal copy concealment is applied to conceal the lost MB. If not, the weighted pixel interpolation is used to conceal the lost MB.



(a) Inter MB concealment in H.264



(b) Refined inter MB concealment for MSVC

Fig. 5. Inter MB concealment methods

3.3 Refined inter MB concealment for MSVC

In H.264 reference software, the lost MB in the inter frame is concealed by estimating the lost motion vector from the neighbor MBs and applying motion-compensated prediction (Varsa et al., 2001). When an inter MB is lost, the motion vector of the missing MB is predicted from one of the neighbor MBs or zero motion vector as shown in Fig. 5(a). The motion vector that has the minimum side match distortion is used for motion-compensated concealment. The reference frames used to conceal the lost MB are the same as the reference frames for correctly received MBs.

In MSVC, even frames and odd frames are encoded independently, which means the correlation between the reference frame and current frame is reduced. If we only use the frames in the same description as reference frames to conceal the lost MB, it may not perform as well as using reference frames from the other description for concealment. Therefore, we propose to explore the information from both descriptions in MSVC to enhance the inter error concealment; that is, we use two reference frame lists from each description for the motion-compensated concealment. The reference list that results in better side match distortion is used as the reference to recover the lost MBs.

In order to perform motion compensated concealment for inter MBs, we need to estimate the lost motion vector and the corresponding reference frame. As shown in Fig. 5(b), instead of using only frames in the same description as reference frames, we add reference frame list 2 from the other description. Then we use MVs from four neighboring MBs and zero MV (shown in Fig. 5(b)) as MV candidates, and apply motion compensated concealment by using the corresponding reference frame from reference frame list 1. When reference frame list 2 is used for concealment, we need to scale these MV candidates accordingly because the estimated motion vector is corresponding to the reference frame in list 1. Assume one of the MV candidates is $MV_{candidate}$ with reference frame n_1, and its corresponding reference frame in reference frame list 2 is n_2, then the scaled MV for reference frame list 2 can be calculated by

$$S(MV_{candidate}) = \frac{n_c - n_2}{n_c - n_1} MV_{candidate} \tag{3}$$

where n_c is the current frame number. Similarly, motion-compensated concealment is applied based on the scaled MVs and reference frames in reference frame list 2. Finally, we choose the estimated motion vector and reference list that minimizes the side match distortion to conceal the lost inter MB.

4. Rate-distortion optimized mode Selection for MSVC

In this section, we propose to estimate the end-to-end distortion for MSVC and incorporate the estimated distortion with optimal mode selection to improve the robustness of MSVC under both random and burst losses.

4.1 Related work

In wireless networks, video transmission may suffer from packet loss due to link errors, node failures, route changes, interference and fading in the wireless channel. Packet loss can seriously degrade the received video quality, especially due to the propagated errors in the motion-compensated prediction loop. Therefore, it is challenging to provide error resilient video coding for reliable video communications over such lossy networks. A number of techniques have been proposed to increase the robustness of video communications to packet loss, such as intra/inter mode selection (Zhu & Kerofsky, 1998; Hinds et al., 1998; Cote & Kossentini, 1999; Cote et al., 2000; Zhang, Regunathan & Rose, 2000; Stockhammer et al., 2002; Eisenberg et al., 2006; Zhang et al., 2007), reference picture selection (Wiegand et al., 2000; Budagavi & Gibson, 2001), and multiple description video coding (Wang, Reibman & Lin, 2005).

Intra coding is an important technique for mitigating error propagation due to packet loss and makes the video stream more robust to errors. However, using more number of intra-coded MBs can greatly reduce the coding efficiency since an intra-coded MB generally requires more bits than an inter-coded MB. Therefore, to select the optimal intra/inter mode that can achieve the best tradeoff between error robustness and coding efficiency has become a widely addressed problem. There are some simple intra updating methods such as refreshing contiguous intra blocks periodically (Zhu & Kerofsky, 1998), or intra-coding blocks randomly (Cote & Kossentini, 1999).

A more advanced category of intra refresh algorithms estimates the end-to-end distortion due to both compression and packet loss, and incorporates mode selection with rate-distortion (RD) optimization (Hinds et al., 1998; Cote & Kossentini, 1999; Stockhammer et al., 2002; Cote et al., 2000; Zhang, Regunathan & Rose, 2000; Eisenberg et al., 2006; Zhang et al., 2007). An early work of RD-based mode selection method is proposed in (Hinds et al., 1998), in which the distortion is roughly estimated. In Cote & Kossentini (1999), the encoder considers the effects of error concealment and encodes the area that is severely affected by packet loss in the intra mode. However, the error propagation beyond one frame is ignored during the estimation procedure. In Cote et al. (2000), the authors further incorporate the distortion due to error concealment of a current block with the distortion due to error propagation from concealed blocks to optimize mode selection. One drawback of the methods proposed in Hinds et al. (1998), Cote & Kossentini (1999), and Cote et al. (2000) is that the estimated distortion at the encoder is not very accurate.

A more precise approach to estimate the end-to-end distortion is proposed by Stockhammer et al. (2002). The authors generate K copies of the channel behavior at the encoder and calculate the decoder reconstruction to estimate the expected end-to-end distortion. This approach can accurately estimate the distortion if K is large enough. However, it has extremely high computational complexity. In Zhang, Regunathan & Rose (2000), an algorithm called "Recursive Optimal Per-pixel Estimate" (ROPE) is proposed to compute the distortion by recursively calculating the first and second moments of each pixel due to compression, error concealment, and error propagation. This algorithm provides an accurate estimation of end-to-end distortion at the cost of a modest increase in computational complexity. Since the ROPE algorithm achieves substantial gains over competing methods, extensive work has been proposed based on the ROPE algorithm. For example, Eisenberg et al. (2006) estimates the variance of expected distortion by calculating the first four moments of each pixel and incorporates these moments to allocate channel resources. In Zhang et al. (2007), the overall distortion is divided into several separable distortion items to reduce the computing complexity. Heng et al. (2006) estimate the expected end-to-end distortion to select multiple description modes on a frame basis.

All of these techniques only consider a simple network condition in which an average packet loss rate is assumed. However, Liang et al. (2008) has shown that not only average packet loss rate but also the specific pattern of the loss affects the expected distortion; specifically, they prove that burst loss has a great impact on the distortion. Because of the likelihood of both random packet loss and burst losses in video communications over wireless networks, we propose a method that combines rate-distortion optimized mode selection with MSVC to enhance the error resilience of video (Liao & Gibson, 2009).

4.2 Preliminaries

Most of the video standards provide different intra and inter modes to encode a MB. For example, H.264 supports various coding modes such as Intra_16×16, Intra_4×4, Inter_SKIP, Inter_16×16, Inter_8×16, Inter_16×8, and Inter_8×8. In order to decide the best mode for each MB, a Lagrangian optimization technique is used to minimize the distortion subject to a rate constraint (Wiegand et al., 1996). That is, the coding mode that minimizes the Lagrangian cost in the following equation is chosen to encode the MB,

$$\min_{mode}(J_{MB}) = \min_{mode}(D_{MB} + \lambda_{mode} R_{MB}) \tag{4}$$

where R_{MB} denotes the bits needed for coding the MB in the specific mode, which includes the bits for the MB header, the motion vector, the reference frame, and the transformed coefficients. D_{MB} represents the distortion of the MB, and λ_{mode} is the Lagrangian multiplier for the mode decision given by Eq. (5) in H.264,

$$\lambda_{mode} = 0.85 \times 2^{(QP-12)/3} \tag{5}$$

where QP is the quantization parameter of the MB.

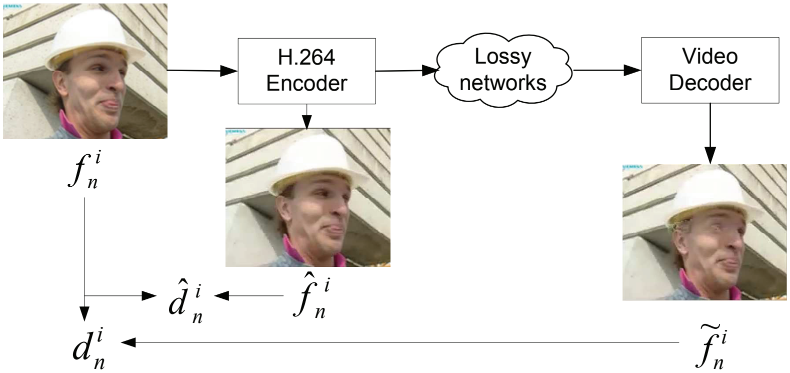


Fig. 6. Video Coding and Transmission System

To determine the optimal mode for each MB, we need to estimate the distortion of the MB. In an H. 264 video encoder, the distortion is defined as the mean square error between the original video pixel value f_n^i and the encoded pixel value $\hat{f}_n^i$ as shown in Fig. 6. The coding mode chosen to encode the video is optimal for the compressed video without losses. However, for the video transmitted over lossy networks, the decoded pixel value $\tilde{f}_n^i$ suffers from packet losses and is not equal to $\hat{f}_n^i$. To select the optimal coding mode for the video that suffers losses, the encoder needs to estimate the distortion d_n^i between the original pixel value f_n^i and the decoder-reconstructed pixel value $\tilde{f}_n^i$ as shown in Fig. 6.

Table 1 defines the notations used in the derivation of the distortion. The distortion of each MB is defined as the sum of the end-to-end distortion of the pixels in the MB,

$$D_{MB} = \sum_{i \in MB} d_n^i \tag{6}$$

In the next section, we derive an approach to estimate the distortion d_n^i for MSVC by considering the quantization, packet losses, error propagation, and multiple state recovery.

4.3 Estimation of end-to-end distortion for MSVC

From Section 2, we know that MSVC transmits two independently decodable descriptions over two different paths to reduce the loss of consecutive frames. Burst losses in one description only cause the loss of consecutive odd (even) frames, which can be well concealed by the even (odd) frames in the other description. On the other hand, burst losses can cause severe degradation to all the subsequent frames in SDC. Therefore, MSVC is more robust to burst losses than SDC. However, when MSVC experiences random packet loss, the distortion

	Definitions
d_n^i	End-to-end distortion of pixel i in frame n
f_n^i	Original value of pixel i in frame n
$\hat{f}_n^i$	Encoder-reconstructed value of pixel i in frame n
$\tilde{f}_n^i$	Decoder-reconstructed value of pixel i in frame n (after error concealment)
$\hat{r}_n^i$	Quantized residue of pixel i in frame n (Inter mode)

Table 1. Notation

due to random loss not only propagates to subsequent frames in the same description, but may also affect frames in the other description because of multiple state recovery. In order to mitigate the error propagation due to random loss in MSVC, we propose a rate-distortion optimized mode selection method for MSVC, which adaptively encodes MBs in different modes to reduce the impact of error propagation.

The idea is similar to the ROPE method, except that MSVC uses multiple state recovery to conceal the error and the encoder needs to consider this during the estimation process. We assume that the refined error concealment methods on a MB basis are applied (Liao & Gibson, 2008). We estimate the first and second moments of $\tilde{f}_n^i$ by considering the packet loss rate p, and the multiple state recovery, and calculate the expected end-to-end distortion for each MB. When applying RD-based mode selection, the proposed method can better recover from random loss.

The expected end-to-end distortion for the pixel f_n^i is given by

$$d_n^i = E[(f_n^i - \tilde{f}_n^i)^2] = (f_n^i)^2 - 2f_n^i E[\tilde{f}_n^i] + E[(\tilde{f}_n^i)^2] \tag{7}$$

Notice that the value of $\tilde{f}_n^i$ is a random variable at the encoder. In order to estimate the expected distortion d_n^i at the encoder, we need to calculate the first and second moments of $\tilde{f}_n^i$ for an intra and an inter MB separately for MSVC.

4.3.1 Pixel in an intra-coded MB

To compute the first and second moments of $\tilde{f}_n^i$ for an Intra MB, we need to consider the following scenarios:

1. The packet for f_n^i is correctly received with probability $1 - p$ and thus we have $\tilde{f}_n^i = \hat{f}_n^i$.
2. The packet for f_n^i is lost and the neighbor group of blocks (GOB) is received with probability $p(1-p)$. In this case, we estimate the motion vector of lost pixel from one of the available neighbor MBs and use motion-compensated concealment to recover the lost pixel. We choose one frame as the reference from each description and get two reconstructed values $\tilde{f}_{n-1}^{j_1}$ and $\tilde{f}_{n-2}^{j_2}$. Then pixel $\tilde{f}_n^i$ is recovered from $\tilde{f}_{n-1}^{j_1}$ or $\tilde{f}_{n-2}^{j_2}$ depending on which reconstructed value is closer to $\hat{f}_n^i$, i.e. $\tilde{f}_n^i = \tilde{f}_{n-m}^{j_m}$, where $m = \arg\min_{x\in\{1,2\}}(\tilde{f}_{n-x}^{j_x} - \hat{f}_n^i)^2$.
3. The packet for f_n^i and the neighbor GOB are both lost with probability p^2. Then either $\tilde{f}_{n-1}^i$ or $\tilde{f}_{n-2}^i$ is used to conceal $\tilde{f}_n^i$. Thus, $\tilde{f}_n^i = \tilde{f}_{n-k}^i$, where $k = \arg\min_{x\in\{1,2\}}(\tilde{f}_{n-x}^i - \hat{f}_n^i)^2$.

Based on the above cases, we can calculate the first and second moments of $\tilde{f}_n^i$ in an intra MB by Eqs. (8) and (9),

$$E[\tilde{f}_n^i] = (1-p)(\hat{f}_n^i) + p(1-p)E[\tilde{f}_{n-m}^{j_m}] + p^2 E[\tilde{f}_{n-k}^i] \tag{8}$$

$$E[(\tilde{f}_n^i)^2] = (1-p)(\hat{f}_n^i)^2 + p(1-p)E[(\tilde{f}_{n-m}^{j_m})^2] + p^2 E[(\tilde{f}_{n-k}^i)^2] \tag{9}$$

$$\text{where} \quad m = \arg\min_{x\in\{1,2\}}(E[\tilde{f}_{n-x}^{j_x}] - \hat{f}_n^i)^2, \quad \text{and} \quad k = \arg\min_{x\in\{1,2\}}(E[\tilde{f}_{n-x}^i] - \hat{f}_n^i)^2$$

4.3.2 Pixel in an inter-coded MB

For MSVC, the odd frame is predicted from previous odd frames and the even frame is predicted from previous even frames. Therefore, the quantized residue $\hat{r}_n^i = \hat{f}_n^i - \hat{f}_{n-2}^j$ for MSVC, where pixel i in frame n is predicted from pixel j in frame $n-2$. Assume that $j_m(m=1,2)$ is the pixel corresponding to the estimated concealment motion vector for pixel i in frame $n-m$. Then we can calculate the first and second moments of $\tilde{f}_n^i$ according to the three cases similar to those in Section 4.3.1,

$$E[\tilde{f}_n^i] = (1-p)(\hat{r}_n^i + E[\tilde{f}_{n-2}^j]) + p(1-p)E[\tilde{f}_{n-m}^{j_m}] + p^2E[\tilde{f}_{n-k}^i] \quad (10)$$

$$E[(\tilde{f}_n^i)^2] = (1-p)E[(\hat{r}_n^i + \tilde{f}_{n-2}^j)^2] + p(1-p)E[(\tilde{f}_{n-m}^{j_m})^2] + p^2E[(\tilde{f}_{n-k}^i)^2] \quad (11)$$

$$\text{where} \quad m = \underset{x\in\{1,2\}}{\arg\min}(E[\tilde{f}_{n-x}^{j_x}] - \hat{f}_n^i)^2, \quad \text{and} \quad k = \underset{x\in\{1,2\}}{\arg\min}(E[\tilde{f}_{n-x}^i] - \hat{f}_n^i)^2$$

5. Experimental results

5.1 Simulation setup

In this section, we discuss the model we used to simulate the packet losses in wireless networks, the performance metrics to evaluate performance of different methods, and the parameters and methods to encode the video for comparison.

5.1.1 Packet loss model

In wireless networks, packet loss may occur due to numerous reasons, including link/node failures, route changes, and bit errors. These factors can cause both random packet loss and burst losses over the network. To investigate the video communications over such lossy networks, we introduce a packet loss model that captures packet loss features in the network. As shown in Fig. 7, this model considers both random packet loss and burst losses during transmission and can be used to generate different loss patterns over the wireless network.

In this model, time is divided into Δt intervals and k frames are transmitted during an interval. Each interval may be either in a good state with probability $(1-p_b)$ or in a down state with probability p_b, which is independent and identically distributed. The packets transmitted in a down state are all lost while the packets transmitted in the good state may suffer from a random packet loss. Therefore, the packet loss model can be determined by three parameters:

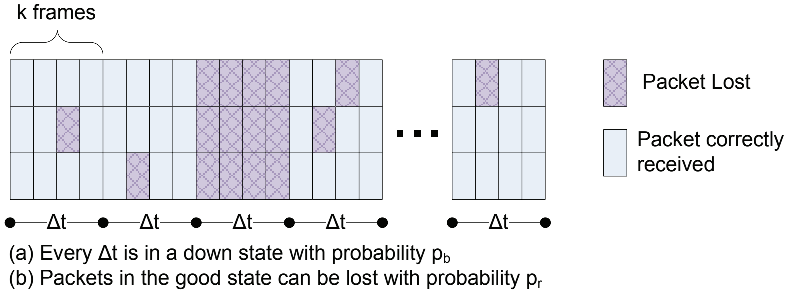

k frames
Packet Lost
Packet correctly received
Δt
Δt
Δt
Δt
Δt
(a) Every Δt is in a down state with probability p_b
(b) Packets in the good state can be lost with probability p_r

Fig. 7. Packet Loss Model

the burst loss rate p_b, the burst length k (frames), and the random packet loss rate p_r in a good state. The total packet loss rate p in the networks can be calculated by

$$p = p_b + (1 - p_b)p_r = p_b + p_r - p_b p_r \tag{12}$$

5.1.2 Performance metrics

To analyze the performance of the decoded video sequences, we use the average peak signal-to-noise ratio (PSNR) of all frames over all realizations to evaluate the objective video quality, because it is the most widely used objective video quality metric. PSNR represents the mean squared errors (MSE) of the distorted videos and is defined by

$$PSNR = 10 \cdot \log_{10} \frac{255^2}{MSE} \tag{13}$$

where MSE is the mean square error between the original pixel and the distorted pixel. However, due to the non-linear behavior of human visual system, video sequences with close average PSNR may reveal different perceptual video quality for human viewers. Therefore, we also introduce $PSNR_{r,f}$ proposed by Hu et al. (2008) to evaluate the perceptual video quality for multiple channel uses. $PSNR_{r,f}$ is defined as the PSNR achieved by f% of the frames for r% of realizations, which shows the video quality guaranteed for r% of realizations among f% frames. The definition of $PSNR_{r,f}$ can be written as

$$PSNR_{r,f} = \arg_x P_{real}(P_{frame}(PSNR > x) \geq f\%) \geq r\%) \tag{14}$$

Here, $P_{frame}(PSNR > x)$ is the percentage of frames that have PSNR higher than x in a realization and $P_{real}(\Omega)$ is the percentage of realizations that satisfy the condition Ω. For example, $PSNR_{r=80\%,f=90\%} = 35$ dB means that there are 80% of the realizations having 90% of frames with PSNR higher than 35 dB. We use $PSNR_{r,f}$ as a multiuser perceptual video quality indicator because of two reasons. First, $PSNR_{r,f}$ captures the lowest PSNR achieved by f% of the frames in each realization, which can be used to measure the perceptual video quality of a single realization due to two observations in video quality assessment (Wang et al., 2003): (1) The bad-quality frames in a video dominates users' experience with the video; (2) For PSNRs higher than a certain threshold, increasing PSNR does not help to enhance the perceptual video quality. Unlike average PSNR that treats every frame equally, $PSNR_{r,f}$ captures the performance loss due to damaged frames in a video sequence (f%). Second, due to the time-variant network conditions, multiple users or a user in multiple uses may have different experience. $PSNR_{r,f}$ can capture the performance experienced by a user in multiple uses (r%), or alternatively, it indicates the percentage of video users that experience a specific video quality.

5.1.3 Compared schemes and simulation settings

We implement our proposed methods by modifying H.264 reference software JM13.2. The temporal copy method is used to estimate the end-to-end distortion for mode selection. To evaluate the performance of the refined error concealment method for MSVC, we compare the following three coding schemes:

1. **SDC**: The video sequence is coded into a single description and transmitted over one path over the network.
2. **MSVC**: The video sequence is coded into two descriptions using MSVC and transmitted over two independent paths over the network.

3. **MSVC_REC**: The video sequence is coded and transmitted the same as MSVC, while the refined error concealment method introduced in Section 3 is applied to decode the corrupted video.

Furthermore, we investigate the performance of MSVC with the optimal mode selection and refined error concealment methods by comparing the following three coding schemes:

1. **SDC_ROPE**: The video sequence is coded into a single description with ROPE proposed in Zhang, Regunathan & Rose (2000) and transmitted over one path over the network.
2. **MSVC_REC**: The video sequence is coded into two descriptions using MSVC and transmitted over two independent paths over the network. At the decoder, refined error concealment is applied to decode the corrupted video.
3. **MSVC_OMS**: The video sequence is coded into two descriptions using MSVC with the optimal mode selection introduced in Section 4. Then the encoded bitstream is transmitted over two paths and decoded using the refined error concealment method for MSVC.

We evaluate six video sequences including Carphone, Claire, Foreman, Hall-monitor, Mother-daughter, and News. Each sequence consists of 300 frames at QCIF format. The sequences are encoded as IPPP format with $GOP = 30$ at 30 fps and each frame is packetized to 4 RTP packets. We examine the video performance under various bitrates (128 - 384 kbps). The video packet size is defined by

$$l = \frac{r}{8f \cdot n}(bytes/packet) \tag{15}$$

where r is the bitrate of the encoded video, f denotes the frame rate, and n represents the number of packets per frame. Based on the above settings, the video packet sizes under different bitrates are in the range of 133 - 400 bytes, which are reasonable packet sizes for wireless transmission (Wenger, 2003). The packet loss model in Section 5.1.1 is used to simulate the random and burst losses in wireless networks and we simulate each video sequence over 500 different realizations for each network setting.

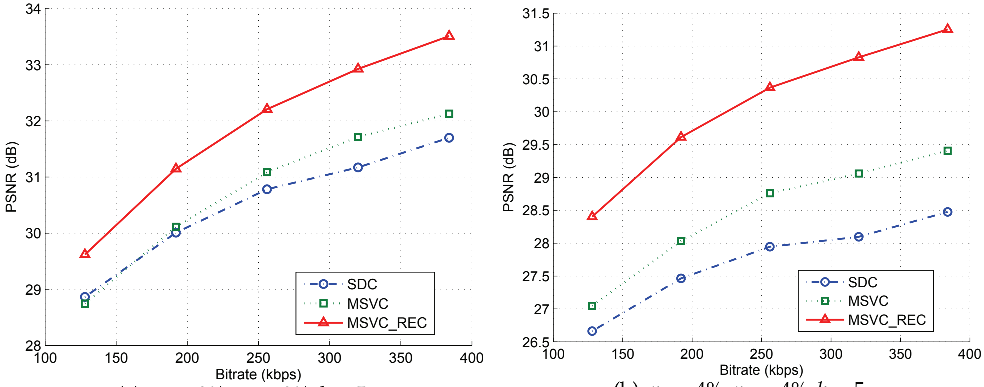
5.2 Performance evaluation

5.2.1 Refined error concealment for MSVC

In Fig. 8, we show the PSNR performance under different bitrates for Foreman sequence. Figure 8(a) compares SDC, MSVC, and MSVC_REC under $p_r = 2\%$, $p_b = 2\%$, and $k = 5$. Under this network condition, SDC and MSVC have similar PSNR performance at low bitrates. This is because even though the usage of multiple descriptions and path diversity enhances the robustness of MSVC, the decreased correlation between adjacent frames in each description reduces its coding efficiency. Therefore, MSVC outperforms SDC at a higher packet loss rate as shown in Fig. 8(b). Compared to SDC and MSVC, our proposed MSVC_REC achieves consistent gains under different bitrates and network conditions. The gains achieved by MSVC_REC are in the range of 0.8-1.8 dB at a 4% packet loss rate (Fig. 8(a)) and in the range of 1.4-2.8 dB at a 8% packet loss rate (Fig. 8(b)). This shows that our proposed method can effectively improve the error concealment for MSVC by utilizing the information from both descriptions to conceal the packet loss on a MB basis.

In Fig. 9(a), the random loss rate is fixed at 1% and the PSNR performance of SDC, MSVC, and MSVC_REC under different burst loss rates is shown.

We notice that the performance of SDC drops more quickly than MSVC and MSVC_REC as the burst loss rate increases, which means SDC is more vulnerable to burst losses. That is



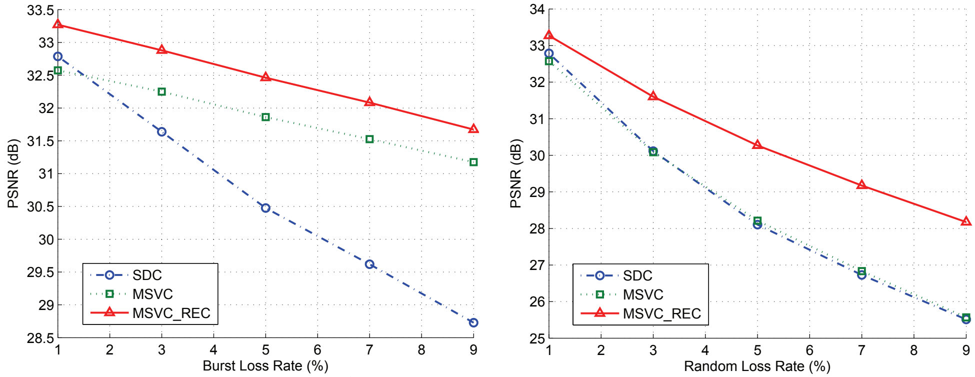
(a) $p_r = 2\%, p_b = 2\%, k = 5$ (b) $p_r = 4\%, p_b = 4\%, k = 5$

Fig. 8. Average PSNR vs Bitrate for SDC, MSVC, and MSVC_REC, Foreman sequence at 30 fps

because even if one description for MSVC is totally lost, the other description can still be correctly decoded and used to recover the lost description. Figure 9(a) shows that MSVC and MSVC_REC are more effective to combat burst losses than SDC. Meanwhile, MSVC_REC has higher PSNR than MSVC of about 0.6 dB under various burst loss rates.

Figure 9(b) investigates the PSNR performance of SDC, MSVC, and MSVC_REC under different random loss rates with a fixed burst loss rate at 1%. We see that MSVC_REC achieves up to 2.6 dB gains in PSNR and the performance gains of MSVC_REC increase as the random packet loss rate increases. This is because with the refined error concealment methods, MSVC_REC better exploits the correctly received information from both descriptions to conceal the random lost MBs.

In Table 2, we compare SDC, MSVC, and MSVC_REC for different video sequences. The results demonstrate that MSVC_REC can provide better reconstructed video quality for video communications over lossy networks.



(a) $p_r = 1\%, p_b = 1\% - 9\%, k = 5$ (b) $p_r = 1\% - 9\%, p_b = 1\%, k = 5$

Fig. 9. Average PSNR vs Packet Loss Rate for SDC, MSVC, and MSVC_REC, Foreman sequence at 30 fps, 256 kbps

Sequence	$p_b = 2\%, p_r = 2\%, k = 5$			$p_b = 4\%, p_r = 4\%, k = 5$		
	SDC	MSVC	MSVC_REC	SDC	MSVC	MSVC_REC
Carphone	31.11	31.93	33.21	27.91	29.21	31.02
Claire	40.37	39.84	42.31	37.03	36.61	40.42
Foreman	30.78	31.09	32.21	27.95	28.76	30.37
Hall-monitor	36.68	36.68	38.46	33.36	33.61	36.52
Mother-daughter	38.11	37.69	39.20	35.64	35.42	37.89
News	36.32	35.77	37.08	32.77	33.15	35.20

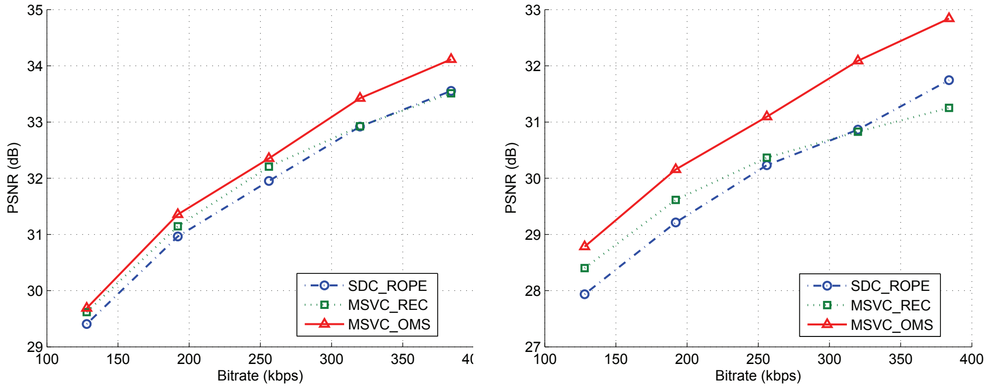
Table 2. Average PSNR (dB) for different video sequences, 30 fps, 256 kbps

5.2.2 Rate-distortion optimized mode selection for MSVC

In this section, we evaluate the performance of our optimal mode selection method for MSVC. First, we show the average PSNRs of SDC_ROPE, MSVC_REC, and MSVC_OMS under different bitrates for two sets of network conditions in Fig. 10.

We see that MSVC_OMS achieves more gain than MSVC_REC at a higher packet loss rate as shown in Fig. 10(b), since MSVC_OMS adaptively selects the optimal coding mode based on the packet loss rate in the network. Figure 10 also shows that MSVC_OMS has the best PSNR performance across different bitrates.

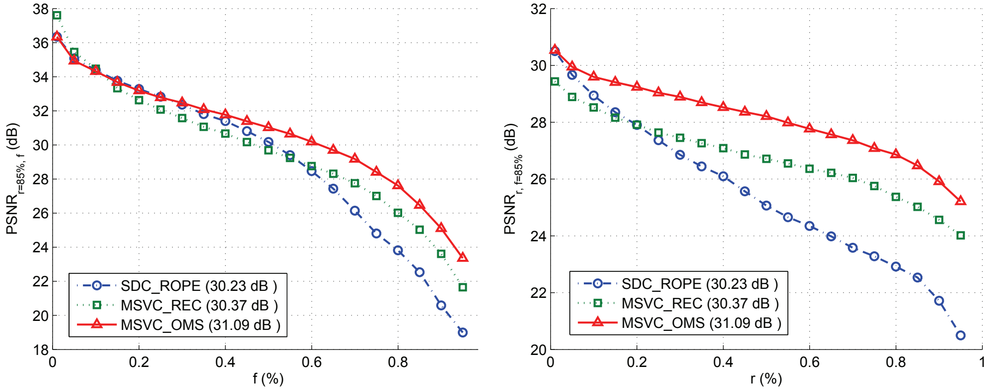
We have shown the objective performance of three methods in Fig. 10. Now we look into a case to investigate the perceptual video quality of the three methods. Figure 11 shows $PSNR_{r,f}$ of SDC_ROPE, and MSVC_REC, and MSVC_OMS for Foreman sequence under network condition ($p_r = 4\%, p_b = 4\%, k = 5$). The average PSNR of these three methods are 30.23 dB, 30.37 dB, and 31.09 dB respectively. Figure 11(a) presents the $PSNR_{r,f}$ values with a fixed $r = 85\%$. Compared to SDC_ROPE and MSVC_REC, MSVC_OMS has the fewest number of low-quality frames in 85% of the realizations. For example, as shown in Fig. 11(a), about 25% of frames in 85% of the realizations for SDC_ROPE have a PSNR lower than 25 dB, while fewer than 10% of frames in 85% of the realizations for MSVC_OMS achieve a PSNR lower than 25 dB.



(a) $p_r = 2\%, p_b = 2\%, k = 5$

(b) $p_r = 4\%, p_b = 4\%, k = 5$

Fig. 10. Average PSNR vs Bitrate for SDC_ROPE, MSVC_REC, and MSVC_OMS, Foreman sequence at 30 fps



(a) $PSNR_{r=85\%,f}$ (b) $PSNR_{r,f=85\%}$

Fig. 11. $PSNR_{r,f}$ for Foreman sequence at 30 fps, 256 kbps, $p_r = 4\%$, $p_b = 4\%$, $k = 5$

Figure 11(b) plots $PSNR_{r,f}$ of SDC_ROPE, MSVC_REC, and MSVC_OMS with fixed $f = 85\%$. In the figure, we see that MSVC_OMS has the highest $PSNR_{r,f=85\%}$ under most values of r. This means that MSVC_OMS can guarantee a higher PSNR than SDC_ROPE and MSVC_REC for 85% of frames in all of the realizations. For example, the PSNRs guaranteed for 85% of the frames in 85% of the realizations for SDC_ROPE, MSVC_REC, and MSVC_OMS are 22.53 dB, 25.02 dB, and 26.47 dB respectively. This indicates that MSVC_OMS guarantees a higher video quality for a user in multiple channel uses (r%) or provides better video experience for multiple users in the network.

Table 3 presents the average PSNR and $PSNR_{r=85\%,f=85\%}$ results for SDC_ROPE, MSVC_REC, and MSVC_OMS for different video sequences. The results show that MSVC_OMS not only provides best objective video quality among the three methods but also has the best perceptual performance for multiple users.

6. Conclusion

We explore approaches to combine source coding diversity with path diversity to support video communications over wireless ad-hoc networks. There are several benefits of transmitting multiple independent source descriptions over different paths:

1. Traffic dispersion and load balancing: Sending a video bitstream across multiple paths reduces the per-path bandwidth, thus relieves congestion at hotspots and enhances network utilization.
2. Reduced burst losses: Distributing video packets through multiple paths increases the time interval to send video packets on each path, therefore, for a given duration of congestion, fewer packets are lost.
3. Improved error resilience: As long as the descriptions do not have simultaneous losses, the system can reconstruct the video with acceptable video quality.

In this article, we focus on a source coding diversity method called MSVC, because this method is easy to implement and compatible with different video standards. We propose methods at both the encoder and decoder sides to improve the error robustness of MSVC. The

Sequence	$p_b = 2\%, p_r = 2\%, k = 5$					
	SDC_ROPE	MSVC_REC	MSVC_OMS	SDC_ROPE	MSVC_REC	MSVC_OMS
	Average PSNR (dB)			$PSNR_{r=85\%,f=85\%}$		
Carphone	33.23	33.21	33.88	23.97	27.86	28.50
Claire	41.37	42.31	42.66	27.22	37.31	38.84
Foreman	31.95	32.21	32.35	24.29	27.99	28.64
Hall-monitor	38.40	38.46	39.09	25.76	32.58	33.77
Mother-daughter	38.90	39.20	39.45	30.69	35.91	36.82
News	37.84	37.08	37.31	25.21	30.58	31.39
	$p_b = 4\%, p_r = 4\%, k = 5$					
	Average PSNR (dB)			$PSNR_{r=85\%,f=85\%}$		
Carphone	31.25	31.02	32.38	21.63	25.17	26.31
Claire	38.68	40.42	41.17	24.40	32.88	35.28
Foreman	30.23	30.37	31.09	22.53	25.02	26.47
Hall-monitor	35.66	36.52	37.72	22.17	29.21	29.60
Mother-daughter	37.25	37.89	38.38	26.82	33.18	34.40
News	35.26	35.20	35.80	22.67	26.66	26.98

Table 3. Average PSNR and $PSNR_{r=85\%,f=85\%}$ for different video sequences, 30 fps, 256 kbps

encoder estimates the end-to-end distortion for MSVC by considering the network conditions, error propagation, and error concealment, then applies RD-based mode selection to select the optimal coding mode. This helps to alleviate error propagation due to packet loss in MSVC. The decoder applies refined error concealment methods for MSVC by using the information available across two descriptions to recover lost MBs.

We study the performance of our proposed methods over a wireless network with both random and burst losses. The results show that our proposed methods have PSNR gain between 0.3 dB and 2.9 dB under various bitrates and network conditions for different video sequences. We also use a multiuser perceptual video quality indicator $PSNR_{r,f}$ to evaluate the perceptual performance for multiple users. The results demonstrate that our proposed methods provide a better video experience for multiple users in the network.

7. References

Aign, S. & Fazel, K. (1995). Temporal and spatial error concealment techniques for hierarchical MPEG-2 video codec, *IEEE International Conference on Communications*, Vol. 3, pp. 1778 –1783.

Apostolopoulos, J. G. (2001). Reliable video communication over lossy packet networks using multiple state encoding and path diversity, *SPIE Proceedings on Visual Communications and Image Processing*, Vol. 4310, pp. 392–409.

Apostolopoulos, J. G. & Trott, M. D. (2004). Path diversity for enhanced media streaming, *IEEE Communications Magazine* 42(8): 80–87.

Atzori, L., De Natale, F. & Perra, C. (2001). A spatio-temporal concealment technique using boundary matching algorithm and mesh-based warping (BMA-MBW), *IEEE Transactions on Multimedia* 3(3): 326–338.

Belfiore, S., Crisa, L., Grangetto, M., Magli, E. & Olmo, G. (2002). Robust and edge-preserving video error concealment by coarse-to-fine block replenishment, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 4, pp. IV–3281–3284.

Budagavi, M. & Gibson, J. D. (2001). Multiframe video coding for improved performance over wireless channels, *IEEE Transactions on Image Processing* 10(2): 252 –265.

Chen, Y., Hu, Y., Au, O. C., Li, H. & Chen, C. W. (2008). Video error concealment using spatio-temporal boundary matching and partial differential equation, *IEEE Transactions on Multimedia* 10(1): 2–15.

Cote, G. & Kossentini, F. (1999). Optimal intra coding of blocks for robust video communication over the internet, *Signal Processing: Image Communication*, Vol. 15, pp. 25–34.

Cote, G., Shirani, S. & Kossentini, F. (2000). Optimal mode selection and synchronization for robust video communications over error-prone networks, *IEEE Journal on Selected Areas in Communications* 18(6): 952–965.

Dumitrescu, S. & Wu, X. (2009). On properties of locally optimal multiple description scalar quantizers with convex cells, *IEEE Transactions on Information Theory* 55(12): 5591–5606.

Eisenberg, Y., Zhai, F., Pappas, T. N., Berry, R. & Katsaggelos, A. K. (2006). VAPOR: variance-aware per-pixel optimal resource allocation, *Image Processing, IEEE Transactions on* 15(2): 289–299.

Franchi, N., Fumagalli, M., Lancini, R. & Tubaro, S. (2005). Multiple description video coding for scalable and robust transmission over IP, *IEEE Transactions on Circuits and Systems for Video Technology* 15(3): 321–334.

Gogate, N., Chung, D.-M., Panwar, S. S. & Wang, Y. (2002). Supporting image and video applications in a multihop radio environment using path diversity and multiple description coding, *IEEE Transactions on Circuits and Systems for Video Technology* 12(9): 777–792.

Goyal, V. K. (2001). Multiple description coding: compression meets the network, *IEEE Signal Processing Magazine* 18(5): 74–93.

Haskell, P. & Messerschmitt, D. (1992). Resynchronization of motion compensated video affected by ATM cell loss, *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 3, pp. 545–548.

Hemami, S. S. & Meng, T. H.-Y. (1995). Transform coded image reconstruction exploiting interblock correlation, *IEEE Transactions on Image Processing* 4(7): 1023–1027.

Heng, B. A., Apostolopoulos, J. G. & Lim, J. S. (2006). End-to-end rate-distortion optimized MD mode selection for multiple description video coding, *EURASIP Journal on Applied Signal Processing* pp. 261–261.

Hinds, R. O., Pappas, T. N. & Lim, J. S. (1998). Joint block-based video source/channel coding for packet-switched networks, *Visual Communications and Image Processing*, Vol. 3309, SPIE, pp. 124–133.

Hu, J., Choudhury, S. & Gibson, J. D. (2008). Video capacity of WLANs with a multiuser perceptual quality constraint, *IEEE Transactions on Multimedia* 10(8): 1465–1478.

Hyun, S. H., Kim, S. S., Kim, B. C., Eom, I. K. & Kim, Y. S. (2008). Efficient directional interpolation for block recovery using difference values of border pixels, *Congress on Image and Signal Processing*, Vol. 3, pp. 565–568.

Kung, W.-Y., Kim, C.-S. & Kuo, C.-C. (2006). Spatial and temporal error concealment techniques for video transmission over noisy channels, *IEEE Transactions on Circuits* *and Systems for Video Technology* 16(7): 789–803.

Kwok, W. & Sun, H. (1993). Multi-directional interpolation for spatial error concealment, *IEEE Transactions on Consumer Electronics* 39(3): 455–460.

Lam, W. M., Reibman, A. R. & Liu, B. (1993). Recovery of lost or erroneously received motion vectors, *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 5, pp. 417–420.

Lee, H. Y., Eom, I. K. & Kim, Y. S. (2004). Error concealment using directional coefficient mask and difference of DC, *30th Annual Conference of IEEE Industrial Electronics Society*, Vol. 3, pp. 2086–2091.

Lee, X., Zhang, Y.-Q. & Leon-Garcia, A. (1995). Information loss recovery for block-based image coding techniques-a fuzzy logic approach, *IEEE Transactions on Image Processing* 4(3): 259–273.

Lee, Y.-C. & Altunbasak, Y. (2002). A collaborative multiple description transform coding and statistical error concealment method for error resilient video streaming over noisy channels, *Proceedings of IEEE International Conference onAcoustics, Speech, and Signal Processing*, Vol. 2, pp. 2077–2080.

Liang, Y. J., Apostolopoulos, J. G. & Girod, B. (2008). Analysis of packet loss for compressed video: Effect of burst losses and correlation between error frames, *IEEE Transactions on Circuits and Systems for Video Technology* 18(7): 861–874.

Liao, Y. & Gibson, J. D. (2008). Refined error concealment for multiple state video coding over ad hoc networks, *Proceedings of the 42nd Asilomar Conference on Signals, Systems and Computers*, pp. 2243–2247.

Liao, Y. & Gibson, J. D. (2009). Rate-distortion based mode selection for video coding over wireless networkswith burst losses, *17th International Packet Video Workshop*, pp. 1–10.

Lim, K. P., Sullivan, G. & Wiegand, T. (2005). Text description of joint model reference encoding methods and decoding concealment methods, *Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, JVT-O079* 6.

Lu, Y., Zhou, R., Cui, H. & Tang, K. (2005). Bi-directional entire frame recovery in MDC video streaming, *IEEE International Symposium on Communications and Information Technology*, Vol. 2, pp. 1058–1061.

Luo, H.-y., Gan, Z.-l. & Zhu, X.-c. (2009). Content-adaptive interpolation for spatial error concealment, *Second International Conference on Information and Computing Science*, Vol. 1, pp. 265–268.

Ma, M., Au, O. C., Guo, L., Chan, S.-H. G. & Wong, P. H. W. (2008). Error concealment for frame losses in MDC, *IEEE Transactions on Multimedia* 10(8): 1638–1647.

Mao, S., Lin, S., Panwar, S. S., Wang, Y. & Celebi, E. (2003). Video transport over ad hoc networks: multistream coding with multipath transport, *IEEE Journal on Selected Areas in Communications* 21(10): 1721–1737.

Park, J. W., Kim, J. W. & Lee, S. U. (1997). DCT coefficients recovery-based error concealment technique and its application to the MPEG-2 bit stream error, *IEEE Transactions on Circuits and Systems for Video Technology* 7(6): 845–854.

Reibman, A., Jafarkhani, H., Wang, Y. & Orchard, M. (2001). Multiple description video using rate-distortion splitting, *Proceedings of International Conference on Image Processing*, Vol. 1, pp. 978–981.

Salama, P., Shroff, N. B. & Delp, E. J. (2000). Error concealment in MPEG video streams over ATM networks, *IEEE Journal on Selected Areas in Communications* 18(6): 1129–1144.

Shirani, S., Kossentini, F. & Ward, R. (2000). A concealment method for video communications in an error-prone environment, *IEEE Journal on Selected Areas in Communications* 18(6): 1122–1128.

Stockhammer, T., Kontopodis, D. & Wieg, T. (2002). Rate-distortion optimization for JVT/H.26L video coding in packet loss environment, *Proceedings of International Packet Video Workshop*.

Sun, H. & Kwok, W. (1995). Concealment of damaged block transform coded images using projections onto convex sets, *IEEE Transactions on Image Processing* 4(4): 470–477.

Tsekeridou, S. & Pitas, I. (2000). MPEG-2 error concealment based on block-matching principles, *IEEE Transactions on Circuits and Systems for Video Technology* 10(4): 646 –658.

Vaishampayan, V. (1993). Design of multiple description scalar quantizers, *IEEE Transactions on Information Theory* 39(3): 821–834.

Varsa, V., Hannuksela, M. & Wang, Y. (2001). Non-normative error concealment algorithms, *ITU-T, VCEG-N62* .

Wang, D., Canagarajah, N., Agrafiotis, D. & Bull, D. (2005). Error concealment for slice group based multiple description video coding, *IEEE International Conference on Image Processing*, Vol. 1, pp. I–769–772.

Wang, D., Canagarajah, N. & Bull, D. (2005). Slice group based multiple description video coding with three motion compensation loops, *IEEE International Symposium on Circuits and Systems*, pp. 960–963.

Wang, Y., Orchard, M. T., Vaishampayan, V. & Reibman, A. R. (2001). Multiple description coding using pairwise correlating transforms, *IEEE Transactions on Image Processing* 10(3): 351–366.

Wang, Y., Reibman, A. R. & Lin, S. (2005). Multiple description coding for video delivery, *Proceedings of the IEEE* 93(1): 57–70.

Wang, Y. & Zhu, Q.-F. (1998). Error control and concealment for video communication: a review, *Proceedings of the IEEE* 86(5): 974–997.

Wang, Y., Zhu, Q.-F. & Shaw, L. (1993). Maximally smooth image recovery in transform coding, *IEEE Transactions on Communications* 41(10): 1544–1551.

Wang, Z., Sheikh, H. R. & Bovik, A. C. (2003). Objective video quality assessment, *The Handbook of Video Databases: Design and Applications*, CRC Press, pp. 1041–1078.

Wenger, S. (2003). H.264/AVC over IP, *IEEE Transactions on Circuits and Systems for Video Technology* 13(7): 645–656.

Wiegand, T., Farber, N., Stuhlmuller, K. & Girod, B. (2000). Error-resilient video transmission using long-term memory motion-compensated prediction, *IEEE Journal on Selected Areas in Communications* 18(6): 1050–1062.

Wiegand, T., Lightstone, M., Mukherjee, D., Campbell, T. G. & Mitra, S. K. (1996). Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard, *IEEE Transactions on Circuits and Systems for Video Technology* 6(2): 182–190.

Wiegand, T., Sullivan, G. J., Bjontegaard, G. & Luthra, A. (2003). Overview of the H.264/AVC video coding standard, *IEEE Transactions on Circuits and Systems for Video Technology* 13(7): 560–576.

Wu, J., Liu, X. & Yoo, K.-Y. (2008). A temporal error concealment method for H.264/AVC using motion vector recovery, *IEEE Transactions on Consumer Electronics* 54(4): 1880–1885.

Zeng, W. & Liu, B. (1999). Geometric-structure-based error concealment with novel applications in block-based low-bit-rate coding, *IEEE Transactions on Circuits and*

Systems for Video Technology 9(4): 648–665.

Zhang, J., Arnold, J. F. & Frater, M. R. (2000). A cell-loss concealment technique for MPEG-2 coded video, *IEEE Transactions on Circuits and Systems for Video Technology* 10(4): 659–665.

Zhang, R., Regunathan, S. L. & Rose, K. (2000). Video coding with optimal inter/intra-mode switching for packet loss resilience, *IEEE Journal on Selected Areas in Communications* 18(6): 966–976.

Zhang, R., Zhou, Y. & Huang, X. (2004). Content-adaptive spatial error concealment for video communication, *IEEE Transactions on Consumer Electronics* 50(1): 335–341.

Zhang, Y., Gao, W., Lu, Y., Huang, Q. & Zhao, D. (2007). Joint source-channel rate-distortion optimization for H.264 video coding over error-prone networks, *IEEE Transactions on Multimedia* 9(3): 445–454.

Zhou, Z.-H., Xie, S.-L. & Xu, Z.-L. (2005). Efficient adaptive MRF-MAP error concealment of video sequences, *Proceedings of International Conference on Machine Learning and Cybernetics*, Vol. 9, pp. 5507–5511.

Zhu, Q.-F. & Kerofsky, L. (1998). Joint source coding, transport processing, and error concealment for H.323-based packet video, *Visual Communications and Image Processing*, Vol. 3653, SPIE, pp. 52–62.

4

Available Bandwidth Estimation and Prediction in Ad hoc Networks

Haitao Zhao, Jibo Wei, Shan Wang and Yong Xi
National University of Defense Technology
China

1. Introduction

Wireless ad hoc networks provide quick and easy networking in circumstances that require temporary network services or when cabling is difficult. With the widespread use of multimedia applications that require Quality of Service (QoS) guarantees, research in providing QoS support in wireless ad hoc networks has received much attention recently (Nafaa, 2007). The term QoS gathers several concepts. Some efforts, like admission control, intend to offer guarantees to the applications on the transmission characteristics, for instance bandwidth, delay, delay jitter, or packet loss. Other solutions, like QoS routing, only select the best path among all possible choices regarding the same criteria. In both cases, an accurate evaluation of the amount of resources available (i.e., available bandwidth) on a given path is necessary. Therefore, obtaining accurate information of available bandwidth is a crucial basis for QoS-aware controls in wireless ad hoc networks. In the followed analysis, the term available bandwidth will be denoted by "AB" for brevity.

Since the IEEE 802.11 Distributed Coordination Function (DCF), based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), is the most popular MAC protocol used in ad hoc networks, the AB estimation problem in 802.11-based ad hoc networks has been a focus of recent research. Some approaches that used to be applied in wired networks have been adopted in wireless scenario, e.g., (Hu & Steenkiste, 2003; Jain & Dovrolis, 2003; Melander, Bjorkman et al., 2000; Ribeiro, Riedi et al., 2003; Strauss, Katabi et al., 2003). Meanwhile, some new proposed approaches that specialize for wireless networks have been proposed, e.g., (de Renesse, Friderikos et al., 2007; Sarr, Chaudet et al., 2008; Wu, Wang et al., 2005). So far, however, there is neither consensus on how to precisely measure the AB in ad hoc networks nor a practical approach that has been widely adopted, which makes all these approaches on the stage of experiment or simulation and no standard is agreed yet. So it's time to rethink the AB estimation in ad hoc networks, and find out the challenges that make it so difficult to arrive at an agreement. In this chapter we'll review the existing approaches for AB estimation, presenting the efforts and challenges to AB estimation in 802.11 or 802.11-alike ad hoc networks, and we will also give some proposals to tackle these challenges. Analyzing these problems will help to not only develop an accurate AB estimation approach but also design QoS support schemes in ad hoc networks.

This chapter is based on our work that previously, in parts, have been published in (Zhao, Garcia-Palacios et al., 2009; Zhao, Wang et al., 2009; Zhao, Wang et al., 2010). And the rest of this chapter is organized as follows. In Section 2, we first give a review of the state-of-the-art of AB estimation in ad hoc networks. Then in Section 3, we present the challenges of sensing-based approaches for AB estimation in 802.11 or 802.11-alike ad hoc networks and also give some solutions to them via analysis and simulation experiments. And in Section 4, we present the model-based approaches for AB prediction. In the end, we conclude this chapter in Section 5.

2. State of the art

In ad hoc networks, AB is defined in the context of end-to-end network path. Specifically, the path is a sequence of nodes, i.e., N_1, N_2, N_3…and N_{n+1} (n is the hop count), that communicate using identical, half-duplex wireless radio based on 802.11 DCF mode. The data packets are relayed from N_1 till N_{n+1}. The link (or hop) between N_i and N_{i+1} is referred to as Link *i*. See the illustration in Fig. 1.

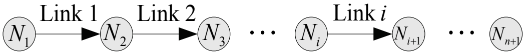


Fig. 1. *n*-hop path to calculate the end-to-end AB

The state of Link *i* at time t is

$$S_i(t) = \begin{cases} 0, & \text{when Link i is idle} \\ 1, & \text{when Link i is busy} \end{cases} \tag{1}$$

Note that the node being busy can be caused by its transmitting, receiving or the neighboring interference. In the time period of [*t*, *t*+*τ*], the utilization of Link *i* is

$$U_i(t) = \frac{1}{\tau}\int_t^{t+\tau} S_i(t)\,dt \tag{2}$$

The AB is defined as the unused bandwidth over the time interval *τ*, here τ is usually referred to as the *estimation period* (namely the time needed for estimating AB once). It is not a constant value and can be changed in different estimation tools, or even in a tool according to the network scenario. Then the AB of Link *i* in the time period of [*t*, *t*+*τ*] can be expressed as

$$AB_i(t) = C_i(1 - U_i(t)) \tag{3}$$

where C_i is the channel capacity of Link *i*. And the end-to-end AB of a path is mainly determined by the link with minimum AB along the path.

AB estimation has generated several contributions in the wired and wireless networking communities. Several classifications of these solutions may be imagined. We chose to classify the approaches that could be adopted to estimate AB in 802.11 ad hoc networks into three categories: probe-based approaches, sensing-based approaches and model-based approaches.

2.1 Probe-based approaches

Probe-based approaches estimate the AB along a path via sending end-to-end probe packets. All these approaches are principally based on Probe Gap Model (PGM) or Probe Rate Models (PRM) (Strauss, Katabi et al., 2003). In PGM, the AB is obtained by first building the mathematical formula of AB regarding the sending gaps and receiving gaps between probe packets, and then measuring the sending gaps and receiving gaps between probe packets to obtain AB. While PRM adopts a more straightforward principle as follows: if the probe packets sending rate is faster than AB, the probe packets will queue at some routers so that end-to-end delay increase gradually; On the other hand, if the probe packets sending rate is slower than AB, the probe packets will experience little delay. Therefore, the AB can be obtained while observing the delay variation and deciding the time when congestion begins. Furthermore, PGM can be cooperated with PRM, for instance in IGI (Initial Gap Increasing) method that proposed in (Hu & Steenkiste, 2003).

In the past decade, many probe-based AB estimation tools have been developed, such as Spruce (Strauss, Katabi et al., 2003), TOPP (Melander, Bjorkman et al., 2000), Pathchirp (Ribeiro, Riedi et al., 2003), IGI (Hu & Steenkiste, 2003), Capprobe (Kapoor, Chen et al., 2004) and Pathload (Jain and Dovrolis, 2003) to name a few. The developing course of these approaches is to build a more accurate relationship between AB and metrics in probe packets and thus increase the AB estimation accuracy. And a survey of them can be found in (Zhou, Wang et al., 2006). This type of approaches is most proposed originally for wired networks, and with the requirement of estimating AB in wireless ad hoc networks they are also adopted in wireless scenario. However, the difference between wired networks and wireless networks, especially that wireless ad hoc networks cannot bare the heavy overhead brought by the probe packets, impulses approaches specifically for wireless ad hoc network to be proposed. These approaches are trying to reduce the amount of probe packets and thus decrease the estimation overhead, among which the representative work are SenProbe (Sun, Chen et al., 2005) and the approach in (Hoang, Shao et al., 2006). SenProbe uses a return-way technique to estimate the unidirectional path capacity in wireless sensor networks, and thus simplify the path capacity estimation process. To further decrease the estimation overhead, authors in (Hoang, Shao et al., 2006) use a one-way probe (The destination node sends the probe to the source node and the source node estimates the AB). In theory, (Hoang, Shao et al., 2006) can reduce half of the overhead comparing to SenProbe. Whereas, reducing the probe packets will inevitably decrease the estimation accuracy.

Though the research on probe-based approaches is still moving on to find a balance between accuracy and overhead, some practical drawbacks of this type of approaches make it difficult to break through in its application in wireless ad hoc networks. First, the accuracy of probe-based approaches is not satisfactory. C. Dovrolis etc. (Dovrolis, Ramanathan et al., 2004) proved that PGM model actually estimates the Asymptotic Dispersion Rate (ADR) instead of the AB (The ADR is asymptotic dispersion rate between AB and channel capacity, and is an upper bound of the AB). At the same time, authors in (Lakshminarayanan, Padmanabhan et al., 2004) showed that in a CSMA- based wireless networks, a new probe packet enqueued at one of the stations might in fact be transmitted sooner than the older cross-traffic packets waiting at other stations. So the probe packet may not experience a delay commensurate with the total volume of cross-traffic, leading to over-estimation of the AB. On the other hand, (Lao, Dovrolis et al., 2006) arrived to a conclusion that contrary to the former two research: in general cases PGM can significantly under-estimate the AB of an end-to-end path. Maybe this can explain why Lakshminarayanan etc. (Lakshminarayanan, Padmanabhan et al., 2004) had arrived to the conclusion, via experiments, that most of probe-based approaches can only used in wired networks, and if they are used in 802.11 wireless networks, the measurement result will have big error without obvious disciplinarian. Second, there are some practical problems when deploying existing probe-based bandwidth measurement approaches in ad hoc networks. It was observed(Johnsson, Melander et al., 2005), for instance, that the measured link capacity show dependence on the probe packet size and a smaller probe packet size will result in a lower bandwidth estimation. Besides, the source node is supposed to have the ability to send probe packets at a higher rate than AB via using PGM model. And the most but not the last drawback is that when every node in an ad hoc network needs to perform such an estimation for several destinations, the number of probe packets introduced in the network can be important and interact on the traffic as well as on other probes.

2.2 Sensing-based approaches

With the effort to avoid the presented problems in probe-based approaches, recent research contributes to estimate the AB on a given wireless link via sensing-based approaches. These approaches need not to send probe packets, but sense nodes' channel utilizations and eventually exchange this information via local broadcasts to calculate the AB. Usually these local broadcasts are performed using Hello packets that are used in many routing protocols to discover local topology. If these exchanges are not too frequent, this technique can be reasonably considered as non intrusive (Sarr, Chaudet et al., 2008). And thus sensing-based approaches are very suitable for wireless networks.

In (Zhai, Chen et al., 2006), the authors proposed the index CBR (channel busyness ratio) for AB estimation, which is easy to obtain and can timely represent channel utilization. Though this algorithm is proposed for single-hop WLAN, it is straightforward to get the idea that the AB of the multi-hop path as illustrated in Fig. 1 can be got by calculating $\min\{1-CBR_i, i=1,2,...n+1\}$, where CBR_i is the channel busy ratio that sensed by node N_i. K. Xu, etc. (Xu, Tang et al., 2003) adopted this idea and added a smoothing factor to mask transient effects. But they only considered the AB estimation at each node within the path and did not consider any possible distant interfering nodes. To fix this problem, QoS-AODV proposed in (de Renesse, Ghassemian et al., 2004) also performs such per-node AB estimation, but the bandwidth available to a node is computed as the minimum of the AB over its single-hop neighborhood. However, with 802.11 protocol, two nodes within carrier sense range share the medium and thus the bandwidth, even if they cannot directly communicate. To consider carrier sense range interfering, most existing literatures such as FAT (Wu, Wang et al., 2005) and CACP (Yang & Kravets, 2005), approximate the carrier sense area by the two-hop neighborhood. The basic ideas of them are alike: each node provides information about the total bandwidth it uses to route flows and about its one-hop neighbors as well as their usage of the bandwidth, by periodically broadcasting a Hello message containing this information. Then, each node can compute the bandwidth usage and then derive the AB in its two-hop neighborhood.

Since the interference ranges of nodes within the same multi-hop path overlap, this phenomenon prevents a node from forwarding transmissions while any path members within its interference range are sending. Thus multiple links on the path of a flow contend for bandwidth, which is known as the *intra-flow contention problem* and was first studied in (Sanzgiri, Chakeres et al., 2004). Because of intra-flow contention, the actual AB for a flow should be further divided by the *contention count* (*CC*), namely the number of nodes that contend for bandwidth. The *CC* at node N_i can be represented as

$$CC_i = |CSN_i \cap NoP| + 1 \tag{4}$$

where CSN_i is the set of nodes that within N_i's carrier sense range, *NoP* is the set of the nodes in the path. This problem is nontrivial because of the difficulty to find out the nodes within one node's carrier sense range. In literature, four methods were proposed to calculate *CC*: (1) based on the assumption that the interference range is the two-hop range, *CC* equals the hop count if the hop count is not more than 4, or 4 otherwise. This is the most popular approach in literature, e.g., (Chen & Heinzelman, 2005; Sarr, Chaudet et al., 2008); (2) increase the transmit power so that the packet can be successfully received by all the nodes in carrier sense range, e.g., CACP (Yang & Kravets, 2005); (3) sense the duration of the packet to determine the nodes in its carrier sense range, e.g., (Sanzgiri, Chakeres et al., 2004); or (4) use localization information with the help of Global Position System (GPS)(Gupta, Musacchio et al., 2007).

In the end, besides considering the carrier sense range media usage, the recent study ABE (Available Bandwidth Estimation) (Sarr, Chaudet et al., 2008) further considered the overlap probability of two adjacent nodes' idle time (P_o), packet collision probability (P_c) and the proportion of bandwidth (K) consumed by the waiting process of 802.11 to improve the accuracy of AB estimation. Then the end-to-end AB of the path {N_1, N_2, N_3… N_{n+1}} at time *t* is

$$AB(t) = min\left\{ P_o \cdot (1 - P_c) \cdot (1 - K) \cdot \frac{AB_i(t)}{CC_i}, i = 1,2,...,n \right\} \tag{5}$$

Sensing-based approaches were first proposed for single-hop wireless networks and then extended to multi-hop wireless scenarios. In this research, improving the estimation accuracy acts as the main driver. And so far, there is still work to do. For instance, AAC (Adaptive Admission Control) protocol (de Renesse, Friderikos et al., 2007) and ABE scheme (Sarr, Chaudet et al., 2008) are recently proposed schemes for AB estimation in 802.11-based ad hoc networks, but both of them need further improving in the consideration of the overlap probability of two adjacent nodes' idle time, i.e., P_o. In AAC, the transmitter and receiver are assumed to obtain perfectly synchronization, i.e., $P_o = 1$. But, in fact, there is possibility that when the transmitter is sensing idle the receiver is busy and thus cannot receive the packets from the transmitter, and vice versa. Under this case, AAC will over-estimate AB on the link between this transmitter-receiver pair. To resolve this problem, ABE use probability analysis to calculate P_o under the assumption that each node's surrounding medium occupancy is a uniform random distribution and independent to each other. This assumption, however, ignores the factual dependence of the interfering around the sender and the receiver, and thus will also result in inaccurate estimation of AB. This observation inspired our work in (Zhao, Garcia-Palacios et al., 2009) to calculate the overlap probability for two adjacent nodes' idle periods while taking into consideration the factual dependence of the interfering around them. And consequently improve the accuracy of AB estimation in IEEE 802.11-based ad hoc networks.

2.3 Model-based approaches

The AB estimation approaches that utilize currently sensed information are often insufficient because they lack predictive power and scalability, just considering that the entrance of a new flow will result in the change of network parameters (i.e. collision probability) and further the real AB. We need an approach that with predictive power and has the ability and scalability to find the quantitive consequences of the entrance of new flows, and to achieve this goal, a proper model is necessary.

In the seminal work of Bianchi (Bianchi, 2000), the authors provided an analysis model for the behavior of 802.11 DCF protocol assuming a two dimensional Markov model at the MAC layer. The main assumptions in this work are (i) every node is saturated (i.e. always has a packet waiting to be transmitted), (ii) transmission error is a result of packets collision and is not caused by channel errors and (iii) the network is homogeneous (i.e. each node acts the same). Provided that these assumptions are satisfied, the resulting model is remarkably accurate. However, these assumptions are not necessarily true in practical networks.

First, the saturation assumption is unlikely to be valid in real multi-hop wireless networks. And even in WLANs, it is proved that the optimal work point[1] of a network lies before it entering saturation (Zhai, Chen et al., 2005). Thus more recent studies have shifted the focus onto 802.11 networks operating in non-saturated conditions, such as (Malone, Duffy et al., 2007) and (Kun, Fan et al., 2007), where the authors extended the underlying model in order to consider unsaturated traffic conditions by introducing a new idle state that accounts for the case in which the node buffer is empty, after a successful packet transmission.

To relax the dependence on the second assumption in (Bianchi, 2000), authors in (Chatzimisios, Boucouvalas et al., 2003) deal with the extension of Bianchi's Markov model in order to account for channel errors. And in (Qiao, Choi et al., 2002), the authors look at the impact of channel errors and the received SNR (Signal-to-Noise Ratio) on the achievable throughput in a system with rate adaptation, where the transmission rate of the terminal is adapted based on the link quality. (Daneshgaran, Laddomada et al., 2008) extends the previous works on this subject by looking at a more realistic channel condition for unsaturated traffic, and their assumptions are essentially similar to those of Bianchi's with the exception that they do assume the presence of both channel errors and capture effects due to the transmission over a Rayleigh fading channel.

To relax the dependence on the third assumption in (Bianchi, 2000), authors in (Ergen and Varaiya, 2005) propose a novel Markov model for the 802.11 DCF in a scenario with various nodes contending for the channel and transmitting with different transmission rates. An admission control mechanism is also proposed for maximizing the throughput while guaranteeing fairness to the involved transmitting nodes. And (Qiu, Zhang et al., 2007) develops a more general model to estimate the throughput, based on SNR or RSSI (Received Signal Strength Index) measurements from the underlying network itself and thus is more accurate than abstract propagation models based on distance. Their model also takes into account the general case of heterogeneous nodes with different traffic demands and different radio characteristics. While management decisions can be based on SNR or RSSI measurements from the PHY layer, it is known that these may be only weakly correlated with the actual channel behavior perceived at the MAC layer (Aguayo, Bicket et al., 2004).

Model-based approaches are very useful for network performance analysis, but the challenge is that to build an accurate analysis model for multi-hop wireless networks is not an easy job.

[1] The optimal work point is the turning point that the network should work around. Before that point, as the input traffic increases, the throughput keeps increasing, the delay and delay variation does not change much. After that point, the throughput drops quickly and the delay and delay variation increase dramatically.

3. Sensing-based approaches for AB estimation

We already mentioned that the probe-based approaches presented above do not yield accurate results in a wireless ad hoc context because of their practical drawbacks. In this section, we will mainly focus on sensing-based approaches, considering the challenges for accurate AB estimation and then presenting some solutions to them.

3.1 Identification of the nodes in the carrier sense range

Under the DCF mode, a transmission within one node's carrier sense range will interfere its receiving, which means while estimating one node's busy time (which is the first step to obtain the AB) we have to consider the interference in the carrier sense range. Thus identifying the nodes in one node's carrier sense range, which is represented by *CSN* in (4), is important to accurately estimate end-to-end AB.

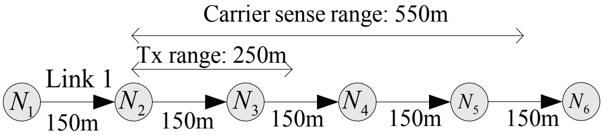


Fig. 2. Example scenario

In the majority approaches to identify the *CSN*, the node's carrier sense range is commonly expressed in terms of number of hops, k. And then use hello messages broadcast over the k-hop range to identify *CSN*. The most popular value of k is 2, e.g., (Chen & Heinzelman, 2005; Sarr, Chaudet et al., 2008; Wu, Wang et al., 2005) and the CACP-multihop in (Yang & Kravets, 2005) . However, this is not necessary true in real wireless scenarios. Take the case shown in Fig. 2 as an example. Node N_5 is within the carrier sense range of N_2, but 3 hops away from it. When N_5 is transmitting to N_6, the transmission will reduce the AB on Link 1 (N_2 will sense the transmission from N_5 and thus shut itself down according to 802.11 protocol, which prevents Link 1 being on). But this effect is not counted by the aforementioned approach which assumes the carrier sense range is two-hop range. To resolve this problem, in AAC (de Renesse, Friderikos et al., 2007), the value of k switches between 2 and 3 with respect to the roughness of the path. And it is claimed that the roughness of the path is very likely to depend on the network node density. A high node density involves more paths existing between two mobile hosts. If nodes are uniformly distributed, there is a high possibility that the shortest path will be the smoothest. However, this theory holds only when assuming all nodes have an identical circular propagation region.

In (Yang & Kravets, 2005), Yang and Kravets also proposed two other different approaches, CACP-power and CACP-CS, to identify CSN. CACP-power assumes that the transmission power is variable. This approach consists of increasing the transmission power for bandwidth query messages such that all carrier sense nodes are able to decode it, which implies high power consumption and potential interferences as drawbacks. CACP-CS analyzes channel activity at a power threshold called *Neighbor-carrier- sensing Threshold*, which is set much lower than the carrier sensing threshold. Thus, each node is able to derive the bandwidth consumption of all its CSN. This technique minimizes overhead but could include isolated nodes that do not belong to any carrier sensing range. In real scenarios where noise interferes with almost any signal, such a low power threshold detection system might wrongly interpret channel activity.

Without increasing the transmission power or decreasing the analysis power threshold, K. Sanzgiri, etc. (Sanzgiri, Chakeres et al., 2004) propose two methods, Pre-Reply Probe (PRP) and Route Request Tail(RRT), to obtain the number of *CSN* along a multi-hop path, i.e., *CC* in (4). The highlight of the proposed solutions is that carrier-sensing metrics such as the duration of sensed transmissions, is used to deduce the information of neighbors within carrier sense range, and no high power transmissions are necessary. But they either requires an additional message (PRPM) to be transmitted during route discovery (in PRP) or a tail is attached to RREQ packets (in RRT), which will increase the network overhead. Furthermore, counting sensed packets of a particular duration can enhance computing complexity and thus increases the route acquisition latency.

3.2 Estimation of the collision probability under unsaturated ad hoc networks

Collision is one important characteristic of wireless networks (Bianchi, 2000; Zhai, Chen et al., 2005). There are two main reasons to bring collision: (1) after each node's backoff, two nodes start to transmit packet to a same node at the same time; (2) the collision brought by the problem of hidden node. After collision, a node has to backoff and waits to retransmit. So the airtime taken by collision and backoff can not be used to transmit data, thus should be eliminated from the AB, see (5). And to do that we have to first estimate the collision probability.

The overwhelming majority of the analysis on collision probability is based on saturated scenario, i.e., Bianchi's landmark Markov model (Bianchi, 2000) and some research following it (Kuan & Dimyati, 2006). Unfortunately, as aforementioned that a very important task of network control is to avoid the network from saturation and keep it work at an unsaturated "optimal work point" (Zhai, Chen et al., 2005). It means the controlled network will work under unsaturated scenario. Thus recent research is focus on the analysis or estimation of the collision probability under the unsaturated scenario.

The first inspiration is that we can rely on the models for non-saturated networks. In (Malone, Duffy et al., 2007), (Ahn, Campbell et al., 2002) and (Ergen & Varaiya, to appear), modifications of (Bianchi, 2000) are considered where a probability of not transmitting is introduced that represents a node which has transmitted a packet, but has none waiting. With these models, we can derive the collision probability in non-saturated networks. However, it is important to note that the Markov chain's evolution is not real-time, and so the estimation of collision probability and throughput requires an estimate of the average state duration. Furthermore, as aforementioned that accurate analysis models for multi-hop networks are difficult to build up and maintain in distributed networks.

Without relying on analysis model for network behavior, ABE (Sarr, Chaudet et al., 2008) considered the real-time estimation of collision probability via calculating the collision rate of Hello packets. The idea is that since every node knows how many Hello packets should be received from one neighbor during a specific period (usually defined by the routing protocol) thus it can measure the collision rate of Hello packets, p_{Hello}, via keeping an account on the number of Hello packets it actually received. And then obtain the collision probability of data packets, p_c, by multiplying a Lagrange interpolate polynomial to compensate the different packet size between data packets and Hello packets as follows,

$$p_c = f(m) \cdot p_{Hello} \tag{6}$$

$$f(m) = am^3 + bm^2 + cm + d \tag{7}$$

where *m* is the size of data packets; a, b, c and d are polynomial parameters, which are obtained after different simulations varying data packet sizes and network load in (Sarr, Chaudet et al., 2008). However, there are two main shortcomings when using this approach. The first issue is that Hello packets are sent at a much lower rate than that of data packets. For instance, when considering AODV routing protocol (Perkins, Royer et al., 2001), Hello packets are usually sent at the frequency of 1 packet per second. Only to recognize that at a rate of 2 Mbps and assuming packet sizes of 1K bytes around 250 data packets are sent for just 1 Hello packet. In theory we could adopt a wider measurement period (e.g., 10 seconds), however the ratio to data packets is still as high and the measurement period can be too long for fast changing topology scenarios that are likely to emerge in Ad hoc networks. Therefore, although collisions on Hello packets can give some idea on collisions rates of Data packets, it is impossible to get an accurate estimate. A second issue is that experiments have to be run in advance in order to get a relative accurate expression of the Lagrange interpolating polynomial (*f*(*m*)) for a given scenario, and this expression will change when varying the scenario (e.g. number of stations, topology and packet size profile).

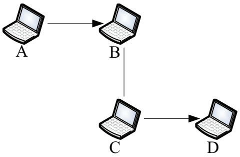


Fig. 3. A typical collision scenario

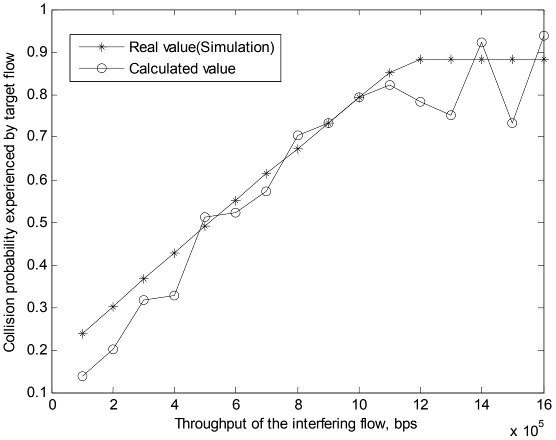


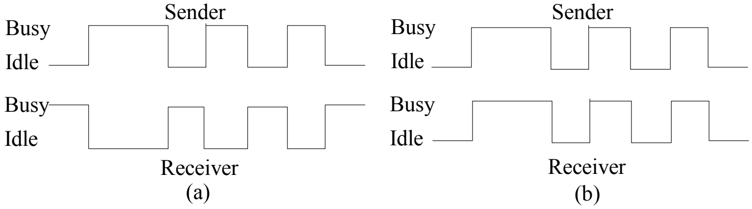
Fig. 4. Evaluation results

Let's consider the same typical collision scenario as in (Sarr, Chaudet et al., 2008), shown in Fig. 3, where C is a hidden node to A. Our aim is to use the aforementioned approach to estimate the collision probability over the target link A-B, which caused by the interfering flow C-D. As the throughput in the interfering flow changes, the collision probability over our target link changes. The simulated medium capacity is set 2 Mbps with a packet size of

1K bytes. And the results are plotted in Fig. 4, which shows the instability and the inaccuracy of estimating *p* by using Hello packets as in (Sarr, Chaudet et al., 2008).

3.3 Airtime synchronization

For the communication to happen, the medium has to be free on the sender's side so that the sender gains access the medium. On the receiver's side, the medium has to be free during the time required to transmit the whole data frame to avoid colliding. In other words, the medium availability on the sender and receiver sides has to somehow synchronize for the communication to take place. Fig. 5 (Sarr, Chaudet et al., 2008) shows two extreme case of the airtime availability at the sender side and the receiver side: (a) when they are never overlapped and (b) when they are totally overlapped. We can see that though in both cases, the idle airtime values measured at each node are the same but in case (a), the periods of airtime availability of both peers never overlap and the AB on the link is null. In the opposite case, the scenario depicted in (b) offers several communication opportunities on the link, when both sides are idle. So we have to consider this synchronization problem of the airtime at the sender and receiver side when estimate the AB.



(a) when they are never overlap; (b) when they are totally overlap

Fig. 5. Airtime at the sender side and the receiver side

Many existing publications, e.g., AAC (de Renesse, Friderikos et al., 2007) and (Wu, Wang et al., 2005), calculating the idle time ratio on Link *i*, enoted as R_i, as

$$R_i = \min\{r_i, r_{i+1}\} \tag{8}$$

where r_i and r_{i+1} are the idle time ratio sensed by N_i and N_{i+1}. These works actually assume that the airtime is totally overlapped. In order to obtain a more accurate consideration of the airtime synchronization, ABE (Sarr, Chaudet et al., 2008) assumes that the each node's surrounding medium occupancy is a uniform random distribution and the idle time ratio on Link *i* is represented by

$$R_i = r_i \cdot r_{i+1} \tag{9}$$

But for adjacent nodes, neither their airtime could be synchronized naturally nor are they independent to each other. In our recent research (Zhao, Garcia-Palacios et al., 2009), we have evaluated the approaches proposed in AAC and ABE to reveal the insufficiency of them. And we further proposed an AB estimation approach IAB (Improved Available Bandwidth estimation) which achieves more accurate estimation in [25]. The main contribution of IAB is that it considered the natural dependence between the medium state that sensed by two adjacent nodes. And this consideration is realized as follows. We first differentiate the channel busy state caused by Transmitting/Receiving and the channel Sensing state. And this differentiation results in a more accurate estimation of the overlap probability of the idle times between two adjacent nodes and consequently a more accurate estimation of the AB between these nodes.

3.4 Intra-flow contention

Intra-flow contention prevents a node from forwarding transmissions while any path members within its interference range are sending, thus reduce the end-to-end AB of a multi-hop path. Therefore, we have to take this phenomenon into consideration when estimate the end-to-end AB in multi-hop ad hoc networks.

As described in Section 2.2, the overwhelming approaches to consider the intra-flow contention is to divide the AB further by the contention count, i.e., *CC*, and in literature there are three methods to calculate the *CC*. These methods are not satisfactory in that they either too simplified to reflect the real scenario (Chen & Heinzelman, 2005; Sarr, Chaudet et al., 2008), or increase the power consumption (Yang & Kravets, 2005) or complexity (de Renesse, Friderikos et al., 2007; Sanzgiri, Chakeres et al., 2004) of the network control. And what makes the problem worse is that even if we can accurately get the value of *CC*, dividing the bottleneck link's AB by *CC* (this method is referred to as the *average-based method* for brevity) cannot provide the accurate information of the end-to-end AB. The reason lies in that there is a throughput deviation of each link which leads to overlapping of simultaneously transmitting packets and collisions. Recently, authors in (Jae-Yong & JongWon, 2007) use the central limit theorem to model the throughput deviation, assuming the summation of uniformly distributed backoff times, which is referred to as the *deviation-based method*. Unfortunately, the proposed throughput deviation model can only consider the collisions due to two simultaneous transmissions along a path. This is invalid when there are more than 4 hops, in which case more collision scenarios exist. For clarity, let's consider an *n*-hop path, as illustrated in Fig. 1. The distance between two adjacent nodes is kept 200m in order to ensure that a flow generated in N_1 will go through each intermediate node and reaches the destination while we set transmitting range and carrier sense range respectively as 250m and 550m. In Fig. 6, we vary the number of hops, i.e., *n,* and plot the calculated end-to-end AB using average-based method, deviation-based method as well as the real value via simulation (The simulated medium capacity is also set 2 Mbps with a packet size of 1K bytes as in Fig. 4).

Fig. 6 clearly shows that the average-based method does not reflect the real value when the hop count is 4 or more because it neglects the collision due to throughput deviation of each link. Note that there is no collision for 3 or less hops, in which case the throughput average method matches the real value. Deviation-based method still accurately estimates the AB when the hop count is 4. However, it disagrees with the real AB after the path exceeds 4 hops, which keeps on decreasing. This estimation error can be explained by that more collisions appear when the hop count exceeds 4 which will further decrease the AB. (The average collision probability is also showed in Fig. 6.) The results demonstrate that the model in (Jae-Yong & JongWon, 2007) which only considers the collisions due to two simultaneous transmissions will produce an inaccurate AB calculation when the hop count is more than 4.

To solve this problem, our recent study (Zhao, Wang et al., 2010) proposes and validates a model to analyze the intra-flow contention of a given path in multi-hop wireless networks.

The basic idea is twofold: (i) We consider the intra-flow contention problem with an analysis model that account for the contenting links' behavior, instead of just calculating the contention count. (ii) The model envelops important factors for intra-flow contention, i.e., neighboring interference, hidden-node collision and possible multi-rate scenario, which make it approach reality and obtain accurate results. (The results obtained by our proposed model under the aforementioned scenario are also shown in Fig. 6, with the legend of *Model-based AB estimation*.)

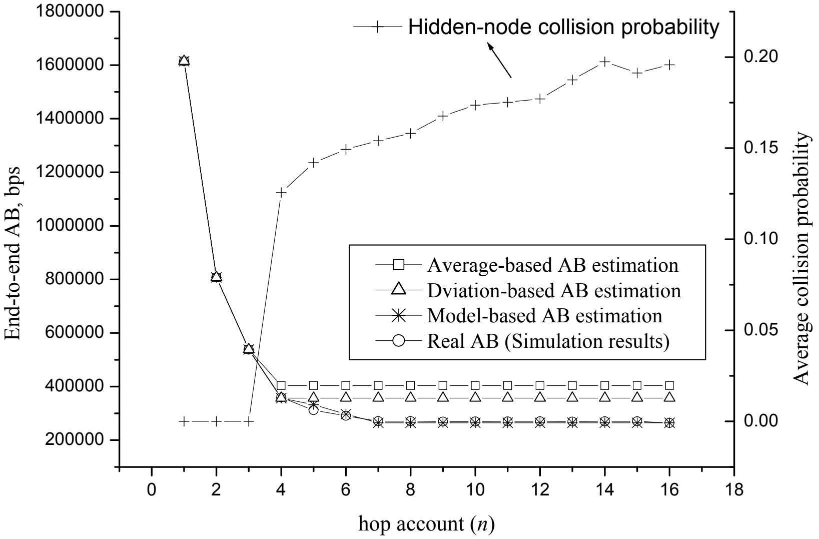


Fig. 6. End-to-end AB while varying the hop count

4. Model-based approaches for AB prediction

The model-based approaches are of redictive power and the current challenge is to derive more accurate and scalable analysis model. We will show our effort on this topic in this section.

4.1 Analytical model

For a better understanding, we give an overview of our model as shown in Fig. 7. Our model takes network information (topology and existing traffic), radio-dependent parameters and incoming traffic throughput demands as input and outputs the predictive throughputs of both the incoming flow and existing flows. Such a model is a powerful tool for performing what-if analysis and facilitating network optimization and diagnosis. Although in this chapter we focus on the throughput demands, or bandwidth requirement, of the flow, there is coupling of bandwidth and delay over a wireless link as shown in (Chen, Xue et al., 2004). So the model in this chapter can potentially be extended to analyze other QoS requirements, such as delay, by relating them to the network parameters, however this is out the scope of this chapter.

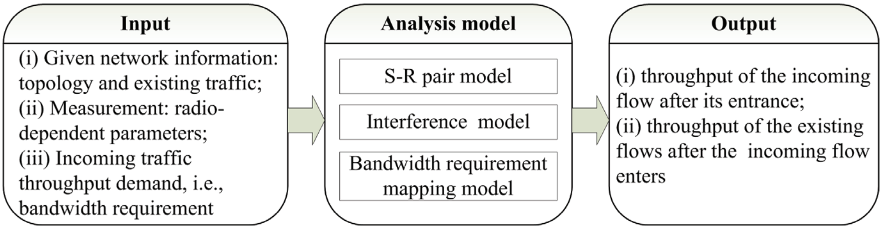

Input
(i) Given network information: topology and existing traffic;
(ii) Measurement: radio-dependent parameters;
(iii) Incoming traffic throughput demand, i.e., bandwidth requirement
Analysis model
S-R pair model
Interference model
Bandwidth requirement mapping model
Output
(i) throughput of the incoming flow after its entrance;
(ii) throughput of the existing flows after the incoming flow enters

Fig. 7. Model structure

The model consists of three major components: S-R (i.e., sender-receiver) pair model, interference model and bandwidth requirement mapping model. These models will be covered in Sections 3.2, 3.3 and 3.4 respectively. The S-R pair model gives the link state from the view of an S-R pair, and considers important probabilities such as the transmission probability, the unsuccessful transmission probability, the sense busy probability and the non-empty transmission buffer probability. The interference model constructs the contention graph of the network, in order to analyze the interference of contending links. The bandwidth requirement mapping model relates the network parameters in the S-R pair model and interference model to the bandwidth requirement of the incoming flow(s). It is also important to initiate some key parameters that used in this model, which is explained in Section 3.5.

4.1.1 S-R pair model

The behavior of an S-R pair that employs an 802.11 protocol is dictated by the occupation of the 'air' around it (the channel). We denote the sender and receiver respectively as N_{k-1} and N_k, and the link between them as Link *k*.

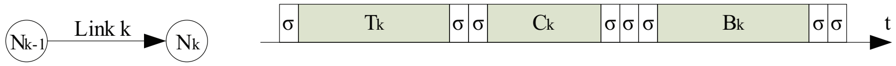
We adopt the concept of generic *slot* used in (Dao & Malaney, 2008) (which is also denoted as variable length slots (VLS) in (Li, Qiu et al., 2008)), thus for the channel sensed by the Link *k*, 4 different states can be identified:

i. Idle—N_{k-1} has seen the medium as idle and, either it has no data to send or its backoff counter has not reached 0 (i.e. backoff is in process).
ii. Successful transmission—N_{k-1} has transmitted a packet, received an ACK from N_k and is about to resume backoff.
iii. Unsuccessful transmission—N_{k-1} has transmitted, timed-out while waiting for an ACK from N_k and is about to resume its backoff.
iv. Sense busy—N_{k-1} has detected the medium busy due to one or more other nodes transmitting, by means of either physical or virtual carrier sensing (i.e., the Network Allocation Vector, NAV), and has suspended its backoff until the NAV and DIFS/EIFS indicate that the backoff can resume.

The average time intervals during which Link k remains in idle, successful transmission, unsuccessful transmission and sense busy are denoted by σ, T_k, C_k, and B_k, respectively. σ is constant, equal to the backoff slot. The duration of the other intervals can be variable, depending on the access mechanism, the frame size, and the sending rate. From the perspective of the S-R pair, the evolution of the channel state of Link k can be abstractly represented by a temporal diagram such as the one exemplified in Fig. 8(b).

So the average length of the Generic slot of link k can be expressed as:

$$E_k = \tau_k p_k C_k + \tau_k (1 - p_k) T_k + (1 - \tau_k) b_k B_k + (1 - \tau_k)(1 - b_k)\sigma \tag{10}$$



(a) The S-R pair; (b) The state of the channel between the S-R pair

Fig. 8. S-R pair model

where τ_k represents the transmission probability on one time slot; p_k is the unsuccessful transmission probability. b_k is the channel busy probability. Then the normalized channel utilization ratio (i.e., the normalized transmitting airtime whether successfully or not, represented by x_k) and the successful transmission time ratio (represented by y_k) of Link *k* can be expressed as:

$$x_k = \frac{\tau_k p_k C_k + \tau_k(1-p_k)T_k}{E_k} \tag{11}$$

$$y_k = \frac{\tau_k(1-p_k)T_k}{E_k} \tag{12}$$

The throughput of Link k is, in pkt/s

$$S_k = \frac{\tau_k(1-p_k)\Lambda}{E_k} \tag{13}$$

where Λ is the effective load fraction.

In equation (10), the average durations of a successful transmission and of an unsuccessful one are known a priori according to the 802.11 DCF standard (see (Bianchi, 2000), here we neglect the propagation delay). They are as follows under the Basic mode and RTS/CTS mode:

$$\begin{cases} T_k^{(Basic)} = DIFS + DATA + SIFS + ACK \\ C_k^{(Basic)} = DIFS + DATA + ACK_{timeout} \end{cases} \tag{14}$$

$$\begin{cases} T_k^{(RTS/CTS)} = DIFS + RTS + CTS + DATA + ACK + 3 \cdot SIFS \\ C_k^{(RTS/CTS)} = DIFS + RTS + CTS_{timeout} \end{cases} \tag{15}$$

In single-hop 802.11 networks all nodes are synchronized and the duration of a busy period equals the sum of the other nodes' transmitting duration. However, in the multi-hop case, transmissions of different nodes can overlap randomly due to the lack of coordination, which makes the determination of one node's busy period more complex. We take the assumption that if two links, for instance Link *i* and Link *j*, cannot sense each other, their action is independent to each other, this assumption is shown reasonable in (Gao, Chiu et al., 2006). So the overlap probability, denoted by $P_{overlap}(i,j)$, of these two links' transmitting airtime can be approximated as

$$P_{overlap}(i,j) = \frac{x_i \times x_j}{1 - \sum_{c \in v(i,j)} x_c} \tag{16}$$

where $v(i)$ represents the set of contending links (i.e., the links that contend with each other, and we will present them in Section 3.3) of Link i and $v(i,j)$ the set of common contending links of Link i and Link j. In Eq. (16), the numerator is the normalized probability that they transmit at the same time. When their common contending links are transmitting, neither of them can transmit, therefore the denominator represents the total time that they can use to transmit. Eq. (16) is referred to as the *second-order approximation*, which will be used again in our future analysis. Thus the sense busy time of Link k can be obtained via

$$B_k = \left(\sum_{i \in v(k)} x_i - \sum_{\substack{i_1, i_2 \in v(k); \\ i_1 \notin v(i_2) \cup i_2}} \frac{x_{i_1} x_{i_2}}{1 - \sum_{c \in v(i_1, i_2)} x_c} \right) E_k \tag{17}$$

A. Calculating the transmission probability τ

We should keep in mind that to support an application throughput along one route, the nodes on this route may have different transmission probabilities considering they may experience different collision probabilities. But in this section we temporarily drop the subscript, k, of the symbols for brevity.

A node can begin transmission when the following three conditions are satisfied: i) the node has data to transmit; ii) the link is idle; and iii) its random backoff counter reaches 0. The first one is related to the transmission queue. The last two are related to the interference by neighboring nodes. More specifically, one node's backoff counter is related to the unsuccessful transmission probability it experiences.

The transmission probability τ is a function of unsuccessful transmission probability p, which is first given in (Bianchi, 2000) under saturated situations. Recently, in (Kumar, Altman et al., 2007)and (Malone, Duffy et al., 2007) similar expressions of τ as a function of p are derived respectively for a large class of backoff mechanisms and for unsaturated situations. The complete expression of **τ** for 802.11 that takes into account the maximum retransmission limit jointly with the maximum window size and non-saturation case is given by

$$\tau = \eta \cdot \left(\frac{q^2 W_0}{(1-q)(1-p)(1-(1-q)^{W_0})} - \frac{q^2(1-p)}{1-q} \right) \tag{18}$$

where η is the stationary probability of a node being in the state where the backoff process is complete, but the node's transmission queue is empty (Malone, Duffy et al., 2007).

$$\begin{aligned} \frac{1}{\eta} &= (1-q) + \frac{q^2 W_0 (W_0+1)}{2(1-q)(1-(1-q)^{W_0})} + \frac{q(W_0+1)(p(1-q)-q(1-p)^2)}{2(1-q)} \\ &+ \frac{pq^2}{2(1-q)(1-p)} \left(\frac{W_0}{1-(1-q)^{W_0}} - (1-p)^2 \right) \left(\frac{2W_0(1-p-p(2p)^{m-1})}{(1-2p)} + 1 \right) \end{aligned} \tag{19}$$

And q is the probability that there is at least one packet in the queue after a transmission, which is mainly related to the traffic load and it will be discussed in Subsection D. W_0 and $2^m W_0$ are respectively the node's minimum and maximum contention window.

B. Calculating the unsuccessful transmission probability p

The unsuccessful transmission probability p may arise from collisions or channel failure. We identify three different categories of unsuccessful transmissions as follows: (i) due to collision between synchronized nodes, which occurs with the probability of l_{sc}; (ii) due to hidden nodes, which occurs with the probability of l_{hc}; (iii) due to channel errors, which occurs with the probability of l_e. And we assume that these three probabilities are statistically independent, then a transmission is successful if it does not suffer from any of the three types of unsuccessful transmission mentioned above (they may occur simultaneously) and thus the unsuccessful transmission probability is:

$$p = 1 - (1 - l_{sc})(1 - l_{hc})(1 - l_e) \tag{20}$$

Collisions between synchronized nodes are the traditional type of packet losses due to the MAC protocol considered in single-hop 802.11 networks (Bianchi, 2000). Indeed, when all senders are in range of each other, the DCF function is able to synchronize all nodes in such a way that all transmission attempts happen at well defined slot boundaries recognized by all nodes. As a result, in this network scenario the conditional unsuccessful transmission probability for Link k is simply given by

$$p_{sc}^{k} = 1 - \prod_{i \neq k, i \in v(k)} (1 - \tau_i) \tag{21}$$

If each node has the same transmission probability then we will obtain the same result as in (Bianchi, 2000): $1 - (1 - \tau)^{n-1}$, where n is the total number of nodes in the WLAN. However, in a multi-hop topology the DCF function fails to synchronize all nodes and the hidden node collision usually account for an important component of the overall packet collision probability. The hidden node collision has been modeled in (Zhao, Wang et al., 2010). If node j is node k's hidden node, the collision probability experienced at node k due to node j is as follows (using $p_{hc}^{k,j,(1)}$ and $p_{hc}^{k,j,(2)}$ to respectively denote the case when node j is the Type I and Type II hidden node[2] to node k)

$$p_{hc}^{k,j,(1)} = \frac{x_j}{1 - \sum\limits_{c \in v(j,k)} x_c + \sum\limits_{\substack{m,n \in v(j,k); \\ m \notin v(n) \cup n}} \dfrac{x_m \times x_n}{1 - \sum\limits_{c \in v(m,n)} x_c}} \tag{22}$$

$$p_{hc}^{k,j,(2)} = \frac{x_k + x_j}{1 - \sum\limits_{c \in v(j,k)} x_c + \sum\limits_{\substack{m,n \in v(j,k); \\ m \notin v(n) \cup n}} \dfrac{x_m \times x_n}{1 - \sum\limits_{c \in v(m,n)} x_c}} \tag{23}$$

Once we know the type of hidden node to Link i, the overall hidden node collision probability is the union of $p_{hc}^{k,j}$, $j \in h(k)$ ($h(k)$ represents the set of hidden node to Link k), namely:

[2] Please refer to (Zhao, Wang et al., 2010) for further detail.

$$p_{hc}^{k} = \sum_{j \in h(k)} p_{hc}^{k,j} - \sum_{\substack{m,n \in h(k);\\ m \notin v(n) \bigcup n}} \frac{p_{hc}^{k,m} \times p_{hc}^{k,n}}{1 - \sum_{c \in v(m,n)} x_c} \tag{24}$$

Here, we also use the *second-order approximation* to unfold the union expression.

Note that the collision may not necessarily result in packet loss, considering the *capture effect*. The capture effect is the ability of certain radios to correctly receive a strong signal from one transmitter despite significant interference from other transmitters. It means that even when two nodes simultaneously transmit, the one with stronger power still has chance to be correctly received. We introduce a parameter $0 \le \alpha \le 1$ to reflect the average impact of the capture effect, which is referred to as the *capture indicator* in this chapter, thus

$$l_{sc} = (1-\alpha) p_{sc} \tag{25}$$

$$l_{hc} = (1-\alpha) p_{hc} \tag{26}$$

To obtain p, the problem is reduced to obtaining the channel error probability l_e and the capture indicator α. We show how to obtain them via measurement in Section 4.1.4.

C. Calculating the sense busy probability b

The sense busy probability, b, is the probability that the channel becomes busy after an idle slot due to the activity of other nodes, under the condition that link k does not start its own transmission. It is equal to the probability that at least one contending link is transmitting, whether it is successful or not

$$b_k = 1 - \prod_{i \in v(k) \bigcup k} (1 - \tau_i) \tag{27}$$

D. Calculating the non-empty transmission buffer probability q

The variable q represents the probability that there is at least one packet in the queue after a transmission. In the previous models, to analyze the performance of saturated wireless networks, each node in the network is assumed to always have a packet to transmit (i.e., q=1). But according to the work in (Zhai, Chen et al., 2006), the network does not perform best when it is saturated and extensive research has been undertaken to prevent the network from saturation. So the effect of q must be considered in the model.

We introduce a parameter λ representing the rate at which packets arrive at the node buffer from the upper layers, and measured in *pkt/s*. The mean time between two packet arrivals is defined as the *mean inter-packet time*, and thus its value can be calculated as $1/\lambda$.

A crude approximation in the unsaturated setting is to assume that packet arrivals are uniformly distributed across slots and set

$$q = \min\left\{ \frac{E}{mean\ inter\text{-}packet\ time},\ 1 \right\} = \min\{\lambda \cdot E,\ 1\} \tag{28}$$

where E is the average length of the Generic slot obtained via Eq. (1) and measured in *seconds*.

If the traffic arrives in a Poisson distribution, then probability q can be well approximated in a situation with small buffer size through the following relations as (Malone, Duffy et al., 2007) and (Daneshgaran, Laddomada et al., 2008) revealed:

$$q = 1 - e^{-\lambda E} \tag{29}$$

Here the packet arrival probability is assumed independent to the channel state. A more accurate model can be derived upon considering different values of q for each backoff state. However, it has been proved in (Malone, Duffy et al., 2007) that as state-dependent models are more computational involved, there seems little advantage in employing a state-dependent model instead of the state-independent model. Thus it is a reasonable solution using a mean probability valid for the whole Markov model.

Note that, in (29), placing the node in saturation by taking the limit q->1, the model is reduced to a model for saturated scenarios.

4.1.2 Interference model

Given a set of wireless nodes, a network can be mapped into a contention graph (Chen, Low et al., 2005). This contention graph is used to represent interference (i.e. which link is interfering with which link) which has a consequent impact upon throughp4ut. We use contention graphs to model the interference between contending links. In the literature, contention graph models have not considered contention due to hidden nodes which is an important difference in our work.

The process of mapping a network topology into a contention graph is introduced in (Chen, Low et al., 2005) and (Gao, Chiu et al., 2006). To illustrate this concept, we take the 4-hop chain network in Fig. 9(a) as a simple example, where nodes on the route are placed with the transmission distance R_{tx}. And R_{CS} represents the carrier-sense range.

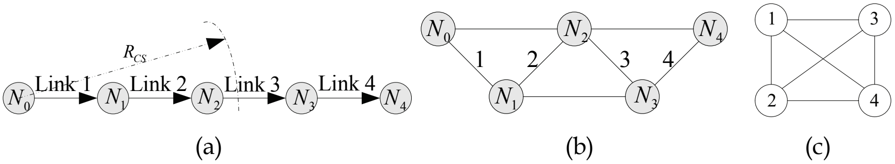


Fig. 9. Process of mapping a multi-hop route to its contention graph: (a) Example network; (b) undirected graph of the network; (c) contention graph

In Fig. 9 (b), nodes that can sense each other are connected. For instance, N_0 is connected to N_1 and N_2 because these two nodes are within the carrier-sense range of N_0 and they are considered neighbors of N_0. However N_3 and N_4 cannot be sensed by N_0 and therefore are not connected to N_0. The numbers beside each edge are used to label all active links in the wireless network, i.e., Link 1, Link 2, Link 3 and Link 4. Finally, in the contention graph in Fig. 9 (c), all active links are transformed into vertices. An edge between two vertices denotes contention between two links. This can be deduced from Fig. 9(b). Two links contend with each other when either the sender or the receiver of one link is within the R_{CS} distance of the sender or the receiver of the other, thus they are called *contending link* to each other. Note that previous work on contention graph only considered the interference due to neighboring nodes; while hidden node interferences were not modeled (i.e. in previous work there is no edge between Vertex 1 and Vertex 4 in Fig. 9(c)). In this research, we will consider interference due to both, neighboring and hidden nodes.

Note that the aggregate successful transmission time ratio of contending links in the network should not be more than 1, thus we have the following interference constraint

$$\sum_{i \in v(k)} y_i \le 1 \text{ , } \forall k \in \mathbb{N} \tag{30}$$

where $\mathbb{N}$ is the set of all active links in the given network.

4.1.3 Mapping bandwidth requirement to the model parameters

In this section, we related the bandwidth requirement of a flow, to the network parameters. For instance, to satisfy the application bandwidth requirement (*BW*, bps), given the traffic packet size (*PS*, bits), the packet arrival rate is

$$\lambda = \frac{BW}{PS \cdot \Lambda} \tag{31}$$

And according to (13), we can easily obtain that the transmission probability used for this application by a link (Link k) along the path of this application is at least

$$\tau_k = \frac{BW \cdot E_k}{PS \cdot (1 - p_k) \cdot \Lambda} = \frac{\lambda \cdot E_k}{(1 - p_k)} \tag{32}$$

Recalling equations (18) and (21), the transmission probability will further affect the packet collision thus the unsuccessful transmission probability p, which will in turn affect the transmission probability, see (32). The coupling of the network parameters relates the bandwidth requirement of a flow to all the network parameters.

4.1.4 Parameters initialization

We still need to obtain two radio-dependent parameters to complete the model. Those are the conditional capture indicator α and the channel failure probability l_e. In this section, we estimate these two parameters by conducting broadcast measurement. The key idea is that we can estimate unicast interference using broadcast packets.

First, we have one node, Node i, broadcasts packets and we keep track of the delivery rate of the packets at all other nodes in the network. Only one node is active at a time. We denote the broadcast rate as R_i and the delivery rate from Node *i* to Node *j* as R_{ij}. Then each node broadcasts in turn. We then select a pair of nodes, Node *i* and Node *k*, and have them broadcast packets together. All remaining nodes measure the delivery rate of packets they receive from each of the two broadcasting nodes. For example, at node *j*, the delivery rate of packets from *i* is denoted by $R_{ij}^{i,k}$. Then each pair of nodes simultaneously broadcast in turn. Thus, we have carried out a total of $o(n^2)$ experiments, where *n* is the number of nodes in the network.

Using the data gathered from the above methodology, we can obtain the maximum-likelihood estimators for the channel error probability for the channel from node *i* and node *j* (denoted by $l_e^{i \to j}$) and the average capture effect experienced by the link from node *i* to node *j* (denoted by α_{ij}) as:

$$l_e^{i \to j} = \frac{R_i - R_{ij}}{R_i} \tag{33}$$

$$\alpha_{ij} = \sum_{k \in \mathbb{N}, k \ne i} \tau_i \tau_k \frac{R_{ij}^{i,k}}{R_{ij}} \tag{34}$$

4.2 Model-based algorithms for AB prediction

We have built up a model considering the bandwidth requirement of a new flow and some other parameters: transmission probability, collision probability. After constructing the contention graph for a given network, we can easily perform admission control and end-to-end AB estimation in order to guarantee throughputs to applications in multi-hop wireless networks.

4.2.1 Admission control

```
▷ Input: bandwidth requirement, i.e., BW, of the incoming flow;
         given route Γ={$N_0$, $N_1$, ..., $N_r$}
▷ Output: whether the flow can be admitted
**Initialization:** $admission = 0; \tau_k = \tau_k^{old} + \frac{BW \cdot E_k^{old}}{PS \cdot (1 - l_e) \cdot \Lambda}, k = 1,2,...,r$
  // iterative admission control ($MaxIter = 20$ and $THD = 0.01$ by default)
**for** $iter = 1$ to $MaxIter$
    $update(\{p_{i \in v(k)}, \tau_{i \in v(k)}\}$ and $\{p_k, \tau_k\})$   // according to (18) and (20)
    $calculate(y_{i \in \mathbb{N}})$   // according to (12)
    **if** any $k$ satisfies $(\sum_{i \in v(k)} y_i > 1)$   // interference constraint is violated
        $admission = 0$; break   // early stop: reject
    **end if**
    $\tau_k{}^{\circ} = \frac{BW \cdot E_k}{PS \cdot (1 - p_k) \cdot \Lambda}, k = 1,2,...,r$
    **if** $(\max_{k=1,2,...,r} \{|\tau_k{}^{\circ} - \tau_k|\} < THD)$   // convergence test
        $admission = 1$; break   // early stop: admit
    **end if**
**end for**
**return** $admission, \tau_k{}^{\circ}$
```

Table 1. Admission control

Given the bandwidth requirement of a coming flow, the goal of admission control is to make a decision on whether the requesting flow can be admitted without impairing the QoS of existing flows. The main challenge is that we cannot make the accurate decision according to the network states before the flow entered because the entrance of the flow will change the transmission probability and collision probability. So the idea in this research is to adopt a what-if analysis, namely to check what will happen if the new flow is admitted. Since there is strong inter-dependency between the transmission probability and the loss rate of contending links: the transmission probability of Link k, τ_k, depends on its packets loss probability as well as the transmission probability of its contending links, which in turn depends on τ_k . (refer to Eq. (18) and (21)). To address the inter-dependency, we use an iterative procedure to jointly estimate the transmission probabilities and loss probabilities.

We initialize that after a new flow entering, the collision probabilities (including collisions due to both synchronized nodes and hidden nodes) at all links for this flow are zero. We then iteratively update link transmission probabilities and packet loss probabilities based on the other links' transmission probabilities and loss probabilities derived in the previous iteration. The iterative procedure continues until the number of iterations reaches a threshold (MaxIter), or the transmission probability no longer change significantly (Less than a threshold THD), or a interference constraint (see (30)) is violated. The algorithm is outlined in Table 1.

In line 1, τ_k^{old} and E_k^{old} are the corresponding parameters estimated on Link k before the entrance of the new flow. If there is no traffic on Link k before the entrance of the new flow, then $\tau_k^{old} = 0$ and $E_k^{old} = T_k$. This algorithm performs the admission control along a given route, and it also calculates the sending rate of the sender to guarantee the bandwidth requirement (obtained via Line 8). This algorithm can also help to perform route selection, namely find a route that can support the requested bandwidth.

4.2.2 End-to-end AB prediction

```
▷ Input: given route $\Gamma=\{N_0, N_1,..., N_r\}$;
▷ Output: $\lambda$ (the available bandwidth of $\Gamma$)
initialization : $\lambda = \lambda_0 / 2$   // $\lambda_0$ is the theoretic maximum capacity
  // iterative process (MaxIter = 20 and THD = 1kbps by default)
for $i = 2$ to MaxIter
    $\lambda° = \frac{1}{2^i}\lambda_0$
    if ($\lambda° < THD$) // convergence test
        break // early stop
    end if
    if (admission control($\lambda, \Gamma$))
        $\lambda = \lambda + \lambda°$
    else
        $\lambda = \lambda - \lambda°$
    end if
end for
return $\lambda$
```

Table 2. End-to-end AB prediction

Let's exploit the following property in 802.11 networks (Kun, Fan et al., 2007): if the throughput of λ is feasible along a given route without violating the QoS of ongoing traffic, all throughputs smaller than λ are also feasible; while if the throughput is unfeasible, all the values larger than λ are also unfeasible. Thus, we can increase the value of λ until it is not feasible to find the end-to-end AB of path Γ without breaking the QoS demands of all existing traffic. Hence the solution can be obtained with logarithmic complexity by applying a binary search algorithm (half the search space each time).

It is worth mentioning that to find the end-to-end AB is different to performing admission control, the latter is only the answer to whether a flow along a given route with a specific bandwidth requirement can be admitted, while the former need to further find out the maximum bandwidth of a flow that can be admitted. Table 2 outlines the algorithm, which takes the admission control as a sub function.

In Line 1, λ_0 is the theoretic maximum capacity, which is the upper bound of our algorithm's searching space. Since the algorithm will converge very fast, the accuracy of this value will not affect the result significantly only if it is bigger than the estimated end-to-end AB. In an n-hop network, representing C as the channel physical capacity, λ_0 is set according to the following equations (i.e., the maximum capacity is limited by the number of hops due to the existence of intra-flow contention):

$$\lambda_0 = \begin{cases} C/n, & 1 \le n \le 4 \\ C/4, & n > 4 \end{cases} \tag{35}$$

5. Conclusion

With the IEEE 802.11-based ad hoc networks deployed as the vital extension to wired networks and the widespread use of multimedia applications that require QoS support, AB estimation is such an important operation that it is very necessary for research community to create an effective, general-purpose estimation method. This chapter reviews the state-of-the-art of AB estimation in IEEE 802.11-based ad hoc networks, gives an analysis of the challenges on this topic. The analysis mainly focuses on fundamental problems, which rise from the nature of wireless networks and operation of DCF mode. To develop estimation tools that can work accurately in 802.11 or 802.11-alike ad hoc networks, researchers are expected to think over all these challenges. It then gives some solutions to these challenges. In particular, it presents our solutions to improve the accuracy of sensing-based AB estiamtion and model-based AB predictation. We hope that this analysis can help to spur further work on this topic.

6. Acknowledgements

This work is partly supported by the National Natural Science Foundation of China (Grant No. 61002032).

7. References

Aguayo, D.;Bicket, J., et al. (2004). Link-level measurements from an 802.11b mesh network. *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, Portland, Oregon, USA, ACM.

Ahn, G.-S.;Campbell, A. T., et al. (2002). Supporting service differentiation for real-time and best-effort traffic in stateless wireless ad hoc networks (SWAN). *IEEE Transactions on Mobile Computing* 1(3): 192–207.

Bianchi, G. (2000). Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications* 18(3): 535-547.

Chatzimisios, P.;Boucouvalas, A. C., et al. (2003). Influence of channel BER on IEEE 802.11 DCF. *Electronics Letters* 39(23): 1687-9.

Chen, K.;Xue, Y., et al. (2004). Understanding bandwidth-delay product in mobile ad hoc networks. *Computer Communications* 27(10): 923-934.

Chen, L. and Heinzelman, W. B. (2005). QoS-aware routing based on bandwidth estimation for mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications* 23(3): 561-572.

Chen, L.;Low, S. H., et al. (2005). Joint congestion control and media access control design for ad hoc wireless networks. *Proceedings of IEEE INFOCOM*.

Daneshgaran, F.;Laddomada, M., et al. (2008). Unsaturated Throughput Analysis of IEEE 802.11 in Presence of Non Ideal Transmission Channel and Capture Effects. *IEEE Transactions on Wireless Communications* 7(4): 1276-1286.

Dao, N. T. and Malaney, R. A. (2008). A New Markov Model for Non-Saturated 802.11 Networks. *5th IEEE Consumer Communications and Networking Conference (CCNC)*.

de Renesse, R.;Friderikos, V., et al. (2007). Cross-layer cooperation for accurate admission control decisions in mobile ad hoc networks. *IET Communications* 1(4): 577-586.

de Renesse, R.;Ghassemian, M., et al. (2004). QoS enabled routing in mobile ad hoc networks. *Fifth IEE International Conference on 3G Mobile Communication Technologies (3G 2004)*

Dovrolis, C.;Ramanathan, P., et al. (2004). Packet-dispersion techniques and a capacity-estimation methodology. *IEEE/ACM Transaction on Networking* 12(6): 963-977.

Ergen, M. and Varaiya, P. (2005). Throughput analysis and admission control for IEEE 802.11a. *Mobile Network Applications* 10(5): 705-716.

Ergen, M. and Varaiya, P. (to appear). Throughput analysis and admission control in IEEE 802.11a. *ACM-Kluwer Mobile Networks and Applications, Special Issue on WLAN Optimization at the MAC and Network Levels*.

Gao, Y.;Chiu, D.-M., et al. (2006). Determining the end-to-end throughput capacity in multi-hop networks: methodology and applications. *Proceedings of ACM SIGMETRICS*.

Gupta, R.;Musacchio, J., et al. (2007). Sufficient rate constraints for QoS flows in ad-hoc networks. *Ad Hoc Networks* 5(4): 429–443.

Hoang, V. D.;Shao, Z., et al. (2006). A New solution to Estimate the available Bandwidth in MANETs. *IEEE 63rd Vehicular Technology Conference (VTC 2006-Spring)*.

Hu, N. and Steenkiste, P. (2003). Evaluation and characterization of available bandwidth probing techniques. *IEEE Journal on Selected Areas in Communications* 21(6): 879-894.

Jae-Yong, Y. and JongWon, K. (2007). Maximum End-to-End Throughput of Chain-Topology Wireless Multi-Hop Networks. *Proceedings of IEEE WCNC*.

Jain, M. and Dovrolis, C. (2003). End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput. *IEEE/ACM Transactions on Networking* 11(4): 537-549.

Johnsson, A.;Melander, B., et al. (2005). Bandwidth Measurement in Wireless Network. Sweden, Malardalen University.

Kapoor, R.;Chen, L.-J., et al. (2004). Capprobe: A simple and accurate capacity estimation technique. *Proc. of ACM SIGCOMM*.

Kuan, C. and Dimyati, K. (2006). Analysis of collision probabilities for saturated IEEE 802.11 MAC protocol. *Electronics Letters* 42(19).

Kumar, A.;Altman, E., et al. (2007). New insights from a fixed-point analysis of single cell IEEE 802.11 WLANs. *IEEE/ACM Transaction on Networking* 15(3): 588-601.

Kun, W.;Fan, Y., et al. (2007). Modeling path capacity in multi-hop IEEE 802.11 networks for QoS services. *IEEE Transactions on Wireless Communications* 6(2): 738-749.

Lakshminarayanan, K.;Padmanabhan, V. N., et al. (2004). Bandwidth estimation in broadband access networks. *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement (IMC)*, Taormina, Sicily, Italy, ACM.

Lao, L.;Dovrolis, C., et al. (2006). The probe gap model can underestimate the available bandwidth of multihop paths. *SIGCOMM Computer Communication Review* 36(5): 29-34.

Li, Y.;Qiu, L., et al. (2008). Predictable performance optimization for wireless networks. *Proceedings of the ACM SIGCOMM 2008 conference on Data communication (SIGCOMM)*, Seattle, WA, USA, ACM.

Malone, D.;Duffy, K., et al. (2007). Modeling the 802.11 Distributed Coordination Function in Nonsaturated Heterogeneous Conditions. *IEEE/ACM Transactions on Networking* 15(1): 159-172.

Melander, B.;Bjorkman, M., et al. (2000). A new end-to-end probing and analysis method for estimating bandwidth bottlenecks. *Proceedings of IEEE GLOBECOM.*

Nafaa, A. (2007). Provisioning of multimedia services in 802.11-based networks: facts and challenges. *IEEE Wireless Communications* 14(5): 106-112.

Perkins, C. E.;Royer, E. M., et al. (2001). Ad hoc on-demand distance vector (AODV) routing.

Qiao, D.;Choi, S., et al. (2002). Goodput analysis and link adaptation for IEEE 802.11a wireless LANs. *IEEE Transactions on Mobile Computing* 1(4): 278-292.

Qiu, L.;Zhang, Y., et al. (2007). A general model of wireless interference. *Proceedings of ACM Mobicom*, Montral, Qubec, Canada, ACM.

Ribeiro, V. J.;Riedi, R. H., et al. (2003). PathChirp: efficient available bandwidth estimation for network paths. *Passive and Active Measurement Workshop.*

Sanzgiri, K.;Chakeres, I. D., et al. (2004). Determining intra-flow contention along multihop paths in wireless networks. *Proceedings of First International Conference on Broadband Networks (BroadNets)*

Sarr, C.;Chaudet, C., et al. (2008). Bandwidth Estimation for IEEE 802.11-Based Ad Hoc Networks. *IEEE Transactions on Mobile Computing* 7(10): 1228-1241.

Strauss, J.;Katabi, D., et al. (2003). A measurement study of available bandwidth estimation tools. *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement (IMC)*, Miami Beach, FL, USA, ACM.

Sun, T.;Chen, L.-J., et al. (2005). SenProbe: path capacity estimation in wireless sensor networks. *the third Intl. Workshop on Measurement, Modelling, and Performance Analysis of Wireless Sensor Networks (SenMetrics).*

Wu, H.;Wang, X., et al. (2005). SoftMAC: layer 2.5 MAC for VoIP support in multi-hop wireless networks. *Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON).*

Xu, K.;Tang, K., et al. (2003). Adaptive bandwidth management and QoS provisioning in large scale ad hoc networks. *IEEE Military Communications Conference (MILCOM).*

Yang, Y. and Kravets, R. (2005). Contention-aware admission control for ad hoc networks. *IEEE Transactions on Mobile Computing* 4(4): 363-377.

Zhai, H.;Chen, X., et al. (2005). How well can the IEEE 802.11 wireless LAN support quality of service? *IEEE Transactions on Wireless Communications* 4(6): 3084-3094.

Zhai, H.;Chen, X., et al. (2006). A call admission and rate control scheme for multimedia support over IEEE 802.11 wireless LANs. *ACM Wireless Networks* 12(4): 451-463.

Zhao, H.;Garcia-Palacios, E., et al. (2009). Accurate Available Bandwidth Estimation in IEEE 802.11-Based Ad Hoc Networks. *Computer Communications* 32(6): 1050-1057.

Zhao, H.;Wang, S., et al. (2009). Challenges to Estimate End-to-end Available Bandwidth in IEEE 802.11-based Ad hoc Networks *Proc. of 2009 IEEE Youth Conference on Information, Computing and elecommunication* Beijing, China

Zhao, H.;Wang, S., et al. (2010). Modeling Intra-Flow Contention Problem in Wireless Multi-hop Networks. *IEEE Communications Letters* 14(1): 18-20.

Zhou, H.;Wang, Y., et al. (2006). Difficulties in Estimating Available Bandwidth. *Proc. of IEEE International Conference on Communications (ICC).*

5

Mathematic Models for Quality of Service Purposes in Ad Hoc Networks

Khalil Amine
Department of Mathematics
Faculty of Sciences and Techniques, University of Sidi Mohammed Ben Abdellah
Fès, Morocco

1. Introduction

The quality of service, according to a networking context, is the degree of users' satisfaction of services that a communication system provides. It aims at improving communication behaviour under a correct data transmission and an optimal use of resources. According to this concept, quality of service is typically the performance criteria that evaluate the service provided.

Wireless multi-hop networks, including ad hoc networks, with their complex nature impose many constraints than in wired networks. Besides, the quality of service concerns the behaviour of the network, and is dealt with from different points of view. It typically addresses a set of metrics relevant to delay, bandwidth, jitter, packet loss rate, energy consumption, stability, security, and so on. It is worth noting, accordingly, that some criteria are very difficult to discern and can be still considered challenging. In this regard, security is not sufficiently addressed in a QoS context in ad hoc networking studies.

Inside the ad hoc networking field, the quality of service issues concern different layers on the network architecture. We distinguish between too main optics of quality of service studies: QoS models and QoS routing. A QoS model defines all mechanisms that the network should respect in order to guarantee the quality of service on the network. The model is based on and characterizes the architecture of the network. It includes all protocols that organize communication and connectivity between the different layers or components. In this respect, QoS routing presents a critical component in the model and a rich field for algorithm development. QoS routing consists of finding the best path to relay a source to a destination and guarantee the quality of service in parallel. In a general case, QoS routing consists to define metrics (usually one) that control the decision making to choose a path. The metrics nature affects the mathematical model and then the algorithm approach used to solve the problem of finding the best path.

Mathematical modelling of quality of service in ad hoc networking aims at improving the decision making on networks in an operational meaningful way. It addresses many concerns of the QoS and allows to benefit from various modalities and techniques of optimization theory.

This work reviews and discusses several mathematical models developed for or oriented to quality of service in ad hoc networks, and highlights the different forms that a model may take. Because we believe that several metrics of quality of service in ad hoc networks are in contradiction and thus the multicriteria optimization gives more opportunities in decision making and in reflecting the realistic system; we opt for multicriteria formulations in all our studies of QoS in ad hoc networks. In this regard, we present and analyze the contributions of such studies on improving ad hoc networks' behaviour.

2. Quality of service issues and ad hoc networks challenges

The quality of service presents a rich field for study in ad hoc networking until *the Internet QoS protocols cannot be migrated to the wireless environment* (Sarkar et al., 2007) and in general because of the difficulty to support all communication patterns that are usual in wired networks. The quality of service concerns the behaviour of the network, and is dealt with from different points of view. Furthermore, it refers to different notions at different networking layers (Sinha, 2005). We distinguish between two main optics of quality of service studies: QoS models and QoS routing. QoS models are architectures providing all mechanisms which governs some properties as time, scheduling and reliability (Brahma, 2006). Several models have been developed in the literature with different intentions, as FQMM (Xiao et al., 2000), SWAN (Ahn et al., 2002), INSIGNIA (Lee et al., 2000), dRSVP (Mirhakkak et al., 2000), etc. They are based on or a hybridization of the classical models, namely, InteServ and DiffServ which have proved their limits in ad hoc network environment[1]. As for QoS routing, it consists to find the route relaying source and destination and insuring the quality of service in parallel. Many issues are considered recently giving more areas of research in QoS concerns.

Essentially, quality of service concerns four topics in the current studies, namely, QoS models, QoS resource reservation, QoS routing, and medium access control protocol. However, from a mathematical point of view, QoS presents an economic problem which consists to optimize an objective cost under several constraints imposed by the network nature and proprieties. In this connection, quality of service typically addresses a set of metrics relevant to delay, bandwidth, jitter, packet loss rate, energy consumption, stability, security, and so on.

QoS in ad hoc networks contends with several vulnerabilities of the networks nature, as contention, waiting, and intrusion phenomena (see Fig. 1). Each one of these vulnerabilities may present a field of study in the context of QoS in ad hoc networking. Indeed, investigating intrusion in ad hoc networks may improve the connectivity which gives more guarantee of the QoS in this field.

In the same respect, contention may present a wide area in QoS studies in particular in the resource allocation strategies. Indeed, the wireless channel is shared by nodes with the same neighborhood. As shown in Fig. 1, node L gets into competition with the node D which receives both signals coming from node S (data source) and node L (noise source).

3. Mathematical QoS models

A mathematical model is a virtual maquette which represents a concrete phenomenon with mathematical symbols and semantics. In other words, it consists to describe the real-world phenomena by using scientific conceptions. Each modelling process starts with a question that can be investigated in order to describe, explain and predict the evaluation of a phenomenon. The nature of this question defers according to the approach adopted to study the phenomenon. Therefore, mathematical models are considered inside a set of mathematical fields or contexts, as operations researcher, control theory, automatic engineering, etc.

[1] IntServ model (Integration of Service) poses a problem of high volume of flow treatment and signalization, as for DiffServ model (Differentiation of Service), it requires a static topology and a network core with high bandwidth.

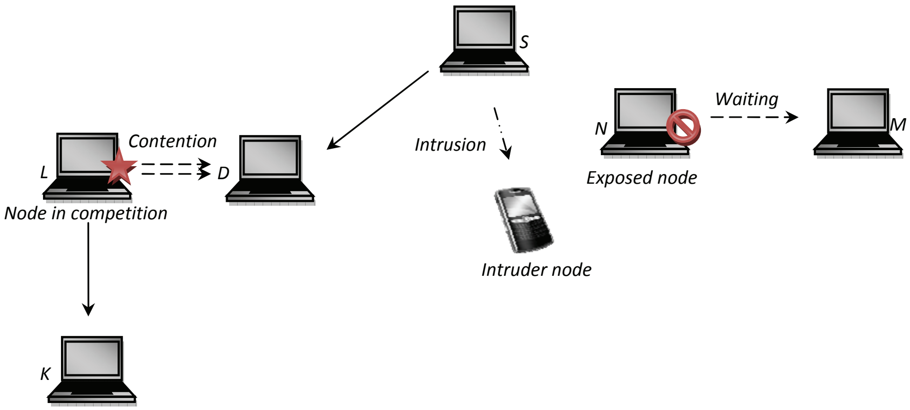


Fig. 1. Ad hoc networks challenges: contention, waiting, and intrusion phenomena (Amine, 2008).

Furthermore, the mathematical models differ according to the objectives and the constraints considered in the formulation. In a general case, the modelling process is described in Fig. 2. Mathematical modelling of quality of service in ad hoc networking aims at improving the decision making on networks in an operational meaningful way. A mathematical modelling consists of determining objective-functions (criteria), which reflect the cost of communication in the network, as well as the constraints which limit the decision domain. In this context, metrics of QoS are considered as objective functions, and services, network proprieties and topology requirement are considered as constraints. Various studies were developed in the literature which follow this methodology with various modalities of modelling and optimization.

In networking science, protocols are defined as all mechanisms which govern the network behaviour in term of communication, transmission, routing, resource allocation, and so on. In this connection, a routing protocol deals with three issues, namely, the routes discovery, routes maintenance and data transmission. We distinguish according to the first issue two categories of protocols: Topology based and spatial position based (Amine, 2008). Besides, all protocols should respect free loop, sleep period considerations, security, and so on. From a mathematical point of view, protocols investigation should focus on the decision phase which takes place, evidently, before the transmission data process.

QoS routing consists of finding the best path to relay a source to a destination and guarantee the quality of service in parallel. It represents an advanced concept of best effort routing previously developed in ad hoc networking. In a general case, QoS routing consists to define metrics (usually one) that control the decision making to choose a path. The metrics nature affects the mathematical model and then the algorithm approach used to solve the problem of finding the best path.

Ad hoc networks are usually represented as a weighted directed complete graph $G(V, A)$ where V represents the set of vertices associated with the nodes of the network and A the set of edges representing all the possible communication links; i.e. $A = \{(i,j)/i,j \in V, i \neq j\}$. The aim of studying QoS is, in general, to determine an optimal path according to well defined objectives; subject to a set of well defined constraints that characterize the nature of the network.

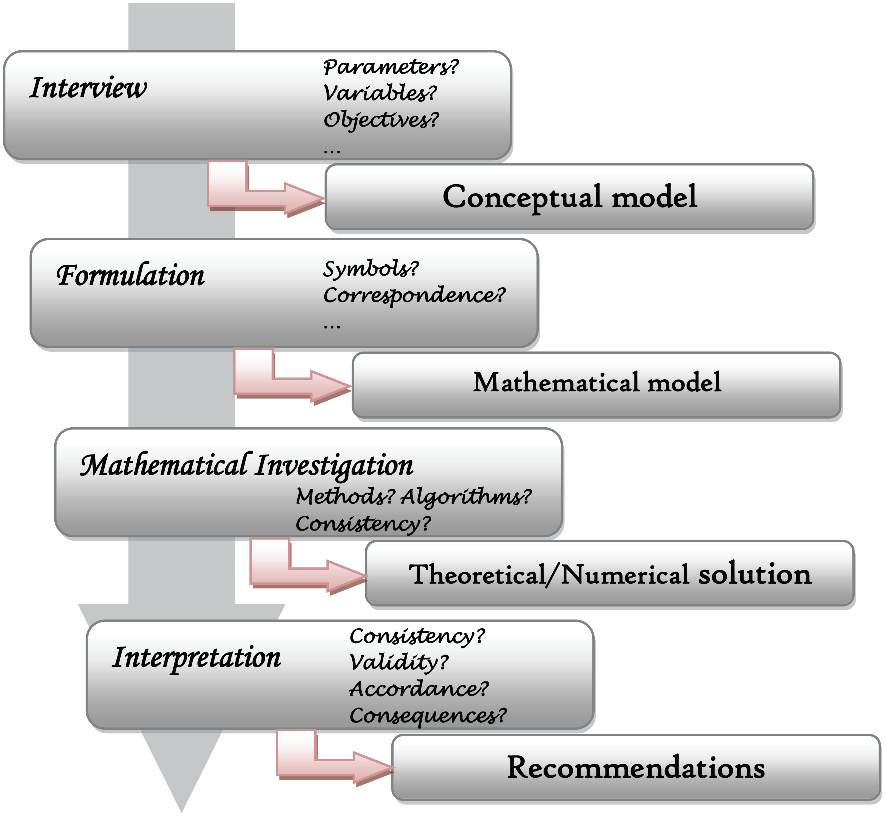


Fig. 2. Modelling process: a technical target leading from a phenomenon observation to a scientific result.

Let $\mathcal{P}$ stand for routing path from a source s of data (massage) to a destination d . We introduce the decision variable x defined as

$$x_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{P} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The transmission model usually adopted in ad hoc networking is defined as

$$p_{ij} = d_{ij}^{\alpha} \tag{2}$$

where p_{ij} is the energy per bit required to reach j from i and d_{ij} is the distance from i to j . α is an environment-dependent coefficient typically between 2 and 5 (Hashemi et al., 2007). The transmission energy is well considered to be symmetric, i.e.

$$p_{ij} = p_{ji} \qquad ; \qquad \forall (i,j) \in A \tag{3}$$

4. Multicriteria approaches: overview and contributions

A multicriteria approach consists of taking into account all objective costs (criteria) that reflect the conceptual system problematic[2]. The consistency of a multicriteria model, takes from the contradiction between objectives functions; the optimization of a criterion affects the others. Thus, the optimization of all criteria cannot be done separately.

A mathematical multicriteria model can be represented in a general case as

$$\begin{cases} \min \left(f_1(x), f_2(x), \ldots, f_m(x)\right) \\ \text{Subject to} \\ \quad h_i(x) = 0 \qquad , \qquad i = 1, \ldots, k \\ \quad k_i(x) \leq 0 \qquad , \qquad i = 1, \ldots, l \\ \qquad x \in \mathcal{D} \subset \mathbb{R}^n \end{cases}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$ are the objective functions, h_i and k_i are equality and inequality constraints respectively, and $\mathcal{D}$ is the domain in which the functions f_i are defined. The multicriteria optimization consists of finding the best compromise between all functions. This compromise is actually a set of solutions that are nondominated[3] by any feasible (realizable) solution. This set of all optimal solutions (in nondominance sense) is called *Pareto Front*.

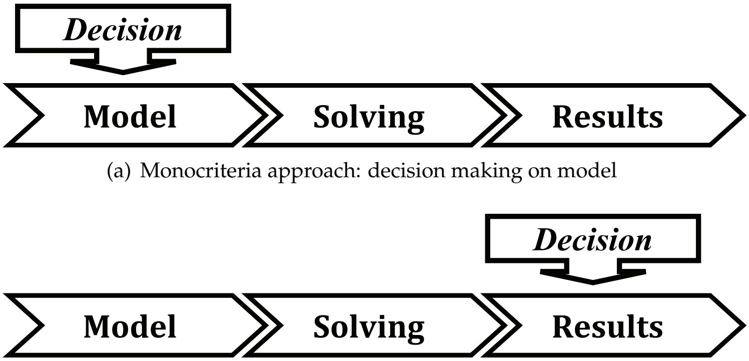
Because quality of service refers at the first stage to the users' satisfaction, QoS strategies should give more choices at different ways. Besides, multicriteria optimization provides all alternatives representing optimal solutions according to different preference structures. In this, each alternative/solution is called *best* or *efficient* solution. Despite goal programming, aggregation methods, and scalarisation process which provide one solution. Multicriteria optimization involves providing all solutions representing the best compromise between all considered criteria.

The difference between monocriteria and multicriteria approach lies in time of decision making. Indeed, the monocriteria formulation consists of making decision on mathematical model (Fig. 3(a)) and process with a classical mathematical programming methods in order to have the optimal solution. Despite of multicriteria formulation for which making decision

[2] The systemic theory shows that each concrete system is governed by a set of parameters that are related to each other.

[3] For more information about nondominance order in the multicriteria optimization field, see (Ehrgott, 2005).

consists of the choice of an adequate solution from all efficient (best) solutions given by a multicriteria programming (Fig. 3(b)).



(a) Monocriteria approach: decision making on model

(b) Multicriteria approach: diversity of choice of solutions

Fig. 3. Modality of modelling: difference between monocriteria and multicriteria approach (Amine et al., 2009).

Techniques of transformation of a multicriteria model to a monocriteria formulation limit the choice of solutions. As against, multicriteria formulation makes possible making decision on the basis of local requirements. That is, the decision making is related especially to the node instead to all the network.

The choice of the adequate path is granted to the node (or the user) in order to respect local requirements which are difficult to be integrated into formal model and which change according to the circumstances. In the same connection, the diversity of solutions given, makes possible to have a backup path for each transmission in the network. Therefore, the approach may give more robustness and reliability to transmission behaviour, and in this context, an architecture, on the basis of a decision making system which controls the ad hoc network behaviour and guarantees the QoS, can be proposed.

5. Mathematical formulations

Several studies in ad hoc networking dealing especially with energy consumption, delay, fair scheduling, and so on, are considered as QoS oriented studies. This studies focus on a metric or a problematic from the ad hoc challenges, and aim at improving this issue.

In this respect, several studies have addressed the energy consumption, in ad hoc networks, that aim at improving techniques for energy conservation. These studies concern an important issue that affects the network behaviour. Indeed, the embedded energy in nodes of ad hoc networks are limited and hardly recharged or replaced, especially for military applications and sensor networks, therefore a meaningful strategy of energy control is desirable.

Energy constraint, from an algorithmic view, has been studied around two areas: minimal energy broadcasting problem and minimal energy multicasting problem. Broadcasting and multicasting represent a crucial issue in most of protocols in ad hoc networking field, because of their usefulness in discovering routes and transmission process.

Minimum Energy problem of Broadcast (MEB) and Multicast (MEM) is considered from a conceptual perspective as the problem of finding the arborescence with the minimum power cost.

5.1 Minimum energy problems

5.1.1 Minimum energy problem of broadcast

Broadcasting is the process to send a message from a source to all active nodes in the network. It is characterized by no requirement of acknowledgement messages. The source is therefore called deaf source. This aspect of transmission, also known in some contexts as flooding, is an important issue in many network techniques as topology discovery and routing table construction.

Minimum Energy Broadcast problem, marked as MEB, consists to find a tree (arborescence) originated in the source s and relaying all active nodes; subject to minimizing the tree energy cost.

A probabilistic formulation of the MEB was presented in (Montemanni et al., 2008). It consists in associating each node i with a value $q_i \in]0,1[$ representing a probability that the node i still be active during the operability of the network. The approach is based on a level $\alpha \in]0,1[$ where a path is regarded as feasible if his probability, which is equal to the product of node probabilities, is greater or equal to α. The model developed is presented bellow:

$$\left\{ \begin{array}{lll} \min \sum\limits_{i \in V} y_i & & \\ \text{subject to} & & \\ \qquad y_i \geq p_{ij} z_{ij} & ; & \forall (i,j) \in A \\ \sum\limits_{\substack{(i,j)\in A \\ i \in S, j \in V \setminus S}} z_{ij} \geq 1 & ; & \forall S \in V \quad , \quad s \in S \\ \sum\limits_{(i,j)\in \mathcal{P}} z_{ij} \leq |P| - 1 & ; & \forall \mathcal{P} \in \mathcal{U} \\ \qquad z_{ij} \in \{0,1\} & ; & \forall (i,j) \in A \\ \qquad y_i \in I\!R^+ & ; & \forall i \in V \end{array} \right.$$

This model uses notations described in Tab. 1 above.

Symbol	Description
y_i	Transmission power of each node i
z_{ij}	Variable characterizing the optimal tree $\mathcal{T}$, i.e., $z_{ij} = \left\{ \begin{array}{lll} 1 & , & \text{if } (i,j) \in \mathcal{T} \\ 0 & , & \text{otherwise} \end{array} \right.$
$\|P\|$	Number of arcs belonging to path $\mathcal{P}$
$\mathcal{U}$	Set of all infeasible paths originated in s

Table 1. Minimum energy broadcast problem model description.

This model was the subject to develop two other models with more probabilistic aspect, namely cumulative probability formulation and multi-commodity flow formulation.

5.1.2 Minimum energy problem of multicast

Multicasting can be considered as a particular case of broadcasting; it consists to send a message from a source to more than one node in the network. The acknowledgment messages are required in this aspect though. Multicasting advantage consists in saving the embedded energy when the destination nodes are many, face to strategies that repeat transmission as many times as the number of destinations.

A QoS-MEM was formulated in (Guo & Yang, 2004) as a mixed integer linear programming model on the basis of the idea to extract a subgraph T_s^*, initiated in the source s and representing the bandwidth constrained multicast tree with minimum energy consumption, from the undirected graph $G(V, A)$. Let's consider the binary variable z characterizing T_s^* as

$$z_{ij} = \begin{cases} 1 & , \text{ if } (i,j) \in T_s^* \\ 0 & , \text{ otherwise} \end{cases} \tag{4}$$

and $t_{ij\kappa}$ is a binary variable which is equal to one if node i is scheduled to transmit to node j at slot κ, and zero otherwise. Let F_{ij} be a non-negative continuous variable representing fictitious flow produced by the multicast initiator s going through arc (i,j), and $q_{i\kappa}$ be a non-negative continuous variable representing the transmission power of the node i at slot κ.
Let's consider the following notation summarized in Tab. 2.
The model is so presented as follows

$$\begin{cases}
\min \sum_{i \in N} \sum_{i \in FS_i} q_{i\kappa} & \\
\text{subject to} & \\
\sum_{\{j\,/\,(j,i) \in A\}} z_{js} = 0 & \\
\sum_{\{j\,/\,(j,i) \in A\}} z_{ji} = 1 & , \forall i \in M \setminus \{s\} \\
\sum_{\{j\,/\,(j,i) \in A\}} z_{ji} \leq 1 & , \forall i \in N \setminus M \\
\sum_{\{j\,/\,(j,i) \in A\}} z_{ij} \leq (n-1) \sum_{\{j\,/\,(j,i) \in A\}} z_{ji} & , \forall i \in N \setminus M \\
\sum_{\{j\,/\,(j,i) \in A\}} F_{ji} - \sum_{\{j\,/\,(i,i) \in A\}} F_{ij} = \sum_{\{j\,/\,(j,i) \in A\}} z_{ji} & , \forall i \in N \setminus \{s\} \\
z_{ji} \leq F_{ji} \leq (n-1) z_{ji} & , \forall i \in N \setminus \{s\}, (j,i) \in A \\
t_{ji\kappa} \leq z_{ji} & , \forall (j,i) \in A, \forall \kappa \in FS_j \\
\sum_{\{j\,/\,(j,i) \in A\}} \sum_{\{\kappa \in FS_j\}} t_{ji\kappa} = B \sum_{(j,i) \in A} z_{ji} & , \forall u \in N \\
\sum_{\{j\,/\,(j,i) \in A\}} t_{ij\kappa} \leq (n-1) \left(1 - \sum_{\{j\,/\,(j,i) \in A\}} t_{ji\kappa}\right) & , \forall \kappa \in S, \forall i \in N \\
\frac{q_{ji}}{d_{ji}^{\alpha}} - \gamma \left(\eta + \sum_{\substack{\{x\,/\,(x,i) \in A\} \\ x \neq j}} \frac{q_{x\kappa}}{d_{xi}^{\alpha}}\right) \geq \beta \left(t_{ji\kappa} - 1\right) & , \forall \kappa \in S, \forall (j,i) \in A \\
q_{i\kappa} \leq p_i^{max} & , \forall i \in N \\
z_{ju}, t_{ji\kappa} \in \{0,1\} & , \forall (j,i) \in A, \quad \forall \kappa \in S
\end{cases}$$

This model contains two main categories of constraints: *rooted tree constraints* which characterize the multicast tree relaying the source s with all destinations. And *bandwidth constraints* defining the bandwidth QoS constraint in the MEM context; they reflect the conditions that bandwidth allocated on each link of the multicast tree should be conflict-free and meet the bandwidth requirement.
The Branch and Cut and Cutting Planes are proposed to provide efficient solution in static ad hoc networks with few nodes. This work, among other, was uncovered the challenge to cope with, in the energetic context, large scale networks, dynamic topology (in the mobility

Symbol	Description
n	Cardinality of V set.
M	The set of multicast nodes including the source node and all destination nodes.
α	Propagation loss exponent.
S	Set of time slots into which the bandwidth is partitioned.
TS_i	Set of transmission schedule of node i, defined as the power assignment in each time slot.
FS_i	A set of free time slots at node i defined as $FS_i = \{\kappa / P_{i\kappa} > 0, \kappa \in S\}$.
p_i^{max}	A maximum power value that a node i can use.
λ	The minimum signal to interference plus noise ratio.
η	The thermal noise at every receiver.

Table 2. Minimum energy multicast problem model description.

sense), and networks equipped with directional antenna.

This model presents a kind of QoS constrained MEB/MEM problem in static ad hoc networks or slowly-mobile ad hoc networks. In the same respect, several studies was addressed in the literature (Hashemi et al., 2007; Leggieri et al., 2008; Li et al., 2007; Montemanni & Gambardella, 2005; Yuan et al., 2008; etc) taking into account to find the covering tree with the exigency that all nodes still connected for broadcasting or multicasting purposes under the objective to minimize the energy consumption. In other words, it is the problem of assigning transmission power to the nodes in such a way that the total energy consumption over the network is minimized, subject to all the nodes still be connected. In this respect, a simplified formulation presented in (Leggieri et al., 2008) is stated as

$$\begin{cases} \min \sum\limits_{i \in V} p_{ij} z_{ij} \\ \text{subject to} \\ \sum\limits_{i \in S} \sum\limits_{j \in K^i(S)} z_{ij} \quad \geq \quad 1 \quad , \quad \forall S \subset V, s \in S, D \cap S^c \neq \emptyset \\ \qquad\qquad x_{ij} \quad \in \{0,1\} \, , \qquad\qquad \forall (i,j) \in A \end{cases}$$

where the decision variable z_{ij}, characterizing the multicast tree as defined before, equal one if the energy available in node i can reach the node j, and zero otherwise.

Different solution methods were proposed in the literature for MEB/MEM problem with both exact and heuristic algorithms, discussing few difficulties of the proposed formulations as the number of some constraints and the efficiency of the returned solutions.

For more information about MEB/MEM problem, (Guo & Yang, 2007) provides a better understanding of the research challenges of energy-aware in ad hoc networks and gives an overview of methods and protocols developed according to this context.

5.2 Bicriteria model

In (Guerriero et al., 2009), authors have developed a model on the basis of the minimum path model introduced in (Skriver & Andersen, 2000), and accounts for the energy reserve and the link stability of mobile nodes.

$$\left\{\begin{array}{l}
\min \sum\limits_{(i,j)\in A} m_{ij}(t)x_{ij} \\
\min \sum\limits_{(i,j)\in A} n_{ij}(t)x_{ij} \\
\text{subject to} \\
\qquad\qquad x_{ij}\,T_{ij}\,P_{ij}(t) \le E_{res_i} \qquad , \qquad \forall\,(i,j)\in A \\
\qquad\qquad x_{ij}\,T_{ij}\,Q_{ij}(t) \le \delta E_{res_j} \qquad , \qquad \forall\,(i,j)\in A \\
\sum\limits_{\{j\,/\,(i,j)\in A\}} x_{ij} - \sum\limits_{\{j\,/\,(i,j)\in A\}} x_{ji} = \begin{cases} 1 & \text{if} \quad i=s \\ 0 & \text{if} \quad i\in\mathcal{N}\setminus\{s,d\} \\ -1 & \text{if} \quad i=d \end{cases} \\
\qquad\qquad x_{ij} \in \{0,1\} \qquad , \qquad \forall\,(i,j)\in A
\end{array}\right.$$

where

$$\begin{array}{lll}
m_{ij}(t) & = \frac{P_{ij}(t)}{PR_j(t)} & , \quad \forall\,(i,j)\in A \\
d_{ij}^{avg} & = \frac{\sum_{k=1}^{|O_{ij}|} d_{ij}^{(k)}}{|\mathcal{O}_{ij}|} & , \quad \forall\,(i,j)\in A \\
n_{ij}(t) & = \frac{d_{ij}^{avg}}{P_{ij}(a_{ij})k} & , \quad a_{ij}\in\{0,\ldots,A_{max}\} \quad , \quad \forall\,(i,j)\in A
\end{array}$$

The key notation that has been used in this model is described in Tab. 3.

Symbol	Description
m_{ij}	Energy coefficient of link (i,j)
n_{ij}	Stability of link (i,j) coefficient
$d_{ij}^{(k)}$	Observation of link (i,j) at time k
T_{ij}	Time required to send a packet of information from node i to node j
P_{ij}	Power dissipated in transmission over link (i,j)
Q_{ij}	Power wasted in the reception
δ	A parameter from $]0,1[$
E_{res_i}	Residual energy in node i
PR_i	Propensity of node i
R_{ij}	Residual life time of link (i,j)
d_{ij}^{avg}	Average traveled distance between nodes i and j
$\mathcal{O}_{ij}$	Flow on link (i,j)

Table 3. Bi-criteria model description.

Authors have transformed this model to a monocriteria model by combining the two objectives into a weighted convex sum

$$\min \sum_{(i,j)\in A} \Big(p_1\,m_{ij}(t) + p_2\,n_{ij}(t)\Big)\,x_{ij}$$

Like any classical scalarisation method, the variation of the parameters p_1 and p_2 (subject to $p_1 + p_2 = 1$) provides few solutions where each one refers to a preference structure. Authors

used the greedy hop by hop approach with the aim to balance the opposite effects of an energy aware routing that tries to select longer routes and a routing algorithm that desires to find a more stable path through the selection of a shorter route. Simulations show the balanced weight p_1 and p_2 which can offer a better network behaviour and which allows to give the optimal trade-off between low energy consumption and high stability link level.
The main observation on this approach lies in few QoS criteria ignored in the model as well as QoS constraints that characterize the ad hoc network nature. All these parameters should be taken into account in order to guarantee best quality of service.

5.3 Multicriteria model based on four criteria

The model developed in (Amine et al., 2009) consists of a model with four criteria; namely, energy consumption, delay, bandwidth, and packets loss rate:

$$\begin{cases} \min \sum\limits_{(i,j)\in A} x_{ij}p_{ij} \\ \max\limits_{\mathcal{P}} \min\limits_{(i,j)\in A} x_{ij}B_{ij} \\ \min \sum\limits_{(i,j)\in A} x_{ij}D_{ij} \\ \min \tau_{\mathcal{P}} \\ \text{subject to} \\ \qquad p_{ij} = p_{ji} \quad , \quad \forall (i,j) \in A \\ \qquad B_P \geq B_{\min} \\ \qquad D_P \leq D_S \\ \qquad p_{ij} = t_{ij} + \varepsilon_{ij} \quad , \quad \forall (i,j) \in \mathcal{P} \\ \qquad D_{ij}B_{ij} \geq f_{ij} \quad , \quad \forall (i,j) \in \mathcal{P} \\ \qquad \tau_{\mathcal{P}} < \pi \\ \qquad x_{ij} \in \{0,1\} \quad , \quad \forall (i,j) \in A \end{cases}$$

The key notation that has been used in this model is described in Tab. 4 below.

Symbol	Description
$\mathcal{P}$	Routing path
x_{ij}	Decision variable
p_{ij}	Energy affected to the link (i,j)
B_{ij}	Bandwidth available on link (i,j)
D_{ij}	End to end delay on link (i,j)
D_S	A maximal value that D_{ij} must not exceed
τ_P	Packets loss rate
π	A maximal value that $\tau_{\mathcal{P}}$ must not exceed
t_{ij}	Energy required to reach a node j from a node i
ε_{ij}	Insurance factor depending of link (i,j)
f_{ij}	Flow on link (i,j)

Table 4. Four-criteria model description.

This model takes into account a number of criteria that are not treated before. However, the consideration of the bandwidth and the delay in a contradiction hypothesis in not generally correct. Indeed, in a perfect network, the delay and the bandwidth are related by the formula

$$D_{ij}B_{ij} = f_{ij} \tag{5}$$

where D_{ij} and B_{ij} are, respectively, the delay and the bandwidth available on a link (i,j), and f_{ij} is the flow to be transmitted over (i,j). Thus, this model may be compatible only with networks with more disturbance according to extra delay affected, eventually, by space conditions and bit errors occurring in nodes (Amine et al., 2009; Korhonen & Wang, 2005). Simulations show that this approach, until it returns numerous paths according to different QoS preferences, gives effectively more choice to the user in the routing process.

6. Conclusion

The quality of service presents a rich field for study in ad hoc networking. It concerns the behaviour of the network, and is dealt with from different points of view. In the current studies, QoS models, QoS resource reservation, QoS routing, and medium access control protocol are stated specifically. From a mathematical point of view, QoS presents an economic problem which consists to optimize an objective cost under several constraints imposed by the network nature and proprieties. In this connection, several models are developed in the literature for or oriented to quality of service.

Mathematical models of quality of service give virtual maquettes representing some aspects aiming at improving the quality of service with mathematical techniques. Mathematical models of quality of service in ad hoc networking involve two different formalities: monocriteria formulations that focus on one aspect of QoS (energy, bandwidth, delay, etc), and multicriteria formulations that take into account a set of no less than two contradictory aspects of QoS and give more realistic representation.

Monocriteria formulations were largely addressed in the literature and gave efficient approach for many concerns of quality of service in ad hoc networking. Furthermore, several results have been already integrated in many standard protocols. In the same respect, various challenging issues were raised which provide currently a large scope for investigation.

Multicriteria formulations present a realistic feature to study the quality of service in ad hoc networking since the QoS criteria are affected by each other and cannot be optimized separately. These formulations are usually dealt with by adopting few techniques in order to transform the formulation into a monocriteria one for which classical methods and algorithms can be used. Solving each transformed formulation gives an efficient solution according to a preference structure. As against, few studies aim at finding the Pareto Front of the original multicriteria formulation in order to give all preference structures which are related to all users' satisfaction. This point of view gives more opportunities in decision making. For this, more attention should be given to the diversity of choice in multicriteria approaches in order to satisfy all quality of service levels related to all classes of users. A crucial feature to fulfill this intention is firstly to develop rigorous mathematical formula characterizing the relationship between parameters in ad hoc networking field as delay, bandwidth, packet loss rate, and so on. Secondly, to improve the conception and the modelling process in the sense that the diversity of choice aims at proposing a totalitarian approach which guarantees the quality of service in an extended meaningful way.

7. References

Ahn, G.-S., Campbell, A. T., Veres, A. & Sun, L.-H. (2002). SWAN : Service differentiation in stateless wireless ad hoc networks, *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'02)*, Vol. 2, New York, USA, pp. 457–466.

Amine, Kh. (2008). *Approche multicritère pour la qualité de service dans les réseaux ad hoc*, Master thesis, Faculty of Sciences, University of Moulay Ismaïl, Meknès, Morocco.

Amine, Kh., El Yassini, Kh. & El Ouadghiri, D. (2009). Multicriteria formulation for the quality of service in ad hoc networks, *Proceedings of the IEEE International Conference on Multimedia Computing and Systems (ICMCS'09)*, Ouarzazate, Morocco, pp. 395–399.

Brahma, M. (2006). *Étude de la QoS dans les réseaux ad hoc: Intégration du concept de l'ingénierie du trafic*, PhD thesis, University of Haute Alsace, Mulhouse, France.

Ehrgott, M. (2005). *Multicriteria optimization*, 2^{nd} edn, Springer Berlin Heidelberg, Germany.

Guerriero, F., De Rango, F., Marano, S. & Bruno, E. (2009). A biobjective optimization model for routing in mobile ad hoc networks, *Applied Mathematical Modelling* 33(3): 1493–1512.

Guo, S. & Yang, O. W. W. (2004). QoS-aware minimum energy multicast tree construction in wireless ad hoc networks, *Ad Hoc Networks* 2(3): 217–229.

Guo, S. & Yang, O. W. W. (2007). Energy-aware multicasting in wireless ad hoc networks: A survey and discussion, *Computer Communications* 30(9): 2129–2148.

Hashemi, S. M., Rezapour, M. & Moradi, A. (2007). Two new algorithms for the Min-Power Broadcast problem in static ad hoc networks, *Applied Mathematics and Computation* 190(2): 1657–1668.

Korhonen, J. & Wang, Y. (2005). Effect of packet size on loss rate and delay in wireless links, *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'05)*, Vol. 3, New Orleans, Louisiana, USA, pp. 1608–1613.

Lee, S.-B., Ahn, G.-S., Zhang, X. & Campbell, A. T. (2000). INSIGNIA: An IP-based quality of service framework for mobile ad hoc networks, *Journal of Parallel and Distributed Computing* 60(4): 374–406.

Leggieri, V., Nobili, P. & Triki, Ch. (2008). Minimum power multicasting problem in wireless networks, *Mathematical Methods of Operations Research* 68(2): 295–311.

Li, D., Liu, Q., Hu, X. & Jia, X. (2007). Energy efficient multicast routing in ad hoc wireless networks, *Computer Communications* 30(18): 3746–3756.

Mirhakkak, M., Schult, N. & Thomson, D. (2000). Dynamic quality of service for mobile ad hoc networks, *Proceedings of the 1st ACM International Symposium on Mobile Ad hoc Networking & Computing (MobiHoc 2000)*, Boston, Massachusetts, USA, pp. 137–138.

Montemanni, R. & Gambardella, L. M. (2005). Exact algorithms for the minimum power symmetric connectivity problem in wireless networks, *Computers & Operations Research* 32(11): 2891–2904.

Montemanni, R., Leggieri, V. & Triki, Ch. (2008). Mixed integer formulations for the probabilistic minimum energy broadcast problem in wireless networks, *European Journal of Operational Research* 190(2): 578–585.

Sarkar, S. K., Basavaraju, T. G. & Puttamadappa, C. (2007). *Ad hoc mobile wireless networks: principles, protocols, and applications*, Auerbach Publications, Taylor & Francis, New York.

Sinha, P. (2005). QoS issues in ad-hoc networks, *in* P. Mohapatra & S. V. Krishnamurthy (eds), *Ad hoc networks: Technologies and Protocols*, Springer, pp. 229–247.

Skriver, A. J. V. & Andersen, K. A. (2000). A label correcting approach for solving bicriterion shortest-path problems, *Computers & Operations Research* 27(6): 507–524.

Xiao, H., Seah, K. G., Lo, A. & Chua, K. C. (2000). A flexible quality of service model for mobile ad-hoc networks, *Proceedings of the 51st IEEE Vehicular Technology Conference (VTC 2000-Spring)*, Vol. 1, Tokyo, Japan, pp. 445–449.

Yuan, D., Bauer, J. & Haugland, D. (2008). Minimum-energy broadcast and multicast in wireless networks: An integer programming approach and improved heuristic algorithms, *Ad Hoc Networks* 6(5): 696–717.

6

Towards Reliable Mobile Ad Hoc Networks

Ricardo Lent and Javier Barria
Imperial College London
United Kingdom

1. Introduction

It is expected that future networks will interconnect an even larger number of devices then today, ranging from servers to micro-devices embedded in objects. These devices will provide useful services thanks to the possibility of a networked operation, for example, localization services to support a variety of situation-aware applications. A very significant number of these devices will be carried by users-on-the-move. While a wireless infrastructure could serve to provide a near permanent network access to these devices, structural network dynamics and service demand patterns could impact the main features of the solutions relying on this network.

Mobile ad hoc networks (MANETs) may serve to bridge these devices to the network in situations where a connection to a wireless infrastructure may not be feasible or desirable because of coverage limitations, network failures, congestion, policies, or cost. MANETs can be quickly created for a wide variety of applications and whenever needed to operate on virtually any environment. A main feature of a MANET is its self-organizing ability over a network that is assumed by temporarily linking each mobile with other nodes within wireless coverage. In this situation, nodes can serve as routers, at least temporarily, to forward packets for other nodes. One of the main technical drawbacks of these type of networks is that the network tends to change quite often. Nodes may arrive at or depart from the network without notice and direct node-to-node communication may or may not be possible at any given time due to node mobility and changes on the surrounding environment. These characteristics determine a highly dynamic network that makes difficult a reliable forwarding of packets on multi-hop routes over long periods of time. Communications tend to be very unreliable and inefficient, because a route break not only disrupts immediately a communication, but also can introduce additional overhead into the network because of the potential need for retransmissions and re-routing operations.

In this chapter, we discuss a feasible approach to obtain improved routing reliability on a MANET. The approach consists in identifying and using links with the most availability to setup and maintain routes. Link availability is related to the residual lifetime of links, which can be calculated from various sources, including signal power, packet transit times, or nodes' location and moving trends. We focus mainly on the latter possibility as localization services are becoming widespread for mobile devices and the trend is expected to continue in the future. In particular, we explore the case where the localization services are provided by a sensor network purposely deployed to track mobile nodes. Most of the ideas discussed in this chapter are widely applicable to the other cases as well. A simulation-based evaluation under realistic assumptions suggests that the proposed routing approach can significantly improve MANET routing reliability, in particular, for highly dynamic networks.

2. Related work

MANETs are subject of intensive research and many works have been devoted to research their properties and operation (1). Some of the principal works that have explicitly addressed MANET reliability, or are in close relation to this discussion, are mentioned below. The list is not intended to be exhaustive, but representative of the related work previously done.

A possibility that have been explored by various authors is on the selection of the longest-lived links to create stable paths. These works are based on the observation that most randomly moving nodes are likely to drift apart from one another over time (2), so that their main assumption is that a link between two nodes that had survived for a significant long time would be unlikely to change any time soon and so, the link could be classified as stable. In fact, even in static networks wireless links may fail (3; 4; 5). In the *Associativity Based Routing* (ABR) (6), a link lifetime is measured by counting the number of beacons received from neighboring nodes and the links associated with the highest beacon counts are preferred. In the *Signal Stability Adaptive Routing* (SSA) (7), routes are created by giving preference to the selection of strong connected nodes. Nodes are classified as strongly or weakly connected on the basis of their signal strength as measured from beacons, which are exchanged periodically between neighboring nodes. McDonald and Zanti (8) investigated a clustering approach for MANETs and the probability of two nodes remaining within a distance threshold of one another over time.

Another possibility for stable routing is to select links based on estimations of the future network state, as done by Su et al. (10; 11) and a previous work (9). In the *Route Lifetime Assessment Based Routing* (RABR), the average change in received signal strength is calculated and used to predict the time when a link would fail (12). A similar approach is used to define link affinity and path stability metrics from the received signal strength (13). A statistical approach was proposed by Gerharz et al. (14; 15) based on observations of link durations for various mobility models. On the other hand, the network availability as a whole can be improved by avoiding routing traffic through nodes with a low remaining energy (16; 17).

On the definition of adequate metrics for describing a path reliability or link availability, a probabilistic measure was introduced (18) to help in the selection of stable paths. A prediction-based link availability calculation was also proposed and used to develop a metric for path selection in terms of path reliability (19). An approach to evaluate the signal strength variations between neighbors has also been proposed (20).

Most works estimate link lifetimes based on the signal strength of beacon packets or by using the nodes location acquired with a GPS receiver. The beaconing scheme relies on knowledge of a radio propagation model to associate a signal loss to a travelled distance (31). The free space propagation model is commonly used (21) and beacons are transmitted with the highest power level (22). However, the fluctuation in signal strength of the transmitter as perceived by the receiver may not depend only on distance in practice (23). Hence, the distance estimation between transmitter and receiver based solely on signal strength may not be accurate (13). On the other hand, the GPS scheme could produce better distance estimations between nodes (24). However, there are some drawbacks in using GPS receivers. The use of GPS receivers implies an extra power consumption to the nodes and an extra implementation cost, and in some cases the reception of GPS signals may not be possible, for example in some indoor locations or under adverse weather. It is interesting to mention that node localization can also be used to route packets to a given geographic area (25; 26; 27).

An alternative to the use of GPS receivers is to use a sensor network (28; 29; 30; 32) to track and localize mobiles. A practical example is the Cricket Indoor Location System (33) which can provide fine-grained localization information including coordinates and orientation. An optimal sensor network structure would have the minimum number of sensors activated at a particular time (coverage problem) to transmit a minimum amount of acquired data (information accuracy problem). The concept of coverage is environment dependent and is subject to a wide range of interpretation, but in the more general case can be considered as the measure of QoS of a sensor network. A definition of the coverage problem from several points of view including deterministic, statistical, worst, and best case is presented in (34). In (35) the tolerance of a sensor network against both random failure and battery exhaustion from the viewpoint of stochastic node placement is evaluated. In (36) a strategy is presented that maximizes the coverage of the most vulnerable regions under surveillance as well as maintaining an average coverage. In (37) the aim is to optimize the number of sensors and determine their placement to cope with constraints of imprecise detection and terrain properties (i.e., number of sensor vs. miss probability). In (38) the miss probability that quantifies the likelihood that an active sensor fails to detect the mobile target using a low beam sensing radius is studied. Self-organized sensor networks have also been proposed in (39). And in (40) a self-organizing technique for enhancing the coverage of wireless micro-sensor networks after an initial random placement of sensors is proposed.

The other important aspect that could affect sensor networks usefulness to MANET routing relates to the level of information accuracy and error incurred by the sensors' measurements. In this respect, density and structural characteristics, and data acquisition and fusion strategies of the sensor network are relevant. An overview of an information-driven approach to sensor collaboration in ad hoc sensor networks is presented in (41). The idea is that the network should be determined by dynamically optimizing data utility for given communication and computation costs. In (42) the tradeoff relating to sensor accuracy and energy consumption of a grid infrastructure is studied. In (43) the problem of optimal sensor selection and fusion is solved with a Bayesian framework under the tradeoffs of low-power consumption and collaborative information processing, while in (44) energy-quality tradeoffs for target tracking in wireless sensor networks is studied.

3. Model

We assume a mobile ad hoc network (MANET) where wireless nodes communicate using a common broadcast channel by omni-directional antennas. A MANET can be represented by an undirected graph $G = (V, E)$, where V expresses the set of vertices (nodes) and $E \subseteq V \times V$ the set of edges (wireless links).

In this model, we assume that nodes have the same transmission range and so, the graph is undirected: $(u, v) \in E \Rightarrow (v, u) \in E$, i.e., nodes are *neighbors* and it is possible a communication in either way although not at the same time. In more detail, the radio signal encoding a packet sent from a node u with a power level P_t may be received and decoded (with a certain probability) by another node v as long as the reception power $P_r > P_s$, i.e., it is above the *receiver sensitivity* P_s. All nodes $v \in (V - u)$ for which this condition is true are neighbors of u. The set of neighbors of u is denoted by N_u.

The value of P_s is determined by the characteristics of the radio receiver and the communication bit rate. On the other hand, P_r depends on P_t and the path loss, which in turn depends on the distance between nodes and the surrounding environment.

The location vector of node i is denoted by $L_i = (x_i, y_i, z_i)$. In a MANET, the location of each node is not constant. Please note that we omit time indices to improve notation clarity. We assume initially that nodes can learn their location vector precisely. However, the routing algorithm discussed later tolerates localization error.

Each node is identified with a unique number and all its packet transmissions bear this number. Likewise, packet transmissions carry either the identifier of the destination node or broadcast. Note that although packets can carry the destination's identifier, transmissions are done as a physical broadcast, so all neighbors will in fact receive the packet even so it could be intended for a particular node. When a packet is received by a node although not intended to it, the packet is normally discarded at its link or routing layer. Also, the link layer (a MAC protocol) is assumed to resolve media contention by temporal channel reservation, which can be done for any non-broadcast addresses.

G is a function of time. V may change over time because node departures or arrivals to the network, which may occur at any time without notice. E may change as a consequence of node mobility, variations in the surrounding environment, and changes in V.

While it is possible to communicate neighbors with a single transmission hop, a packet transmission between non-neighbors must be relayed between neighbors for a number of steps. Any path is a sequence of vertices that the packets of a particular flow must visit to be delivered: $p = v_1 \rightarrow v_2 \rightarrow . . . \rightarrow v_k$. The condition for the path to be feasible is that $(v_i, v_{i+1}) \in E$, $1 \leq i < k$, that is, each hop must be between neighbors. We denote the set of all possible paths between two nodes s to d by Π_{sd}.

A flow f represents a data communication and is expressed by the tuple $f = (s, d, b)$, where $s, d \in V$ are the source and sink of the flow and b the sending rate. We assume routing on demand, so for the purposes of this discussion, routing is the process of associating a flow to a path, i.e., $R : (f, G) \rightarrow p, p \in \Pi_{s,d}$.

3.1 Link lifetime

Link lifetime (also known as link duration) has been suggested previously as a metric to determine stable (long living) paths in ad hoc networks to replace the hop count metric commonly used and which is implicitly approximated by the use of standard flooding. A link's lifetime can be estimated from the receiving signal strength, packet transmission times, or from the distance and mobility trends of nodes, as either a probabilistic or deterministic value.

The lifetime of a path is concave and limited to the lifetime of the weakest link among the ones composing the path. For path $p = v_1 \rightarrow v_2 \rightarrow \cdot \cdot \cdot \rightarrow v_k$ and link lifetime function $\Phi : (v_i, v_{i+1}) \rightarrow \Re$, the lifetime Φ_p of path p is:

$$\Phi_p = \min_{(u,v) \in E_p} L(u,v)$$

In this work, we are interested in expressing link lifetimes in deterministic terms calculating them from the nodes' location information. Localization services for mobiles are becoming widespread, so it makes sense to explore their use to improve MANET routing for future networks. In particular, we look at the use of the *link residual lifetime*, i.e., the remaining time for a link before it is expected to fail rather than the link age (as done in ABR, SSA, etc.), which do not necessarily perform well in all cases.

The residual link lifetime between two neighbors i and j of interest can be calculated from their current separating distance $|D_{ij}|=|L_i - L_j|$, $i \in N_j, j \in N_i$ and their relative speed D'_{ij} :

$$\Phi = \frac{D_m - \alpha |D_{ij}|}{|D_{ij'}|} \quad ;|D_{ij'}| > 0$$

where D_m is the maximum wireless coverage (can be estimated apriori from the properties of both radio transceivers and the surrounding environment).

Φ provides an estimate of the link's time to break when nodes move and diverge. If nodes tend to converge, $\alpha = -1$ allows to add the convergence time to the expected divergence time. Otherwise, $\alpha = 1$. Note that the function is undefined at $|D'| = 0$ (when nodes move in parallel). This situation can be handled as a special case (with a low value) when defining a cost function for routing purposes.

3.2 Localization

The basic assumption is that nodes can learn their own location in a three-dimensional space precisely through an external mechanism. Various alternatives exist to let mobile nodes acquire their location. The Global Positioning System (GPS) is a navigation satellite system that provides physical location information free-of-charge to any GPS receiver, but requires line of sight to at least four of the 24–32 GPS satellites. The information accuracy depends on various factors and could range from 5 m to 100 m in civilian receivers. Another possibility is through trilateration, which allows a node to determine its location from measurements of the transmission time from at least three known references. In contrast, an external system could be deployed to implement hyperbolic positioning (multilateration), which can determine the location of a node by using three or more receivers and computing the time difference of arrival of signals emitted by the node of interest. Multilateration is used by GSM systems and so it is of particular interest for implementing ad hoc network of smart mobile phones. Another alternative is to use a system with antenna diversity, where nodes' location can be determined by triangulation. These localization techniques could be implemented by the mobiles themselves of by an external system, such as the sensor network that we consider in this study.

Regardless of the method used, the localization system that is used would provide periodic updates to each mobile informing them of their relative or absolute coordinates. The mobiles will then estimate their location whenever needed from the data available, for example, from their calculated velocity vector and the previous location update. Note that the length of time between location updates can determine the accuracy of the location estimations. The impact of using imprecise information to MANET routing will be addressed in the simulation study discussed in a later section.

4. Problem formulation

The problem is to find the most durable path p^* from s to d for each flow f:

$$p^* = \underset{p \in \Pi_{sd}}{\operatorname{argmax}} \, \Phi_p$$

By selecting the most durable paths for the flows, less path repairs would be needed, which implies less protocol overhead and a better use of the nodes' energy.

4.1 Evaluation under ideal conditions

To support the idea, we conducted a simulation study to find out the average route lifetime on a mobile ad hoc network to determine whether there would be any reliability improvement over flooding (calculated as the shortest path) with the use of either the oldest links or the links with the longest residual lifetime metrics. The simulation was done at the topology level and assuming ideal conditions, which imply that links are determined solely based on the distance between nodes and that route calculation can be done with full knowledge of the location of nodes and their mobility patterns. Although these assumptions do not hold in practice, the results would suggest the best metric from a route reliability standpoint. We defer to a later section the use of more realistic assumptions.

Nodes move according to the random waypoint point (RWP) model without pause times and at a given speed that is randomly selected in the range [1, *S*]. Nodes move on a rectangular field of 400 × 100 units, all with equal wireless coverage of 50 units. For each simulation instance and after a suitable time (2.5 simulated hours) to let the statistical properties of the RWP model emerge, a route is established between two randomly selected nodes. Routes are established with either a hop count, link age, or link residual lifetime criterium. The results are depicted in Figure 1 for all three cases as a function of the maximum moving speed of the nodes. The plots also indicate the 95% confidence interval resulting from the Monte Carlo simulations.

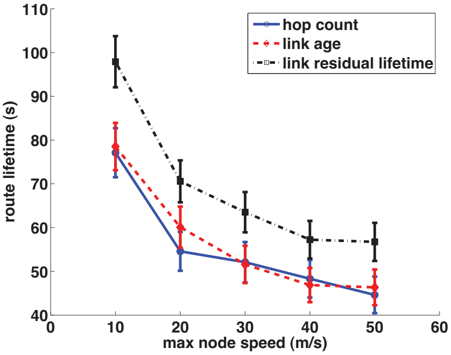


Fig. 1. Average route lifetime with ideal wireless transmissions: radio coverage only limited by the distance between nodes.

From the results, the use of link residual lifetimes clearly produced the most reliable routes. However, these routes tend to be longer than the shortest path (Figure 2).

5. A distributed solution: LDR

In a MANET, it is normally impractical to acquire global information about the network state (such as the nodes' location) to drive routing decisions. We discuss a distributed algorithm to allow each source independently find durable paths on demand with only local information. We call the algorithm *Link Durability Routing* (LDR) and has the following properties:

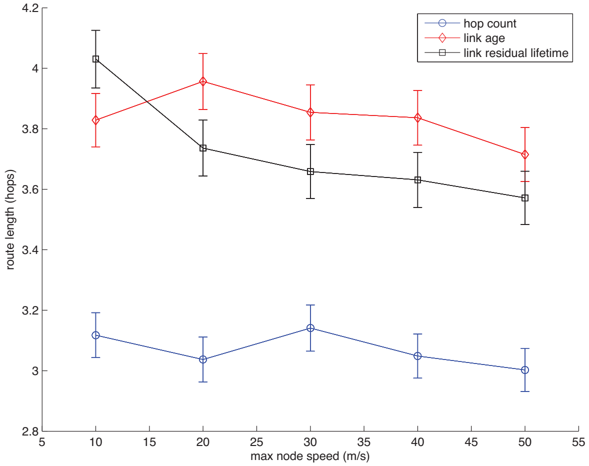


Fig. 2. Average route length (hops) with ideal wireless transmissions.

- LDR uses a modified flooding algorithm, but introduces decisions and actions at each iteration based on the residual lifetime of each link as calculated from local information. Route selection is distributed and not bounded to the source or destination nodes.
- In case localization becomes available partially to some nodes or not available at all in the network, LDR can continue to work, but it would produce less optimal routes.
- Clock synchronization among nodes is not needed.
- Active paths are periodically monitored by piggybacking information into selected data packets, so that preventive re-routing can occur in addition to reactive re-routing in the case of a route failure.
- The algorithm can be incorporated into existing flooding-based MANET protocols (e.g., AODV (45), DSR (46)). However, we will discuss the algorithm in the context of an independent protocol (LDRP), given that some particular operations available in other protocols are not needed, for example, HELLO beacons for neighbor discovery.
- LDR could also enhance the performance of other location-based protocols. For example, it could be used to extend LAR (47) or GPSR (48) to achieve improved route durability in addition to reduce the search area for routes.

5.1 Route setup

As mentioned above, LDR relies on a modified flooding algorithm to discover routes. The standard flooding algorithm is of common use by many on demand ad hoc routing protocols and works as follows. Whenever a new route to a destination is needed, the source broadcasts a route request message. The message indicates the desired destination and a message identifier, in addition to other pieces of information that could be relevant to each particular algorithm. The identifier and origin addresses of the message allows intermediate nodes to discern new from replicated requests, so that they can select to process only the first arrival of each request. If the node receiving the request if not the destination, the node will append its own address to the packet (and possibly other pieces of information depending on the actual protocol being used) and broadcast again the message without delay. On the other hand, if the receiving node's address matches the destination of the route request, the node will respond to the source with a reply message that will list the path used by the route request to reach the destination. The message is forwarded along the reversed path.

If the destination is reachable, there is a high probability that one of the copies generated by the process will eventually reach the destination. The path produced by the process will tend to the shortest path in number of hops, although, network congestion may induce longer routes.

5.2 Link selection

To implement a selection mechanism that will discern links based on their residual lifetime to allow setting up durable routes, we introduce a decision mechanism that is executed at each node participating in a route discovery process.

With standard flooding, each node receiving for the first time a route request message broadcast immediately the message to its neighbors. In LDR, a route request in retained at each node for a certain time before doing a new broadcast. By making the retaining time inversely proportional to the durability of the preceding link, LDR can delay the messages traveling on the less desirable routes and favor the best route request replicas (so, those traveling on the most durable links) to reach first the destination. Before the node re-broadcast the request, it will continue processing other route request arrivals sharing the same request identifier. However, each node will at most broadcast one request per route request ID as in the standard flooding algorithm. Since the destination replies only to the first arriving route request, a robust path will be selected for the flow. We define the route request defer time (δ_{uv}) for link (u, v) as follows:

$$\delta_{uv} = K_0 e^{-K_1 \Phi_{uv}} \tag{1}$$

where K_0 and K_1 are positive scaling constants that are experimentally chosen and Φ_{uv} represents the residual lifetime of link (u, v). K_0 is the maximum defer time that can be introduced to a route request.

The general idea is similar to the one developed by Cheng and Heinzelman (2), but with a different defer function. Also, our approach introduces the monitoring of active routes so that preventive re-routing can occur before a path breaks. If no localization is available, LDRP assumes $\delta = 0$, so that request will be broadcast without delay. LDR on a network without localization would produce results identical to standard flooding.

5.3 LDR protocol

Route selection with LDR is distributed by definition and link selection is implicit by introducing a temporal behavior to the way route requests are handled by nodes rather than by defining a spatial selection of the next hop or by explicitly selecting a route among the choices available at the source or destination nodes.

To calculate the durability of a link, a node requires its current location and velocity vector as well as the vectors from the predecessor node. This information can be easily obtained by augmenting route requests with two fields: `location` and `velocity`. Therefore, in addition to appending its network address, an intermediate node updates these two fields with its own data. Note that these two fields are fixed given that only information from the predecessor is needed and not from the rest of nodes in the path. This procedure allows each intermediate node to obtain fresh information to compute the residual lifetime of links.

After the defer time for a new route request is determined, the message is scheduled at a target time for broadcast or to be delivered to upper layers. The target time is the current time plus the calculated defer time. Note that the defer time is a minimum time that the message is forced to wait in a node. The actual residence time in the node could be longer due to other factors that may occur after the target time, such as queue waiting before transmission. A new route request arrival may replace an existing scheduled message transmission whenever the new target time is less than the existing target time. If the target time for a message is reached, the request is considered processed so any further arrival with the same request id will be dropped.

5.4 Route maintenance

The route setup phase allows to setup durable paths. However, the residual lifetime of each link on a path is likely to change over time as a consequence of node mobility and changes in the operating environment. To reduce the risk of a route break, active routes are periodically monitored by LDRP. For this purpose, selected data packets are augmented to carry the position and velocity vectors of the predecessor node. On arrival of an augmented data path, the node calculates the new residual lifetime of the preceding link. The link would be assumed to be at risk of failure if its residual lifetime is less than parameter `ttb_thr`. If so, a *route information* message will be sent to the source to initiate a preventive re-routing action.

LDRP limits the creation rate (per flow) of control messages, which include augmented data packets, route requests, and route error messages, to place a cap to the monitoring overhead that could be generated. The inverses of the maximum sending rate limits are defined by `rdata_limit`, `rreq_limit`, and `route_error_limit`.

6. Test case: LDR and sensor network for mobile localization

The discussion so far has considered that nodes were able to determine their location accurately. In this section, we evaluate LDR under less favorable assumptions. Furthermore, to enrich the test case we consider that MANET nodes lack a GPS receiver, but that can be localized with the help of a sensor network.

Both the MANET (using LDR) and the sensor network share the same working area but operate nearly independently of each other: MANET nodes route their traffic independently of the sensor's activities and sensor nodes track the location of MANET nodes and pass them the information but unaware of any other MANET activity. MANET nodes use the localization updates sent by the sensor nodes to determine their velocity, so that their location can be calculated when needed. If no updates from the sensors are received within a predefined time, it would be assumed that no localization is available and any new route request arrival will be broadcast without a defer time. Each type of network operates on its own radio channel. However, it is assumed that MANET nodes are able to receive packets from the sensor nodes, so that they have access in fact to both radio channels.

We are interested in observing MANET route reliability, which will be measured in terms of the packet delivery ratio for a test flow. The effect of most factors on packet routing (link failures, congestion, channel contention) are summarized in the packet delivery ratio (and its complement, the packet error ratio). To measure this metric, we consider the packet exchange between two stationary nodes that must relay on mobile nodes to communicate.

The two nodes are located far apart on the test field and are kept stationary to prevent any direct (single hop) communication with their simulated radios. All aspects of LDR have been integrated into a packet-level simulator (49) for this study including support for the concurrent simulation of the two independent wireless networks. Each wireless network was IEEE 801.11 DCF-based at the MAC layer with RTS/CTS enabled only for MANET unicasts. These assumptions produce various sources of localization error for MANETs. Other than the inherent localization error introduced during the sensing phase, MANET nodes can only receive their location estimates at irregular times. To receive a location estimate, a MANET node must be in the vicinity of sensors and their transmissions must be successful (e.g., must not collide). All route reliability measurements will be taken under these less than ideal conditions.

In addition to account for packet transmissions, the simulator also keeps track of the power consumption for communication related tasks (of both networks). The radio transceiver state (transmitting, receiving, sleeping, or idle) is associated with a power consumption as in Table 1.

Transceiver state	Power consumption (MANET)	Power consumption (sensor)
idle	0.035 W	0.0001 W
transmit	(0.532 +Tpwr) W	(0.03 +Tpwr) W
receive	0.395 W	0.0354 W
sleep	0.001 W	3e-06 W

Table 1. Power consumption parameters for mobile nodes and sensor nodes. *Tpwr* represents the transmission power.

The evaluation was done under two cases: obstacle-free and a more realistic obstructed scenario. In all cases, AODV (45) was used as a reference protocol for performance comparison purposes.

6.1 Scenario 1: obstacle-free case

The first scenario consists of 30 MANET nodes (28 mobiles, 2 stationary) that reside on a 300m x 200m field. Mobility is modeled by the random way point model with pause times selected in the range 0 to 10 seconds. The data traffic corresponds to a video stream and is modeled as a single 80 Kbps constant-bit-rate flow of packets. Source and sinks are centered on the field but separated by 200m.

Other simulation parameters were defined as follows. The ideal wireless range is provided as a reference in the table. It is not used in the simulations. Instead, a packet reception is modeled realistically with a probability that depends on the received signal power.

A simulation run consisted in starting node mobility and the video stream, and in measuring the number of packets delivered at the destination plus other relevant observations for 10 simulated minutes. Sensor nodes are placed at fixed random locations to track mobiles within their sensing area. Each τ seconds plus a random jitter to reduce the collision probability, sensors emit their localization results (whenever available). Either 50 or 100 sensor nodes are deployed at the beginning of each simulation run with a τ (average time between sensor broadcasts) of either 10 or 20 s. MANET nodes have two simulated wireless interfaces (one connected to the MANET itself and another to the sensor network). Mobiles listen to the sensor channel and determine their location when needed from observations of their moving trend. The accuracy of the localization will be determined, among other factors, by the sensor density and the probability to remain within a sensor broadcast coverage when a transmission occurs. The expected time between 2 successful receptions of localization broadcasts is depicted in Figure 3 as a function of the sensor node density in the scenario. On the other hand, the location error (ε) is depicted in Figure 4 also as a function of the sensor density.

Parameter	Value
moving speed	from 1to Sm/s
pause times	[0,10] s
Power consumption	Physical layer model
Trans. power (mobile)	Fixed: 8mW
Trans. power (sensor)	Fixed: 2mW
Trans. rate (mobile)	1Mbps
Trans. rate (sensor)	100 Kbps
Ideal wireless range (mobile)	62.82 m
Ideal wireless range (sensor)	31.41 m
max_update_time	10 sec
rreq_limit	0.1
route_error_limit	0.1
K0, K1	0.005, 0.5
ttb_thr	1.0
max_rreq_interval	1.0
update_interval	1.0
τ	10 or 20 sec

Table 2. Simulation parameters used in the test scenario.

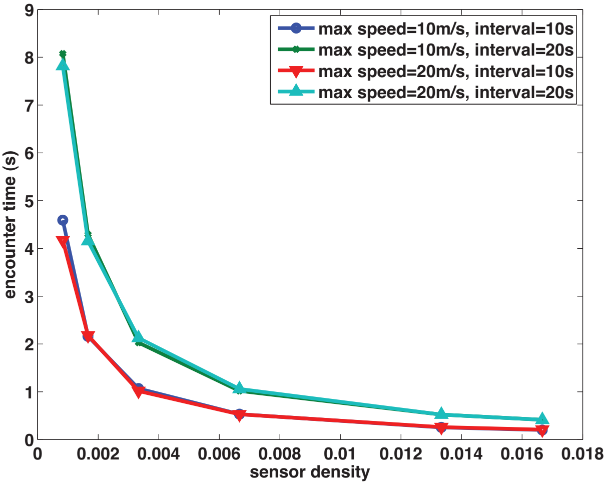


Fig. 3. Expected inter-arrival time between 2 consecutive location updates from the sensor network to any given MANET node.

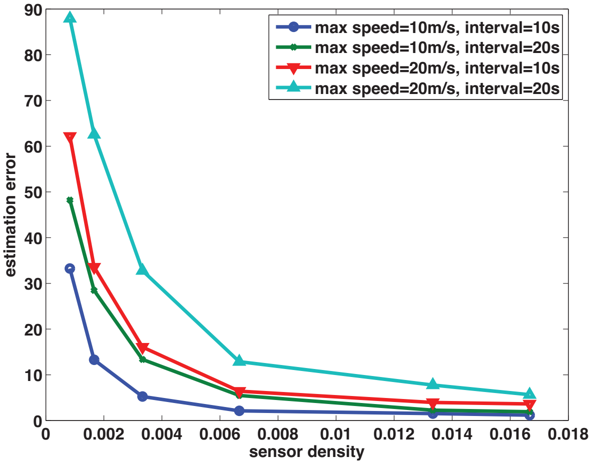


Fig. 4. Localization error of the estimates with respect to the real node location.

For comparison purposes, in the next set of results we include an evaluation of the system when using ideal GPS receivers, which unlike the sensor network case, can provide accurate localization at any time. A large number of runs were conducted to achieve high confidence (confidence intervals were very close to average values, so that there are not shown in the figures for visual clarity).

Figure 5 depicts the average packet loss ratio as a function of the maximum node speed. We naturally observe an increase in the packet error ratio as nodes move faster under all routing cases. LDRP results are labeled as follows: "sn: X I: τ", where "X" indicates the number of sensor nodes and τ the average time interval between sensor broadcasts.

The results suggest that route reliability can be improved with LDRP by exploiting the localization provided by sensors as compared to the standard flooding-based approach of AODV. The improvement is particularly significant when ideal GPS receivers are available to localize nodes. When using a sensor network to localize nodes, the less frequent localization updates produce localization errors than impact route reliability. Nevertheless, the difference between these two cases is small.

On the other hand, LDRP produced longer paths than AODV (Figure 6). The distributed link selection process used by LDRP implicitly takes into account link congestion in addition to link lifetime, given that route requests are transmitted via regular broadcasts. Indirectly, LDRP balances the selection of durable and less congested links. The downside of this approach is that longer paths would tend to increase average packet latency (depicted in Figure 7). However, the increase in individual end-to-end packet latency can be compensated by the reduced number of retransmissions needed to successfully transmit a certain amount of data. The power consumed for communication-related tasks, excluding the power for GPS readings or mobile tracking, is depicted in Figure 8 in terms of the ratio power to throughput. This ratio accounts for the energy required to successfully deliver a certain number of bytes to the destination. The results suggest that LDRP can be more efficient than the standard AODV, even when including the energy costs of the supporting sensor network.

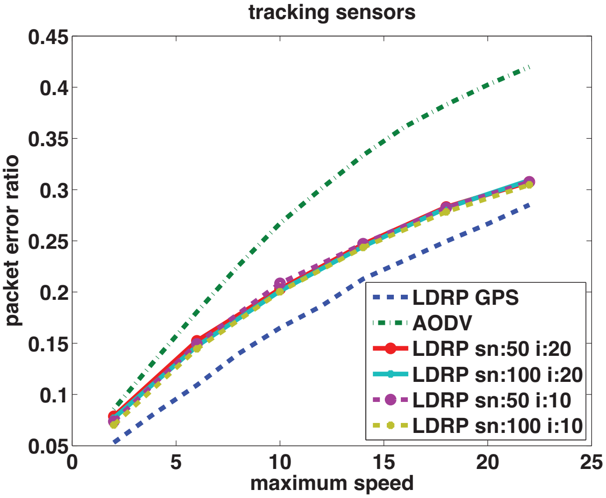


Fig. 5. The packet loss ratio naturally increases in proportion to the maximum speed of mobiles. LDR is tolerant to localization errors but nevertheless the deviations from the real values impact the reliability of MANET paths.

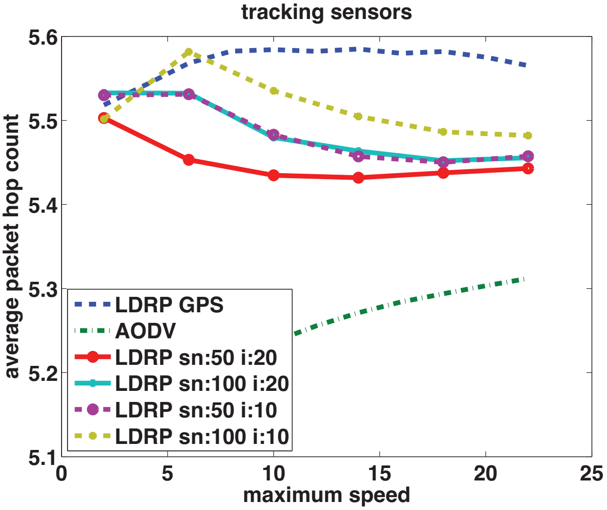


Fig. 6. Average path length in number of hops.

It is interesting to note that localization accuracy when using a sensor network to localize nodes can be improved either by increasing the broadcast rate or by deploying a large number of sensors on the environment. In both cases, there will be an increase in the number of localization update messages arriving at the mobiles, which would result in better location estimates. Figure 9 depicts this situation in terms of the packet delivery rate for the same scenario as a function of the sensor density. The downside of adding more sensors is that the overall energy consumption of the system will also increase as shown in Figure 10.

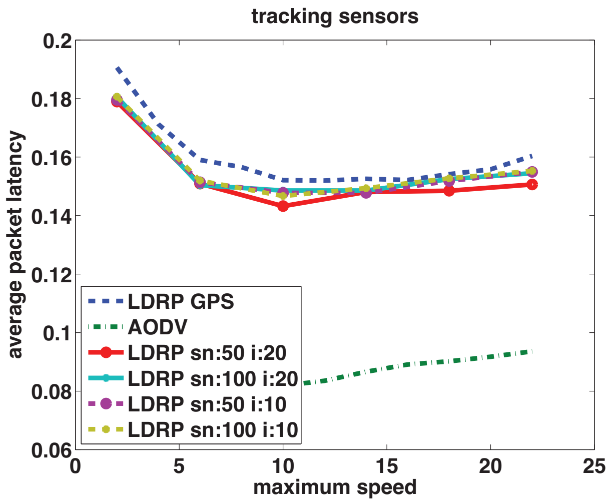


Fig. 7. The reduced need for retransmissions gained from more reliable routing would compensate the higher individual end-to-end latency of packets.

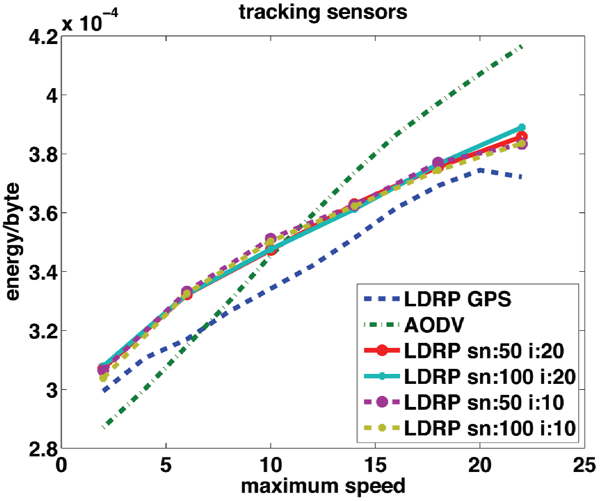


Fig. 8. The power consumption–throughput ratio (Joule/byte) gives an indication of the energetic cost of the network (lower is better).

6.2 Scenario 2: obstructed case

The environment where a MANET operates can affect packet reception leading to a worst routing performance than expected as predicted by the use of ideal unobstructed environments.

To evaluate LDR under more realistic assumptions, we consider the field with obstacles (e.g., buildings) represented in Figure 11. The scenario hosts a hypothetical rescue operation

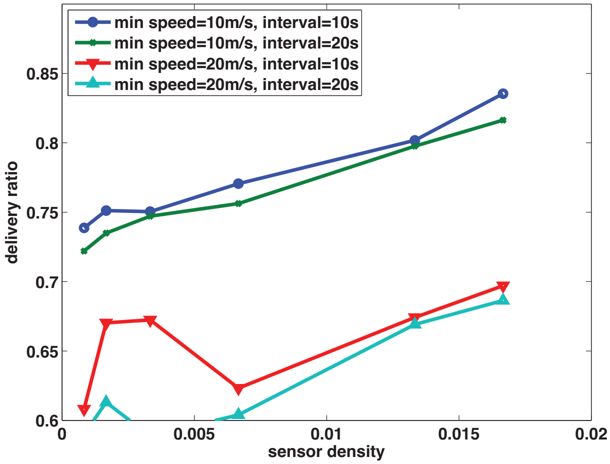


Fig. 9. Delivery ratio as a function of the sensor density (in sensors per square meter) . A larger number of sensors can produce more accurate localization for mobiles, which can directly benefit the reliability of MANET routes.

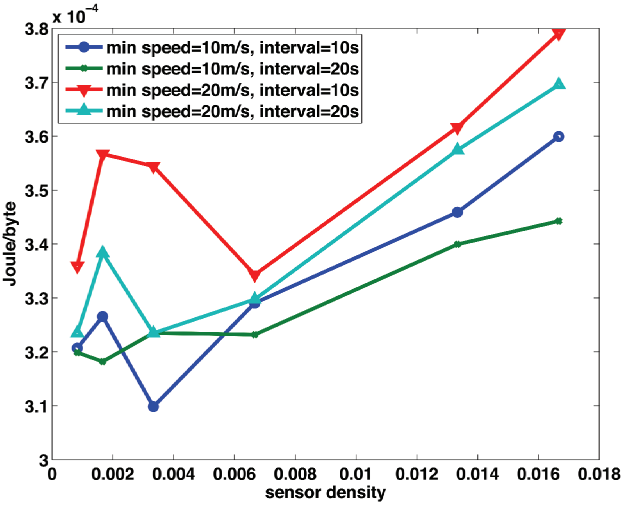


Fig. 10. Energy consumed per delivered byte as a function of the sensor density (in sensors per square meter) in the scenario.

where a number of sensors could have been deployed to gather information relevant for the rescue efforts and at the same time help to localize mobiles. The mobiles on the other hand are carried by the rescuers that need to work on the area.

As in the previous case, we are interested in observing the route reliability of a test traffic flow modeled by a constant bit rate transmission of 40 Kbps between two distant stationary nodes. For this second scenario, we consider 50 MANET nodes (48 mobile) on a 300x200m field. A set of 400 sensor nodes are as well randomly deployed.

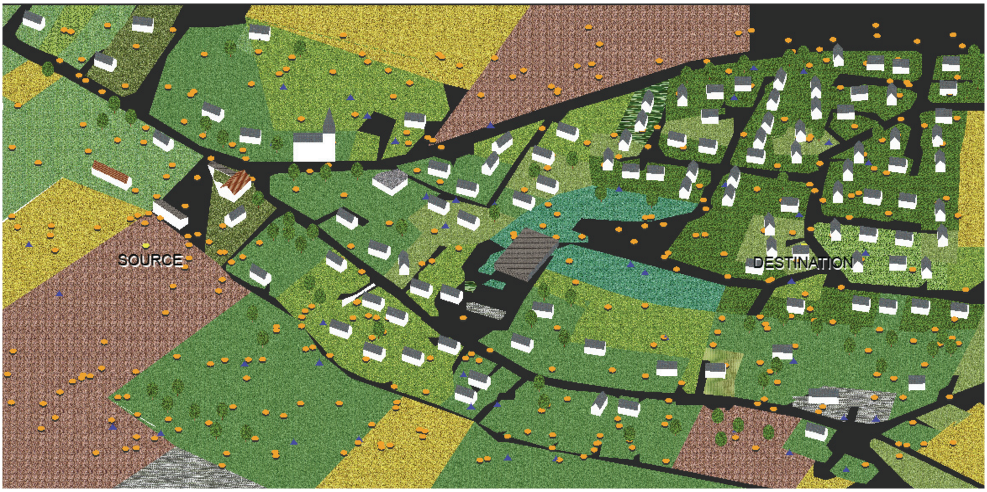


Fig. 11. Test case for LDR representing an obstructed simulated field. Sensors are represented by a circular shape and mobiles with a triangular shape.

The field contains a number of different obstacles that may affect both node mobility and packet reception. The field geometry is a (modified) user-contributed model available from Google 3D warehouse. For each packet transmission, the receiving power at each mobile is computed by the simulator. Obstacles that appear on the ray that connects the transmitter and receiver will reduce the receiving power by a pre-determined amount, depending on the predefined obstacle material (concrete walls, wood, etc.) The receiving power determines the probability of a successful packet reception.

On the other hand, node mobility is modeled with an extended random way-point (RWP) model that supports the inclusion of mobility attractors (RWPA). As with the RWP, the destination of each mobile is randomly selected on the field (but not inside an obstacle) and they move at a random speed towards the selected destination. Once they arrive at their destination, mobiles stay there for a random "pause" time before selecting a new random destination to repeat the process. In RWPA, nodes may select with probability p one of the attractors as destination instead of the random destination. If a node decides to move to an attractor, it will move to the point located $\gamma = C + q$ from the attractor (on the line connecting the current mobile location and the attractor location). C is a constant and q is an exponential random variable of parameter Q. γ therefore models how close the mobiles can get to the attractor. In the test case, the attractors represent areas of interest for the rescue operation.

Other simulation parameters are identical to the previous scenario.

Because of the high complexity of this second scenario, we restrict the evaluation scope to a single case of nodes moving with speeds in the range [1, 20] m/s The average packet delivery ratio is depicted in Figure 12.

As with the unobstructed case, path lengths and individual packet latency were higher with LDRP than with AODV (figures 13 and 14). About 5% longer paths and 30–40% higher delay. Finally, results for power consumption indicated similar figures when using AODV or LDRP for this scenario to deliver the same amount of data (Figure 15).

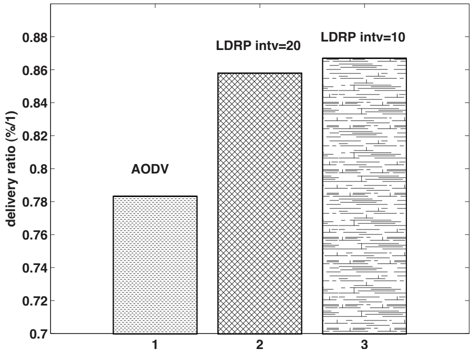


Fig. 12. Delivery ratio of the test flow on the obstructed scenario with nodes moving with speeds from 1 to 20 m/s.

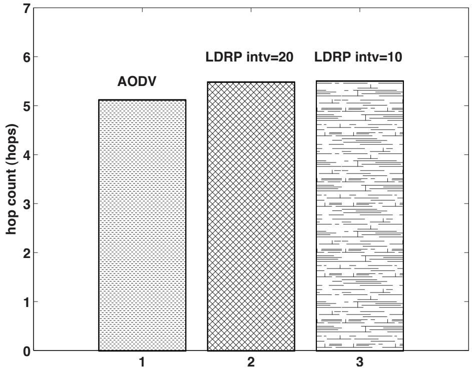


Fig. 13. Path length in number of hops for the test flow between two stationary nodes located at both ends of the test scenario.

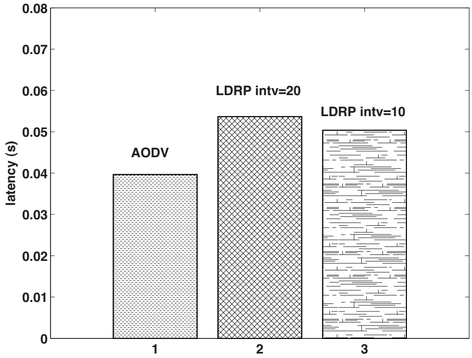


Fig. 14. The individual packet latency is also expected to be higher for LDR in the obstructed scenario.

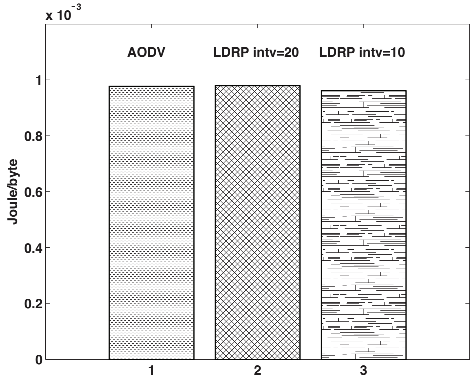


Fig. 15. Energy consumption per byte delivered (Joule/byte)

7. Final remarks

Mobile ad hoc networks can complement existing wireless infrastructure-based networks and bring a plethora of novel services to mobile users. While the lack of need for an existing infrastructure and centralized control, allows MANETs to be quickly created or destroyed as needed, their multihop nature makes them quite sensitive to changes in both the structure of the network and the surrounding environment.

We have discussed reliability issues in MANETs and elaborated on a low-overhead solution to improve the reliability of routes by introducing a mechanism that allows the identification and selection of links with the most availability as measured by their residual lifetime. We have also suggested a realization of the approach whereby the residual lifetime of links are calculated based on node location. We call the algorithm Link Durability Routing (LDR). In addition to a reliable path establishment, the algorithm takes advantage of existing packet flows to constantly monitor the expected availability of links. The algorithm relies solely on local information to operate and without needing a periodic local or global exchange of network information. By means of the continuous monitoring of active paths, LDR can detect paths at risk of become unavailable and enforce preventive or corrective re-routing.

Finally, we have evaluated LDR in the context of a realistic scenario where node localization is acquired from either a GPS receiver of from tracking sensors. The results suggest that path reliability can be significantly increased with the proposed algorithm as compared to a reference case (AODV). The improvement was particularly noticeable in networks where nodes can move at high speeds. While the GPS-based case performed the best in terms of route reliability, the system based on tracking sensor nodes produced results close to the GPS case. On the downside, the routes produced by the algorithm tend to be longer than the shortest path, which could impact the individual end-to-end latency of packets. However, the overall impact to the flows would be small or even non-existing in most cases given that the higher reliability of paths will reduce the need for packet transmissions as suggested by our relative energy consumption comparison results.

8. References

[1] M. Abolhasan, T. Wysocki and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks", Ad Hoc Networks, Vol. 2, January 2004, pp.1-22.

[2] Z. Cheng and W. B. Heinzelman, "Discovering long lifetime routes in mobile ad hoc networks", Ad Hoc Networks, Vol. 6, January 2005, pp.661-679.

[3] X. Li, Y. Wang, H. Chen, X. Chu, Y. Wu, and Y. Qi, "Reliable and energy-efficient routing for static wireless ad hoc networks with unreliable links", IEEE Trans. Parallel Distrib. Syst. 20, 10 (Oct. 2009), pp. 1408-1421

[4] H. Pishro-Nik, K. Chan, and F. Fekri, "Connectivity properties of large-scale sensor networks", Wirel. Netw. 15, 7 (Oct. 2009), pp. 945-964

[5] G. Treplan, L. Tran-Thanh, A. Olah, and J. Levendovszky, "Reliable and energy aware routing protocols for wireless sensor networks", In Proceedings of the 17th international Conference on Software, Telecommunications and Computer Networks (Hvar, Croatia, September 24 - 26, 2009). IEEE Press, Piscataway, NJ, pp. 171-175

[6] C. K. Toh, "Associativity-based routing for ad-hoc mobile networks", Wireless Personal Communications, Vol. 4, 1997, pp. 103–139.

[7] R. Dube, K. Wang, C. D. Rais, and S. K. Tripathi, "Signal stability-based adaptive routing (SSA) for ad hoc mobile networks", IEEE Personal Communications, Vol. 4, No. 1, February 1997, pp. 36-45

[8] A. B. McDonald, and T. F. Znati, "A mobility-based framework for adaptive clustering in wireless ad hoc networks", IEEE Journal on Selected Areas in Communications, Vol. 17, No. 8, August 1999, pp. 1466–1487.

[9] J.A. Barria and R. Lent, "MANET route discovery using residual lifetime estimation", 2006, IEEE International Symposium onWireless Pervasive Computing ISWPC 2006.

[10] W. Su, S. J. Lee, and M. Gerla, "Mobility prediction in wireless networks", in Proceedings of the 2000 Military Communications Conference, 2000.

[11] W. Su, S. Lee, and M. Gerla, "Mobility prediction and routing in ad hoc wireless networks", International Journal of Network Management, Wiley & Sons, 11:3-30, 2001

[12] Sulabh Agarwal, Ashish Ahuja, Jatinder Pal Singh, and Rajeev Shorey, "Route-lifetime assessment based routing (RABR) protocol for mobile ad-hoc networks", In Proc. IEEE International Conference on Communications 2000 (ICC'00), pp. 1697-1701

[13] K. Paul, S. Bandyopadhyay, A. Mukherjee, and D. Saha, "A stability-based distributed routing mechanism to support unicast and multicast routing in ad hoc wireless network", Computer Communications Volume: 24, Issue:18, December 1, 2001, pp. 1828-1845

[14] Michael Gerharz, Christian de Waal, Matthias Frank, and Peter Martini, "Link stability in mobile wireless ad hoc networks", in Proceedings of the 27th Annual IEEE Conference on Local Computer Networks (LCN'02), Tampa, FL, November 2002, pp. 30-39

[15] M. Gerharz, C. de Waal, and P. Martini, "Strategies for finding stable paths in mobile wireless ad hoc networks", in Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks (LCN'03), 2003

[16] E. Gelenbe and R. Lent, "Link quality-aware routing", in Proceedings of the 1st ACM international Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (Venezia, Italy, October 04 - 04, 2004). PE-WASUN '04. ACM, New York, NY, pp. 87-90

[17] W. Naruephiphat and C. Charnsripinyo, "Routing algorithm for balancing network lifetime and reliable packet delivery in mobile ad hoc networks", in Proceedings of the 2009 Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing (July 07 - 09, 2009). UIC-ATC. IEEE Computer Society, Washington, DC, pp. 257-262

[18] A. McDonald and T. Znati, "A path availability model for wireless ad hoc networks", in Proc. IEEE WCNC, September 1999, pp. 35-40

[19] S. Jiang, D. He and J. Rao, "A prediction-based link availability estimation for mobile ad hoc networks", in Proc. IEEE Infocom, April 2001, pp. 1745-1752

[20] Y. Taj and K. Faez, "Signal strength based reliability: a novel routing metric in MANETs", In Proceedings of the 2010 Second international Conference on Networks Security, Wireless Communications and Trusted Computing - Volume 01 (April 24 - 25, 2010). NSWCTC. IEEE Computer Society, Washington, DC, pp. 37-40

[21] T. S. Rappaport, "Wireless communications principles and practice", Prentice Hall PTR, New Jersey, 1996

[22] C. Tang, C. Raghavendra, and V. Prasanna, "Energy efficient adaptation of multicast protocols in power controlled wireless ad hoc networks", in Proc. of the International Symposium on Parallel Architectures, Algorithms and Networks IEEE ISPAN'02, pp 91-98, 2002.

[23] K. Moaveninejad, W-Z Song, and X-Y Li, "Robust position-based routing for wireless as hoc networks", Ad Hoc Networks, Vol. 3, January 2005, pp.546-559.

[24] Y. B. Ko and N. H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks", Wirel. Netw. 6, 4 (Jul. 2000), 307-321

[25] I. Stojmenovic, "Position-based routing in ad hoc networks", Open Call Article, IEEE Communications Magazine, July 2002

[26] S. Giordano, I. Stojmenovic, and L. Blazevic, "Position based routing algorithms for ad-hoc networks: a taxonomy", Ad Hoc Wireless Networking, 2003

[27] S. Lee and Y. Ko, "Efficient geocasting with multi-target regions in mobile multi-hop wireless networks", Wirel. Netw. 16, 5 (Jul. 2010), pp. 1253-1262

[28] G. He and J. C. Hou, "Tracking targets with quality in wireless sensor networks", IEEE 13th ICNP 2005, 6-9 Nov 2005, Boston, MA, US, pp 63-74

[29] S. Pattem, S. Poduri and B. Krishnamachan, "Energy quality tradeoffs for target tracking in wireless sensor networks", IPSN, LNCS 2634, pp. 32-46, 2003 Springer Verlag.

[30] S. Pattem and B. Krishnamachan, "Energy quality tradeoffs in sensor tracking: selective activation with noisy measurements", in proc of SPIE 17th Annual Intl. Symposium on Aerospace/Defense Sensing, Simulation, and Controls, Aerosense, April 2003.

[31] C. Chen, C.Weng, and Y. Kuo, "Signal strength based routing for power saving in mobile ad hoc networks", J. Syst. Softw. 83, 8 (Aug. 2010), pp. 1373-1386

[32] H. Yang and B. Sikdar, "A protocol for tracking mobile targets using sensor networks", Proceedings of IEEE Workshop on Sensor Network Protocols and Applications, Anchogare, A.K

[33] D. Moore, J. Leonard, D. Rus, and S. Teller. "Robust distributed network localization with noisy range measurements", in Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys '04). Baltimore, MD. November 3-5, 2004. pp. 50-61

[34] S Meguerdichian, F Koushanfar, M Potkonjak, and M B Srivastava, "Coverage problems in wireless ad hoc sensor networks", IEEE Infocom 2001, pp. 1380-1387.

[35] M. Ishizuka, M. Aida, "Performance study of node placement in sensor networks", 24th ICDCSW'04, 2004

[36] Y. Zou and K. Chakrabarty, "Uncertainty-aware sensor deployment algorithms for surveillance applications", Globecom 2003, pp. 2972-2976, 2003

[37] S S Dhillon and K Chakrabarty, "Sensor placement for Effective Coverage and Surveillance in Distributed Sensor Networks", IEEE 2003

[38] H. Yang and B. Sikdar, "A protocol for tracking mobile targets using sensor networks", in Proceedings of IEEE Workshop on Sensor Network Protocols and Applications, Anchogare, A.K

[39] G Wang, G Cao, and T LaPorta, "A bidding protocol for deploying mobile sensors", 11th ICNP'03 2003

[40] T Wong, T Tsuchiya, and T Kikuno, "A self-organising technique for sensor placement in wireless micro-sensor networks", n Proceedings of the 18th international Conference on Advanced information Networking and Applications - Volume 2 (March 29 - 31, 2004). AINA. IEEE Computer Society, Washington, DC, 78

[41] F Zhao, J Shin and J Reich, "Information-driven dynamic sensor collaboration", IEEE Signal Processing Magazine, March 2002

[42] V Hingne A Joshi E Houstis, and J Michopoulos, "On the grid and sensor networks", IEEE/ACM International Workshop on Grid Computing, 2003

[43] D Guo and X Wang, "Dynamic sensor collaboration via sequential montecarlo", IEEE 2004

[44] S. Pattem, S. Poduri, and B. Krishnamachari, "Energy-quality tradeoffs for target tracking in wireless sensor networks", the 2nd Workshop on Information Processing in Sensor Networks (IPSN 2003), April 2003

[45] D. B. Johnson, D.A. Maltz, Y-C. Hu, and J. G. Jetcheva, "The dynamic source routing protocol for mobile ad hoc networks", IETF draft, Mar 2001

[46] C. E. Perkins and E. M. Royer, "Ad hoc on demand distance vector routing", Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, Feb 1999, pp. 90-100

[47] Y-B. Ko, N.H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks", Proceedings of Mobicom, pp. 66-75, 1998.

[48] B. Karp and H.T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks", Proceedings of the 6th annual international conference on Mobile computing and networking, pp. 243–254 2000.

[49] R. Lent, "INES: Network simulations on virtual environments", in Proceedings of International Conference on Simulation Tools and Techniques for Communications, Networks and Systems, March 2008.

7

ADHOCTCP: Improving TCP Performance in Ad Hoc Networks

Seyed Mohsen Mirhosseini and Fatemeh Torgheh
Islamic Azad University-HidajBranch, Islamic Azad University-AbharBranch
Iran

1. Introduction

A mobile ad-hoc network (MANET) is a special type of wireless networks. It consists of a collection of mobile nodes that are capable of communicating with each other without help from a fixed infrastructure. The interconnections between nodes are capable of changing on a continual and arbitrary basis. Nodes within each other's radio range communicate directly via wireless links, while those that are far apart use other nodes as relays in a multi-hop routing fashion. The typical applications of MANETs include conferences or meetings, emergency operations such as disaster rescue, and battlefield communications.

Transmission Control Protocol (TCP) [1] is a reliable, connection-oriented, full-duplex, transport protocol widely used in wired networks. TCP’s flow and congestion control mechanisms are based upon the assumption that packet loss is an indication of congestion. While this assumption holds in wired networks, it does not hold in the case of mobile wireless networks.

In addition to congestion, a transport protocol in an ad hoc network must handle mobility-induced disconnection and reconnection, route change-induced packet out-of-order delivery for mobile hosts, and error/contention prone wireless transmissions. Reaction to these events might require transport control actions different from congestion control. It might be better to periodically probe the network during disconnection than to back off exponentially [2], and it makes more sense simply to re-transmit a packet lost to random channel error than to multiplicatively decrease the current congestion window [3]. Even if the correct action is executed in response to each type of network event, it is not immediately obvious how to construct an engine that will accurately detect and classify events. Packet loss alone cannot detect and differentiate all these new network events [4].

In this paper, we first describe the necessary network states in an ad hoc network to be identified by TCP and use an end-to-end approach for identification of congestion state in ad hoc network then examine metrics that can be measured end-to-end. Two metrics are devised to detect congestion, IDD (Inter Delay Difference) and STT (Short Term Throughput).The approach we propose in this paper utilizes network layer feedback (from intermediate hops) for identification of disconnection state to put TCP sender into persist mode. Therefore we use from advantage of both end to end measurements and network layer feedback.

The remainder of the chapter is organized as follows: It starts with describing TCP’s challenges in MANETs environment in Section 2. Section 3 provides an overview of related works. The design and implementation of ADHOCTCP are presented in Section 4. Simulations results are given in Sections 5.we conclude the chapter in Section 6.

2. Challenges for TCP in MANETs

TCP assumes that network congestion has happened whenever a packet is lost. It then invokes appropriate congestion control actions including window size reduction. Although this assumption is reasonable for wired networks, it is questionable for wireless networks especially MANETs. Other than congestion, possible causes of packet losses in MANETs include, wireless link errors, MAC layer losses due to channel contention, and link breakages due to node mobility. All those causes that are not related to congestion can result in unnecessary congestion control, which will degrade the TCP performance.

Unlike wired networks, some unique characteristics of mobile ad hoc networks seriously deteriorate TCP performance. These characteristics include the unpredictable wireless channels due to fading and interference, the vulnerable shared media access due to random access collision, the hidden terminal problem and the exposed terminal problem, and the frequent route breakages due to node mobility. Undoubtedly, all of these pose great challenges on TCP to provide reliable end-to-end communications in mobile ad hoc networks. From the point of view of network layered architecture, these challenges can be broken down into six categories: lossy channels, hidden and exposed stations, network partitions, path asymmetry, route failures, and Energy Efficiency.

2.1 Lossy channels

Wireless links posses high bit error rates that cannot be ignored. But TCP interprets packet losses caused by bit errors as congestion. As a result, its performance suffers in wireless networks when TCP unnecessarily invokes congestion control, causing reduction in throughput and link utilization.

The main causes of errors in wireless channels are the following:

- Signal attenuation: This is due to a decrease in the intensity of the electromagnetic energy at the receiver (e.g. due to long distance), which leads to low signal-to-noise ratio (SNR).
- Doppler shift: This is due to the relative velocities of the transmitter and the receiver. Doppler shift causes frequency shifts in the arriving signal, thereby complicating the successful reception of the signal.
- Multipath fading: Electromagnetic waves reflecting off objects or diffracting around objects can result in the signal traveling over multiple paths from the transmitter to the receiver. Multipath propagation can lead to fluctuations in the amplitude, phase, and geographical angle of the signal received at a receiver.

In order to increase the success of transmissions, link layer protocols implement Automatic Repeat reQuest (ARQ) or Forward Error Correction (FEC), or both. For example, IEEE 802.11 implements ARQ, so when a transmitter detects an error, it will retransmit the frame; error detection is timer based.

Bluetooth implements both ARQ and FEC on some synchronous and asynchronous connections.

Note that packets transmitted over a fading channel may cause the routing protocol to incorrectly conclude that there is a new one-hop neighbor. This one-hop neighbor could provide a shorter route to even more distant nodes. Unfortunately, this new shorter route is usually unreliable.

2.2 Hidden and exposed stations

In ad hoc networks, stations may rely on physical carrier-sensing mechanisms to determine an idle channel, such as in the IEEE 802.11 DCF function[5]. Contention-based medium access control (MAC) schemes, such as the IEEE 802.11 MAC protocol, have been widely studied and incorporated into many wireless testbeds and simulation packages for wireless multi-hop ad hoc networks, where the neighboring nodes contend for the shared wireless channel before transmitting. There are three key problems, the hidden terminal problem, the exposed terminal problem, and unfairness[6].

Before explaining these problems, we need to clarify the term "transmission range." The transmission range is the range, with respect to the transmitting station, within which a transmitted packet can be successfully received.

A hidden node is the one that is within the interfering range of the intended receiver but out of the sensing range of the transmitter. The receiver may not correctly receive the intended packet due to collision from the hidden node. As shown in Fig. 1, a collision may occur, for example, when terminal A and C start transmitting toward the same receiver, terminal B in the figure. A typical hidden terminal situation is depicted in Fig. 1. Stations A and C have a frame to transmit to station B. Station A cannot detect C's transmission because it is outside the transmission range of C. Station C (resp. A) is therefore "hidden" to station A (resp. C). Since the transmission areas of A and C are not disjoint, there will be packet collisions at B. These collisions make the transmission from A and C toward B problematic. To alleviate the hidden station problem, virtual carrier sensing has been introduced. It is based on a two-way handshaking that precedes data transmission. Specifically, the source station transmits a short control frame, called Request-To-Send (RTS), to the destination station. Upon receiving the RTS frame, the destination station replies by a Clear-To- Send (CTS) frame, indicating that it is ready to receive the data frame. Both RTS and CTS frames contain the total duration of the data transmission. All stations receiving either RTS or CTS will keep silent during the data transmission period (e.g. station C in Fig. 1).

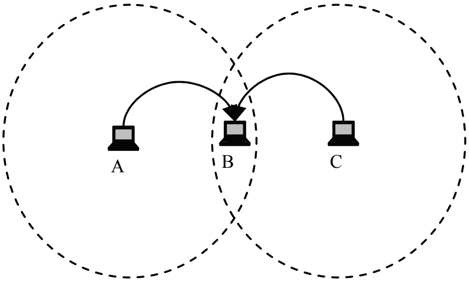


Fig. 1. Hidden terminal problem

However, as pointed out in, the hidden station problem may persist in IEEE 802.11 ad hoc networks even with the use of the RTS/CTS handshake, because the power needed to interrupt a packet reception is much lower than that required to deliver a packet successfully[7,8]. In other words, a node's transmission range is smaller than the sensing node range.

An exposed node is the one that is within the sensing range of the transmitter but out of the interfering range of the receiver. Though its transmission does not interfere with the receiver, it could not start transmission because it senses a busy medium, which introduces spatial reuse inefficiency. The binary exponential backoff scheme always favors the latest successful transmitter and results in unfairness.

The exposed station problem results from a situation where a transmission has to be delayed because of the transmission between two other stations within the sender's transmission range. In Fig. 2 we show a typical scenario where the exposed terminal problem occurs. Let us assume that A and C are within B's transmission range, and A is outside C's transmission range. Let us also assume that B is transmitting to A, and C has a frame to be transmitted to D. According to the carrier sense mechanism, C senses a busy channel because of B's transmission. Therefore, station C will refrain from transmitting to D, although this transmission would not cause interference at A. The exposed station problem may thus result in a reduction of channel utilization.

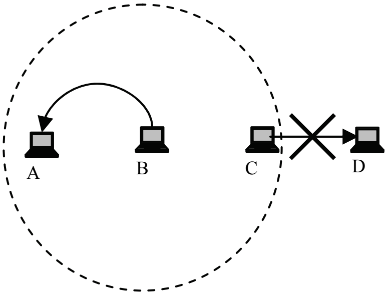


Fig. 2. Exposed terminal problem

It is worth noting that hidden terminal and exposed terminal problems are correlated with the transmission range. By increasing the transmission range, the hidden terminal problem occurs less frequently. On the other hand, the exposed terminal problem becomes more important as the transmission range identifies the area affected by a single transmission.

When TCP runs over 802.11 MAC, as pointed out, the instability problem becomes very serious. It is shown that collisions and the exposed terminal problem are two major reasons for preventing one node from reaching the other when the two nodes are in each other's transmission range. If a node cannot reach its adjacent node for several times, it will trigger a route failure, which in turn will cause the source node to start route discovery. Before a new route is found, no data packet can be sent out. During this process, TCP sender has to wait and will invoke congestion control algorithms if it observes a timeout. Serious oscillation in TCP throughput will thus be observed. Since large data packet sizes and back-to-back packet transmissions both decrease the chance of the intermediate node to obtain the channel, the node has to back off a random period of time and try again. After several failed tries, a route failure is reported.

2.3 Network partition

An ad hoc network can be represented by a simple graph *G*. Mobile stations are the "vertices." A successful transmission between two stations is an undirected "edge."

Network partition happens when *G* is disconnected. The main reason for this disconnection in MANETs is node mobility.

Mobility may induce link breakage and route failure between two neighboring nodes, as one mobile node moves out of the other's transmission range. Link breakage in turn causes packet losses. As we said earlier, TCP cannot distinguish between packet losses due to route failures and packet losses due to congestion. Therefore, TCP congestion control mechanisms react adversely to such losses caused by route breakages. Meanwhile, discovering a new route may take significantly longer time than TCP sender's RTO. If route discovery time is longer than RTO, TCP sender will invoke congestion control after timeout. The already reduced throughput due to losses will further shrink. It could be even worse when the sender and the receiver of a TCP connection fall into different network partitions. In such a case, multiple consecutive RTO timeouts lead to inactivity lasting for one or two minutes even if the sender and receiver finally get reconnected[9].

Another factor that can lead to network partition is energy constrained operation of nodes. An example of network partition is illustrated in Fig. 3. In this figure dashed lines are the links between nodes. When node D moves away from node C this movement, cause to network partition into two separate components. Clearly, the TCP agent of node A cannot receive the TCP ACK transmitted by node F. Originally, TCP does not have an indication about the exact time of network reconnection.

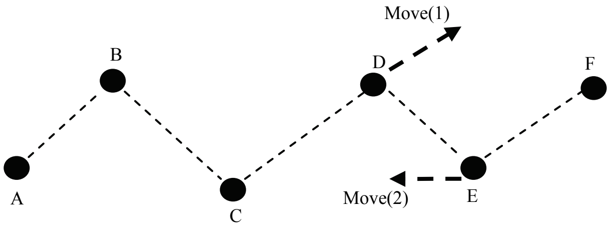


Fig. 3. Example for Network partition

This lack of indication may lead to long idle periods during which the network is connected again, but TCP is still in the backoff state.

2.4 Path asymmetry

Path asymmetry in ad hoc networks may appear in several forms as bandwidth asymmetry, loss rate asymmetry, and route asymmetry.

Bandwidth Asymmetry: Satellite networks suffer from high bandwidth asymmetry, resulting from various engineering tradeoffs (such as power, mass, and volume), as well as the fact that for space scientific missions, most of the data originates at the satellite and flows to the earth. The return link is not used, in general, for data transferring. For example, in broadcast satellite networks the ratio of the bandwidth of the satellite-earth link over the bandwidth of the earth-satellite link is about 1000 [10]. On the other hand, in ad hoc networks, the degree of bandwidth asymmetry is not very high. For example, the bandwidth ratio lies between 2 and 54 in ad hoc networks that implement the IEEE 802.11 version g protocol [11]. The asymmetry results from the use of different transmission rates. Because of this different transmission rates, even symmetric source destination paths may suffer from bandwidth asymmetry.

Loss Rate Asymmetry: This type of asymmetry takes place when the backward path is significantly more lossy than the forward path. In ad hoc networks this asymmetry occurs because packet losses depend on local constraints that can vary from place to place. Note that loss rate asymmetry may produce bandwidth asymmetry. For example, in multi-rate IEEE 802.11 protocol versions, senders may use the Auto- Rate-Fallback (ARF) algorithm for transmission rate selection [12]. With ARF, senders attempt to use higher transmission rates after consecutive transmission successes, and revert to lower rates after failures. So, as the loss rate increases the sender will keep using lower transmission rates.

Route Asymmetry: Unlike the previous two forms of asymmetry, where the forward path and the backward path can be the same, route asymmetry implies that distinct paths are used for TCP data and TCP ACKs. This asymmetry may be an artifact of the routing protocol used. Route asymmetry increases routing overheads and packet losses in the case of a high degree of mobility,1 because when nodes move, using a distinct forward and reverse route increases the probability of route failures experienced by TCP connections. However, this is not the case with static networks or networks that have a low degree of mobility, as in the case of a network with routes of high lifetime compared to the session transfer time. So it is up to the routing protocols to select symmetric paths when such routes are available in the case of ad hoc networks of high mobility.

In the context of satellite networks, there has been much research on how to improve TCP performance. However, since satellite networks are out of the scope of this article, we will limit ourselves to list three techniques introduced by these proposals, which we believe might be useful in ad hoc networks.

2.5 Routing failures

In wired networks route failures occur very rarely. In MANETs they are frequent events. The main cause of route failures is node mobility. Another factor that can lead to route failures is the link failures caused by the contention on the wireless channel, which is the main cause of TCP performance degradation in SANETs. The route reestablishment duration after route failure in ad hoc networks depends on the underlying routing protocol, mobility pattern of mobile nodes, and traffic characteristics. As already discussed, if the TCP sender does not have indications on the route re-establishment event, the throughput and session delay will degrade because of the large idle time. Also, if the new route established is longer or shorter in term of hops, than the old route TCP will face a brutal fluctuation in round trip time (RTT)[13].

In addition, in ad hoc networks, routing protocols that rely on broadcast Hello messages to detect neighbors' reachability may suffer from the "communication gray zones" problem. In these zones data messages cannot be exchanged, although broadcast Hello messages and control frames indicate that neighbors are reachable. So on sending a data message, routing protocols will experience routing failures.

2.6 Energy efficiency

As power is limited at mobile nodes, any successful scheme must be designed to be energy efficient. In some scenarios where battery recharge is not allowed, energy efficiency is critical for prolonging network lifetime[14]. Because batteries carried by each mobile node have limited power supply, processing power is limited. This is a major issue in ad hoc networks, as each node is acting as an end system and as a router at the same time, with the implication that additional energy is required to forward and relay packets. TCP must use this scarce power resource in an "efficient" manner. Here, efficiency means minimizing the number of unnecessary retransmissions at the transport layer as well as at the link layer.2 In general, in ad hoc networks there are two correlated power problems: the first problem is "power saving," which aims at reducing power consumption; the second problem is "power control," which aims at adjusting the transmission power of mobile nodes[31]. Power saving strategies has been investigated at several levels of a mobile device, including the physical-layer transmissions, the operation systems, and the applications. Power control can be jointly used with routing or transport agents to improve the performance of ad hoc networks. Power constraints on communications also reveal the problem of cooperation between nodes, as nodes may not participate in routing and forwarding procedures in order to save battery power[32].

3. Current approaches to improving TCP performance in MANETs

In this section we present some schemes that have been proposed to improve TCP performance in ad hoc networks. There are some approaches for classifying these proposals that we introduce two of the most common of these classifying approaches. In first classifying scheme we classify these proposals in two categories: cross layer proposals and layered proposals. In layered proposals, the adaptation involves only one OSI layer, whereas in cross layer proposals at least two OSI layers are involved.

We classify layered proposals according to which layer the adaptation is done: at the TCP layer or at the link layer. On the other hand Cross layer proposals can be classified in three types :(1)TCP and network cross layer, (2)TCP and physical cross layer, and(3) network and physical cross layer.

Another classifying method can be as follow:

1. Modified TCP: This represents a class of transport layer approaches, where minor modifications are made to the TCP protocol to adapt it to the characteristics of an ad-hoc network, but the fundamental elements of TCP are still retained.
2. TCP aware Cross Layer Solutions: This represents a class of lower layer approaches that hide from TCP the unique characteristics of ad-hoc networks, and thus necessitate minimal changes to TCP. Such approaches can be used in tandem with the approaches in the previous class.
3. Ad-hoc Transport Protocols: Finally, this represents a class of new built-from-scratch transport protocols that are built specifically for the characteristics of an ad-hoc network, and are not necessarily TCP-like.

In the rest of this section we discuss in detail specific protocol instances of the different approaches and highlight the main features of each one. We classify these selected approaches in terms of usage from network layer feedback or not (using feedback means the proposal is cross layer solution). We terminate this section with describing a proposal that are not a modification from TCP but a new transport protocol that is suitable for ad hoc environments.

3.1 TCP with feedback solutions

Route changes are triggered by link breakages at some intermediate nodes (possibly the sender itself). Detecting these link Breakages is a basic requirement for any ad-hoc routing protocol. If the intermediate nodes, where the breakages happen, can convey this information back to the sender, the TCP controller at the sender will be able to detect the event. We call this a network layer feedback mechanism. The majority of the existing approaches employ this detection mechanism, namely TCP-F (TCP-Feedback)[5], ELFN(Explicit Link Failure Notification)[16], ATCP (AdhocTCP)[7],and TCP-BuS[8].

3.1.1 TCP-F

TCP-F [15] relies on the network layer at an intermediate node to detect the route failure due to the mobility of its downstream neighbor along the route. A sender can be in an active state or a snooze state. In the active state, transport layer is controlled by the normal TCP. As soon as an intermediate node detects a broken route, it explicitly sends a route failure notification (RFN) packet to the sender and records this event. Upon reception of the RFN, the sender goes in to the snooze state, in which the sender completely stops sending further packets, and freezes all of its timers and the values of state variables such as RTO and congestion window size. Meanwhile, all upstream intermediate nodes that receive the RFN invalidate the particular route to avoid further packet losses. The sender remains in the snooze state until it is notified of the restoration of the route through a route reestablishment notification (RRN) packet from an intermediate node. Then it resumes the transmission from the frozen state.

3.1.2 TCP-ELFN

Holland and Vaidya proposed this feedback-based technique, the Explicit Link Failure Notification (ELFN)[16,19].The goal is to inform the TCP sender of link and route failures so that it can avoid responding to the failures as if congestion occurs. ELFN is based on the dynamic source routing (DSR)[20]routing protocol. To implement ELFN message, the route failure message of DSR is modified to carry a payload similar to the “host unreachable” ICMP (Internet Control Message Protocol) message. Upon receiving an ELFN, the TCP sender disables its congestion control mechanisms and enters in to a “stand-by” mode, which is similar to the snooze state of TCP-F mentioned above. Unlike TCP-F using an explicit notice to signal that a new route has been found, the sender, while on stand-by, periodically sends a small packet to probe the network to see if a route has been established. If there is a new route, the sender leaves the stand-by mode, restores its RTO and continues as normal. Recognizing most of popular routing protocols in ad hoc networks are on demand and route discovery/rediscovery is event driven, periodically sending a small packet at the sender is appropriate to restore routes with mild overhead and without modification to the routing layer.

3.1.3 ATCP protocol

ATCP [17] does not impose changes to the standard TCP itself. Rather it implements an intermediate layer between network and transport layers in order to lead TCP to an enhanced performance and still maintain inter operation with non-ATCP machines. In particular, this approach relies on the ICMP protocol and ECN scheme to detect network partition and congestion, respectively. In this way, the intermediate layer keeps track of the packets to and from the transport layer so that the TCP congestion control is not invoked when it is not really needed, which is done as follows. When three duplicate ACKs are detected, indicating a lossy channel, ATCP puts TCP in “persist mode” and quickly retransmits the lost packet from the TCP buffer; After receiving the next ACK the normal state is resumed. In case an ICMP “Destination Unreachable” message arrives, pointing out a network partition, ATCP also puts the TCP in “persist mode” which only ends when the connection is reestablished. At last, when network congestion is detected by the receipt of an ECN message, the ATCP does nothing but forwards the packet to TCP so that it can invoke its congestion control normally.

This model was implemented in a test bed and evaluated under different constraints such as congestion, lossy scenario, partition, and packet re ordering. In all cases the transfer time of a given file by ATCP yielded better performance comparatively to TCP. However, again the used scenario was somewhat special, since neither wireless links nor ad hoc routing protocols were considered. In fact, such experiments relied on a simple ethernet networks connected in series in which each node had two ethernet cards. Moreover, some assumptions such as ECN-capable nodes as well as sender node being always reachable might be somehow hard to be met. In case the latter is not fulfilled, for example, the ICMP message might not even reach the sender which would retransmit continuously instead of entering "persists mode". Also, ECN scheme deployment raises security concerns [ECN], and it might compromise the viability of this scheme.

In summary, as shown by the simulations, these feedback-based approaches improve TCP performance significantly while maintaining TCP's congestion control behavior and end-to-end TCP semantics. However, all these schemes require that the intermediate nodes have the capability of detecting and reporting network states such as link breakages and congestion. Enhancement at the transport layer, network layer, and link layer are all required. It deserves further research on the ways to detect and distinguish network states in the intermediate nodes.

3.1.4 TCP-BuS

TCP-BuS[18]is similar to TCP-F in detection mechanisms. Two control messages (ERDN and ERSN) related to route maintenance are introduced to notify the TCP sender of route failures and route reestablishment. These indicators are used to differentiate between network congestion and route failures as a result of node movement. ERDN (Explicit Route Disconnection Notification) message is generated at an intermediate Node upon detection of a route disconnection, and is propagated toward the sender. After receiving an ERDN message, the sender stops transmission. Similarly, after discovering a new partial path from the failed node to the destination, the failed node returns an ERSN (Explicit Route Successful Notification) message back to the sender. On receiving ERSN Message, the sender resumes data transmission.

TCP-BuS considers the problem of reliable transmission of control messages. If a node A reliably sends an ERDN message to its upstream node B, the ERDN message subsequently forwarded by node B can be overheard by A (assuming same transmission ranges of A and B). Thus, if a node has sent an ERDN message but cannot overhear any ERDN message relayed by its upstream node during a certain period, it concludes the ERDN message is lost and retransmits it. The reliable transmission of ERSN is similar. To summarize, these mechanisms all rely on the intermediate nodes, where the route Failures are detected, to send some control messages to notify the TCP sender. We categorize and call them the network layer feedback mechanisms.

3.2 TCP without feedback solutions

3.2.1 TCP-DOOR

TCP-DOOR [21] attempts to improve TCP performance by detecting and responding to out-of-order (OOO) packet delivery events and thus avoiding invoking unnecessary congestion

control by definition, OOO occurs when a packet sent earlier arrives later than a subsequent packet. In ad hoc networks, OOO may happen multiple times in one TCP session because of route changes. In order to detect OOO, ordering information is added to TCP ACKs and TCP data packets.OOO detection is carried out at both ends: the sender detects the Out-of-Order ACK packets and the receiver detects the Out-of-Order data packets. If the receiver detects OOO, it should notify the sender, considering the fact that it is the sender who takes congestion control actions. Once the TCP sender knows of an OOO condition, it may take one of the two responsive actions: temporarily disabling congestion control and instant recovery during congestion avoidance. The first action
means that, whenever an OOO condition is detected, TCP sender will keep its state variables such as RTO and the congestion window size constant for a time period T. The second action means that, if during the past time period T the TCP sender has already entered the state of congestion avoidance, and it should recover immediately to the state prior to such congestion avoidance. The main reason is the detection of OOO condition implies that a route change event has just occurred. However, OOO can be detected only after a route has recovered from failures. As a result, TCP-DOOR is less accurate and responsive than a feedback-based approach that is able to determine whether congestion or route errors occur, and hence report to the sender at the very beginning. Furthermore, it may not work well with multi-path routing since multi-path routing may cause OOO as well. Therefore, it is concluded that TCP-DOOR may work as an alternative to the feedback-based approach to improve TCP performance over ad hoc network, if the latter is not available.

3.2.2 Fixed RTO

Fixed RTO [22] is a very simple responding mechanism, originally coming from the consecutive time outs heuristic. If the sender encounters two consecutive Retransmission timeouts, it assumes some events other than congestion happen. Then the Value of retransmission timeout is fixed, without incurring exponential backoff. The RTO Remains fixed until the route is re-established and the retransmitted packet is acknowledged. This simple technique is particularly effective when network partition happens. Without fixing the RTO, it will become longer and longer exponentially, which implies that the chance to probe a valid route is smaller and smaller. An improved approach is, not only to fix the RTO, but also to reset it to the initial value which is a short time period. In other words, it is better to probe the network frequently after a network partition is believed to have happened in order to avoid wasting time idling.

3.3 Ad-hoc transport protocols

In this section we describe a novel transport protocol for MANETs. Unlike other proposals, this protocol is not a modification of the TCP but is specifically tailored to the characteristics of the MANET environment. It is able to manage efficiently route changes and route failures. Furthermore, it includes a completely re-designed congestion control mechanism. Finally, it is designed in such a way to reduce as much as possible the number of useless retransmissions. This is extremely important since retransmissions consume energy.

3.3.1 ATP (Ad hoc Transport Protocol)

ATP (ad-hoc transport protocol) is tailored toward the characteristics of ad-hoc networks. ATP, by design, is an antithesis of TCP and consists of: rate based transmissions, quick-start during connection initiation and route switching, network supported congestion detection and control, no retransmission time outs, decoupled congestion Control and reliability, and coarse grained receiver feedback. Briefly, just as in TCP, ATP primarily consists of mechanisms at the sender to achieve effective congestion control and reliability. However, unlike in TCP, ATP relies on feedback not just from the receiver, but also from the intermediate nodes in the connection path. In terms of specific functionality, the intermediate nodes provide congestion feedback to the sender, while the receiver provides feedback for both flow control and reliability. The receiver also acts as a collator of the congestion information provided by the intermediate nodes in the network before the information is sent back to the sender. The receiver provides the reliability, flow control, and collated congestion control information through periodic messages. The sender on the other hand, is responsible for connection management, start-up rate estimation (with network feedback), congestion control, and reliability.

4. ADHOCTCP

In this section for description o f new proposed approach we first determine the network states that TCP must monitor. Identifying three network states is necessary to improve TCP performance over ad hoc networks that states are: CONGESTION, CHANNEL ERROR, and DISCONNECTION. These states should be our identification target. We use end-to-end measurements to identify the presence of congestion in the network; we must then determine what available end-to-end metrics can be used to accurately identify congestion state in the network. The goal of the identification algorithm is therefore a mapping from metric measurements to the target states that in ADHOCTCP we describe the identification algorithm to decide that network is congested or not. We first assume a situation in which TCP knows why its packets are being lost and consider what TCP should do to improve its performance. First, if the packet loss is due to congestion, TCP should apply the congestion control mechanisms; but if not, TCP might do better not to slow down and exponentially backoff its retransmission timeout. Therefore knowing whether the current state of network is congested or not is important. As it turns out, proper congestion identification proves to be the biggest improvement to TCP in ad hoc networks. Second, if the packet is lost due to reasons other than congestion, TCP can benefit if it further knows whether the loss is due to channel errors or network disconnection. If the loss is due to channel errors, a simple retransmission is adequate. However, if it is due to disconnection, some special probing mechanisms might be needed for a prompt transmission recovery upon network reconnection.

4.1 Congestion

TCP attempt to fully utilize the network bandwidth makes ad hoc networks easily go into congestion. In addition, due to many factors such as route change and unpredictable variable MAC delay, the relationship between congestion window size and the tolerable data rate for a route is no longer maintained in ad hoc networks. The congestion window size computed for the old route may be too large for the newly found route, resulting in network congestion if the sender still transmits at the full rate allowed by the old congestion window size congestion/overload may give rise to buffer overflow and increased link contention, which degrades TCP performance. As a matter of fact, [23] showed the capacity of wireless ad hoc networks decreases as traffic and/or competing nodes arise.

When network congestion occurs, ad hoc transport should adopt the same congestion control actions as conventional TCP [24]. Here, we define congestion as queue build-up and packets being dropped due to buffer overflow at some nodes.

4.1.1 Identifying congestion

There are two types of approaches in detecting network congestion in the Internet. One is based on end-to-end measurement and the other on feedback from intermediate gateways in the network. Standard TCP [25] uses end-to-end measurement of RTT and packet loss to detect congestion; RED/ECN [26] provides congestion notification by monitoring the instantaneous queue size at the network gateways.

The end-to-end approach is easy to implement and deploy, requires no network support, and provides the flexibility for backward compatibility. However, using single metric measurements, the probability of false congestion detection in an uncongested ad hoc network is quite high. This sort of false detection can lead to serious throughput degradation. In ADHOCTCP we use of multi-metric joint identification for identifying congestion in ad hoc networks. By exploiting the degree of independence in measurement noise of individual metrics, the probability of false identification can be significantly reduced by cross-verification.

Two metrics are devised to detect congestion, IDD (Inter Delay Difference) and STT (Short Term Throughput). They each exhibit a unique pattern upon congestion; and in non-congestion states, they are influenced by different network conditions in such a way that their respective measurement noise is largely independent.

Inter-packet delay difference (*IDD*) Metric: IDD measures the delay difference between consecutive packets that calculate as fallow:

$A^{i+1} - A^{i} - (S^{i+1} - S^{i})$, where A^{i} is the arrival time of packet i and S^{i} is its sending time from the sender

It reflects the congestion level along the forwarding delivery path by directly sampling the transient queue size variations among the intermediate nodes.

Short-term throughput (*STT*): STT metric calculate as fallow:

Np(T)/T , where Np(T) is the # of received packets during interval T

Compared with IDD, STT is also intended for network congestion identification. It provides observation over a time interval T, and is less sensitive to short term out-of-order packet delivery than IDD. Therefore, STT is more robust to transient route changes, which can be very frequent in a mobile ad hoc network. However, using STT alone to detect network congestion can be susceptible to measurement noise introduced by bursty channel error, network disconnections or altering TCP source rates. We combine STT and IDD to jointly identify network congestion.

We identify a congestion state when both IDD is HIGH and STT is LOW, and non-congestion state if otherwise. We define a value to be HIGH or LOW if respectively it is within the top or bottom 30% of all samples.

The identification module is plugged into the receiver side. Space is allocated for storing metrics samples. Identifying high or low related calculations are performed after normal processing of each incoming data packet. One bit used for representing congestion state. We introduce an option field in the TCP header and set the corresponding bit in each outgoing ACK packet. Algorithm 1 shows receiver side algorithm:

Algorithm1: Receiver Side Algorithm: Upon packet arrival

```
process data and generate ACK packet
compute sample value for two metrics
estimate HIGH/LOW for each metric
network state identification (congested or not)
set state bit in option field of out-going ACK packet
transmit ACK
```

Logic to process the ADHOCTCP option field of a TCP header is introduced at the sender side to read in this bit from incoming ACK packets. We extend the code that handles third duplicate ACKs and retransmission timeouts to follow the ADHOCTCP design. In particular, when a sender goes into the probing state, it caches its current transmission state and begins using small packets (8 bytes payload) to probe the receiver until it receives an acknowledgement of the reception of the probing packet. Upon leaving the probing state, the previous transmission state is then restored.

4.2 Disconection

4.2.1 Impact of mobility

Mobility may induce link breakage and route failure between two neighboring nodes, as one mobile node moves out of the other's transmission range. Link breakage in turn causes packet losses and TCP cannot distinguish between packet losses due to route failures and packet losses due to congestion. Therefore, TCP congestion control mechanisms react adversely to such losses caused by route breakages [18,22,27]. Meanwhile, discovering a new route may take significantly longer time than TCP sender's RTO. If route discovery time is longer than RTO, TCP sender will invoke congestion control after timeout. The already reduced throughput due to losses will further shrink. It could be even worse when the sender and the receiver of a TCP connection fall into different network partitions. In such a case, multiple consecutive RTO timeouts lead to inactivity lasting for one or two minutes even if the sender and receiver finally get reconnected. Fu et al. conducted simulations considering mobility, channel error, and shared media-channel contention [4]. They indicated that mobility-induced network disconnections and reconnections have the most significant impact on TCP performance comparing to channel error and shared media-channel contention. As mobility increases, compared to a reference TCP, TCP NewReno suffers from a relative throughput drop ranging from almost 0% in a static case to 90% in a highly mobile case (when moving speed is 20m/s). In contrast, congestion and mild channel error (say 1%) have less visible effect on TCP (with less than 10% performance drop compared with the reference TCP).

4.2.2 Disconnection identification and reaction

It is likely that the ad hoc network may periodically get partitioned for several seconds at a time. If the sender and the receiver of a TCP connection lie in different partitions., all the sender's packets get dropped by the network resulting in the sender invoking congestion control. If the partition lasts for a significant amount of time (say several times longer than the RTO), the situation gets even worse because of a phenomena called *serial timeouts*.

The goal is to inform the TCP sender of link and route failures so that it can avoid responding to the failures as if congestion occurs. Our disconnection identification method is similar to TCP-ELFN [16], we use from EPLN (Explicit Packet Loss Notification) for inform sender. EPLN is based on the dynamic source routing (DSR) [10] routing protocol. To implement EPLN message, the route failure message of DSR is modified to carry a payload similar to the "host unreachable" ICMP (Internet Control Message Protocol) message. Upon receiving an EPLN, the TCP sender disables its congestion control mechanisms and enters into a "stand-by" mode, the sender, while on stand-by, periodically sends a small packet to probe the network to see if a route has been established. If there is a new route, the sender leaves the stand-by mode, restores its RTO and continues as normal.

4.3 Channel error

Bursty bit errors may corrupt packets in transmission, leading to the loss of TCP data packets or acknowledgments (ACKs). If it cannot receive the ACK within the retransmission timeout (RTO), the TCP sender immediately reduces its congestion window to one packet, exponentially backs off its retransmission, and retransmits the lost packet. Intermittent channel errors may thus cause the congestion window size at the sender to remain small, resulting in low throughput.

If RTO expires or sender receives 3 duplicate ack and network state does not detect as congestion by receiver's end to end measurements, sender assume that packet loss is due to channel error. In such case because packet loss is a random packet loss, without slowing down, the sender will re-transmit the lost packet [3][28].

5. Performance evaluation

We used *ns-2* [29] network simulator with Monarch Project's wireless and mobile extensions [30]. The network interface model provides a 2Mbps transmission rate and a nominal transmission range of 250m; the network interface uses IEEE 802.11 DCF MAC protocol [26]. The mobility model is *random waypoint model* in a rectangular field. In this model, a node starts at a random position, picks a random destination, moves to it at a randomly chosen speed, and pauses for a specified pause time. The node speed was randomly chosen from v m/s, where v is node mean speed. We used pause time 0 s for all simulations. The two field configurations we used were 1500m*1000m field with 50 nodes and 2200m*600m field with 100 nodes. We used TCP-NewReno with the packet size of 1460 bytes. The maximum size of both congestion window and receiver's advertised window is 8. FTP is the application that we used over TCP.

5.1 Simulation results

TCP's congestion window never really has been opportunity to grow in size because losses due to bit error result in congestion control. ADHOCTCP's congestion window on the other hand, never shrinks. This accounts for the dramatic difference in TCP and ADHOCTCP performance in Figure 6. TCP's congestion windows remains small making TCP behave almost like a stop-and-wait protocol (figure 4).

The first experiments we ran did not include disconnection or congestion events. The connection was only subjected to bit error that occurred at a BER of 10^{-5} at each hop.

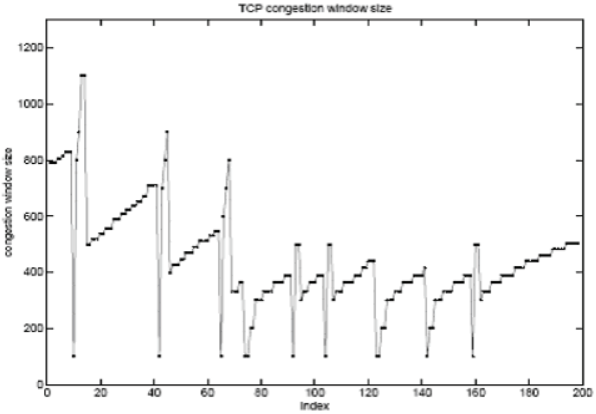


Fig. 4. TCP congestion window in presence of Bit error only

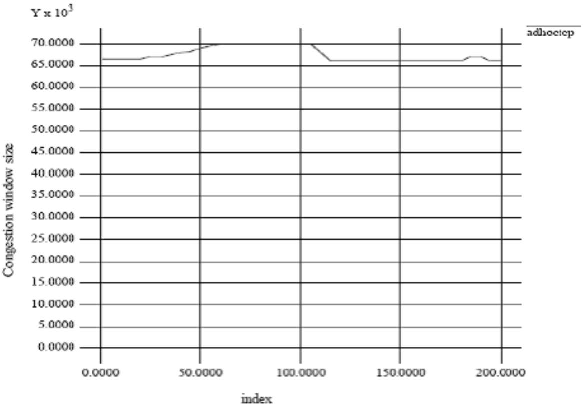


Fig. 5. ADHOCTCP congestion window in presence of bit error only

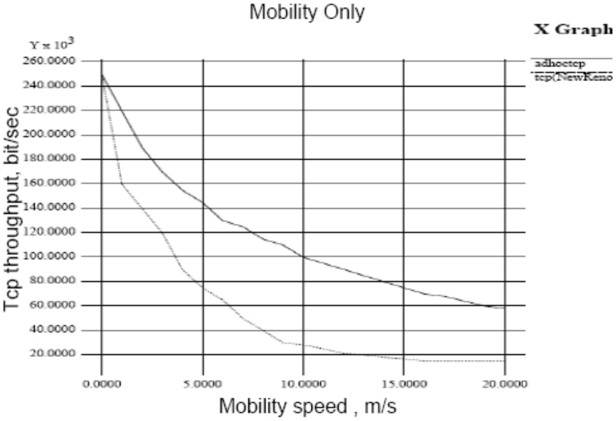


Fig. 6. ADHOCTCP performance in the presence of node mobility

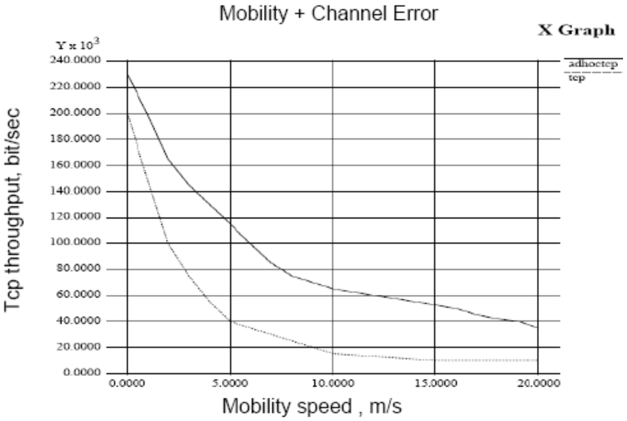


Fig. 7. ADHOCTCP performance in presence of node mobility and channel error

In the next experiment, we introduced periodic congestion in the network that results presented in figure8.

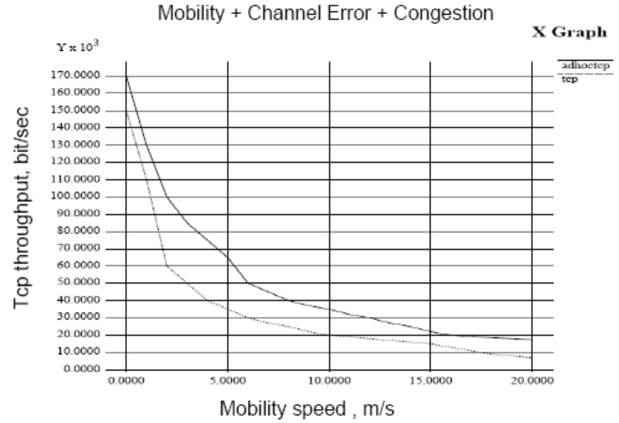


Fig. 8. ADHOCTCP performance in presence of mobility, channel error and congestion

There are a couple of reasons for the difference in performance between TCP and ADHOCTCP .First, the number of time out events in TCP is high because of the high bit error as well as because of loss due to congestion. Thus, TCP does not get much of an opportunity to grow its congestion window. ADHOCTCP, on the other hand, defers to TCP's congestion only when network state identified as congestion by receiver side. In other cases, it retransmits the lost packets from TCP's buffer.

6. Conclusions and future research

As the assumption made by TCP that any packet loss is due to network congestion is not valid in ad hoc networks, either TCP should be capable of distinguishing various reasons of packet losses or such non-congestion related losses should be reduced. To enable TCP to identify various causes of packet losses, there are largely two approaches, depending on whether or not network feedback information is used. Feedback-based schemes seem to be able to react more quickly to non-congestion related packet losses, thus to be more effective in enhancing TCP performance. However, the price to be paid is that they are more difficult to implement, since they require end nodes and intermediate nodes to cooperate with each other. On the other hand, approaches without feedback are relatively simple to implement. However, the performance gain may not be high enough.

This paper explores an alternative approach that relies solely on end-to-end mechanisms. To robustly detect congestion state in the presence of measurement noise, we propose a multiple-metric based joint detection technique. In this technique, a congestion event is signaled only if all the relevant metrics detect it. Our simulations show that ADHOCTCP is able to significantly reduce the probability of false detection while keeping the incompatible detection errors low and thus greatly improves the transportation performance in a TCP friendly way. This demonstrates that the end-to-end approach is also viable for ad hoc networks.

7. References

[1] W. Richard Stevens TCP/IP *Illustrated, Volume 1, The Protocols*, AWL, 1994

[2] G. Holland and N. Vaidya, “Analysis of TCP performance over mobile ad hoc networks,” *MOBICOM’99*.

[3] H. Balakrishnan, S. Seshan, E. Amir, and R. Katz, “Improving TCP/IP performance over wireless networks,” *MOBICOM’95*.

[4] S. Biaz and N.H. Vaidya, ”Distinguishing congestion losses from wireless transmission losses” *IEEE 7th Int. Conf. on Computer Communicationsand Networks*, October 1998.

[5] IEEE 802.11WG, Part 11:Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, Standard, Aug. 1999.

[6] F. Tobagi and L. Kleinrock, “Packet Switching in Radio Channels: Part II — The hidden Terminal Problem in Carrier Sense Multiple-Access Modes and the Busy-Tone Solution,” IEEE Trans. Net., vol. 23, no. 12, 1975, pp. 1417–33.

[7] Z. Fu et al., “The Impact of Multihop Wireless Channel on TCP Throughput and Loss,” Proc. IEEE INFOCOM, San Francisco, USA, Apr. 2003.

[8] K. Xu, M. Gerla, and S. Bae, “Effectiveness of RTS/CTS Handshake in IEEE 802.11-Based Ad Hoc Networks,” Ad Hoc Net. J., Elsevier, vol. 1, no. 1, July 2003, pp. 107–23.

[9] V. Paxson and M. Allman, “Computing TCP’s Retransmission Timer,” RFC 2988, Category: Standard Track, Nov. 2000.

[10] R. Durst, G. Miller, and E. Travis, “TCP Extensions for Space Communications,” Proc. ACM MOBICOM, Rye, NY, 1996, pp. 15-26.

[11] “IEEE 802.11 WLAN standard,” Web site: http://standards. ieee.org/getieee802

[12] A. Kamerman and L. Monteban., “Wavelan II: A High-Performance Wireless LAN for the Unlicensed Band,” Bell Labs Tech. J., Summer 1997, pp. 118–33.

[13] H. Lundgren, E. Nordstro, and C. Tschudin, “Coping with Communication Gray Zones in IEEE 802.11b-Based Ad Hoc Networks,” Proc. ACM Wksp. Wireless Mobile Multimedia, Atlanta, GA, USA, Sept. 2002, pp. 49–55.

[14] C. Jones et al., “A Survey of Energy Efficient Network Protocols for Wireless and Mobile Networks,” ACM Wireless Net., vol. 7, no. 4, 2001, pp. 343–58.

[15] K.Chandran, S.Raghunathan, S.Venkatesan, R.Prakash. A Feedback Based Scheme For Improving TCP Performance In AdHoc Wireless Networks. In Proceedings of International Conference on Distributed Computing Systems- ICDCS '98. pp. 472-479,1997.

[16] G.HollandandN.H.Vaidya,"Analysis of TCP performance over mobile ad hoc networks,"ACMMOBICOM'99, Seattle, August 1999.

[17] J. Liu and S. Singh. ATCP: TCP for mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 19(7):1300{1315, July 2001.

[18] D.Kim,C.-K.Toh,andY.Choi. TCP-BuS: Improving TCP Performance in Wireless Ad Hoc Networks. Journal of Communications and Networks, Vol. 3, No. 2. Jun. 2001.

[19] J.P.Monks, P.Sinhaand V.Bharghavan, "Limitations of TCP-ELFN for ad hoc networks,"MOMUC2000

[20] D. Johnson, D. Maltz, Y.-C. Hu, and J. Jetcheva. The dynamic source routing protocol for mobile ad hoc networks (DSR). IETF Internet-Draft, draft-ietf-manet-dsr-06.txt, work in progress, Nov.2001

[21] F. Wang and Y. Zhang, "Improving TCP Performance over Mobile Ad-Hoc Networks with Out-of-Order Detection and Response," MobiHoc'02, pp. 217-225, Lausanne, Switzerland, Jun 2002.

[22] T.Dyer and R.Boppana. A comparison of TCP performance over three routing protocols for mobile Ad hoc networks. In Proceedingsofthe2001ACM International Symposium on Mobile Ad Hoc Networking & Computing(MobiHoc'01), Long Beach, California, Oct. 2001.

[23] J. Li, C. Blake, D. S. J. De Couto, H. Lee, and R. Morris, "Capacity of ad hoc wireless networks," *ACM MobiCom'01*, Rome, Italy, July 2001.

[24] M.Allman, V.Paxson, and W. Stevens, "TCP Congestion Control" *RFC 2581* April 1999.

[25] S. Floyd. TCP and explicit congestion notification. *ACM Computer Communication Review*, 24(5):8-23, Oct. 1994.

[26] S. Floyd, "TCP and explicit congestion notification," *ACM CCR*, 1994

[27] A. Ahuja, S. Agarwal, J. P. Singh and R. Shorey, "Performance of TCP over different routing protocols in mobile ad-hoc networks," *IEEE VTC 2000*, vol. 3, pp. 2315-2319, Tokyo.

[28] H. Balakrishnan, and R. Katz, "Explicit loss notification and wireless web performances," *Globecom'98*.

[29] K. Fall and K. Varadhan, *ns* notes and documentation, LBNL (August 1998) http://www-mash.cs.berkeley.edu/ns/

[30] J. Broch, D.A. Maltz, D.B. Johnson, Y. Hu and J. Jetcheva, A performance comparison of multi-hop wireless ad hoc network routing protocols, in: *ACM/IEEE International Conference on Mobile Computing and Networking* October 1998) pp. 85–97.

[31] H. Singh and S. Singh, "Energy consumption of TCP reno, newreno, and SACK in multi-hop wireless networks," ACM SIGMETRICS'02, Jul. 2002.

[32] I. Ali, R. Gupta, S. Bansal, A. Misra, A. Razdan and R. Shorey, "Energy efficiency and throughput for TCP traffic in multi-hop wireless networks," IEEE INFOCOM' 02, New York, 2002.

Part 2

Cross Layer Design in Ad Hoc Networks

8

Cross–Layer Design in Wireless Ad Hoc Networks with Multiple Antennas

Ehsan Soleimani-Nasab, Mehrdad Ardebilipour and Mahdi Kashiha
K.N. Toosi University of Technology
Iran

An ad hoc wireless network is a collection of wireless nodes that self-configure to form a network without the aid of any established infrastructure. Some or possibly all of these nodes are mobile. These networks are extremely compelling for applications where a communications infrastructure is too expensive to deploy, cannot be deployed quickly, or is simply not feasible. There are numerous potential applications for ad- hoc wireless networks, ranging from multi-hop wireless broadband Internet access, to sensor networks, to building or highway automation, to voice and video communication for disaster areas.

The lack of established infrastructure, the network and channel dynamics, and the nature of the wireless medium offer an unprecedented set of challenges in supporting demanding applications over ad hoc wireless networks. The wireless channel is inherently a broadcast medium, so transmissions from different nodes interfere with each other. The quality of wireless links vary over time and space due to interference, multipath fading, and shadowing refer to (setton et al., 2005).

Ad hoc networks are harder to design than wired networks because of problems that arise from the every nature of wireless communication. One of these problems, namely the hidden terminal that makes collision. To avoid collisions, a collision avoidance method could be used, as in the well known IEEE 802.11 DCF, which recommends the use of a bidirectional signaling flow made of Request-To-Send (RTS) and Clear-To-Send (CTS) frames before packet transmission, closing the data exchange with an Acknowledgement packet. This scheme forces other nodes in the proximity of the sender and the receiver to defer their own transmissions while a data exchange is in progress, even if they sense a free channel.

Fig. 1. shows a typical scenario of "hidden-terminal". Suppose that the B station is in the range of transmission of both A and C, but A and C do not feel another, and suppose that A is transmitting to B. According to the DCF protocol, if C has a packet to be sent to B, listens to the channel and it senses free, because it can not hear the transmission of A. Then begins to transmit, causing a collision at node B. Many authors are working to solve the hidden terminal problem.

In (Choudhury et al., 2006), the authors focused on purely directional transmission and designed multi-hop MAC (MMAC), a routing-aware protocol that bridges longer distances by both coordinating farther nodes using RTS/CTS exchanges over multiple hops and exploiting the higher gain and lower overall interference achieved by directional communications. In (Gatsis et al., 2010) the authors dealt with optimal cross-layer design for wireless ad hoc networks.

A networking-based approach is carried out in (Park et al., 2005) with MIMA-MAC, an access protocol specifically designed for ad hoc networks with up to two antennas per node. The devised MAC includes a contention-based and a contention-free period, used to set up links among receivers using two antennas to decode data coming from up to two transmitters using one antenna each. The small number of nodes considered and the constraint to use at most one antenna for transmission represent significant limitations.

In (Ramantan et al., 2005) authors proposed a set of integrated MAC, routing, neighbor discovery and signaling protocols for directional ad-hoc networks. In (Chen et al., 2006) authors proposed an access scheme to exploit multi packet reception with CDMA while meeting QOS requirement. Zhang and Lee (2008) toke an information-theoretic approach by defining throughput as the maximum mutual information between a received and a transmitted signal. They analyzed 802.11 multi user detection in one hop scenario.

In (Sundaresan et al., 2004), a centralized controller is able to estimate concurrent resource usage and to schedule links to exploit the benefits of MIMO such as Spatial Multiplexing (SM) and interference suppression refer to (paulraj et al., 2004), along with increased transmit rate. The final objective is a proportional fair scheduling of transmissions, the accounts for bottleneck links, and is achieved by graph coloring. An online algorithm is also designed. This last contribution, although interesting, makes some very strong assumptions on the PHY layer, e.g., that any transmission uses the full channel capacity and that signaling at the MAC level is perfect. Also, in (Hu and Zhang, 2004), some new ideas have suggested.

Unlike wireless ad hoc networks, wireless sensor networks, instead, are tiny objects that face a lot of constraints from the point of view of PHY capabilities, processing and memory resources, and most of all available battery energy. They are often designed for long-time operations, and thus require a careful design that grasps as many performance improvements as possible. In this field, a good cross–layer design could provide the ultimate resource for increasing lifetime without performance loss refer to (Madan et al., 2006).

From the MAC point of view, above works rely on the exchange of signaling messages among separate communications. Unlike IEEE 802.11 standard that uses in ad hoc networks we want the MAC to coordinate transmissions in order to favor parallel communications, while avoiding channel overload. Also, we want to drive the reception of SM signals so that wanted ones are sufficiently protected from interference, using a mechanism to prevent some nodes from transmitting if needed. In order to do this, we let the MAC use the knowledge of ongoing neighboring handshakes to decide whether or not to grant some requested transmissions, so that the interference cancellation capabilities of the MIMO receiver are properly exploited without its being overloaded refer to (Zorzi et al., 2006).

With cross layer design, physical layer at the symbol level and framed MAC layer on top of it lead to decrease in the error and increase in the network throughput. We use MIMO technique to improve MAC in ad hoc networks. MIMO techniques allow exploiting the presence of multiple antennas to improve transmission bit rate through spatial multiplexing or to improve the signal decoding efficiency through diversity reception and interference cancellation. In this study, we provide some framework and results on the reception performance of MIMO link in a multiuser scenario. The results show that the capture capability introduced by MIMO technology is significant and this should be taken into account when designing MAC protocols. In this study, we start the analysis of the performance of the PHY layer in a multiuser context and derive the implication that this

PHY layer would be on the design of higher layer protocols. We continue by designing a MAC layer that makes use of back-and-forth information exchanges with the PHY layer in order to perform multiuser detection. The whole process is driven in order to guarantee a satisfactory throughput and yet protect the wanted signals through active interference detection and cancellation. Results show that our offered protocol provides a large throughput improvement. This is due to the higher number of packets delivered to their final destination.

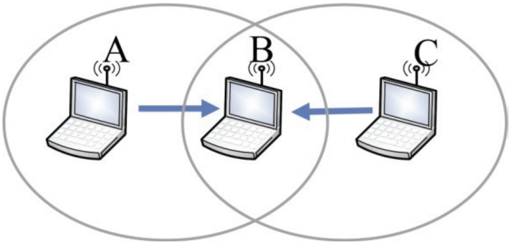


Fig. 1. Hidden terminal problem.

2. Physical layer model

2.1 Problem formulation

As a general line, consider that nodes with multiple antennas are arranged in the network where transmission takes place using packet radio communications. Transmitting nodes build streams of bits and (if necessary) encode them to combat channel impairments. Each user may select the number of antennas to use for transmission that is best suited to its needs. We make the assumption that, in each packet, the number of bits to be sent per transmitting antenna is constant for all users. Each time a multiple transmission has to be decoded, the receiver knows in advance the number of symbols to be simultaneously processed, along with the transmission duration of each of the incoming streams.

At the receiver, multiuser decoding is performed symbol-by-symbol, with a de-correlating layered space-time signal processing technique refer to (Sfar et al., 2003). The receiver is listening to the signals coming from K different users, $l = 1,...,K,$ each using u_l antennas, and thus has to decode a total of $U = \sum_{l=1}^{K} u_l$ incoming symbol per time interval. Let $b = [b_1,...,b_U]^T$ denote the U - length symbol vector where each element is a symbol coming from one of the U transmitting antennas and superscript T denotes transposition. Let S be a matrix with columns containing spreading sequences, one column for each stream. Signals pass through the fading channel that we assume to be frequency non-selective, represented by the channel matrix $H = [h_1,...,h_P]$, where h_p is $1 \times K$ channel coefficient vector between the p-th receiver antenna and all K users. The received signal at antenna p can be written as:

$$r_p = SC_p b + n_p \quad (1)$$

where C_p denotes the complex diagonal channel matrix for the p-th antenna, $diag(h_a)$ .The noise vector n_p is a complex valued zero mean Gaussian random N-vector with a covariance matrix $\sigma^2 I_N$, in which I_N denotes the $N \times N$ identity matrix, where N is length of spreading code for each user.

After the space code match filtering, we obtain the sufficient statistics vector Y_{MU} as:

$$Y_{MU} = \sum_{p=1}^{P} X_p^H r_p = R_{MU} b + n \tag{2}$$

Where $R_{MU} = \sum_{p=1}^{P} X_p^H X_p$ is the $U \times U$ space cross-correlation matrix, with $X_p = SC_p , n = \sum_{p=1}^{P} X_p^H n_p$, H denotes the complex transpose operator. The receiving node may decide to estimate the channel for only a subset of the transmitting users, limiting the stream detection and cancellation to this subset. Thus, the sufficient statistics vector in (2) becomes a sum of two contributions, the first coming from decoded signals, and the other representing a interference term, namely

$$Y_{MU} = \sum_{p=1}^{P} X_p^H (r_p + X_p^{\text{int}} b_{\text{int}}) = R_{MU} b + n + I \tag{3}$$

Where $I = \sum_{p=1}^{P} X_p^H X_p^{\text{int}} b_{\text{int}}$ is the space filtered interfering signal, involving the interference symbol b_{int} and the channel matrix towards interfering users C_{int} which receiver need not know. We report in Fig. 2. a flowchart description of detection algorithm

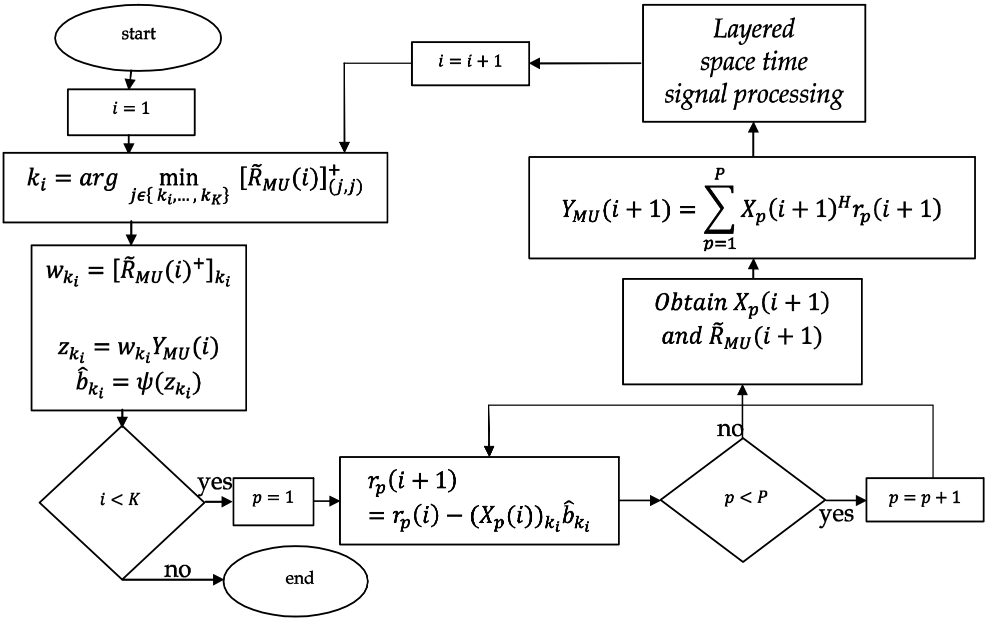

start
$i = 1$
$k_i = arg \min_{j \epsilon \{ k_i,\dots,k_K \}} [\tilde{R}_{MU}(i)]^+_{(j,j)}$
$w_{k_i} = [\tilde{R}_{MU}(i)^+]_{k_i}$
$z_{k_i} = w_{k_i} Y_{MU}(i)$
$\hat{b}_{k_i} = \psi(z_{k_i})$
$i < K$
yes
no
end
$p = 1$
$r_p(i+1) = r_p(i) - (X_p(i))_{k_i} \hat{b}_{k_i}$
$p < P$
yes
no
$p = p + 1$
Obtain $X_p(i+1)$ and $\tilde{R}_{MU}(i+1)$
$Y_{MU}(i+1) = \sum_{p=1}^{P} X_p(i+1)^H r_p(i+1)$
Layered space time signal processing
$i = i + 1$

Fig. 2. Flowchart description of LAyered Space Time Multi User Detection (LAST-MUD) algorithm

The detection algorithm works on a single symbol each time and consists of U iterations. It implies pseudo-inverse calculus over R_{MU} to get R_{MU}^{+} and reordering the received symbols according to their post-detection SNRs (including effects deriving from propagation from different distances). Iteration by iteration, the symbol with the maximum SNR is chosen and isolated from spatially multiplexed signals by linearly weighing the sufficient statistic vector Y_{MU} with a set of coefficients extracted from R_{MU} . The scalar value obtained by this process is fed into a decision block to yield the estimate of the transmitted symbol, and then the sufficient statistics vector Y_{MU} is updated by cancellation of the resulting estimate by striking out the $k_i - th$ column of $X_p(i)$ and the $k_i - th$ row and column of $R_{MU}(i)$. Iterative selection, decoding, and cancellation continue until all U symbols are extracted.

2.2 Physical layer simulations & results

In Fig. 3. we report a graph of bit error rate for all combinations of 4,10,14,16 and 22 users with one antenna each and a receiver with 6 and 8 antennas. The figure contains a performance comparison of BPSK modulation that also processed by taking the real part of R_{MU} during the decoding phase along with a QPSK modulation.

$$Y_{rT} = real(Y_{Tr}) = real(R_{Tr}b + n_{Tr}) = R_{rT}b + n_{rT} \tag{4}$$

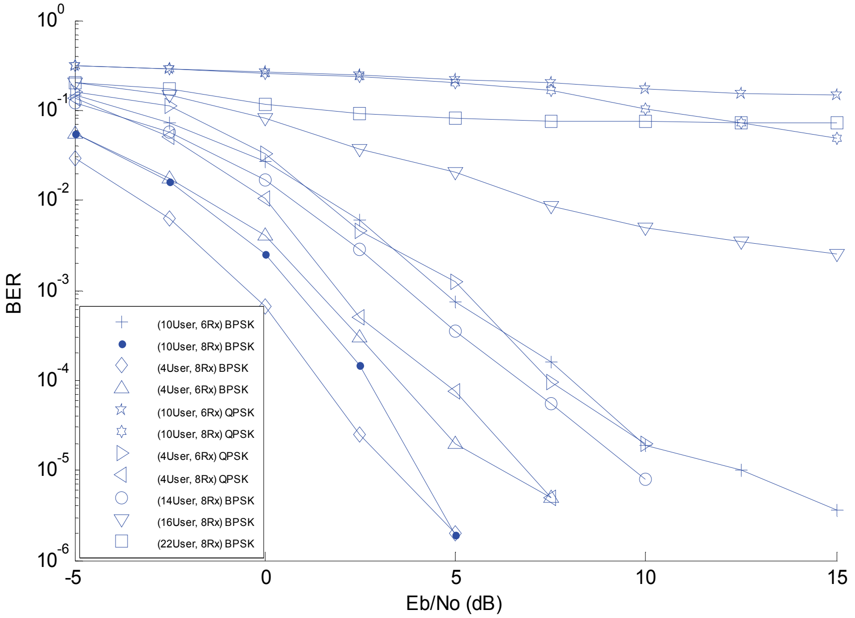


Fig. 3. Comparison of BERs as a function of SNR per receiver

In Fig. 3. and (4), when Y_{MU} is a real value we show our signal by rT and when Y_{MU} is a complex value we show our signal by Tr. As can be inferred from Fig. 3. rT-BPSK gives better results. This is not only a consequence of the constellation simplicity, but also of the fact that it is real. R_{MU}'s imaginary part brings into the detection process a further uncertainty element, namely the noise affecting the imaginary part that may impair the decision over symbols in a way that is unpredictable, due to the nonlinearity of the cancellation process. Fig. 3. also suggests that the loss in spectral efficiency due to the use of BPSK is easily recovered by the higher decoding performance of the system. For instance, with 14 incoming streams the BER for BPSK falls bellow 10^{-5} for 10dB SNR. Note in a more realistic ad hoc network scenario, where the nodes are randomly placed in the area of network, different average received powers would lead to even better performances.

In simulation, model is supposed Hata path loss model which states that a signal which propagates in distance d is multiplied by the constant A, $\frac{A}{d^{\beta}}$, where $A = 0.001$ and $\beta = 4$. Transmission rate and length of signaling packet are 7.5 Mb/sec and 25 byte respectively refer to (Soleimani-Nasab et al., 2009).

Fig. 4. shows signal interference plus noise ratio (SINR) per user for receiving stream by varying the distance, according to the number of antennas are used for transmission, interfered with 8 users are placed at 300m.

Fig. 5. is reproduced probability of errors in the RTS packets which were sent from 1 to 24 transmitters are placed at 100m and 8 interferers are placed at 300m which are sending packets simultaneously.

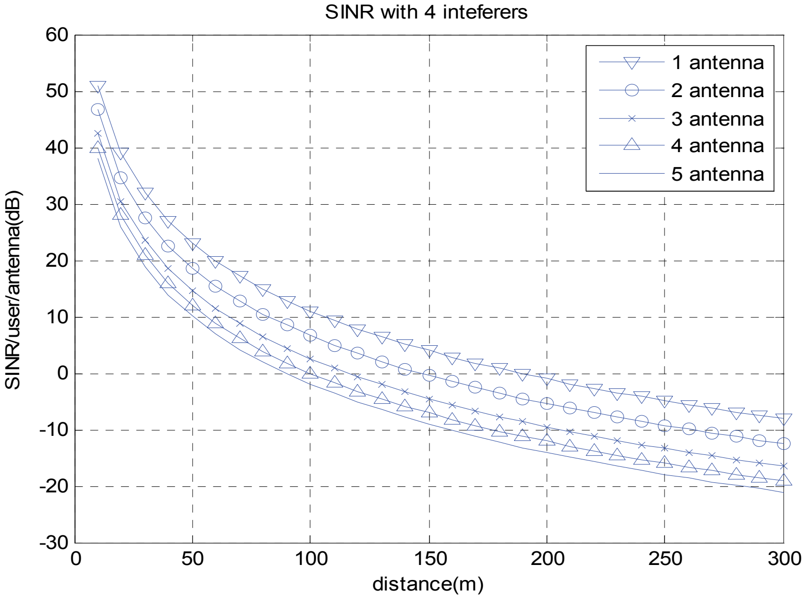


Fig. 4. SINR per user for receiving antenna by varying the distance

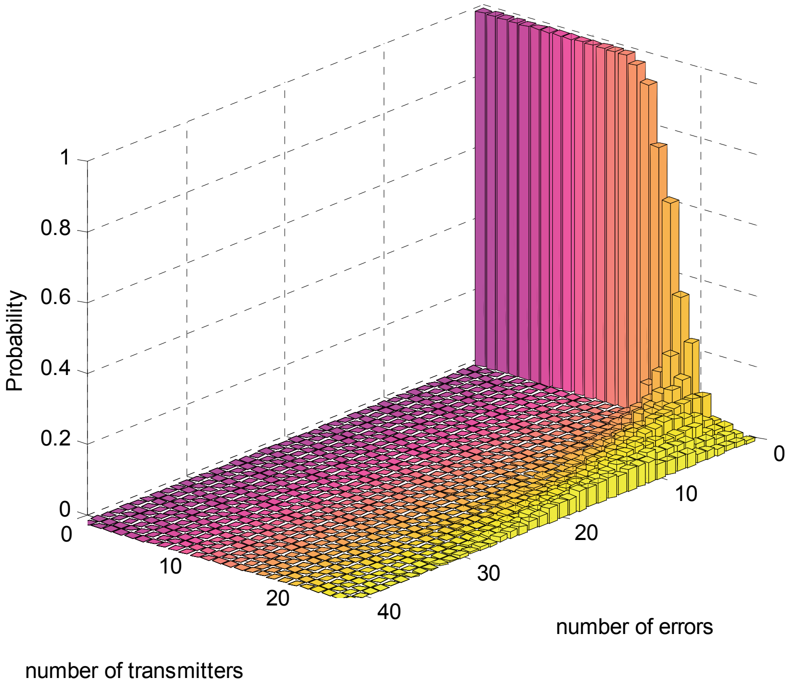


Fig. 5. Probability of errors in the RTS packets sent from 1 to 24 transmitters

Fig. 6. makes distributions of errors for data packets transmitted by 2 users so as each one with 4 antennas by the varying the distance, with 8 interferers which are placed at 300m. It is noted that each user is sending packet with rate equal to 30Mb/sec. The receiver with rate equal to 60Mb/sec is receiving a total of 1000-byte with 8 antennas in the slot. The transmitted power , which associated with any flow that has divided the packet, is 0.25/4 watt. The distances, which they are related to the charts, are: 50,100,110,120,130, 140, 150 and 200m. We observe that increase in distance leads to increase in error.

In Fig. 7. we show the probability of correct decoding RTS packets when they are received from 1 to 24 users simultaneously, corresponding to the probability of having 0 bit wrong, in the same above situation. The results show how to decode correctly a large number of handshake packets, which have the advantage of being short and being transmitted via a single antenna. The transmission via a single antenna gives the advantage of being able to concentrate all of power on a stream and loading a single signal in receiver. In fact, it has a very high probability of decoding up to 15 RTS packets which are simultaneously transmitted from 100m in the conditions of described interference. A station could receive multiple CTSs and decide itself how many antennas to use for spatially-multiplexed data transmission: as a rule of thumb, when more CTSs are heard, more streams will be sent in that part of the network, so more receivers will become overloaded, and fewer streams should be simultaneously sent by each transmitter.

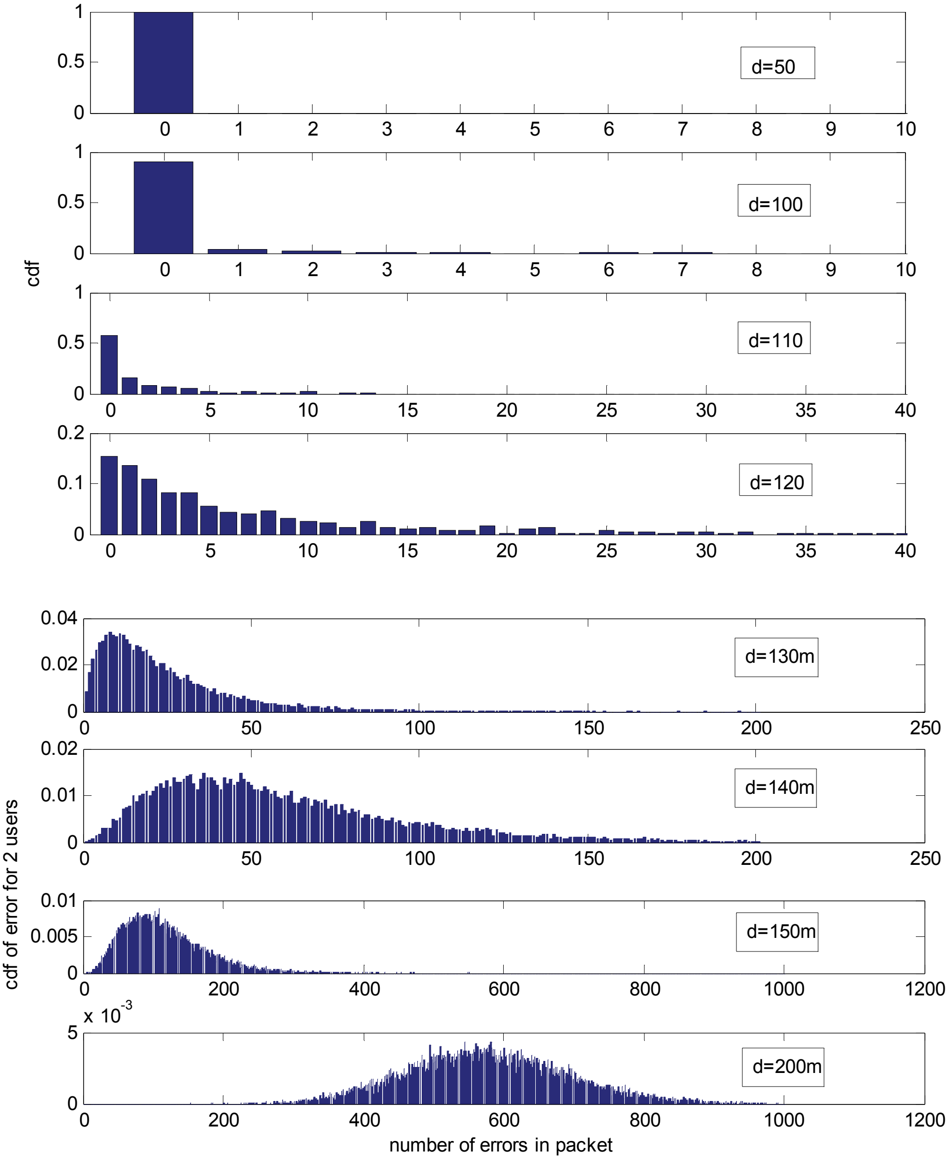


Fig. 6. Distributions of errors for data packets transmitted by 2 users each with 4 antennas by varying the distance

Fig. 8. shows the average error and standard deviation of the number of errors in the RTS packets, which are sent from 1 to 24 transmitters and are placed at 100m with 8 interferers which are placed at 300m are sending packets simultaneously. The standard deviation gives an idea of the magnitude, which distributes the number of errors: the error will increase by growth in the number of packets.

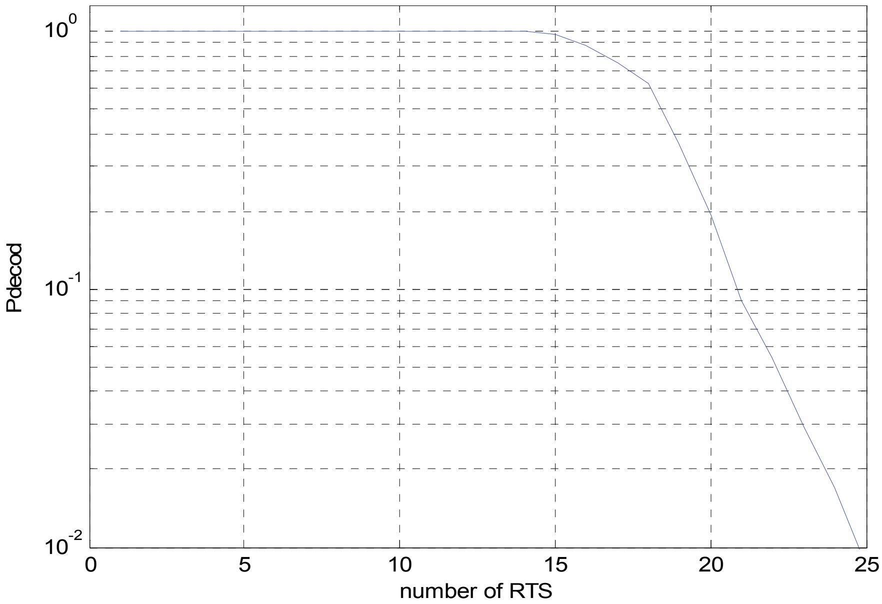


Fig. 7. The probability of correct decoding for 1-24 RTS packets

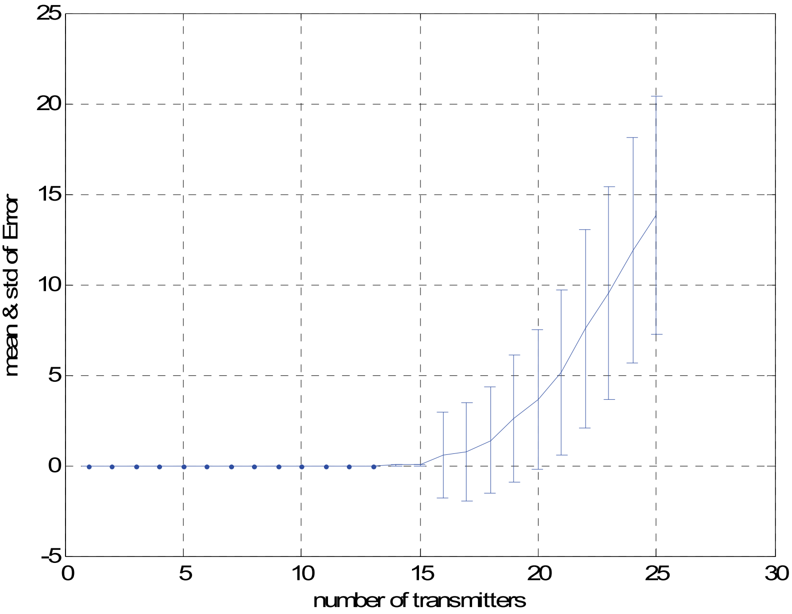


Fig. 8. Mean and standard deviation of bit errors for 1-24 RTS packets

In Fig. 9. the average number of bit errors over two 500-byte packets split into two 4-streams is shown as a function of distance. We suppose each of two users is transmitting four 125-byte streams to a receiver with 8 antennas and 0, 4, 8 interferers are placed in 300m. As we can see, in the low load situation, there are no errors over a large range of distances. The error tolerance is very high up to a distance depending on the number of interfering users, i.e. those whose interference is not eliminated due to unknown channel state. In particular, two tx-rx links may continue to work at distances as far as 100-160m, depending on the interference level.

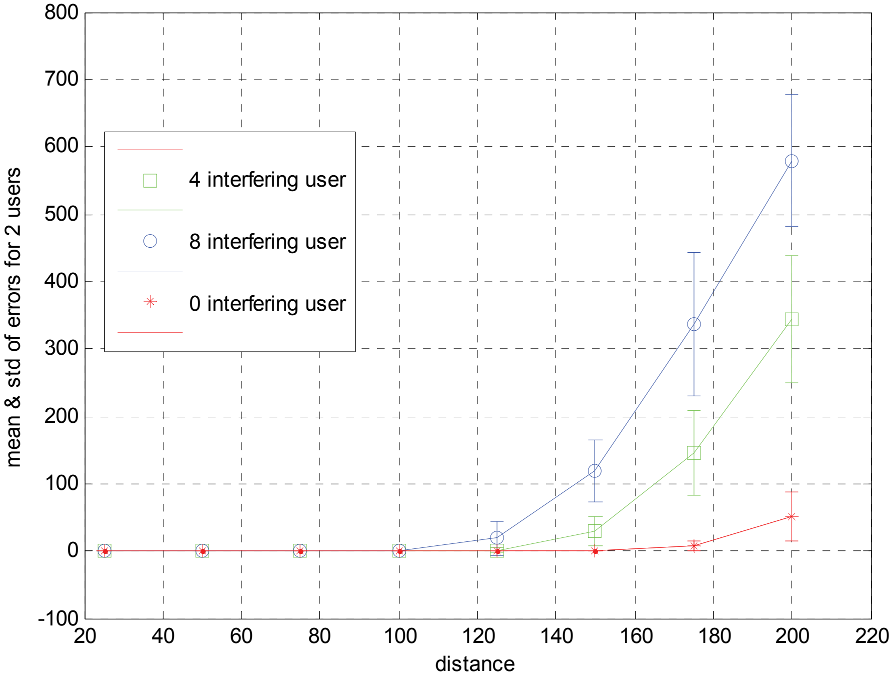


Fig. 9. Average and standard deviation of bit errors over two packets as a function of distance

Fig. 10. lists the CCDF (Complementary Cumulative Distribution Function, which expresses the probability that the number of errors are or exceed in this case, the figure reported in abscissa) the number of errors per transmitted DATA packet by 4 users and each has 4 antennas, with 8 interferers are placed in 300m. We see how the change in distance (20, 50, 70, 80 , 90 , 100m) causes a shift in distribution and the overload which is caused by the large number of streams, makes the performance of decoding very low even at limited distances. The probability that the number of errors is larger than zero for 20m, 50m and 80m is 0.009, 0.07 and 0.35 respectively.

Figs. 11 and 12, show the probability of the decoding of un-coded DATA packets are sent by one user and 2 users with 1, 2, 4, 6, 8 antennas respectively, equivalent to bit-rate of 7.5, 15, 30, 45, 60Mbps, and length of packets are 125, 250, 500, 750, 1000 bytes. For decoding purposes, we use the rate of 1/2 convolutional code is described by the octal coefficients

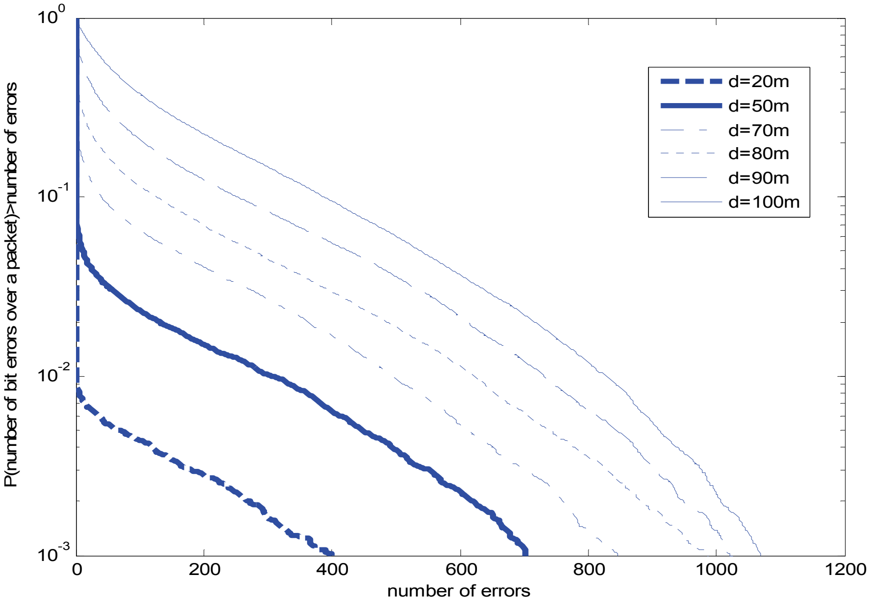


Fig. 10. Complementary cumulative distribution function of the number of errors

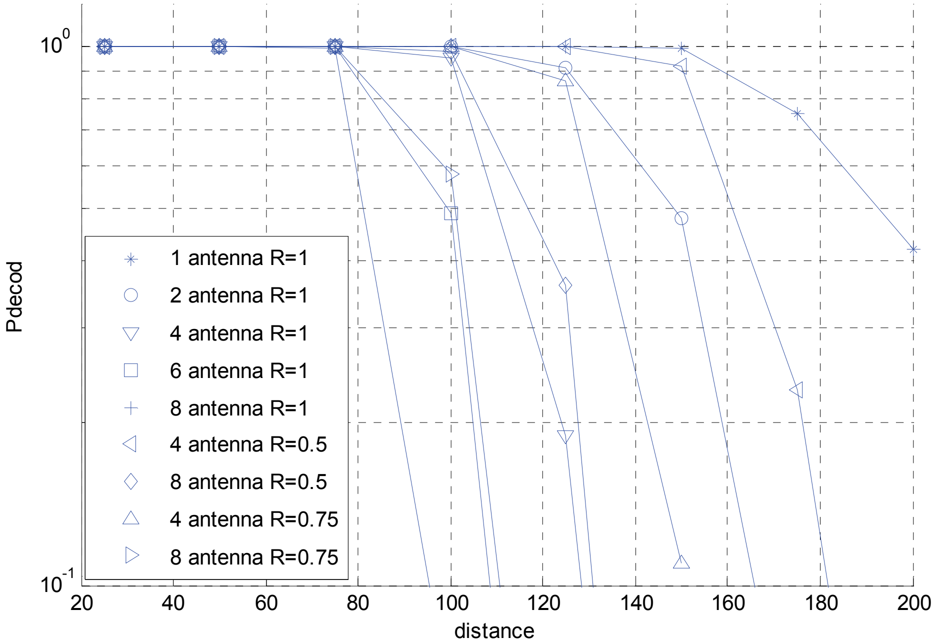


Fig. 11. Probability of correct packet reception for a single data transmission as a function of distance, with and without coding

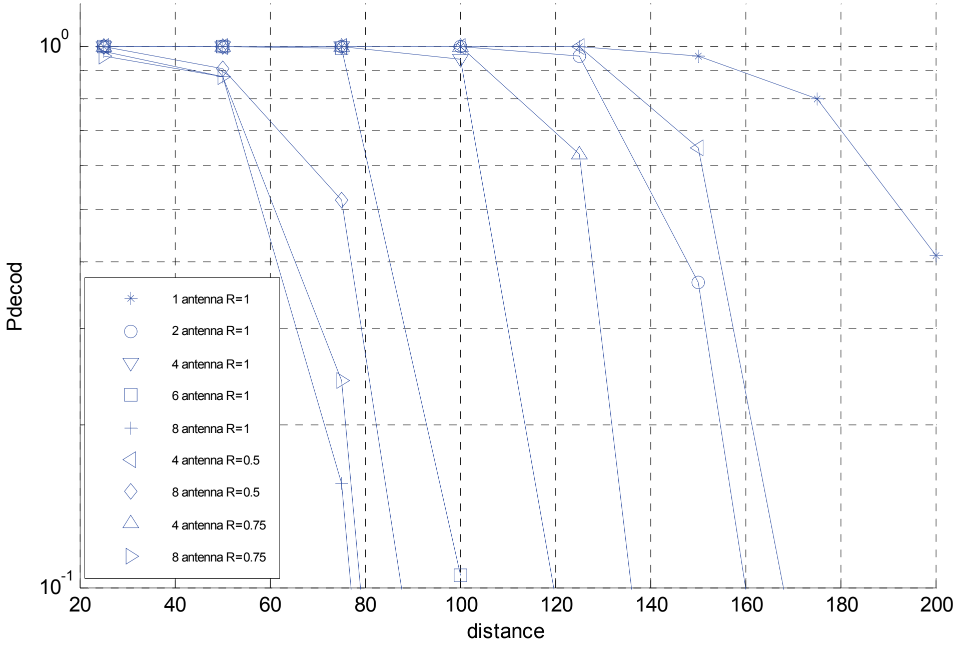


Fig. 12. Probability of correct packet reception for two users data transmission as a function of distance, with and without coding

[133_8, 171_8] as specified, for instance, in the IEEE 802.11 standard. A 3/4 rate version of the code is obtained by puncturing the coded bits. As you can deduce, the distance in which an un-coded transmission becomes excessively error-prone varies as a function of the number of used antennas. The cases with one transmitter and two transmitters show a maximum reachable distance of about 125 and 100 meters respectively (when a single antenna is used) which falls to roughly 75 and 25 meters respectively when the complete set of available antennas are engaged in transmission. Where it has only one transmitter to full the capacity, we have a greater advantage to encode the data flow with further rate instead of reducing the number of antennas (even if we have more power to flow and earn diversity). This means that in low traffic conditions, coding makes it possible to reach farther distances at the price of an increased number of transmitting antennas. A MAC protocol should be able to exploit this favorable condition by forcing users to change adaptively their coding and antenna configuration, according to their own bit rate requirements and taking into account the adjacent nodes' status, which could be extrapolated from signaling packets. From above, if a node requires that at least an average percentage of its data transmission is correctly decoded, it may estimate (through RTS and CTS overhearing) how many its described receiver is loaded, the appropriate curve which corresponds to the required performance and distance to cover is selected from the graphs, hence it is necessary to establish the proper coding and spatial multiplexing scheme that would allow transmission at the desired successful probability, without overloading the receiver.

The information we get from this figure is that the coding cannot help anymore to reduce the interference from other data flow, which we have introduced, when target is to reach farther distance. The system still has a very high performance even for a high number of transmitting antennas, if the distance from the receiver is kept below 25m but when the transmission distance increases, for seeing lesser interference it is better to send un-coded packets over fewer antennas. In addition, we infer that it would be preferable for a MAC protocol to split the longer packets into smaller units and transmit these units sequentially by using fewer antennas, somehow, the system load does not increase. This last result suggests that the use of channel coding (increasing the number of antennas) is not a very good choice. The lower transmit power and the increased receiver load tend to cancel the advantage which is introduced by the coding scheme. A similar problem would be found by using for example space-time codes, refer to (Jafarkhani, 2005; Alamouti, 1998 & Paulraj, 2003). Hence, in the following design, we decide to assume that no stream is actually coded. Our MAC protocol will focus on traffic control among adjacent nodes rather than bit rate and coding scheme adaptation.

Fig. 13. shows the bit-rate transmission versus distance. It is important to note that in the event of 2 users in transmission, the destination node is receiving data at double bit-rate in case of a single user.

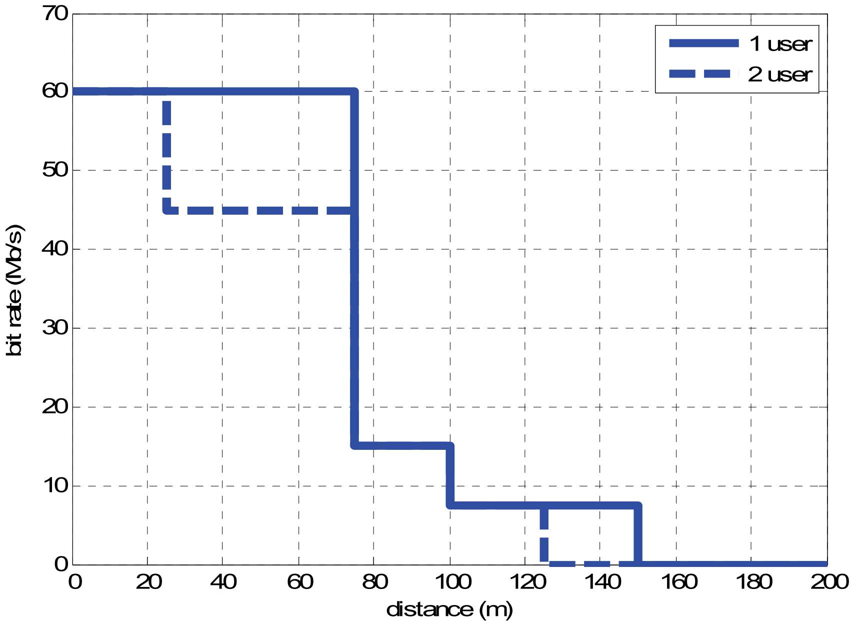


Fig. 13. Bit rate of data packets transmitted by 1 and 2 users by varying the distance

3. Cross layer MAC design for MIMO Ad Hoc networks

3.1 Introduction

The IEEE 802.11 protocol includes a specific mode called ad hoc. This mode operates according to the so-called Distributed Coordination Function (DCF). In turn, DCF defines two different modes, the basic mode (with random access after carrier sensing) and the collision avoidance mode (with four-way handshaking before channel access). We know that preventing collisions would result in loss of data and waste of resource. In this section we want to introduce a good solution for hidden terminal problem in ad hoc network. With some channel knowledge, obtained through training sequences, receiver detects incoming streams separately. Each node have a limited capability of N_s^{max} sequence simultaneously. So the protocol must be aware of the tradeoff existing between the among of wanted data to detect and the interference protection granted to this data. In other word, without enough resources for interference cancellation, the receiver is not aware of interfering nodes nearby and so it can not estimate their channel and cancel them. Indeed, instead of blocking mechanisms, such as 802.11, we want to have simultaneous transmissions. We also want to exploit the spatial demultiplexing capability of MIMO processing.

In our approach, we consider that channel of nodes with a certain distance from receiver can be detected and cancelled and nodes with further distance and low received power can not be cancelled. In Fig. 14. we show the probability of correct receiving a data packet in the presence of interfering traffic versus the distance of the transmitter, for varying number of antenna used by the transmitter. We see that with a 90% minimum success ratio, a transmitter could reach 70m, 90m, 110m, using 8, 4 and 2 antennas respectively. It means that the maximum number of antennas allowed when transmitting to a set of receivers including corresponded neighbor. We use a framed communication structure, with four phases. Theses phases are designed according to standard sequence of messages in a collision avoidance mechanism, and are summarized as follows.

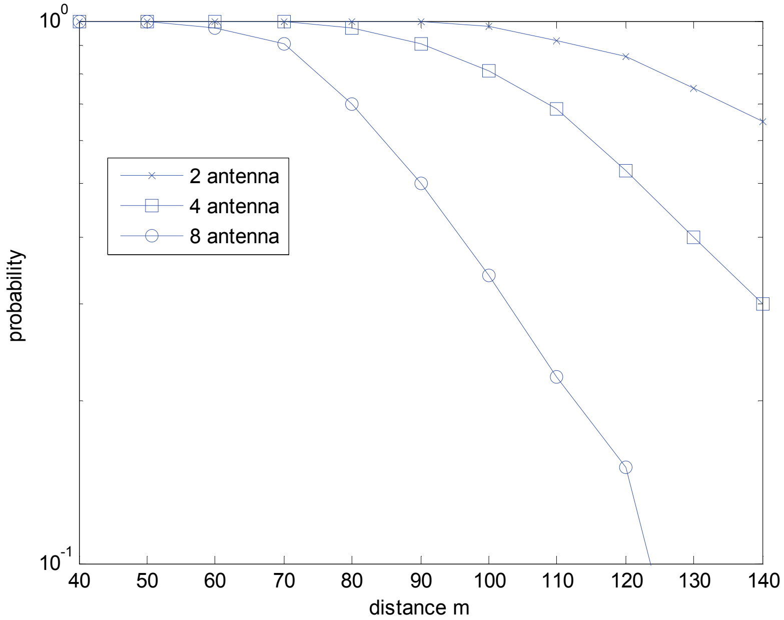


Fig. 14. Probability of correct receiving a data packet by varying the distance and number of transmitter antennas

Sending RTS packet: In this phase, all senders look into their backlog queue, and if it is not empty they compose transmission requests and pack them into a single RTS message. Each packet in the queue is split into multiple streams of fixed length, such that each stream can be transmitted through one antenna. Any RTS has to specify the number of streams to be sent simultaneously, in addition to the intended destination node. How to associate a destination node with a suitable number of transmit antenna depends on the degree of spatial multiplexing sought, as well as the local traffic intensity, thus the queue level of the sender. Any RTS may contain several such requests. Moreover, an RTS is always sent with one antenna and at full power. Each node selects number of antennas according to number of streams of current packet and keeps free other antennas for sending other packets.

Sending CTS packet: During this phase, all nodes that were not transmitters, themselves receive multiple simultaneous RTSs, and apply the reception algorithm of section 2 to separate and decode them. CTSs are also sent out using one antenna and at full power. We use 4 schemes for receiving data and interfering streams to control the number of allowed transmitters and antennas.

Sending DATA packet: All transmitters receive CTSs and, after BLAST detection, they follow CTS indication and send their streams.

Sending ACK packet: After detection, all receivers evaluate which streams have been correctly received and send an ACK back to the transmitters. After the last phase the data handshake exchange is complete, the current frame ends and the next is started.

A random backoff is needed for nodes that do not receive a CTS, as otherwise persistent attempts may lead the system into deadlock. We make use a standard exponential backoff. Accordingly, before transmitting, node wait for a random number of frames, uniformly distributed in the interval $[1, BW(i)]$, where i tracks the current attempts, and $BW(i) = 2^{i-1}W$, with W a fixed backoff window parameter refer to (IEEE 802.11 Standard, 2007).

3.2 RTS and CTS sending schemes

To specify our MAC protocol, we need to introduce a simpler protocol for comparison. The definition of this protocol is necessary, since the approaches described in Section 2 can not be directly compared to our solution, because of either the absence of a specific MAC scheme refer to (chen and Gans, 2005), the optimization of MAC around some fixed PHY parameters such as the number of antenna refer to (Vang and Tureli, 2005), the diverse issue related to different modulation and signaling scheme refer to Hu and Zhang (2004), the attention devoted to achieving full diversity instead of full parallelism refer to (Hu and Zhang, 2004), or the idealized assumptions about a MIMO PHY level and MAC signaling refer to (Sundaresan et al., 2004) This protocol is meant as an example of how a layered networking solution would behave when set up on top of a SM-capable MIMO PHY level. Furthermore, it is directly comparable with our policies, as it can into account the PHY used (unlike (Sundaresan et al., 2004), that focuses on link capacity) and is sufficiently general not to depend on the number of antenna per node (unlike (Vang and Tureli, 2005)). When a node is granted access, it sends an RTS and waits for a CTS. With MIMO transmission, packets are divided in streams, each 125-byte long. To increase bit rate, streams are split in substreams, one per each available antenna and transmitted in parallel through all antenna. If a packet is formed of a number of 125-byte streams and N_a=8, each antenna will send one 125-bit substream per stream. Ack'ed substreams remove from the queue of node and streams with errors are retransmitted. Indeed, Simpler protocol is a CSMA/CA protocol, just using a more powerful MIMO PHY layer.

3.2.1 RTS sending scheme

Consider that the set of neighbors of a given node s be denoted as $\mathcal{V} = \{v_1, v_2, \dots\}$. Let a_{sv_j}be the maximum number of antennas that s can uses when transmitting to any set of nodes that includes v_j. Suppose that node n is current node. At step $i = 1$, a request is created as fellows. The node reads the $k_1 = 1$ packet's destination, d_{k_1}, and the number of unsent streams, p_{k_1}. After that, node compares p_{k_1} with maximum antenna constraint, $a_{nd_{k_1}}$. If $p_{k_1} > a_{nd_{k_1}}$, the streams violate from maximum antenna constraint, hence forbidding any further spatial multiplexing. The request pair $(d_{k_1}, a_{nd_{k_1}})$ is inserted in the RTS packet.

If $p_{k_1} \leq a_{nd_{k_1}}$, the pair (d_{k_1}, p_{k_1}) is inserted in the RTS. Each node keeps indices of all packets selected for transmission in set S_i. The total number of antennas allocated until step i hold in $A(i)$. In the absence of interferes, node d_{k_1}could support $a_{nd_{k_1}} - p_{k_1}$ further antenna. So, the node goes to step 2 and searches its queue , until it finds a packet k_2 that maximum number of destination's antennas match the condition $a_{nd_{k_2}} \geq A(1)$. This means that the d_{k_2}can stand the transmission of the $A(1)$ streams from other node, in addition to its own. The transmitter sets $S_2 = S_1 \cup \{k_2\}$, calculates the number of streams allocated to packet k_2 as $M(2) = min\{\min\{a_{nd_{k_1}}, a_{nd_{k_2}}\} - A(1), p_{k_2}\}$, that not violate the maximum number of antennas constraints $a_{nd_{k_1}} and\ a_{nd_{k_2}}$and $A(1)$ streams have been allocated. Then, it inserts in the RTS packet the pair $(d_{k_2}, M(2))$, and finally updates $A(2) = A(1) + M(2)$. If there is still antenna for transmission without saturating antenna constraints, algorithm goes to next step and so on. In general, at step i, the node searches the queue for a packet k_i with condition $a_{nd_{k_i}} > A(i-1)$. Then $S_i = S_{i-1} \cup \{k_i\}$, $M(i) = min\{min_{j\epsilon S_i} a_{nd_{k_j}} - A(i-1), p_{k_i}\}$, and $A(i) = A(i-1) + M(i)$. The request $(d_{k_i}, M(i))$ is put in the RTS. The algorithm then goes to step $i + 1$ if and only if $min_{j\epsilon S_i} a_{nd_{k_j}} > A(i)$ and a packet such that $a_{nd_{k_{i+1}}} > A(i)$ is found in the queue refer to (Casari et al., 2008). As an example consider Fig. 15. Another example with further request could be found in Fig. 16. In Fig. 17. we show a pseudo code of transmitter protocol.

Rx id	Number of streams	Maximum number of antennas
8	1	4
13	2	8
7	4	2
4	4	4
15	1	8

Request 1		Request 2		Request 3	
Rx id	streams	Rx id	streams	Rx id	streams
8	1	13	2	4	1

Fig. 15. An example of application of RTS sending scheme.

Rx id	Number of streams	Maximum number of antennas
8	1	4
7	2	4
13	1	8
4	1	2
15	4	8

Request 1		Request 2		Request 3		Request 4	
Rx id	streams	Rx id	streams	Rx id	streams	Rx id	streams
8	1	7	2	13	1	15	1

Fig. 16. Another example of application of RTS sending scheme with further request

```
transmitter protocol
// Initialize the step index , the number of allocated antennas $A$ , the set of receivers $S$ and the number of failures $N_{fail}$
$i = 1;\ A(0) = 0;\ S_0 = \phi;\ N_{fail} = 0$
// RTS phase: add users until class constraint are violated
While $min_{j\epsilon S_i}\, a_{nd_j}\ >\ A(i-1)$ do
  // Is there a packet in the queue that complies with the current constraints?
  if ∃ a packet $k_i$ s.t. $a_{nd_{k_i}}\ >\ A(i-1)$ then
    // Add user as receiver
    $S_i = S_{i-1} \cup \{k_i\}$
    // Determine number of streams to send that does not violate any current class constraint
    $M(i)\ = min\{min_{j\epsilon S_i}\, a_{nd_{k_j}}\ -\ A(i-1), p_{k_i}\}$
    $A(i)\ =\ A(i-1)\ +\ M(i)$
    Insert request $(d_{k_i}, M(i))$ in RTS
  end if
end while
Send RTS
// Data phase: check CTS
if one or more CTS received then
  Send data streams according to CTSs
  $N_{fail} = 1$
else
  Backoff for $b$ frames, $b$ uniformly distributed in $[1, W.\,2^{N_{fail}-1}]$
  $N_{fail} =\ N_{fail} + 1$
end if
if ACK received then
  Mark all ACK'ed streams
  Remove from the queue all packets whose streams have been all ACK'ed
end if
```

Fig. 17. Pseudo code of transmitter protocol

3.2.2 CTS sending schemes

In this section we report 4 schemes for receiving data from transmitters. All of these schemes contain two set $\mathcal{W}$ and $\mathcal{U}$. The first set contains all requests directed to the node that names wanted request, the second set all other requests that names unwanted request. We knows that if p_k streams implies to transmitted, the receiver estimates channel of this streams. After that, number of available estimating resources is $N_s^{max} - p_k$. If $N_s^{max} - p_k > 0$ and exist any request in the node queue, process will be continued in the next step and so on.

SNR based receiver protocol: The node grants first highest power request in $\mathcal{W}$ and then considers all other requests in $\mathcal{W} \cup \mathcal{U}$, re-ordered by decreasing received power. In Fig. 18. we report a pseudo code of SNR based receiver protocol. In Fig. 19. an example of application of this protocol is showed.

```
SNR based receiver protocol
//Initialize number of trackable training sequences, $N_s$
$N_s = N_s^{max}$
// CTS phase: apply CTS policy
if one or more RTSs received then
  Create ordered sets $\mathcal{W}$ and $\mathcal{U}$
  Let $I_{\mathcal{W}}$ be the ordered set with the indices of the packets in $\mathcal{W}$
  Let $I_{\mathcal{U}}$ be the ordered set with the indices of the packets in $\mathcal{U}$
  //Grant at least one wanted request
  $i = I_{\mathcal{W}}(1)$
  Read source $s_i$ and number of data streams $p_i$ for the packet with index $i$
  Insert grant $(d_i, p_i)$ in CTS
  $N_s = N_s - p_i$
  $I_{\mathcal{W}} = I_{\mathcal{W}} - \{i\}$
  // Manage other requests in order of decreasing received power
  While $N_s > 0$ & $(I_{\mathcal{W}} \neq \phi$ or $I_{\mathcal{U}} \neq \phi)$ do
    Let $i$ be the request with the greatest power between $I_{\mathcal{W}}(1)$ and $I_{\mathcal{U}}(1)$
    $N = min\ \{p_i, N_s\}$
    $N_s = N_s - N$
    if $i \in I_{\mathcal{W}}$ then
      Insert grant $(d_i, N)$ in the CTS
      $I_{\mathcal{W}} = I_{\mathcal{W}} - \{i\}$
    else
      $I_{\mathcal{U}} = I_{\mathcal{U}} - \{i\}$
    end if
  end while
end if
Send CTS
//Data phase: receive data streams
if Data streams received then
  De-multiplex streams and extract wanted ones
  Send ACK for correctly received streams belonging to requests in $\mathcal{W}$
end if
```

Fig. 18. Pseudo code of SNR based receiver protocol

First wanted based receiver protocol: In this protocol, a node gives priority to wanted transmission. If any estimating resources left , it then begins to consider unwanted requests. In Fig. 20. we report a pseudo code of first wanted based receiver protocol. In Fig. 21. an example of application of this protocol is showed.

Wanted based receiver protocol: In this case, the node grants the requests in $\mathcal{W}$ and does not consider $\mathcal{U}$ at all. In Fig. 22. we report a pseudo code of wanted based receiver protocol. In Fig. 23. an example of application of this protocol is showed.

SNR based receiver protocol without interference cancellation: This scheme operates as SNR based receiver protocol, but does not perform cancellation of interfering requests in $\mathcal{U}$. It means that only powerful interferes could be considered. In Fig. 24. we report a pseudo code of SNR based receiver protocol without interference cancellation. In Fig. 25. an example of application of this protocol is showed.

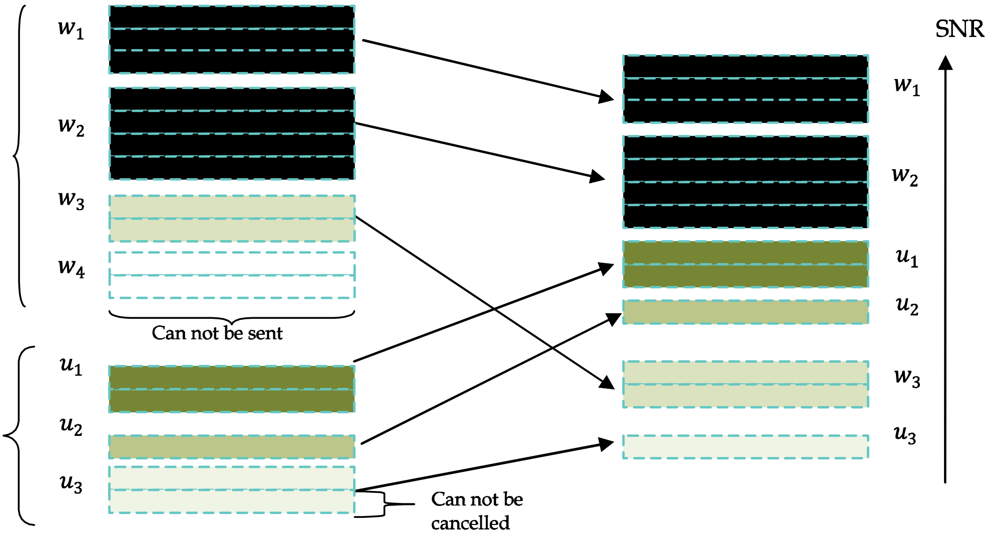


Fig. 19. An example of application of SNR based receiver protocol.

```
First wanted based receiver protocol
//Initialize number of trackable training sequences, $N_s$
$N_s = N_s^{max}$
// CTS phase: apply CTS policy
if one or more RTSs received then
  Create ordered sets $\mathcal{W}$ and $\mathcal{U}$
  Let $I_{\mathcal{W}}$ be the ordered set with the indices of the packets in $\mathcal{W}$
  Let $I_{\mathcal{U}}$ be the ordered set with the indices of the packets in $\mathcal{U}$
  //Grant at least one wanted request
  While $N_s > 0$ & $(I_{\mathcal{W}} \neq \phi)$ do
    $i = I_{\mathcal{W}}(1)$
    $N = min\,\{p_i, N_s\}$
    $N_s = N_s - N$
    Read source $s_i$ and number of data streams $p_i$ for the packet with index $i$
    Insert grant $(d_i, N)$ in the CTS
   $I_{\mathcal{W}} = I_{\mathcal{W}} - \{i\}$
  end while
  While $N_s > 0$ & $(I_{\mathcal{U}} \neq \phi)$ do
    $i = I_{\mathcal{U}}(1)$
    $N = min\,\{p_i, N_s\}$
    $N_s = N_s - N$
   $I_{\mathcal{U}} = I_{\mathcal{U}} - \{i\}$
  end while
end if
Send CTS
//Data phase: receive data streams
if Data streams received then
   De-multiplex streams and extract wanted ones
   Send ACK for correctly received streams belonging to requests in $\mathcal{W}$
end if
```

Fig. 20. Pseudo code of first wanted based receiver protocol

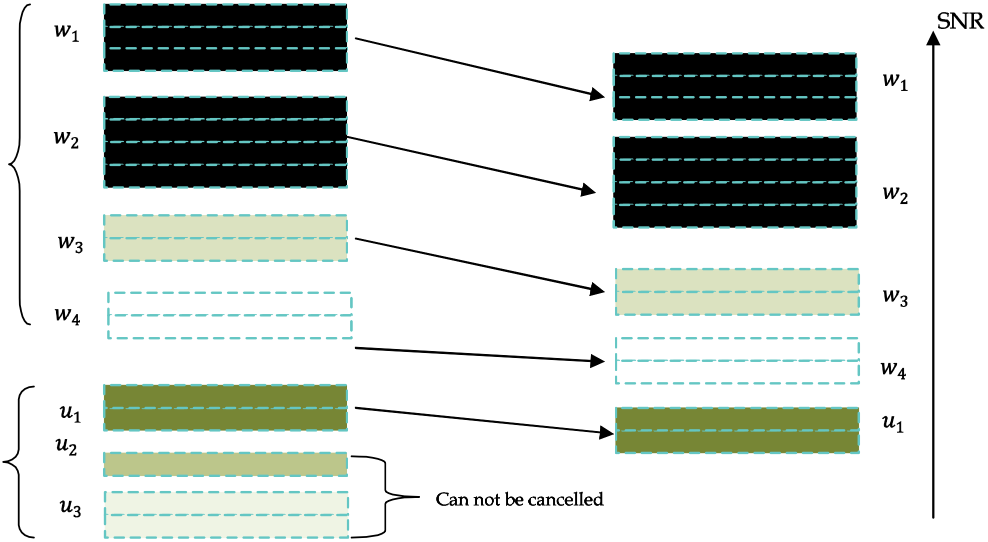


Fig. 21. An example of application of first wanted based receiver protocol.

```
**Wanted based receiver protocol**
//Initialize number of trackable training sequences, $N_s$
$N_s = N_s^{max}$
// CTS phase: apply CTS policy
if one or more RTSs received then
  Create ordered sets $\mathcal{W}$
  Let $I_{\mathcal{W}}$ be the ordered set with the indices of the packets in $\mathcal{W}$
  //Grant at least one wanted request
  While $N_s > 0$ & $(I_{\mathcal{W}} \neq \phi)$ do
     $i = I_{\mathcal{W}}(1)$
     Read source $s_i$ and number of data streams $p_i$ for the packet with index $i$
     $N = min\ \{p_i\ , N_s\}$
     $N_s = N_s - N$
     Insert grant $(d_i, N)$ in the CTS
    $I_{\mathcal{W}} = I_{\mathcal{W}} - \{i\}$
  end while
end if
Send CTS
//Data phase: receive data streams
if Data streams received then
   De-multiplex streams and extract wanted ones
   Send ACK for correctly received streams belonging to requests in $\mathcal{W}$
end if
```

Fig. 22. Pseudo code of wanted based receiver protocol.

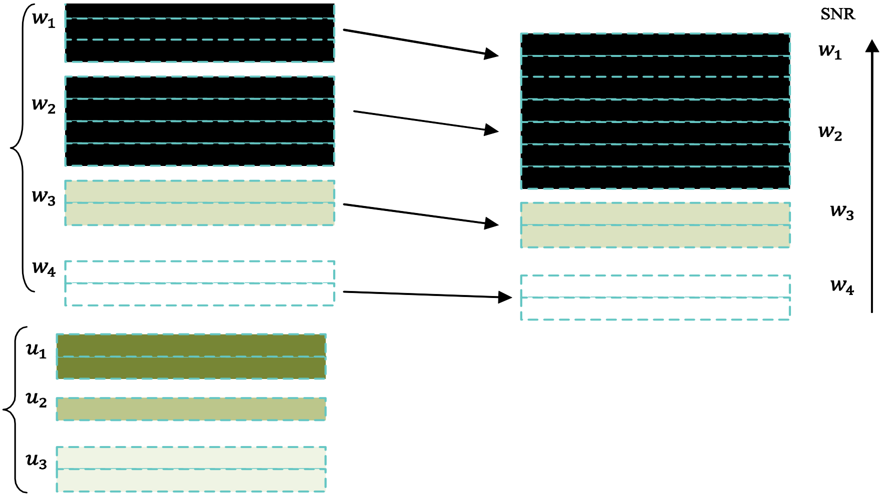


Fig. 23. An example of application of wanted based receiver protocol.

```
SNR based receiver protocol without interference cancellation
//Initialize number of trackable training sequences, $N_s$
$N_s = N_s^{max}$
// CTS phase: apply CTS policy
if one or more RTSs received then
  Create ordered sets $\mathcal{W}$ and $\mathcal{U}$
  Let $I_{\mathcal{W}}$ be the ordered set with the indices of the packets in $\mathcal{W}$
  Let $I_{\mathcal{U}}$ be the ordered set with the indices of the packets in $\mathcal{U}$
  //Grant at least one wanted request
  $i = I_{\mathcal{W}}(1)$
  Read source $s_i$ and number of data streams $p_i$ for the packet with index $i$
  Insert grant $(d_i, p_i)$ in CTS
  $N_s = N_s - p_i$
  $I_{\mathcal{W}} = I_{\mathcal{W}} - \{i\}$
  // Manage other requests in order of decreasing received power
  While $N_s > 0$ & $(I_{\mathcal{W}} \neq \phi$ or $I_{\mathcal{U}} \neq \phi)$ do
    Let $i$ be the request with the greatest power between $I_{\mathcal{W}}(1)$ and $I_{\mathcal{U}}(1)$
    $N = min\{p_i, N_s\}$
    $N_s = N_s - N$
    if $i \in I_{\mathcal{W}}$ then
      Insert grant $(d_i, N)$ in the CTS
      $I_{\mathcal{W}} = I_{\mathcal{W}} - \{i\}$
    end if
  end while
end if
Send CTS
//Data phase: receive data streams
if Data streams received then
  De-multiplex streams and extract wanted ones
  Send ACK for correctly received streams belonging to requests in $\mathcal{W}$
end if
```

Fig. 24. Pseudo code of SNR based without interference cancellation receiver protocol.

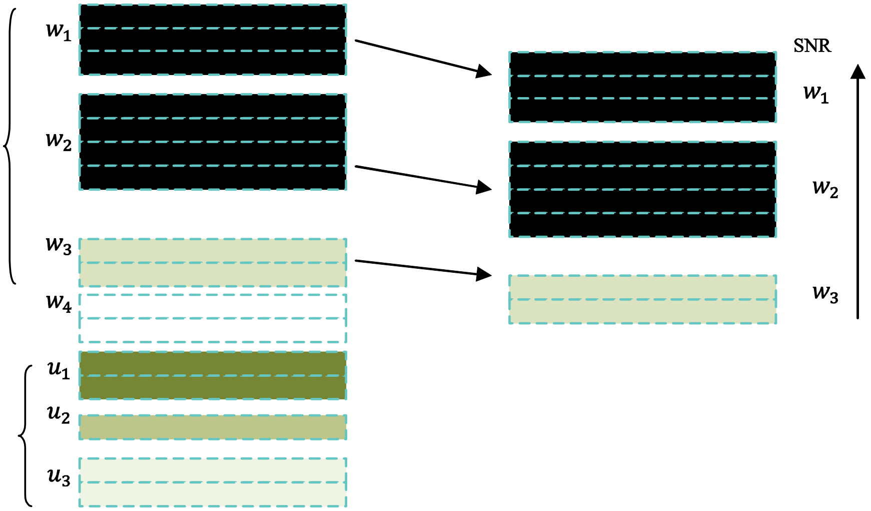


Fig. 25. An example of application of SNR based without interference cancellation receiver protocol.

CTS sending schemes are the only way to reduce data traffic in ad hoc network, since RTS/CTS are not used for channel reservation., but rather as an indication of intention /clearance to transmit, also both RTS and CTS sending schemes favor the creation of multiple point to point links, all potentially making use of SM. This is made possible by inserting multiple requests (grants) in the RTS (CTS), each composed of multiple streams. These schemes can operate on top of any PHY that successively detects multiple signals, cancels their contribution from the received signal. We choose V-BLAST as one such PHY, since it is a good representative and has recently received a lot of attention refer to (Zhang and Lee, 2008).

3. Network simulation setup & results

For evaluating our MAC scheme, we deploy 25 nodes randomly in a square area with 8 antennas each and nearest neighbors 25 m apart. Traffic is generated according to a Poisson process of rate λ packets per second per node. Each generated packet is made of k 125-bytes long streams, with k randomly chosen in the set {1, 2, 3, and 4}. Unsent packets are buffered. Each node has a finite FIFO queue where the packets are stored before being served. We also study the effect of convolutional coding on data packets using the standard 802.11 code refer to (IEEE 802.11 standard, 2007), W and $BW_{\max}$ are 1 and 32 respectively. For our simulation, we used the MATLAB.

Fig. 26. shows the average network throughput defined as a function of the offered traffic λ , defined as the number of correctly detected 125-byte streams per frame for all CTS sending schemes. We see that wanted based receiver protocol has bad performance, because it permits the sending of all requested streams and does not cancel any interferers. First wanted based receiver protocol have better performance than wanted based, because it has a way to cancel highest SNR interfering streams. Indeed, from network load 700, the amount of requested traffic have not enough antenna for cancellation of unwanted signals and lead to decrease in the throughput. In the worst case, one wanted request protected against $N_s^{max}-1$ strongest interferences and lead to best performance of SNR based receiver protocol.

Fig. 27. shows the average queue length as a function of the offered traffic λ for all CTS sending schemes. We see that first wanted based protocol because of lower throughput at network load larger than 800 does not allow sufficient packet sending. Also SNR based protocol have shorter queue length. We observe that other protocol reach to upper bound of delay. SNR based receiver protocol without interference cancellation has bad performance because it hasn't interference cancellation feature. Results show that the SNR based receiver protocol reach to best performance , as it has high throughput and throughput ratio, limited delay and queue length.

4. Conclusions

In this study, we combine MIMO multiuser detection at PHY layer with design of a protocol at MAC layer in a cross layer fashion simultaneously to have a better throughput for mobile ad hoc networks. As we can see in Fig. 26. this approach is able to support up to 12

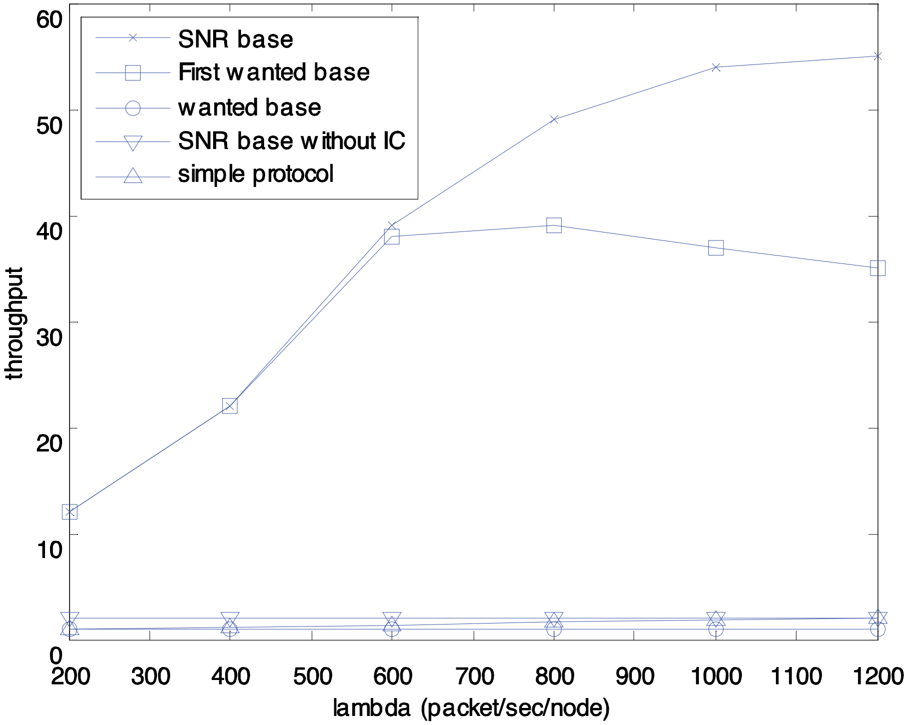


Fig. 26. Network throughput versus network traffic.

successful 125-byte streams per frame on average, which is larger than the maximum number of antennas per node, i.e., 8. This is a very interesting result. It substantiates the need for both a well-designed physical layer and a management protocol, and shows that the number of terminal antenna is a soft limit in MIMO ad hoc networks, if the effective rejection of multiple access interferences is provided. Also in Fig. 27 we show that average queue length is shorter than maximum length of queue, i.e., 120. Future work on this topic may be the extension to routing layer issues. Our scheme can be used on laptops that each one is considered as an ad hoc node and uses 8 antennas with 3 cm distance between the two adjacent antennas.

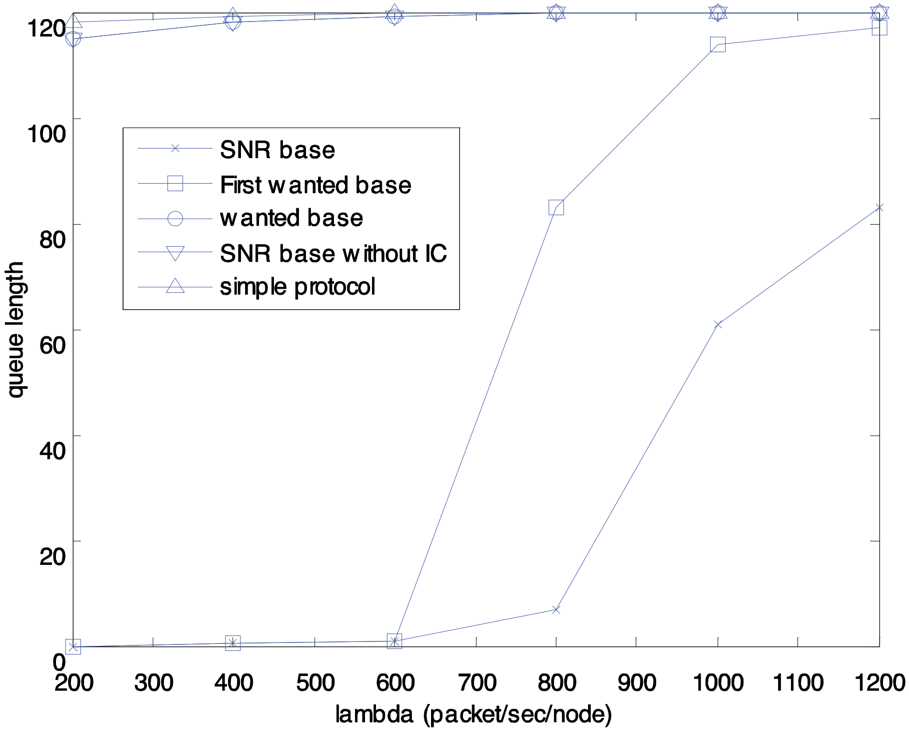


Fig. 27. Queue length versus network traffic.

5. References

Alamouti, S-M. (1998). A simple transmit diversity technique for wireless communication, *IEEE Trans Commun*, 16., pp. 1451-58

Casari, P., Levorato, M. & Zorzi M. (2008). MAC/PHY Cross-Layer Design of MIMO Ad Hoc Networks with Layered Multiuser Detection, *IEEE Transactions on Wireless Communications*, 7. 11., pp. 4596-4607

Chen, B. & Gans, M. (2005). MIMO communication in ad hoc networks, *Proceeding of IEEE VTC Conf*, Sweden, pp. 2434-38

Chen, H., Yu, F., Chan, H. & Leung, V. (2006). A novel multiple access scheme over multi-packet reception channels for wireless multimedia networks, *IEEE Trans Wireless Commun*, 6., pp. 1501-11

Choudhury, R-R., Yang, X., Ramantan, R. & Vaidya, N-H. (2006). On designing MAC protocols for wireless networks using directional antennas, *IEEE Trans Mobile Comput*, 5. pp. 477-491

Gatsis, N., Ribeiro, A. & Giannakis, G. B. (2010) Optimal resource allocation in wireless networks: algorithms and convergence, *IEEE Transactions on Wireless Communications*, submitted.

Hu, M. & Zhang, J. (2004). MIMO ad hoc networks: medium access control, saturation throughput, and optimal hop distance, *Journal of Commun Networks*, pp. 317-30

IEEE Standards Department (2007), ANSI / IEEE Standard 802.11, IEEE Press

Jafarkhani, H. (2005). *Space-Time Coding: Theory and Practice*, Cambridge University Press

Madan, R., Cui, S., Lal, S. & Goldsmith, A. (2006). Cross layer design for lifetime maximization in interference-limited wireless sensor networks, *IEEE Transactions on Wireless Communications*, 5. 11., pp. 3142-3152.

Park, M., Choi, S-H. & Nettles, S-M. (2005). Cross-layer MAC design for wireless networks using MIMO, *Proceeding of IEEE Global Commun Conf*, USA, pp. 938-942

Paularj, A., Nabar, R. & Gore, D. (2003). *Introduction to Space-Time Wireless Communication*, Cambridge University Press

Paulraj, A-J., Gore, D-A., Nabar, R-U. & Boleskei, H. (2004). An overview of MIMO communication: a key to gigabyte wireless, *Proceeding of IEEE*, 92., pp. 198-218

Ramantan, R., Redi, J., Santivanez, C., Viggins, D. & Polit S. (2005). Ad hoc networking with directional antenna: a complete system solution, *IEEE J Selected Areas Commun*, 23., pp. 496-506

Setton, E., Yoo, T., Zhu, X., Goldsmith, A. & Girod, B. (2005). Cross-layer design of adhoc networks for real-time video streaming, *IEEE Trans Wireless Commun*, 12., pp. 59-65

Sfar, S., Murch, R-H. & Letaief, K-B. (2003). Layered space-time multiuser detection over wireless uplink systems, *IEEE Trans Wireless Commun*, 2., pp. 653-668

Soleimani-Nasab, E. & Ardebilipour, M. (2009). Improve efficiency of ad hoc networks with MIMO communication and cross layer MAC design, *Procceeding of IEEE ICACT 2009*, South Korea, pp. 907-912

Sundaresan, K., Sivakumar, R., Ingram, M. & Chang, T-Y. (2004). Medium Access Control in ad-hoc networks with MIMO links: optimization consideration and algorithms, *IEEE Trans. Mobile Comput*, 3., pp. 350-65

Vang, D. & Tureli, U. (2005). Cross layer design for broadband ad hoc networks with MIMO-OFDM, *Proceeding of Signal processing Advances in Wireless Communication*, pp. 630-34

Zhang, J. & Lee, H-N. (2008). Throughput enhancement with a modified 802.11 MAC protocol with multi-user detection support, *Int J Electronics Commun*, 62., pp. 365-73

Zorzi, M., Zeidler, J., Anderson, A., Rao, B., Proakis, J., Swindlehurst, A.L., James, M. & Krishnamurthy, S. (2006). Cross-layer issues in MAC protocol design for MIMO ad hoc networks, *IEEE Wireless Commun Mag*, 13. 4., pp. 62-76

9

Performance Modeling of MAC and Multipath Routing Interactions in Multi-hop Wireless Networks

Xin Wang, J.J. Garcia-Luna-Aceves, Hamid R. Sadjadpour
University of California, Santa Cruz
USA

1. Introduction

For multi-hop wireless networks, the performance experienced by each node is a complex function of the following factors: 1). the signals used at the physical layer; 2). the radio topology of the network; 3). the transmission scheduling established at the MAC layer, 4). the route selection results of the network layer.

The input of any above component is partially decided by the output of the other components, e.g. 1). *transmission scheduling*: radio topology decides whether links interfere with others, and route selection decides which links will be used for transmissions; 2). *route selection*: the radio topology of the network influences the route selection results directly; since routing control packets are transmitted as the data packets at the MAC layer, the transmission scheduling decides how the routing information is propagated throughout the network, etc.

Hence, analyzing the performance of protocol stacks in a wireless network must consider the interactions between the different layers. In fact, a cross-layer perspective to both performance analysis and protocol design brought to attention with recent advances in wireless networks. It is critical for us to treat the entire protocol stack as a single algorithmic construct in order to improve the performance, and in general, it is not meaningful to speak about a MAC or a routing protocol in isolation.

This chapter introduces a modeling framework for the characterization of the performance attained with a MAC protocol working together with different packet forwarding disciplines on top of a realistic physical layer. We also analyzed how different packet forwarding disciplines interact with different channel access schemes to influence the system performance.

2. Related work

A significant amount of work (e.g.,(Gitman, 1975; Tobagi, 1980a;b; Boorstyn et al., 1987; Tobagi & Brazio, 1983; Shepard, 1996; Chhaya & Gupta, 1997; Wang & Garcia-Luna-Aceves, 2002; Wu & Varshney, 1999)) has been reported on the analytical modeling of contention-based MAC protocols. However, there are very few prior works discussing the interaction between MAC and packet forwarding in wireless networks, and most of them are based on the discussion of simulation results focusing on contention-based MAC protocols and single-path routing. Das et al. (Das et al., 2000)(Das et al., 2001) use a simulation model to show that the interplay between routing and MAC protocols affects the performance significantly in the context of AODV and DSR. Royer et al. (Royer et al., 2000) explore the behavior of different unicast routing protocols when run over varying contention-based MAC protocols. They find that table-driven routing protocols behave in much the same way when used with different MAC protocols, while an on-demand routing protocol is more sensitive to the functionality of the MAC protocol, because it requires feedback mechanisms at the MAC layer. Barrett et al. (Barrett et al., 2003) conducted a comprehensive simulation study to characterize the interaction between MAC and routing protocols, node speed, and data rates in mobile ad-hoc networks. They concluded that no combination of MAC and routing protocol was better than other combinations over all mobility models and response variables. Bai et al. (Bai et al., 2003) proposed a framework consisting of various protocol-independent metrics to capture interesting mobility characteristics, including spatial(temporal) dependence and geographic restrictions. They observed that the mobility pattern influences the connectivity graph that in turn influences the protocol performance. In addition, they did a preliminary investigation of the common building blocks of MANET routing protocols, the effect of mobility on these building blocks and how they influence the protocol as a whole. Vadde et al. (Vadde & Syrotiuk, 2004) studied the impact of QoS architectures, routing protocols, and MAC protocols on service delivery in MANETs, using interaction graphs to visualize the two-way interactions between factors. Vadde et al. (Vadde et al., 2006) used statistical design of experiments to study the impact of factors and their interaction on the service delivery in a MANET. They considered the factors of QoS architecture, routing protocols, medium access control protocols, offered loads, and node mobility. Through statistical analysis of the simulation results, they found that the MAC protocol and its interaction with the routing protocol are the most significant factors influencing average delays, and that throughput is not much impacted by the type of routing protocol used. The bulk of the analytical modeling of wireless ad hoc networks has concentrated on the analysis of MAC protocols in fully-connected segments of networks (e.g., satellite networks, cellular networks, or single-hop wireless LANs (WLANs)), because they are simpler to analyze than multihop networks. The majority of this work has followed the formalism and assumptions introduced by Abramson (Abramson, 1970; 1977) for the analysis of the ALOHA protocol, and by Tobagi and Kleinrock (Kleinrock & Tobagi, 1975; Tobagi & Kleinrock, 1975) for the analysis of the carrier sense multiple access (CSMA) protocol. The model typically adopted assumes that all nodes have infinite buffers and transmissions are scheduled according to *independent* Poisson point processes. This implies that packets which were either inhibited from being transmitted or were unsuccessfully transmitted are rescheduled after a "sufficiently long" randomized time out to preserve the Poisson property (i.e., no correlation between new packet arrivals and their rescheduling). Packet lengths are exponentially distributed and are independently generated at each transmission attempt (including retransmissions). In many cases, acknowledgments are assumed to happen instantaneously or, in cases where propagation delay is taken into account, acknowledgment traffic is simply ignored, and periods of collisions are restricted to the propagation time, after which all other nodes are able to perceive any activity in the channel (through the single-hop and perfect-channel assumptions). Regarding the quality of the radio links, they are generally considered error free, and the event of unsuccessful transmission is restricted to packet collisions at the receiver. Examples where such assumptions have been made include (Roberts, 1975) (Kleinrock & Lam, 1975) (Colvin, 1983) (Lo & Mouftah,

1984) (Karn, 1990) (Barghavan et al., 1994), (Fullmer & Garcia-Luna-Aceves, 1995), and (Fullmer & Garcia-Luna-Aceves, 1997).

Other works consider physical-layer aspects more explicitly within the context of single-hop scenarios. Raychauduri (Raychauduri, 1981) analyzed slotted ALOHA with code division; Gronemeyer and Davis (Davis & Gronemeyer, 1980) considered spread-spectrum slotted ALOHA with capture due to time of arrival. Musser and Daigle (Musser & Daigle, 1982) derived the throughput of pure ALOHA with code division. Pursley (Pursley, 1983) studied the throughput of frequency-hopped spread-spectrum communications for packet radio networks. In other cases, the error-free link assumption was relaxed and multipath fading channels where considered while preserving other original assumptions (e.g., Poisson scheduling). This is the case in the works by Arnbak and Blitterswijk, who studied the capacity of slotted ALOHA in Rayleigh-fading channels (Arnbak & Blitterswijk, 1987). More recently, with the advent of the IEEE 802.11 standard for WLANs, its operation. Unfortunately, the vast majority of this effort has considered only single-hop networks under ideal channel conditions (Carvalho & Garcia-Luna-Aceves, 2003), (Bianchi, 2000), (Cali et al., 2000), (Foh & Zukerman, 2002), (Kim & Hou, 2003). A gap still remains on the modeling of multi-hop wireless networks under specific combinations of MAC protocols and packet-forwarding disciplines in a way that the impact of their interactions is taken into account in the performance evaluation of each node.

3. Protocol interactions

In this section we address the interactions between protocol stacks and the classification of different feedback information. The most important modeling factor in the interaction between the MAC layer and the physical layer is the *probability that a frame transmission is successful*, because it is the basis for the scheduling of either transmissions or retransmissions of frames by the MAC protocol. The output of any routing protocol is a subset of nodes in the network, which forms a specific routing path, and this subset varies at different stages of routing protocol. For example, when there is no existing route, the subset includes every nodes that are involved in the route discovery (e.g. initiating route requests, sending route replies or forwarding routing control packets, etc.). After the route is established, the subset consists of the nodes that form a specific routing path or are responsible for the route maintenance. In this paper, we focus on the interaction of routing and MAC protocols that takes place *after* routes have been established. Accordingly, we are mainly interested on the interaction between the MAC protocol and the number of next-hops per destination, which are used according to specific forwarding rules. Our model captures this interplay by means of the *probability that a transmission schedule is collision-free*. We classify the feedback information that flows across layers into two classes: (a) Feedback information that does not depend on the activity of other nodes (e.g., whether a node has data packets to send); and (b) feedback information dependent on the activities of all other nodes (e.g., the successful transmission probability of each frame, or the probability that a transmission schedule is collision-free). The MAC and physical (PHY) layers are coupled with each other tightly at small time scales encompassing just a few packet transmissions. On the other hand, route selections are made based on the end-to-end information between the traffic source and destination; hence, this activity interacts with the MAC layer at large time scales, i.e., hundreds of packet transmissions. Based on the above considerations, we investigate the interaction between protocol layers from small time scales (MAC and PHY) to large time scales (MAC and routing).

4. Model formulation

We assume that each node k transmits frames according to a transmission rate (transmission probability) τ_k, and retransmissions are independent of previous attempts. All nodes along the selected routing path always have packets to send (i.e., the transmission queue of each node is always nonempty). If there are more than one nodes transmit to the same receiver simultaneously, the whole frame transmission is a failure.

4.1 Successful frame reception probability

Let P_k^r denote the received signal power at node r for a signal transmitted by node k. Let V denote the finite set of $|V| = n$ nodes spanning the network under consideration, and $V_r \subseteq V$ the subset of nodes that are in the reception range of node r. $V_r' \subseteq V_r$ is the subset of nodes that are on the selected routing path. V_r incorporates the topology information, while V_r' includes the feedback information from the network layer. At time t, the signal-to-interference-plus-noise density ratio $SINR_i^r(t)$ for a signal transmitted by node i and received at node r is (Tse & Hanly, 1999):

$$\mathrm{SINR}_i^r(t) = \frac{P_i^r(t)}{\sum_{j \in V_r'} \chi_j(t) P_j^r(t) + \sigma_r^2}, \tag{1}$$

where σ_r^2 is the background or thermal noise power at the front end of the receiver r. $\chi_j(t)$ is an on/off indicator,

$$\chi_j(t) = \begin{cases} 1, & \text{if } j \text{ transmits to } r \text{ at time } t, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

$\chi_j(t)$ reflects MAC layer transmission scheduling(contention) results. Let $|V_r'| = n_r$, there are exactly $2^{n_r - 1}$ combinations of active transmitting nodes (*interferers*) in V_r', excluding the transmitter i itself. In what follows, let $\{c_{ik}^r\}_{k=1,\ldots,2^{n_r-1}}$ denote the set of such combinations. Additionally, c_{i0}^r is the combination corresponding to the case when *no* interferers of r transmit. Let $\gamma(c_{i0}^r)$ denote the SINR at node r for a bit transmitted by i when none of r's interferers transmits:

$$\gamma(c_{i0}^r) = \frac{P_i^r L_i}{\sigma_r^2}, \tag{3}$$

where L_i is the spreading gain (or bandwidth expansion factor) of the spread-spectrum system. If K is the length of the frame in bits, and $P_b(\gamma)$ is the bit-error probability for a certain SINR level γ, then the probability of successful frame reception ($f(c_{i0}^r)$) when only the sender transmits in the neighborhood of an intended receiver is:

$$f(c_{i0}^r) = \{1 - P_b[\gamma(c_{i0}^r)]\}^K. \tag{4}$$

The probability q that a transmitted packet does not collide equals the probability that no neighbor of the receiver transmits and the packet is received correctly (we do not consider the partial overlapping case in this paper). The probability that no neighbor transmits equals

$$P\{\text{no neighbor transmits}\} = \prod_{j \in V_r'} (1 - \tau_j) \tag{5}$$

Hence, using conditional probability, q can be expressed as

$$q = f(c_{i0}^r) \prod_{j \in V_r'} (1 - \tau_j) \tag{6}$$

We analyze the performance of the MAC layer following the approach introduced by Carvalho et al. (Carvalho & Garcia-Luna-Aceves, 2004) and Bianchi's model (Bianchi, 2000). The MAC protocols we seek to model adjust their behavior dynamically according to the feedback information of the PHY and network layers to maximize the number of successful transmissions. Accordingly, we approximate the operation of the MAC protocols by assuming that these protocols in steady-state can be represented by a time-invariant function $h_i(\cdot)$ relating the successful transmission probability q_i with the steady-state scheduling rate τ_i,

$$\tau_i = h_i(q_i), i \in V, \tag{7}$$

where the subscript i in the mapping function $h_i(\cdot)$ denotes a node-specific instantiation of the MAC protocol in use. Let $\mathcal{C}_i^r$ denote the random variable that indicates the occurrence of a specific combination c_{ik}^r of interferers. The probability that the set of active interferers is c_{ik}^r, i.e., $P\{\mathcal{C}_i^r = c_{ik}^r\}$ is a function of the MAC-dependent transmission probabilities τ_i,

$$P\{\mathcal{C}_i^r = c_{ik}^r\} = \prod_{m \in \overline{c_{ik}^r}} (1 - \tau_m) \prod_{n \in c_{ik}^r} \tau_n, \tag{8}$$

where $\overline{c_{ik}^r}$ denotes the complement set of c_{ik}^r, $V_r' - \{c_{ik}^r\}$. The probability q_i *that a frame transmitted by i is successfully received* can be obtained as follows by considering the set $\{c_{ik}^r\}_{k=1,\ldots,2^{n_r-1}}$ of all possible combinations of active nodes in V_r':

$$\begin{aligned} q_i &= P\{\text{ successful frame reception }\} \\ &= \textstyle\sum_k P\{\text{ successful frame reception}, \mathcal{C}_i^r = c_{ik}^r\} \\ &= \textstyle\sum_k P\{\text{ succ. frame reception} \,|\, \mathcal{C}_i^r = c_{ik}^r\} P\{\mathcal{C}_i^r = c_{ik}^r\} \\ &= \textstyle\sum_k f(c_{ik}^r) P\{\mathcal{C}_i^r = c_{ik}^r\}, \end{aligned} \tag{9}$$

Recall that c_{i0}^r denotes the combinations corresponding to the case when *no* interferer of receiver r transmit, i.e., $c_{i0}^r = \{\emptyset\}$, meaning that $\overline{c_{i0}^r} = V_r'$, then we can approximate q_i as follows:

$$q_i \quad \approx \quad f(c_{i0}^r) P\{\mathcal{C}_i^r = c_{i0}^r\} \tag{10}$$

From Eq. (8),

$$q_i = f(c_{i0}^r) \prod_{j \in V_r'} (1 - \tau_j). \tag{11}$$

After the linear approximation using the Taylor series expansion (justified in (Carvalho & Garcia-Luna-Aceves, 2004)), we have

$$\tau_i = h_i(q_i) \approx a q_i, \text{ where } a = h_i'(0), \tag{12}$$

From Eq. (12),

$$q_i = f(c_{i0}^r) \prod_{j \in V_r'} (1 - a q_j). \tag{13}$$

If we assume $a << 1$, and because $0 \leq q_i \leq 1$, we can approximate the previous products as follows:

$$q_i \quad \approx \quad f(c_{i0}^r) \left(1 - a \sum_{j \in V_r'} q_j \right) \tag{14}$$

From Eq. (12) and Eq. (14), we can obtain the functional form $h_i(q_i)$ by which the MAC layer relates the steady-state transmission probability τ_i with the successful transmission probability q_i.

4.2 End-to-end throughput

Given that all nodes along an active path are assumed to be saturated, the average MAC layer one-hop throughput for any node i carrying traffic is

$$S_i = \frac{E\{\text{Data Payload}\}}{\overline{T_i}}. \tag{15}$$

where $\overline{T_i}$ is the average service time of node i. We note that since $\overline{T_i}$ varies across different nodes due to the topology information and traffic distributions, S_i is per-node throughput. We denote the end-to-end throughput as

$$S_E = \min_{k=1}^{h_j} \{S_1, S_2, \ldots, S_k, \ldots, S_{h_j}\} \tag{16}$$

where h_j is the hop length of path j, S_k is the average one-hop throughput of hop k, defined in Eq. (15).

4.3 Interaction with number and type of paths

Multipath routing protocols adapt different constraints for the establishment of next hops to destinations. The existing multipath routing protocols can be classified according to the type of paths they use:

1. Node-disjoint paths (Ye et al., 2003), which are paths to a destination in which a node appears in at most one path.
2. Link-disjoint paths (Marina & Das, 2001) (Nasipuri et al., 2001), which are paths to a destination in which the same pair of nodes defining a link can appear in at most one path.
3. Minimum-cost paths (Lee & Gerla, 2001), which are paths to a destination that have the minimum cost amongst all available paths. These paths need not be link or node disjoint.

Because there is no standard definition of *minimum-cost* for multipath routing protocols, we focus on the study of node-disjoint routing and link-disjoint routing. We use Dijkstra's shortest path algorithm to form the multipath routing set. We choose hop-count as the routing distance metric. The first selected path is the one with the shortest distance between the source and the destination. A path will be added to the selected routing set if: 1). it has the shortest distance among all the unselected paths; 2). it satisfies the node-disjoint or link-disjoint constraint with previous selected paths. If there are more than one path with the same distance, we will select the path with the smaller IP address. We continue this process until no more paths can be added. In our modeling framework, the routing information is fed into V_r', c_{ik}^r and S_E^f, separately. We extend the definition of *interference matrix* (Carvalho & Garcia-Luna-Aceves, 2004) to take into account the effect of routing factors. As indicated in Eq. 11, in order to calculate q_i, we need to know the set of interferers for each transmitter-receiver pair. We select a node as a potential interferer if and only if: (a) The received interference signal power at the receiver is above the carrier sensing threshold, as indicated in (Carvalho & Garcia-Luna-Aceves, 2004); and (b) it is on at least one of the routing paths.

4.4 Interaction with packet forwarding disciplines

Once routing paths are formed, nodes use different forwarding rules to select their successors. Opportunistic routing protocols (Biswas & Morris, 2005) (Chachulski et al., 2007) have been proposed to exploit the benefits of cooperative diversity and path diversity techniques. To simplify our analysis, we classify the different routing forwarding rules into the following types:

1. Single-copy forwarding: A node selects its neighbor with the smallest distance to the destination as the successor, and the smallest address is chosen if there are multiple successors with the same distance.
2. Multiple-copy forwarding: A node selects all successors for forwarding to a destination.
3. P-persistent opportunistic forwarding: A node selects a given successor to forward a packet towards a destination with a probability p_f.

As in Section 4.3, the routing forwarding rule impacts the calculation of $SINR_i^r(t)$, c_{ik}^r and q_i, which influences the conditional probability of successful frame reception ($f(c_{i0}^r)$) and the mapping function $h_i(.)$.

5. Modeling contention-based MAC: 802.11 DCF

In this Section, we extend the prior model proposed by Carvalho et al. and Bianchi's model to study the interactions between 802.11 DCF and different packet forwarding methods. Given the backoff time characterization in 802.11 DCF, the *average service time* is $\overline{T} = \overline{T}_B + \overline{T}_S$, where $\overline{T}_B$ is the average backoff time, $\overline{T}_S$ is the average time to successfully transmit a packet at the end of the backoff operation. In order to obtain $\overline{T}_B$, $\overline{T}_S$, we first need to calculate the probability that a transmission is successful (p_s^i), the probability that the channel is idle (p_i^i), and the probability that a collision occurs (p_c^i). The transmission probability τ_i of each node i is

$$\tau_i = \frac{2[1-2(1-q_i)]}{[1-2(1-q_i)](W_{\min}+1)+(1-q_i)W_{\min}(1-(1-q_i)^m)} \tag{17}$$

where $W_{\min}$ is the minimum contention window size specified for the backoff operation, m is the standard-defined maximum power used to set up the maximum contention window size, i.e., $W_{\max} = 2^m W_{\min}$. Eq. (17) gives us the functional form $h_i(q_i)$ by which the MAC layer relates the steady-state transmission probability τ_i with the successful transmission probability q_i. Then we could derive a first order approximation for it using Taylor series expansion and express τ_i in terms of q_i

$$\tau_i(q_i) = \frac{2W_{\min}}{(W_{\min}+1)^2} q_i, \tag{18}$$

when we consider all nodes in the topology, can be rewritten in the matrix notation $\tau = a\mathbf{q}$, where $\boldsymbol{\tau} = [\tau_1\ \tau_2\ \dots\ \tau_n]^T$, $a = 2W_{\min}/(W_{\min}+1)^2$, and $\mathbf{q} = [q_1\ q_2\ \dots\ q_n]^T$. The probability that there exists some node from V_r' transmitting a frame while node i is in backoff is

$$p_{tr}^i = 1 - \prod_{j \in V_r'} (1-\tau_j) \tag{19}$$

The probability p_{suc}^i that a transmission is successful is the probability that some node in V_r' transmits successfully, conditioned on the fact that at least one node in S_i attempted to transmits, i.e.,

$$\begin{aligned} p^i_{suc} &= \frac{\sum_{k\in S_i} P\{k \text{ succeed} \,|\, k \text{ transmits}\} P\{k \text{ transmits}\}}{p^i_{tr}} \\ &= \frac{\sum_{k\in S_i} q_k \tau_k}{p^i_{tr}} \end{aligned} \tag{20}$$

Then, according to Bianchi's model, the probability that a transmission is successful is $p^i_s = p^i_{tr} p^i_{suc}$; the probability that the channel is idle is $p^i_i = 1 - p^i_{tr}$, and the probability that a collision occurs is $p^i_c = p^i_{tr}(1 - p^i_{suc})$. We can further derive $\overline{T}_B$ and $\overline{T}_S$ using p^i_s, p^i_i and p^i_c.

6. Modeling schedule-based MAC: NAMA

We choose NAMA as an example of schedule-based MAC schemes, because it completely eliminates the communication overheads of building the dynamic channel access schedule, except for collecting two-hop neighbor information, which is minimal compared with the task of collecting complete network topology information. In NAMA, a hash function is implemented at each node. The hash function takes a distinctive string of a node as input, and derives a random priority for each neighbor within two hops. The distinctive input string is the concatenation of the corresponding node identifier (collected through periodical HELLO messages) and the current time slot number such that the priority changes in different time slot. The channel access eligibility of each node is then determined by the node comparing its own priority with those of its two-hop neighbors. If a node has the highest priority, the node can access the channel within the corresponding time slot, while its two-hop neighbors are forbidden from channel access because they have lower priorities than the node. In order to find the correlation between the steady-state MAC layer scheduling rate (τ_i) and the successful transmission probability q_i, we first define the probability that the transmission schedule for node i is collision-free (ϕ_i) as follows:

$$\phi_i = P_{\{no\_conflicts|success\_info\}} P_{success\_info} \tag{21}$$

where $P_{success\_info}$ is the probability that the topology information exchange is successful in i's two-hop range. $P_{\{no\_conflicts|success\_info\}}$ is the conditional probability of conflict-free scheduling given the correct neighbor information. For simplicity, We assume that the unsuccessful information exchange leads to transmission collisions. Then

$$\tau_i = \phi_i q_i \tag{22}$$

The time frame of NAMA can be further divided into a signal section and a data section. We denote the length of a time frame as

$$T_f = N_{signal} t_{signal} + N_{data} t_{data}, \tag{23}$$

where t_{signal}, t_{data} are the signal and data slot length; N_{signal}, N_{data} are the number of signal and data slots, respectively.
Then according to Equation 6,

$$P_{success\_info} = f(c^r_{i0}) P\{\text{no neighbor transmits}\} \tag{24}$$

In NAMA, each node randomly picks up a signal slot in the signal section to exchange topology information.

$$P\{\text{no neighbor transmits}\} = (1 - \frac{1}{N_{signal}})^{N_2^i - 1} \tag{25}$$

where N_2^i is the number of neighbors within two hops of i. The conditional probability of node i winning the node election given the correct topology information is:

$$p_s^i = \frac{1}{N_2^i} \tag{26}$$

Because NAMA uses the node identifier and the current time slot number as input to derive a random priority for every neighbor, which is unique within two hops, it eliminates the conflict scheduling given the correct topology information.

$$P_{\{no\_conflicts|success\_info\}} = 1 \tag{27}$$

$$\phi_i = p_s^i P_{success\_info} \tag{28}$$

From Eq. (22) (24) (25) (28), we can obtain the correlation between τ_i and q_i. Given that the average number of times node i could transmit successfully in one time frame is $\lceil \tau_i N_{data} \rceil$, the average service time is

$$\overline{T} = \frac{T_f}{\lceil \tau_i N_{data} \rceil} \tag{29}$$

7. Model validation

802.11 DCF MAC		NAMA MAC		PHY	
$W_{\min}$	15	t_{signal} (μs)	142	Transmission rate (Mbps)	54
$W_{\max}$	1023	N_{signal}	500	Transmission Power (dBm)	16
RTS (bytes)	30	t_{data} (μs)	362.2	Sensitivity of PHY (dBm)	-69
CTS (bytes)	24	N_{data}	1000	Path loss factor (α)	4
ACK (bytes)	24			Transmission range (m)	79.58
MAC Header (bytes)	34			Temperature (Kelvin)	290
Slot Time (μsec)	9			Noise Factor	10
SIFS (μsec)	16				

Table 1. Simulation Parameters

7.1 Simulation settings

We compare the numerical results with the simulation results obtained from Qualnet (Qualnet Simulator, n.d.). The detailed simulation settings can be found in Table I. The packet length used is 1500 bytes. The duration of the simulation is 100 seconds. For the system throughput results, the simulations are repeated with ten different seeds to average the results for each scenario. We validate the numerical results against simulation experiments under two scenarios. The first scenario consists of 50 nodes distributed randomly across a 500 × 500 square meters area. The second scenario consists of 100 nodes distributed across a 800 × 800 square-meter area. The only constraint for the topology generation is that the network needs to be connected. For each topology, we set up multiple multi-hop CBR flows and vary the number of CBR flows to investigate the influence of packet forwarding methods.

7.2 Interaction between multipath routing and MAC

We first examine the interaction of multipath routing formation and different MAC protocols.

7.2.1 802.11 DCF

50 nodes	Node-disjoint (analytical) (Mb/s)	Node-disjoint (simulation) (Mb/s)	Link-disjoint (analytical) (Mb/s)	Link-disjoint (simulation) (Mb/s)
10 flows	32.12	28.24	32.55	33.17
20 flows	29.97	28.13	32.65	30.26
30 flows	25.19	23.37	29.99	27.45
100 nodes	Node-disjoint (analytical)	Node-disjoint (simulation)	Link-disjoint (analytical)	Link-disjoint (simulation)
20 flows	64.01	59.74	81.99	79.23
30 flows	65.21	61.21	77.04	81.49
40 flows	68.43	64.35	82.07	86.34

Table 2. 802.11 DCF system throughput with different multipath packet forwarding

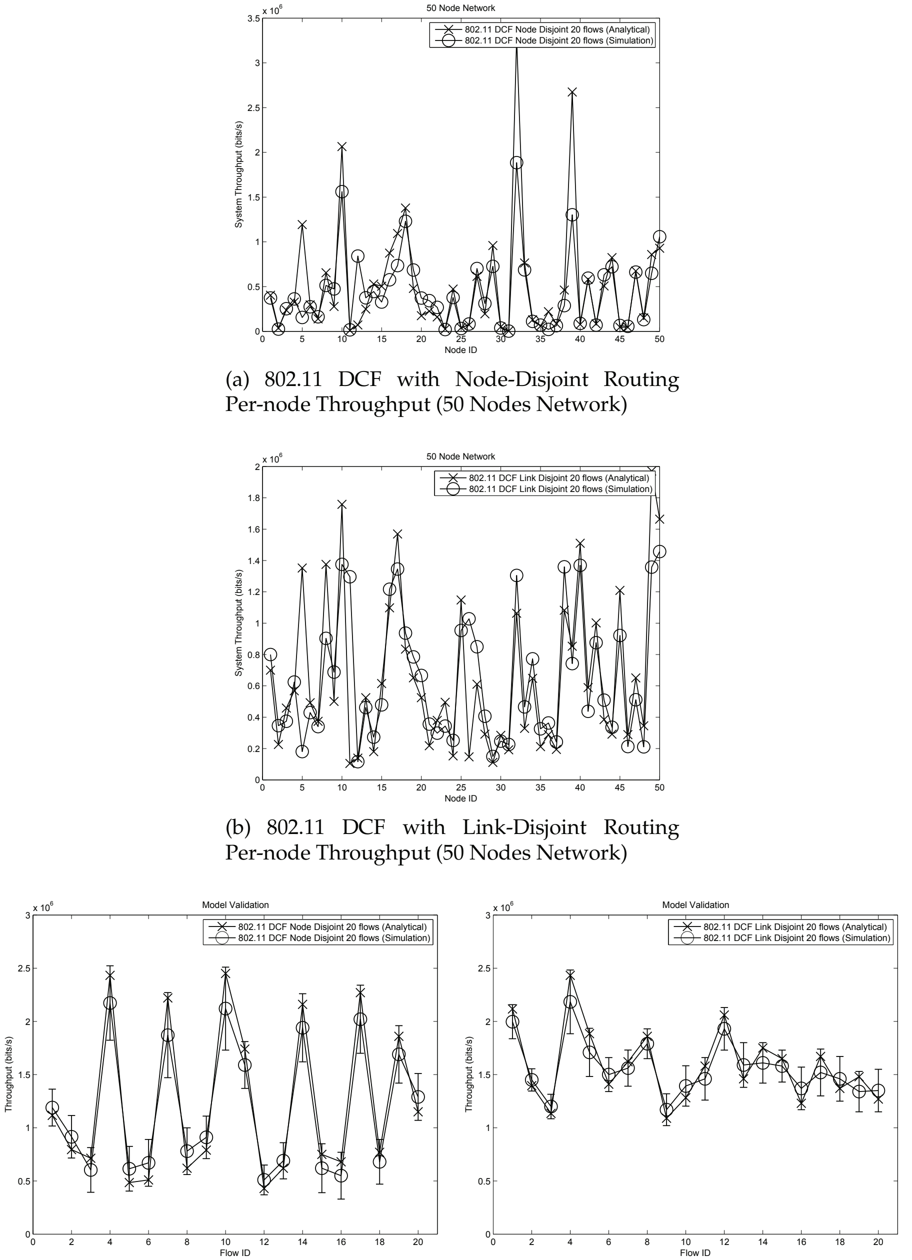
To demonstrate the model accuracy and provide some insights on system performance difference, we first examine the per-node throughput of 802.11 DCF, as Fig. 1 shows. Comparing Fig. 1(a) and Fig. 1(d), we observe that link-disjoint routing balances the traffic more evenly across different nodes. In other words, it is relatively easier to form congestion (bottlenecks) using node-disjoint routing. Because link-disjoint routing has a better spatial reuse throughout the network, it helps to form a better transmission scheduling at the MAC layer. This effect is amplified by a contention-based MAC. When we revisit the analytical model procedure shown in Eq. (1)-(21), the larger the contention neighbor set V_r', C_i^r, the lower the probability that a frame the transmission is successful, the lower the probability that a transmission schedule is collision free. The network-level congestions introduced by the routing protocols will introduce more contentions at the MAC layer, and the contention overheads around the bottlenecks will degrade the system performance significantly. For the above reasons, link-disjoint routing always outperforms node-disjoint routing when interacting with contention-based MACs, as Table 2 shows.

7.2.2 NAMA

50 nodes	Node-disjoint (analytical) (Mb/s)	Node-disjoint (simulation) (Mb/s)	Link-disjoint (analytical) (Mb/s)	Link-disjoint (simulation) (Mb/s)
10 flows	125.54	117.29	123.27	121.03
20 flows	118.81	114.42	118.81	118.98
30 flows	116.02	112.13	115.78	116.37
100 nodes	Node-disjoint (analytical)	Node-disjoint (simulation)	Link-disjoint (analytical)	Link-disjoint (simulation)
20 flows	351.80	341.23	323.07	337.15
30 flows	320.94	313.42	314.85	316.38
40 flows	307.59	309.78	301.97	306.42

Table 3. NAMA system throughput with different multipath packet forwarding

In contrast to contention-based MAC protocols, when a schedule-based MAC interacts with different multi-path packet forwarding disciplines, there is no significant difference between node-disjoint routing and link-disjoint routing. This is shown in Fig. 2 and Table 3. Revisiting the modeling process of the schedule-based MAC (Eq. (21)), its performance is mainly



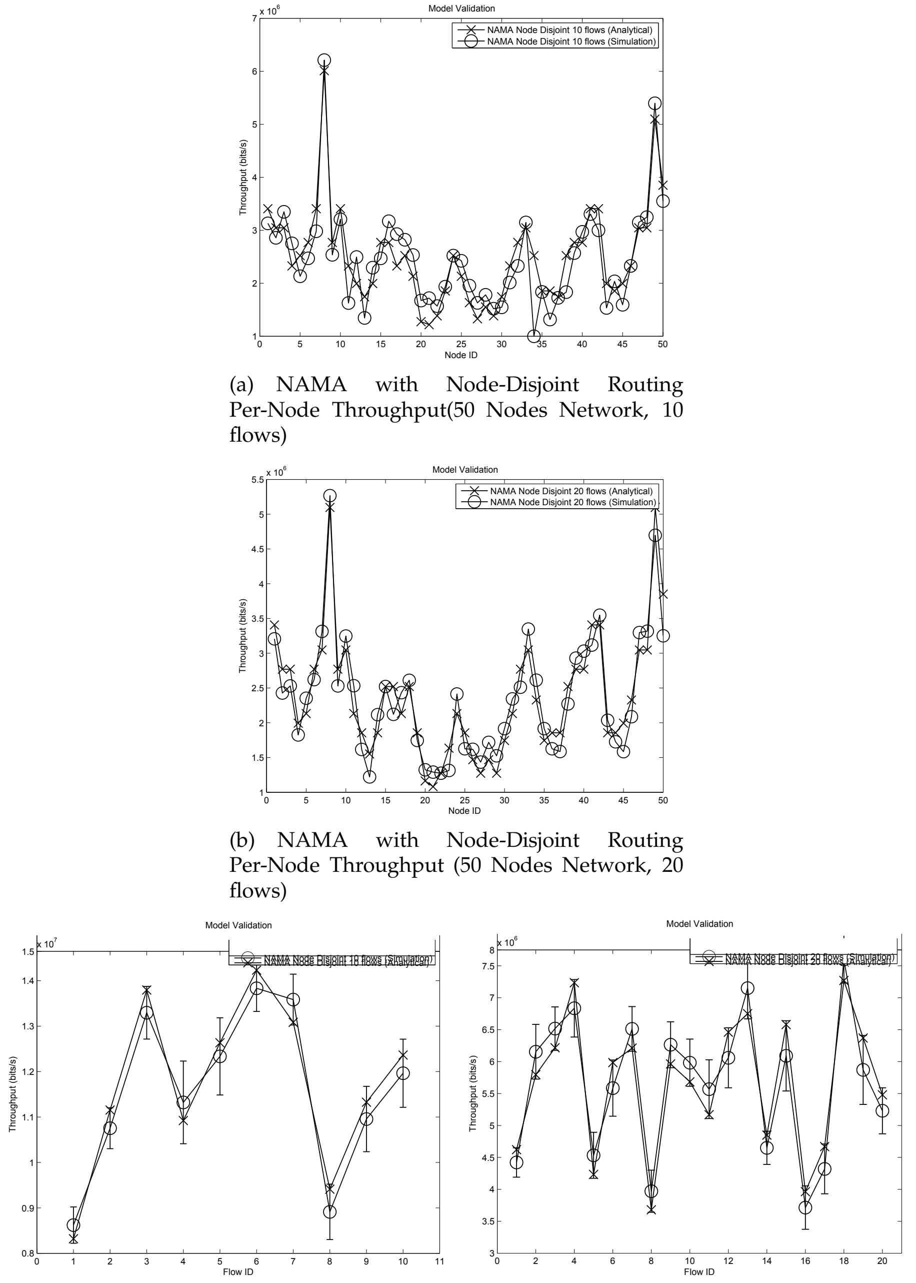
(a) 802.11 DCF with Node-Disjoint Routing Per-node Throughput (50 Nodes Network)

(b) 802.11 DCF with Link-Disjoint Routing Per-node Throughput (50 Nodes Network)

(c) 802.11 DCF with Node-Disjoint Routing Per-flow Throughput (50 Nodes Network, 20 Flows)

(d) 802.11 DCF with Link-Disjoint Routing Per-flow Throughput (50 Nodes Network, 20 Flows)

Fig. 1. Model Validation: 802.11 DCF



(a) NAMA with Node-Disjoint Routing Per-Node Throughput(50 Nodes Network, 10 flows)

(b) NAMA with Node-Disjoint Routing Per-Node Throughput (50 Nodes Network, 20 flows)

(c) NAMA with Node-Disjoint Routing Per-flow Throughput(50 Nodes Network, 10 flows)

(d) NAMA with Node-Disjoint Routing Per-flow Throughput (50 Nodes Network, 20 flows)

Fig. 2. Model Validation: NAMA

dependent on two factors: (a) The probability that the topology information exchange is successful; and (b) the conditional probability that a transmission schedule is collision-free given the correct topology information. Although the first factor is partially decided by the the number of contending nodes, the contention overheads will not increase linearly with the intensity of contentions, as contention-based MACs do. In other words, channel access contention may influence how quickly the collision-free transmission schedule is formed, while it does not influence the system throughput over the long-time run if the schedule mechanism works correctly. Another reason why schedule-based MACs are insensitive to the behavior of the routing protocol in our model is that the schedule rule is to increase the spatial/time reuse in the two-hop range to the largest extent, which alleviates the congestion introduced by routing protocols, if there are any.

7.3 Interaction between opportunistic forwarding and MAC

We now examine the impact of packet forwarding rules on different MAC protocols. For opportunistic forwarding, we vary different p_f values. As Table 4-Table 7 show, multiple-copy forwarding degrades system throughput while opportunistic forwarding could improve system throughput to some extent.

7.3.1 802.11 DCF

The system throughput comparisons of 802.11 DCF under different packet forwarding rules are shown in Table 4 and Table 5. We observe that, when combined with 802.11 DCF, opportunistic forwarding could enhance the system throughput for some p_f.

50 nodes	Single-copy forwarding (analytical) (Mb/s)	Single-copy forwarding (simulation) (Mb/s)	Multiple-copy forwarding (analytical) (Mb/s)	Multiple-copy forwarding (simulation) (Mb/s)
10 flows	22.38	21.75	16.28	16.59
20 flows	20.09	19.26	16.33	15.14
30 flows	18.41	18.78	14.99	13.73
100 nodes	Single-copy forwarding (analytical) (Mb/s)	Single-copy forwarding (simulation) (Mb/s)	Multiple-copy forwarding (analytical) (Mb/s)	Multiple-copy forwarding (simulation) (Mb/s)
20 flows	64.01	59.74	41.99	36.62
30 flows	65.20	61.26	38.52	41.75
40 flows	68.43	64.35	41.04	43.17

Table 4. 802.11 DCF system throughput with different routing forwarding rules

7.3.2 NAMA

The system throughput results for NAMA using different packet forwarding rules are shown in Table 6 and Table 7. We observe that, in contrast to the results shown in Table 4, when combining NAMA with opportunistic forwarding, the improvement of system throughput is quite minor. To understand the reason for the differences in the results obtained with 802.11 DCF and NAMA, we need to revisit how opportunistic forwarding impacts the system performance. First, opportunistic forwarding increases the system reliability by using multiple successors to forward duplicate packets. This is at the cost of consuming more system resources, which is the major reason that single-copy forwarding always outperforms multi-copy forwarding in terms of throughput. Second, one key aspect of opportunistic forwarding is that the node that forwards a packet is determined on-the-fly, which means that the contention neighbor sets V_r^l and C_i^r change over time. This is desirable when a contention-based MAC is used, because it increases the robustness of the end-to-end transmissions and could accommodate channel fluctuations. However, it is more difficult for a schedule-based MAC to build a collision-free transmission schedule. What is more, the schedule-based MAC also alleviates the collisions of transmissions and physical-layer interference to some extent. As a result, the gain of the opportunistic forwarding is reduced when combined with a schedule-based MAC, as Table 6 shows. Given that most opportunistic routing schemes have been evaluated over contention-based MAC (802.11 DCF or its extensions) (Biswas & Morris, 2005) (Chachulski et al., 2007), the results obtained in this paper motivate us to rethink how to leverage opportunistic forwarding using generic MAC protocols. From Table 5 and Table 7, we can also find the system throughput does not increase linearly with p_f. This is because a larger p_f not only increases the reliability of end-to-end delivery, but also the contentions within the two-hop range. For each simulation experiment, there is an optimal p_f, which is dependent on the topology and the traffic pattern.

8. Conclusion

We introduced a novel analytical model to study the interactions of MAC and packet forwarding schemes in multi-hop wireless networks. Our model captures different aspects of the protocol interaction procedure and different information feedback across layers, and permits us to study how the use of multiple paths and packet forwarding rules influence the performance of different MAC protocols. We validated our analytical model by comparing its results against simulation experiments. Given the good match between analytical and simulation results, it follows that the results obtained from the analytical model can provide valuable insights on the interaction between MAC and routing protocol and how protocol stacks could be optimized.

50 nodes	$p_f = 0.2$ (analytical) (Mb/s)	$p_f = 0.2$ (simulation) (Mb/s)	$p_f = 0.4$ (analytical)(Mb/s)	$p_f = 0.4$ (simulation) (Mb/s)
10 flows	26.76	28.09	24.25	22.62
20 flows	25.15	26.17	22.78	24.47
30 flows	25.03	24.72	21.96	24.08
50 nodes	$p_f = 0.6$ (analytical) (Mb/s)	$p_f = 0.6$ (simulation) (Mb/s)	$p_f = 0.8$ (analytical) (Mb/s)	$p_f = 0.8$ (simulation) (Mb/s)
10 flows	21.09	22.55	18.43	19.13
20 flows	19.27	20.76	17.06	17.88
30 flows	18.45	19.87	15.11	16.52
100 nodes	$p_f = 0.2$ (analytical) (Mb/s)	$p_f = 0.2$ (simulation) (Mb/s)	$p_f = 0.4$ (analytical)(Mb/s)	$p_f = 0.4$ (simulation) (Mb/s)
20 flows	76.18	79.69	67.26	71.25
30 flows	75.27	78.85	65.13	69.23
40 flows	78.31	78.26	66.89	69.28
100 nodes	$p_f = 0.6$ (analytical) (Mb/s)	$p_f = 0.6$ (simulation) (Mb/s)	$p_f = 0.8$ (analytical) (Mb/s)	$p_f = 0.8$ (simulation) (Mb/s)
20 flows	59.22	63.54	49.04	48.15
30 flows	60.91	62.08	45.16	41.21
40 flows	58.34	62.99	46.60	43.12

Table 5. 802.11 DCF system throughput with different opportunistic forwarding (p_f)

50 nodes	Single-copy forwarding (analytical) (Mb/s)	Single-copy forwarding (simulation) (Mb/s)	Multiple-copy forwarding (analytical) (Mb/s)	Multiple-copy forwarding (simulation) (Mb/s)
10 flows	96.02	91.08	61.64	66.53
20 flows	92.11	86.39	59.40	55.49
30 flows	86.25	82.01	57.89	53.26
100 nodes	Single-copy forwarding (analytical) (Mb/s)	Single-copy forwarding (simulation) (Mb/s)	Multiple-copy forwarding (analytical) (Mb/s)	Multiple-copy forwarding (simulation) (Mb/s)
20 flows	265.14	254.39	161.54	168.58
30 flows	243.28	231.76	157.43	149.19
40 flows	214.87	203.91	150.99	143.21

Table 6. NAMA system throughput with different routing forwarding rules

50 nodes	$p_f = 0.2$ (analytical) (Mb/s)	$p_f = 0.2$ (simulation) (Mb/s)	$p_f = 0.4$ (analytical) (Mb/s)	$p_f = 0.4$ (simulation) (Mb/s)
10 flows	98.10	104.28	83.37	80.19
20 flows	96.35	100.02	80.29	84.45
30 flows	88.24	96.23	78.06	81.27
50 nodes	$p_f = 0.6$ (analytical) (Mb/s)	$p_f = 0.6$ (simulation) (Mb/s)	$p_f = 0.8$ (analytical) (Mb/s)	$p_f = 0.8$ (simulation) (Mb/s)
10 flows	75.16	79.85	66.26	70.24
20 flows	72.32	71.58	68.84	66.59
30 flows	70.35	68.73	64.56	68.16
100 nodes	$p_f = 0.2$ (analytical) (Mb/s)	$p_f = 0.2$ (simulation) (Mb/s)	$p_f = 0.4$ (analytical)(Mb/s)	$p_f = 0.4$ (simulation) (Mb/s)
20 flows	270.18	262.39	231.04	225.01
30 flows	246.23	234.85	217.50	219.74
40 flows	219.72	231.80	202.59	210.88
100 nodes	$p_f = 0.6$ (analytical) (Mb/s)	$p_f = 0.6$ (simulation) (Mb/s)	$p_f = 0.8$ (analytical) (Mb/s)	$p_f = 0.8$ (simulation) (Mb/s)
20 flows	196.16	182.40	182.55	178.14
30 flows	185.24	170.16	180.61	172.06
40 flows	183.44	174.33	176.18	169.58

Table 7. NAMA system throughput with different opportunistic forwarding (p_f)

9. References

Abramson, N. (1970). The ALOHA system—another alternative for computer communications, *AFIPS Conf. Proc.*, Vol. 37, FJCC, pp. 281–285.

Abramson, N. (1977). The throughput of packet broadcast channels, *IEEE Trans. on Communications* COM-25(1): 117–128.

Arnbak, J. C. & Blitterswijk, W. V. (1987). Capacity of slotted aloha in rayleigh-fading channels, *IEEE Journal on Selected Areas in Communications* SAC-5(2): 261–269.

Bai, F., Sadagopan, N. & Helmy, A. (2003). IMPORTANT: A framework to systematically analyze the Impact of Mobility on Performance of RouTing protocols for Ad Hoc Networks, *Proc. IEEE INFOCOM*, San Francisco, CA, USA, pp. 825–835.

Barghavan, V., Demers, A., Shenker, S. & Zhang, L. (1994). MACAW: A media access protocol for wireless LAN's, *Proc. of ACM SIGCOMM '94*, pp. 212–225.

Barrett, C., Drozda, M., Marathe, A. & Marathe, M. (2003). Characterizing the Interaction between Routing and MAC Protocols in Ad-Hoc Networks, *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc)*, Lausanne, Switzerland, pp. 92–103.

Bianchi, G. (2000). Performance analysis of the IEEE 802.11 distributed coordination function, *IEEE Journal on Selected Areas in Communications* 18(3): 535–547.

Biswas, S. & Morris, R. (2005). ExOR: Opportunistic Multi-hop Routing for Wireless Networks, *Proc. ACM SIGCOMM*.

Boorstyn, R. R., Kershenbaum, A., Maglaris, B. & Sahin, V. (1987). Throughput analysis in multihop CSMA packet radio networks, *IEEE Trans. on Communications* COM-35(3): 267–274.

Cali, F., Conti, M. & Gregori, E. (2000). Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit, *IEEE/ACM Tran. on Networking* 8(6): 785–799.

Carvalho, M. M. & Garcia-Luna-Aceves, J. J. (2003). Delay analysis of IEEE 802.11 in single-hop networks, *Proc. 11th IEEE International Conference on Network Protocols (ICNP)*, Atlanta, USA, pp. 146–155.

Carvalho, M. M. & Garcia-Luna-Aceves, J. J. (2004). A Scalable Model for Channel Access Protocols in Multihop Ad Hoc Networks, *Proc. ACM Mobicom*, Philadelphia, USA.

Chachulski, S., Jennings, M., Katti, S. & Katabi, D. (2007). Trading structure for randomness in wireless opportunistic routing, *Proc. ACM SIGCOMM*.

Chhaya, H. & Gupta, S. (1997). Performance modeling of asynchronous data transfer methods of IEEE 802.11 MAC protocol, *Wireless Networks* 3: 217–234.

Colvin, A. (1983). CSMA with collision avoidance, *Computer Commun.* 6(5): 227–235.

Das, S., Perkins, C. & Royer, E. (2000). Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks, *Proc. IEEE Conference on Computer Communications (INFOCOM)*, Tel Aviv, Israel, pp. 3–12.

Das, S. R., Perkins, C. E., Royer, E. M. & Marina, M. K. (2001). Performance comparison of two on-demand routing protocols for ad hoc networks, *IEEE Personal Communications Magazine, special issue on Mobile Ad Hoc Networks*.

Davis, D. H. & Gronemeyer, S. A. (1980). Performance of slotted ALOHA random access with delay capture and randomized time of arrival, *IEEE Trans. Commun.* COM-28(5): 703–710.

Foh, C. & Zukerman, M. (2002). Performance analysis of the IEEE 802.11 MAC protocol, *Proc. of the European Wireless 2002 Conference*, Florence, Italy, pp. 184–190.

Fullmer, C. L. & Garcia-Luna-Aceves, J. J. (1995). Floor acquisition multiple access (FAMA) for packet-radio networks, *SIGCOMM '95*, Cambridge, MA (USA), pp. 262–273.

Fullmer, C. L. & Garcia-Luna-Aceves, J. J. (1997). Solutions to hidden terminal problems in wireless networks, *Proc. ACM SIGCOMM 97*, Cannes, France.

Gitman, I. (1975). On the capacity of slotted ALOHA networks and some desigh problems, *IEEE Trans. on Communications* COM-23(3): 305–317.

Karn, P. (1990). MACA - a new channel access method for packet radio, *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pp. 134–140.

Kim, H. & Hou, J. C. (2003). Improving protocol capacity with model-based frame scheduling in IEEE 802.11-operated WLANs, *Proc. of the 9th ACM International Conference on Mobile Computing and Networking (MOBICOM)*, San Diego, CA, USA, pp. 190–204.

Kleinrock, L. & Lam, S. S. (1975). Packet switching in a multiaccess broadcast channel: Performance evaluation, *IEEE Trans. on Communications* COM-23(4): 410–423.

Kleinrock, L. & Tobagi, F. A. (1975). Packet switching in radio channels: Part I - carrier sense multiple-access modes and their throughput-delay characteristics, *IEEE Trans. on Communications* COM-23(12): 1400–1416.

Lee, S. & Gerla, M. (2001). Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks, *Proceedings of the IEEE ICC*, pp. 3201–3205.

Lo, W. F. & Mouftah, H. T. (1984). Carrier sense multiple access with collision detection for radio channels, *IEEE 13th Int'l Commun. and Energy Conf.*, pp. 244–247.

Marina, M. & Das, S. (2001). On-demand Multipath Distance Vector Routing in Ad Hoc Networks, *Proceedings of IEEE International Conference on Network Protocols (ICNP)*, pp. 14–23.

Musser, M. & Daigle, J. (1982). Throughput analysis of an asynchronous code division multiple access (CDMA) system, *Proc. ICC'82*, Philadelphia, PA.

Nasipuri, A., Castaeda, R. & Das, S. R. (2001). Performance of Multipath Routing for On-demand Protocols in Mobile Ad Hoc Networks, *Mob. Netw. Appl.* 6(4): 339–349.

Pursley, M. (1983). Throughput of frequency-hopped spread spectrum communications for packet radio networks, *Proc. 1983 CISS*, John Hopkins Univ., Baltimore, MD, USA.

Qualnet Simulator (n.d.). Scalable Network Technologies, http://www.scalable-networks.com/.

Raychauduri, D. (1981). Performance analysis of random access packet-switched code division multiple access systems, *IEEE Trans. Commun.* COM-29(6): 895–901.

Roberts, L. G. (1975). ALOHA packet system with and without slots and capture, *Comput. Commun. Rev.* 5: 28–42.

Royer, E., Lee, S. & Perkins, C. (2000). The Effects of MAC Protocols on Ad hoc Network Communications, *Proc. IEEE Wireless Communications and Networking Conference(WCNC)*, Chicago, IL.

Shepard, T. J. (1996). A channel access scheme for large dense packet radio networks, *Proc. of ACM SIGCOMM*, ACM Press, pp. 219–230.

Tobagi, F. A. (1980a). Analysis of a two-hop centralized packet radio network—part I: Slotted ALOHA, *IEEE Trans. Commun.* COM-28(2): 196–207.

Tobagi, F. A. (1980b). Analysis of a two-hop centralized packet radio network—part II: Carrier sense multiple access, *IEEE Trans. Commun.* COM-28(2): 208–216.

Tobagi, F. A. & Brazio, J. M. (1983). Throughput analysis of multihop packet radio network under various channel access schemes, *Proc. INFOCOM'83*, San Diego, CA.

Tobagi, F. A. & Kleinrock, L. (1975). Packet switching in radio channels: Part II - the hidden terminal problem in carrier sense multiple-access modes and the busy-tone solution, *IEEE Trans. on Communications* COM-23(12): 1417–1433.

Tse, D. & Hanly, S. (1999). Linear multiuser receivers: Effective interference, effective bandwidth and user capacity, *IEEE Trans. Information Theory* 45(2): 641–657.

Vadde, K. K. & Syrotiuk, V. R. (2004). Factor Interaction on Service Delivery in Mobile Ad Hoc Networks, *IEEE Journal on Selected Areas in Communications* pp. 1335–1346.

Vadde, K. K., Syrotiuk, V. R. & Montgomery, D. C. (2006). Optimizing Protocol Interaction using Response Surface Methodology, *IEEE Transactions on Mobile Computing* 5: 627–639.

Wang, Y. & Garcia-Luna-Aceves, J. J. (2002). Performance of collision avoidance protocols in single-channel ad hoc networks, *Proc. of 10th IEEE International Conference on Network*

Protocols (ICNP), Paris, France.

Wu, L. & Varshney, P. (1999). Performance analysis of CSMA and BTMA protocols in multihop networks (I). single channel case, *Information Sciences, Elsevier Sciences Inc.* 120: 159–177.

Ye, Z., Krishnamurthy, S. V. & Tripathi, S. K. (2003). A Framework for Reliable Routing in Mobile Ad Hoc Networks, *Proceedings of the IEEE INFOCOM*, pp. 270–280.

10

A Bandwidth Reservation QoS Routing Protocol for Mobile Ad Hoc Networks

Wen-Hwa Liao
Dept. of Information Management, Tatung University
Taiwan

1. Introduction

The advancement in wireless communication and economical, portable computing devices have made mobile computing possible. One research issue that has attracted a lot of attention recently is the design of mobile ad hoc network (MANET). A MANET is one consisting of a set of mobile hosts which can communicate with one another and roam around at their will. No base stations are supported in such an environment. Due to constraints such as battery power, transmission distance, and channel utilization, a mobile host may not be able to communicate directly with all other hosts in a single-hop fashion. In this case, a multi-hop scenario occurs, where packets may need to be relayed by several intermediate hosts before reaching their destinations. Applications of MANETs occur in situations like battlefields and major disaster areas, where networks need to be deployed immediately but base stations or fixed network infrastructures are not available.

Since MANET is characterized by its fast changing topology, extensive research efforts have been devoted to the design of routing protocols for MANETs (Haas & Pearlman, 1998; Johnson et al., 2002; Liao et al., 2001; Perkins & Bhagwat, 1994; Perkins et al., 2002; Royer & Toh, 1999; Wu & Li, 2001). These works only concern with shortest-path routing and the availability of multitude routes in the MANET's dynamically changing environment. So only best-effort data traffic is provided. Issues related to quality-of-service (QoS) requirements, such as delay and bandwidth bounds, are less frequently addressed.

This paper considers the problem of searching for a route of a given bandwidth in a MANET. This problem has been addressed by several works in the literature. References (Chen & Nahrstedt, 1999; Liao et al., 2002) have discussed this problem by assuming quite an ideal model that the bandwidth of a link can be determined independently of its neighboring links. This is untrue if all mobile hosts share a single common channel, or one needs to assume a costly multi-antenna model where a host can send/receive using several antennas simultaneously and independently. A less stronger assumption made in (Lin, 2001; Lin & Liu, 1999) is the CDMA-over-TDMA channel model, where the use of a time slot on a link is only dependent of the status of its one-hop neighboring links. Reference (Stojmenovic et al., 2000) addresses QoS routing with delay and bandwidth constraints, but still no specific channel model is accounted.

In this paper, we assume a simpler TDMA model on a single common channel shared by all hosts. So it is inevitable to take the radio interference problems into consideration. We consider the bandwidth reservation problem in such environment. A route discovery protocol is proposed, which is able to find a route with a given bandwidth (represented by number of slots). When making reservation, both the hidden-terminal and exposed-terminal problems will be taken into consideration.

The rest of the paper is organized as follows. Section 2 contains backgrounds and preliminaries. Our routing protocol is presented in Section 3. Experimental results are in Section 4. Section 5 concludes this paper.

2. Preliminaries

A MANET is one consisting of a set of mobile hosts which may communicate with one another and roam around at their will. Communication is done through wireless links among mobile hosts by their antennas, but no base stations are supported in such an environment. The MANET distinguishes itself from traditional wireless networks by its dynamic changing topology, no base-station support, and the need of multi-hop communication capability. To support multi-hop communication, a routing protocol is needed to forward data packets.

2.1 QoS transmission in MANET

Guaranteeing QoS is important for networks to support multimedia applications (such as video and audio transmissions). QoS defines nonfunctional characteristics of a system, affecting the perceived quality of the result. For multimedia applications, this may include picture quality, image quality, and speed of response. From technology point of view, QoS characteristics include timeliness (e.g. delay or response time), bandwidth (e.g. bandwidth required or available), and reliability (e.g. normal operation time between failures or down time from failure to restarting normal operation) (Chalmers & Sloman, 1999; Wang & Crowcroft, 1996). Providing QoS is more difficult for MANET due to at least two reasons. First, unlike wired networks, radios have broadcast nature. Thus, each link's bandwidth will be affected by the transmission/receiving activities of its neighboring links. Second, unlike cellular networks, where only one-hop wireless communication is involved, a MANET needs to guarantee QoS on a multi-hop wireless path. Further, mobile hosts may join, leave, and rejoin at any time and any location; existing links may disappear and new links may be formed on-the-fly. All these raise challenges to QoS routing in a MANET.

Issues related to QoS transmission in MANET have received attention recently. In (Sobrinho & Krishnakumar, 1999), this is addressed on the medium access control (MAC) layer, where mobile hosts contend for accessing the common radio channel based on how urgent their data is, and this is determined based on the amount of time that a host has been waiting for the channel to become idle. Once admitted, a host is ensured with high probability to send real-time packets in a collision-free manner. QoS routing is considered in (Chen & Nahrstedt, 1999; Lin, 2001; Lin & Liu, 1999; Stojmenovic et al., 2000). In (Chen & Nahrstedt, 1999), a ticket-based protocol is proposed to support QoS routing. This protocol maintains the end-to-end state information at every node for every possible destination by a distance-vector-like protocol (Perkins & Bhagwat, 1994). A source node *s*, on requiring a QoS route, can issue a number of probing packets each carrying a ticket. Each probe is in charge of searching for one path, if possible. The basic idea of using tickets is to confine the number of route-searching packets to avoid a blind flooding (flooding in a MANET is very costly, according to (Ni et al., 1999)). Each probe, on reaching an intermediate node, should choose one outgoing path that satisfies the QoS requirements. However, this paper assumes that the bandwidth of a link can be determined independently of its neighboring links. The is quite a strong assumption, because a costly multi-antenna model may be needed. Otherwise, such an assumption is generally untrue because neighboring mobile hosts sharing a common channel will interference with each other. In (Lin, 2001; Lin & Liu, 1999), how to calculate the bandwidth of a routing path in a MANET is addressed. A CDMA-over-TDMA channel model is assumed. The code used by a host should be different from that used by any of its two-hop neighbors. So a code assignment protocol should be supported (this can be regarded as an independent problem; references can be found in (Bertossi & Bonuccelli, 1995; Garcia-Luna-Aceves & Raju, 1997; Ju & Li, 1999)). The bandwidth requirement is realized by reserving time slots on links. Based on such assumption, this paper shows how to allocate time slots on each link of a path such that no two adjacent links share a common time slot. Reference (Stojmenovic, 2000) addresses QoS routing with delay and bandwidth constraints. It suggests that the depth-first search be used to find routes. However, no specific channel model is accounted.

2.2 System model and challenges

This paper is concerned with QoS routing in a MANET. Different from the above referenced works, we assume a simpler (and perhaps more realistic) TDMA-based channel model. One single common channel is assumed to be shared by all hosts in the MANET. The channel is time-framed. Each frame is divided into a control phase and a data phase, as shown in Fig. 1. The former supports various kinds of control functions, such as frame synchronization, call setup, call maintenance, and time slot reservation for QoS routing. The latter consists of s time slots, indexed from 1 to s, each being able to carry one data packet, where s is a predefined integer. This model may be emulated by wireless LAN cards which follow the IEEE 802.11 standard (IEEE, 1997).

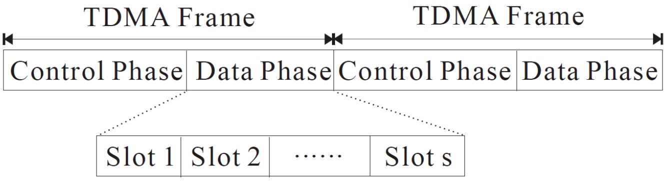


Fig. 1. The TDMA frame structure.

Because an antenna can not send and receive at the same time, bandwidth calculation in a multi-hop route is a non-trivial problem. Take the path from *A* to *C* in Fig. 2(a) as an example, where the white slots associated with each host are free and the gray slots are busy. Matching the free slots between hosts, we obtain five common free time slots {1, 2, 3, 4, 5} between *A* and *B* and four common free time slots {3, 4, 5, 6} between *B* and *C*. One may naively think that the path bandwidth from *A* to *C* is four (= min { 4, 5 }). Unfortunately, this is not true. As shown in Fig. 2(b), if we reserve slots {1, 2, 3} for *A* to transmit and slots {4, 5, 6} for *B* to transmit, the path bandwidth is only three. In fact, it is not hard to see that it is impossible to further increase the path bandwidth in this example. Even worse, as shown in Fig. 2(c), if one reserves slots {3, 4} for *A* to transmit and slots {5, 6} for *B* to transmit, the path bandwidth will degrade to two, and the situation cannot be improved, unless we change the assignment for *A*.

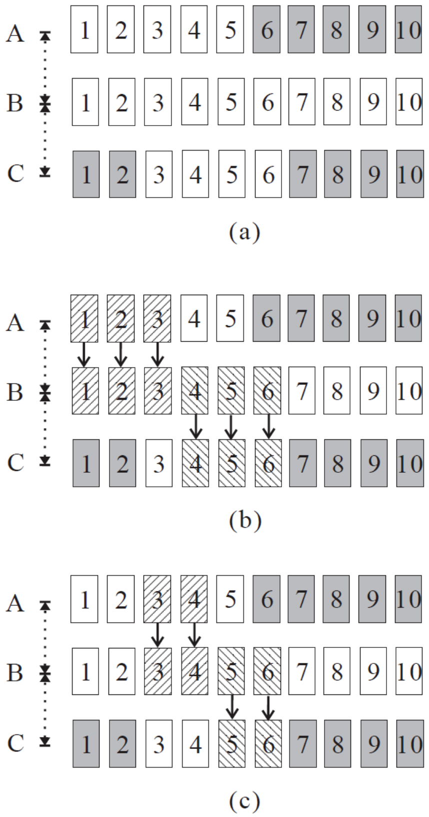


Fig. 2. The bandwidth calculation problem under the TDMA model: (a) free and busy time slots, (b) an assignment with bandwidth = 3 slots from *A* to *C*, and (c) an assignment with bandwidth = 2 from *A* to *C*.

The above discussion has already been simplified by purposely ignoring the transmission and reception activities of individual mobile hosts. At this point, we'd like to recall the hidden-terminal and exposed-terminal problems, which are well-known problems in the literature of radio-based communication. Consider the scenario in Fig. 3, where the status of *A*, *B*, and *C* is the same as the above example. Suppose there is another pair, *D* and *E*, who are currently using slot 2 to communicate. Then two cases will occur. If *D* is a receiver on slot 2, *A* will not be allowed to send on slot 2 because otherwise collision will occur at *D*. This is the hidden-terminal problem. So in the example of Fig. 2, the common free time slots between *A* and *B* should be reduced to {1, 3, 4, 5}. Then the case in Fig. 2(b) will not hold anymore, and the path bandwidth from *A* to *C* has to downgrade to 2 slots. On the contrary, if *D* is a sender on slot 2, *A* will still be allowed to send on slot 2, because this is an exposed-terminal problem. Then the common free time slots between *A* and *B* (and thus the path bandwidth) remain the same.

While the above examples already show the complication in the bandwidth reservation problem, we'd like to comment on the data structure used above. From the discussion, we see that simply indicating a time slot as busy or free is insufficient to resolve the hidden- and exposed-terminal problems. For the busy case, we need to tell whether the host is sending or receiving in this slot. This observation motivates our design in the next section.

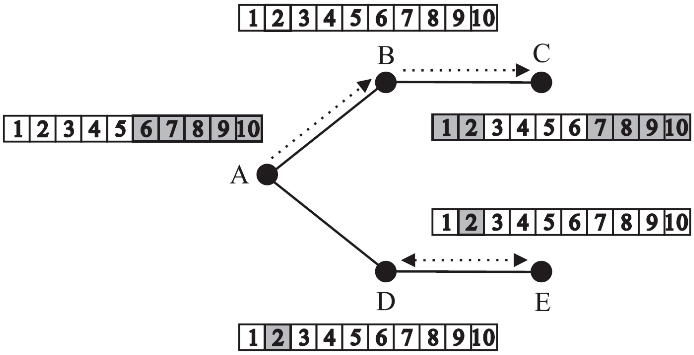


Fig. 3. Example of how bandwidth calculation is interfered by the hidden- and exposed-terminal problem.

3. QoS routing protocol

3.1 Basic idea

Our routing protocol is an on-demand one, so route search is done only when necessary. (The contrary is proactive, which is generally regarded to be more costly.) It is based on source routing, and works similar to the DSR protocol (Johnson et al., 2001) on disseminating route-searching packets, but we need to carefully calculate the bandwidth of each route being searched.

A source host *S*, on requiring a route to a destination *D* with bandwidth *b*, will issue through broadcast a QoS route request packet *QREQ*(…, *b*, *PATH*, *NH*) to its neighbors. The field *PATH* provides the important information to keep track of the partial route and time slots that the *QREQ* packet has discovered so far. The *NH* is a list of hosts, each of which may be used as the next hop to extend *PATH* with one more hop. Any host *x* listed in *NH* hearing this *QREQ* for the first time may rebroadcast this packet, if it has sufficient collision-free time slots (here the route cache design may be raised to reduce the flooding cost; however, we deal this as an independent issue and refer the reader to the literature (Marina & Das, 2001; Perkins et al., 2002). In *x*'s rebroadcast, proper information will be added to *PATH* and *NH*.

When *D* receives the *QREQ* packet, it can reply a QoS route reply packet *QREQ*(…, *PATH*) destined to the source *S*. This packet will be routed, through unicast, along the reverse direction of *PATH*, and on its way back reserve proper time slots at intermediate hosts according to the content in *PATH*. Our protocol relies on the following lemma to choose time slots in a host.

Lemma 1: A time slot *t* can be used by a host *X* to send to another host *Y* without causing collision if the following conditions are all satisfied:

1. Slot *t* is not yet scheduled to send or receive in neither *X* nor *Y*.
2. For any 1-hop neighbor *Z* of *X*, slot *t* is not scheduled to receive in *Z*.
3. For any 1-hop neighbor *Z* of *Y*, slot *t* is not scheduled to send in *Z*.

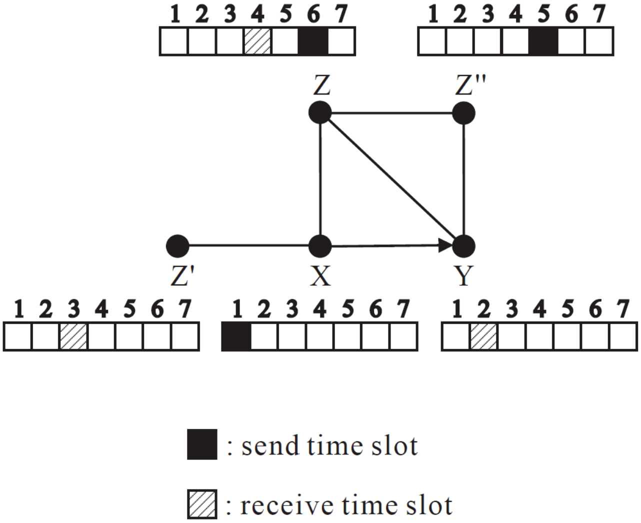


Fig. 4. An example of Lemma 1.

For example, in Fig. 4, a host *X* needs a time slot to transmit to a host *Y*. First, slots 1 and 2 can not be considered because slot 1 is used by *X* to send and slot 2 is used by *Y* to receive. Second, slots 3 and 4 can not be considered because they will cause collision at *Z* and *Z*'. Third, slots 5 and 6 can not be considered because *Z* and *Z*'' are sending on these slots. So we conclude that only slot 7 can be used.

3.2 Data structures

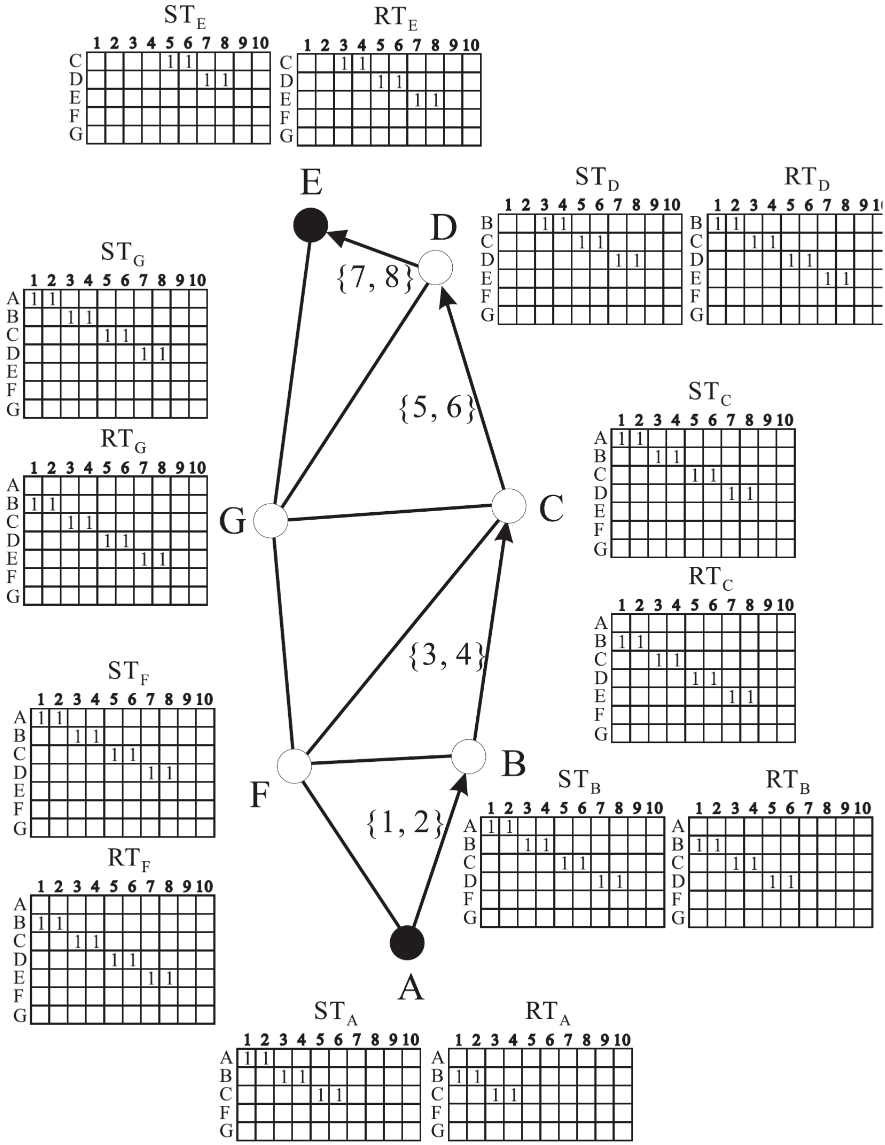
We will index hosts by numbers 1, 2, … , *n*, and time slots by 1, 2, …, *s*. Each host *x* will maintain three tables as follows.

- ST_x[1..*n*, 1..*s*]: the *send table* of *x*, which records on which time slots a host which is within 2 hops from *x* will have sending activities. Specifically, ST_x [*i*, *j*]= 1 if slot *j* of host *i* has been reserved for sending; otherwise, ST_x [*i*, *j*]=0.
- RT_x[1..*n*, 1..*s*]: the *receive table* of *x*, which records on which time slots a host which is within 2 hops from *x* will have receiving activities. Specifically, RT_x[*i*, *j*]= 1 if slot *j* of host *i* has been reserved for receiving; otherwise, RT_x[*i*, *j*]= 0.
- H_x[1..*n*, 1..*n*]: the *hop-count matrix* of *x*, which is to keep track of the mutual distances between hosts in *x*'s neighborhood. Specifically, for each host *i* that is within 1 hop from *x*, H_x[*i*, *j*] = 1 if host *j* is within 1 hop from *i*; otherwise, H_x[*i*, *j*] = ∞.

In Fig. 5, host *A* is sending to *E* on the path *A* → *B* → *C* → *D* → *E* of bandwidth of 2 slots. Slots {1, 2}, {3, 4}, {5, 6}, and {7, 8} are used by *A*, *B*, *C*, *D* to send, respectively. They are reflected on each host's *ST* and *RT* tables. Note that the rows ST_x[*x*, *j*] and RT_x[*x*, *j*] indicate the sending and receiving activities of host *x* itself. In order to create these data structures, a host needs to periodically broadcast its own status to its 2-hop neighbors.

To search for a QoS route, we mainly use the packet *QREQ*(*S*, *D*, *id*, *b*, *x*, *PATH*, *NH*), whose parameters are defined as follows.

- *S*: the source host.
- *D*: the destination host.



ST

ST_E

	1	2	3	4	5	6	7	8	9	10
C					1	1				
D							1	1		
E										
F										
G										

RT_E

	1	2	3	4	5	6	7	8	9	10
C			1	1						
D					1	1				
E							1	1		
F										
G										

ST_D

	1	2	3	4	5	6	7	8	9	10
B			1	1						
C					1	1				
D							1	1		
E										
F										
G										

RT_D

	1	2	3	4	5	6	7	8	9	1
B	1	1								
C			1	1						
D					1	1				
E							1	1		
F										
G										

ST_G

	1	2	3	4	5	6	7	8	9	10
A	1	1								
B			1	1						
C					1	1				
D							1	1		
E										
F										
G										

RT_G

	1	2	3	4	5	6	7	8	9	10
A										
B	1	1								
C			1	1						
D					1	1				
E							1	1		
F										
G										

ST_C

	1	2	3	4	5	6	7	8	9	10
A	1	1								
B			1	1						
C					1	1				
D							1	1		
E										
F										
G										

RT_C

	1	2	3	4	5	6	7	8	9	10
A										
B	1	1								
C			1	1						
D					1	1				
E							1	1		
F										
G										

ST_F

	1	2	3	4	5	6	7	8	9	10
A	1	1								
B			1	1						
C					1	1				
D							1	1		
E										
F										
G										

RT_F

	1	2	3	4	5	6	7	8	9	10
A										
B	1	1								
C			1	1						
D					1	1				
E							1	1		
F										
G										

ST_B

	1	2	3	4	5	6	7	8	9	10
A	1	1								
B			1	1						
C					1	1				
D							1	1		
F										
G										

RT_B

	1	2	3	4	5	6	7	8	9	10
A										
B	1	1								
C			1	1						
D					1	1				
F										
G										

ST_A

	1	2	3	4	5	6	7	8	9	10
A	1	1								
B			1	1						
C					1	1				
F										
G										

RT_A

	1	2	3	4	5	6	7	8	9	10
A										
B	1	1								
C			1	1						
F										
G										

Fig. 5. Example of time slot tables ST and RT.

- *id*: an identity which is unique to each route-searching request issued by *S*. So the triplet (*S*, *D*, *id*) can be used to detect duplicate *QREQ* to avoid endless looping.
- *b*: the bandwidth requirement, represented by an integer number of slots.
- *x*: the host currently relaying the *QREQ* packet.
- *PATH*: the partial path, together with the available time slots, that has been discovered so far. It has the format ((h_1, l_1), (h_2, l_2), …, (h_k, l_k)). Each h_i, *i*=1..*k*, is a host identity, so the

sequence $h_1, h_2, \ldots, h_k, x$ represents the current partial path. Each l_i contains a total of b time slots that are found to be available for h_i to transmit to h_{i+1}, with the exception that h_k's intending receiver is host x.

- *NH*: a list next-hop hosts of the format $((h_1', l_1'), (h_2', l_2'), \ldots)$. Each host h_i' has potential to serve as the next hop of host x to extend the current partial path (so the new path will be $h_1, h_2, \ldots, h_k, x, h_i'$). However, this will depend on whether h_i' has sufficient time slots or not (this will become clear in the protocol). The corresponding parameter l_i' contains b time slots that can be used by x to transmit to h_i' without collision.

When a route is found, we need to initiate from the destination D a packet *QREP*(S, D, *id*, *PATH*) to the source S. This packet will travel on the reverse direction of *PATH* and reserve time slots, as discovered, on the path. These parameters carry the same meanings as above.

3.3 Protocol details

Now suppose a host y receiving a broadcasting packet *QREQ*(S, D, *id*, b, x, *PATH*, *NH*) initiated by a neighboring host x. If the same route request (uniquely identified by (S, D, *id*)) has not be heard by y before, it will perform the following steps:

```
A1. if  (y is not a host listed in NH) then
        exit this procedure.
    else
        Let $(h_i', l_i')$ be the entry in NH  such that $h_i' = y$.
        Construct a list   $PATH\_temp = PATH \mid (x, l_i')$, where | means list concatenation.
    end if.
```

A2. Construct two temporary tables, *ST_temp*[1..n, 1..s] and *RT_temp*[1..n, 1..s], as follows.

i. Copy all entries in ST_y[1..n, 1..s] into *ST_temp*[1..n, 1..s], and similarly copy all entries in RT_y[1..n, 1..s] into *RT_temp*[1..n, 1..s].

ii. Let $PATH = ((h_1, l_1), (h_2, l_2), \cdots, (h_k, l_k))$. For each i = 1..k-1, assign $ST\_temp[h_i, t] = 1$ and assign $RT\_temp[h_{i+1}, t] = 1$ for every time slot t in the list l_i. Assign $ST\_temp[h_k, t] = 1$ and assign $RT\_temp[x, t] = 1$ for every time slot t in the list l_k.

iii. Recall l_i' (the slots for x to send to y). Let $ST\_temp[x, t] = 1$ and $RT\_temp[y, t] = 1$ for every time slot t in the list l_i'.

These temporary tables, *ST_temp* and *RT_temp*, are obtained from *ST*, *RT*, *PATH*, and *NH*. This is because we are in the probing stage, but *ST* and *RT* only contain slot status already confirmed. The information in *PATH* and *NH* has to be introduced into these temporary tables.

```
A3. Let NH_temp= $\phi$ (i.e., an empty list).
    for  each 1-hop neighbor z of y do
        L= select_slot(y, z, b, ST_temp, RT_temp)
        if $L \neq \phi$ then
            NH_temp = NH_temp | (z, L)
        end if
    end for
```

The above step calls for a procedure *select_slot()*, which will return, if possible, b available slots that can be used by y to send to z (the details will be shown later). If the above loop can find at least one host to extend the current path, the *QREQ* will be rebroadcast, as shown below.

```
A4. if $NH\_temp \neq \phi$ then
      broadcast $QREQ(S, D, id, b, y, PATH\_temp, NH\_temp)$
    end if
```

The source host *S* will initiate the *QREQ*. It can be regarded as a special case of intermediate hosts, and can perform similarly to the above steps by replacing host *y* with *S*. We only summarize the modifications required for *S*. First, *S* has not *PATH* and *NH*. So in S1, the checking of *NH* is unnecessary. We can simply set *PATH_temp* =ϕ. Also, step A2 can be simplified to only executing step i. The other steps remain the same.

When the destination *D* receives packet *QREQ*(*S*, *D*, *id*, *b*, *x*, *PATH*, *NH*), a satisfactory path has been formed. *D* can accept the first *QREQ* received, or choose based on other policy. Then following steps will be executed.

```
B1. Let $(h_i', l_i')$ be the entry in $NH$ such that $h_i' = D$.
B2. $PATH\_temp = PATH \mid (x, l_i')$.
B3. Send $QREP(S, D, id, PATH\_temp)$ to $S$.
```

Note that the *QREP* packet will travel in the reverse direction of *PATH* through unicast. Each intermediate host should relay this packet. In addition, proper sending and receiving activities should be recorded in their sending and receiving tables. Specifically, let the whole path be *PATH* = $((h_1, l_1), (h_2, l_2), \ldots, (h_k, l_k))$. For each intermediate host $x = h_i$, the following steps should be conducted.

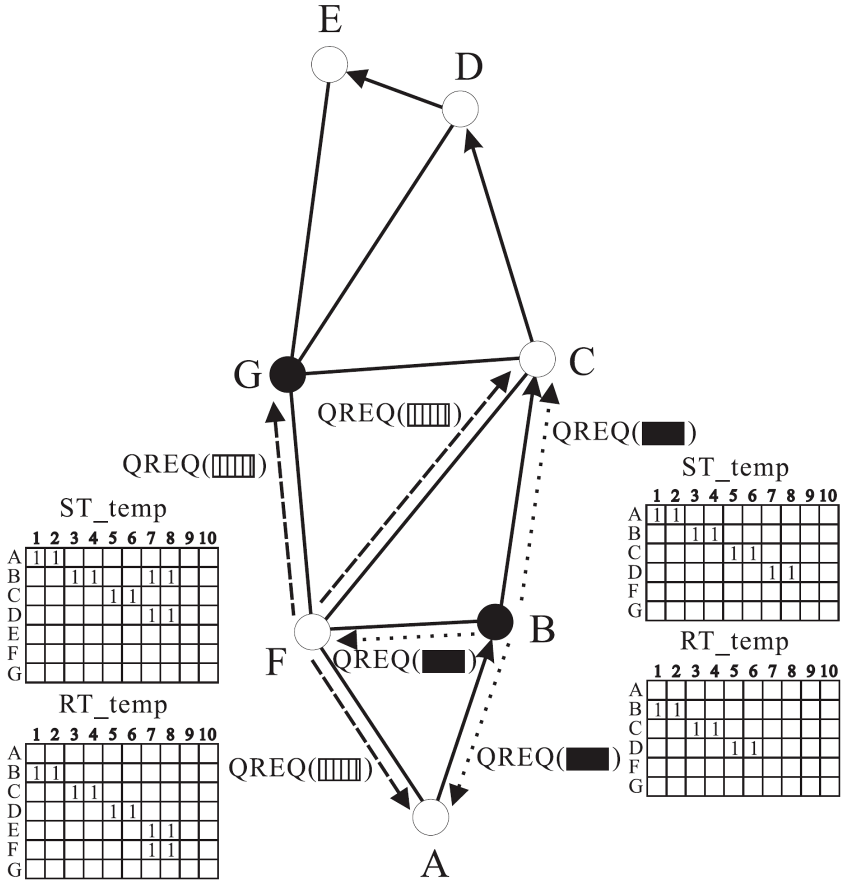
```
C1. for $j = i - 2$ to $i + 2$ do
        Let $ST_x[h_j, t]=1$ for each time slot $t$ in $l_j$.
    end for

C2. for $j = i - 2$ to $i + 2$ do
        Let $RT_x[h_j, t]=1$ for each time slot $t$ in $l_{j-1}$.
    end for
```

3.4 Time slot selection

The procedure *select_slot*(*y*, *z*, *b*, *ST_temp*, *RT_temp*) is for host *y* to choose *b* free time slots to send to *z*. It mainly relies on Lemma 1 to do the selection. Specifically, for each time slot *i*, $1 \leqq i \leqq s$, we check the following conditions D1, D2, and D3. If all conditions hold, slot *i* is a free slot that can be used by *y* to send to *z*.

D1. $(ST\_temp[y, i]=0) \wedge (RT\_temp[y, i]=0) \wedge (ST\_temp[z, i]=0) \wedge (RT\_temp[z, i]=0)$.
D2. $\forall w : (H_y[y, w] = 1) \Rightarrow RT\_temp[w, i]=0$.
D3. $\forall w : (H_y[z, w] = 1) \Rightarrow ST\_temp[w, i]=0$.

To respond the procedure call in A3, if there are at least *b* time slots satisfying the above conditions, we should return a list of *b* free slots to the caller; otherwise, an empty list ϕ should be returned. When there are more than *b* time slots available, we can further choose slots based on some priority. The basic idea is to increase channel reuse (which is generally favorable in almost all kinds of wireless communications). Those slots which have the exposed-terminal problem can be chosen with higher priority. To reflect this, we can give a legal time slot *i* a higher priority such that *ST_temp*[*w*, *i*]=1 for some neighbor *w* of *x*.



ST_temp

	1	2	3	4	5	6	7	8	9	10
A	1	1								
B			1	1			1	1		
C					1	1				
D							1	1		
E										
F										
G										

RT_temp

	1	2	3	4	5	6	7	8	9	10
A										
B	1	1								
C			1	1						
D					1	1				
E							1	1		
F							1	1		
G										

ST_temp

	1	2	3	4	5	6	7	8	9	10
A	1	1								
B			1	1						
C					1	1				
D							1	1		
F										
G										

RT_temp

	1	2	3	4	5	6	7	8	9	10
A										
B	1	1								
C			1	1						
D					1	1				
F										
G										

■ =(B, G, 1, 2, B, PATH=∅,
NH=((A, {7, 8}), (C, {9, 10}), (F, {7, 8})))

▥ =(B, G, 1, 2, F, PATH=(B, {7, 8}),
NH=((A, {9, 10}), (C, {9, 10}), (G, {9, 10})))

Fig. 6. An example of QREQ propagation in our protocal.

3.5 Example

Following the example in Fig. 5, we show in Fig. 6 how *B* searches for a route of bandwidth 2 slots to *G*. Since *B* is the source, the *ST_temp* and *RT_temp* are equal to ST_B and RT_B, respectively. Each of hosts *A*, *C*, and *F* can serve as the next hop by using slots {7, 8}, {9, 10}, and {7, 8}, respectively, as reflected in the packet content. We also show *F*'s *ST_temp* and *RT_temp* when searching for the next host. Hosts that can serve as the next hop of *F* are *A*, *C*, and *G*. The *QREQ* packets sent by other hosts are not shown for clarity. Finally, when *G* receives *F*'s *QREQ*, it may reply a *QREP*(*B*, *G*, 1, (*B*, {7, 8}), (*F*, {9, 10})) to *B*.

4. Experimental results

We have developed a simulator to evaluate the performance of the proposed bandwidth reservation scheme. A MANET in a $1000m \times 1000m$ area with 20 ~ 70 mobile hosts was simulated. Each mobile host had the same transmission range of 300 meters. Hosts might roam around continuously for 5 seconds, and then have a pose time from 0 ~ 8 seconds. The roaming speed is 0 ~ 20 m/s, with a roaming direction which was randomly chosen in every second. A data transmission rate of 11 Mbit/s was used. Each time frame had 16 ~ 32 time slots, with 5 ms for each time slot. Traffic was generated from randomly chosen source-destination pairs with bandwidth requirement of 1, 2, or 4 slots (denoted as QoS1, QoS2, and QoS4, respectively). New calls arrived with an exponential distribution of mean rate 1/12000 ~ 1/500 per ms. Each call had duration of 180 sec. Since our goal was to observe multi-hop communication, we impose a condition that each source-destination pair must be distanced by at least two hops. The total simulation time was 1000 sec.

We make observations from several aspects.

A) Network throughput: When calculating throughput, we only count packets that successfully arrive at their destinations. In Fig. 7, we show the network throughput under various loads, where load is defined to be the bandwidth requirement (which are 1, 2, and 4 for QoS1, QoS2, and QoS4, respectively) times the corresponding call arrival rate. Among the simulated ranges, the throughputs all increase linearly with respect to loads for all QoS types. This indicates that QoS routing can be supported quite well by MANET based on our protocol. As comparing different bandwidth requirements, QoS4 performs slightly worse than QoS1 and QoS2. The reason will be elaborated below.

To understand the above scenarios, we further investigate the call success rate (the probability to accept a new call) under the same inputs. The results are in Fig. 8 .When the traffic load increases, the success rates decrease for all QoS types. The success rate of QoS1 is the largest, which is followed by QoS2, and then QoS4. This is reasonable because larger bandwidth requirements are more difficult to satisfy.

Next, we investigate the average number of hops for all source-destination pairs under different bandwidth requirements. The result is in Fig. 9. We see that QoS4 routes are the shortest in all ranges. One interesting thing is that when the traffic load is higher than 1/1000, the lengths of QoS1 routes will start to increase, while on the contrary those of QoS4 routes will drop significantly. The reason is that it is less likely to find satisfactory, but long QoS4 routes under heavy load. But for QoS1 routes, the chances are actually higher. This is why QoS1 gives the best network throughput.

B) Effect of host density: In this experiment, we vary the total number of hosts. Since the physical area is fixed, this actually reflects the host density (or crowdedness of the environment). The result is in Fig. 10. First, we observe that the network throughput will improve as the network is denser under all QoS types. This is perhaps due to richer choices of routing paths. Second, there will be larger performance gaps between low QoS routes (such as 1 and 2) and high QoS routes (such as 4). So higher host density is more beneficial to low-bandwidth routes.

C) Effect of host mobility: In Fig. 11, we show the throughput under various host mobility. We see that throughput is very sensitive to mobility in all QoS types. In our simulation, whenever a route is broken, an error message will be sent to the source host. Before the source host knows this fact, all packets already sent will still consume time slots without contributing to the real throughput. Furthermore, before a new route is discovered, some time slots will be idle. This is why we see significant drop on throughput as mobility increases, which also indicates a challenging problem deserving further research.

D) Effect of frame length: In Fig. 12, we show the network throughput when a time frame has 16, 24, and 32 time slots. Longer frame length will be more beneficial to requests with higher QoS requirements. This is reasonable because requests with larger QoS requirements get rejected with higher probability as the frame length is shorter.

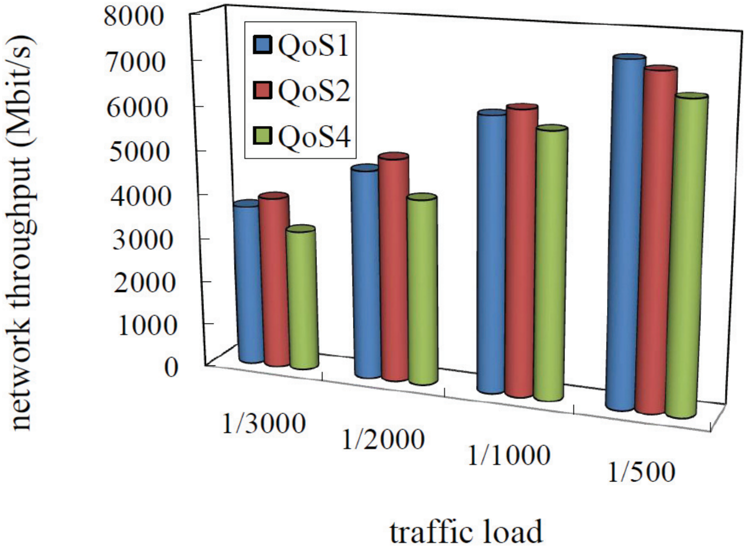


Fig. 7. Network throughput vs. traffic load (= QoS requirement times call arrival rate), where number of hosts=30, number of time slots=16, pose time=0, and mobility=4m/s.

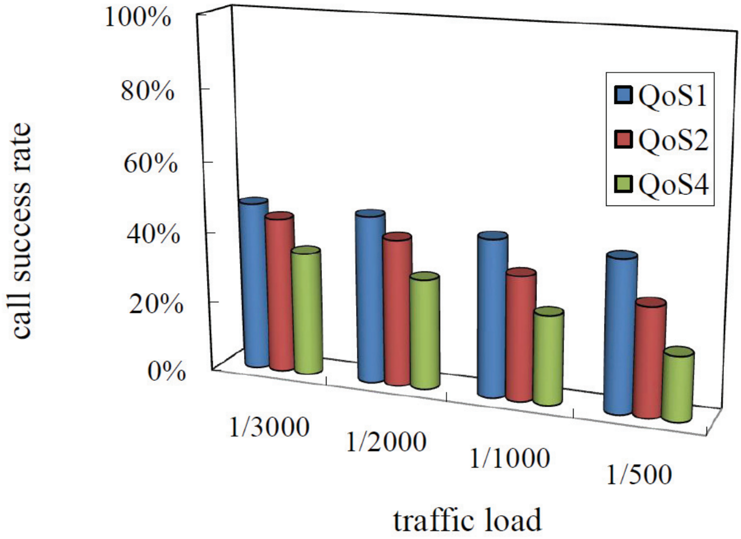


Fig. 8. Call success rate vs. traffic load, where number of hosts=30, number of time slots=16, pose time=0, and mobility=4m/s.

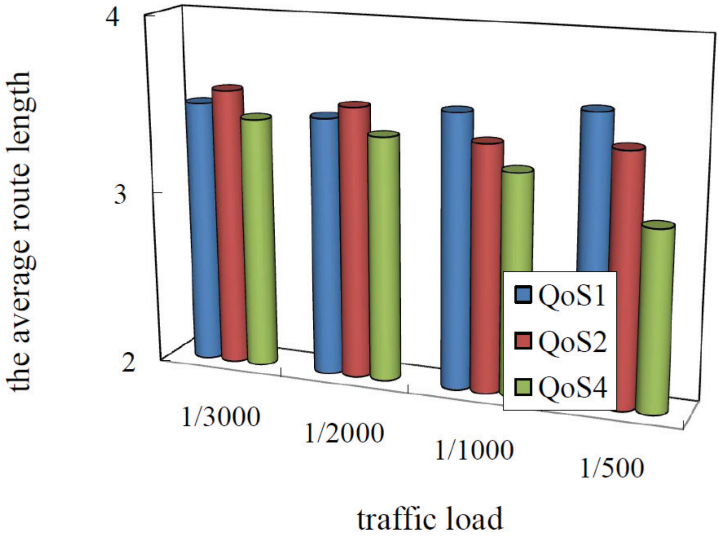


Fig. 9. The average route length v.s. traffic load, where number of hosts=30, number of time slots=16, pose time=0, and mobility=4m/s.

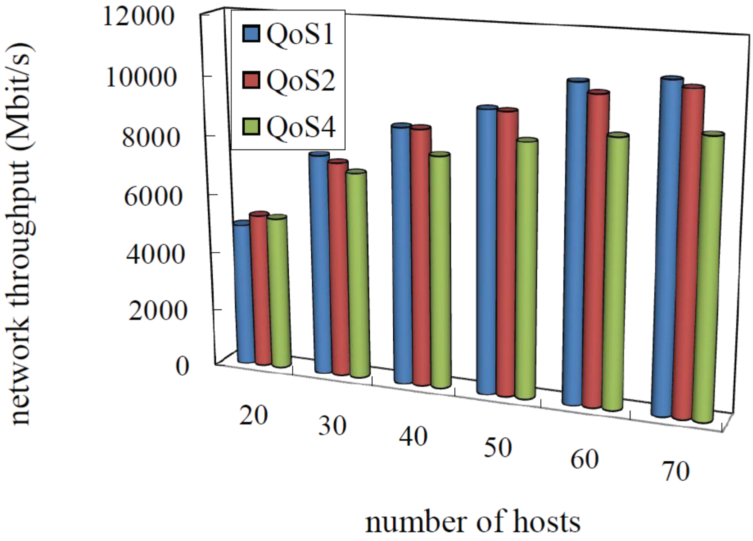


Fig. 10. Network throughput v.s. host density, where traffic load=1/500, number of time slots=16, pose time=0, and mobility=4m/s.

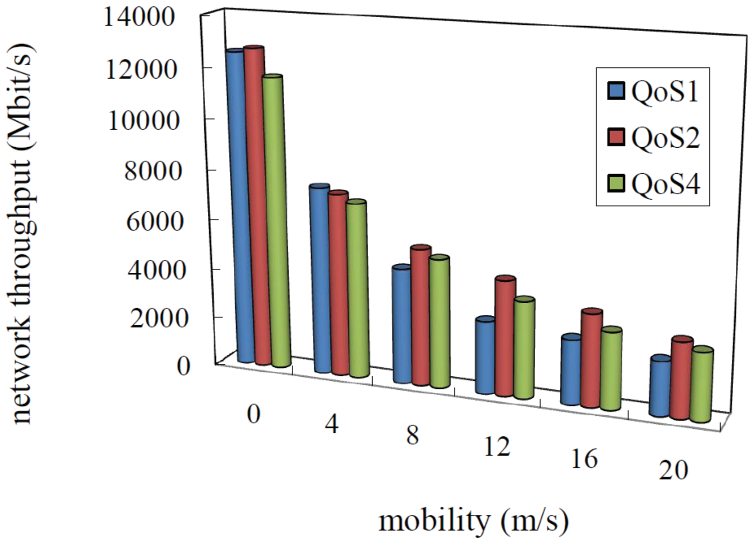


Fig. 11. Network throughput v.s. mobility, where number of hosts=30, number of time slots=16, pose time=0, and traffic load=1/500.

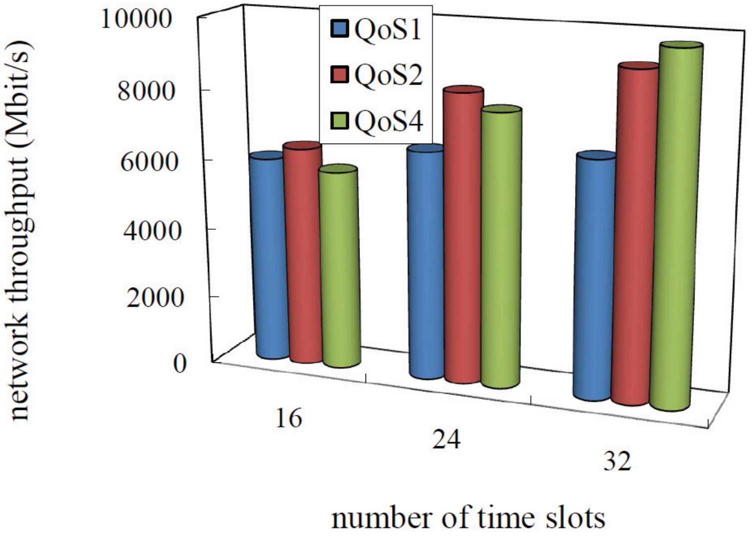


Fig. 12. Network throughput v.s. frame length, where number of hosts=30, pose time=0, mobility=4m/s, and traffic load=1/1000.

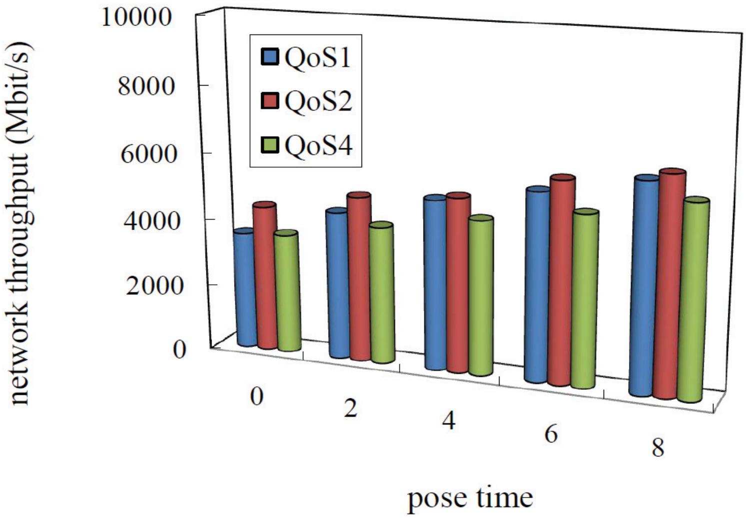


Fig. 13. Network throughput v.s. pose time, where number of hosts=30, number of time slots=16, mobility=8m/s, and traffic load=1/1000.

E) Effect of pose time: Recall that we adopt a roaming model that a host will continue move for 5 seconds, and then pose for 0 to 8 seconds. In Fig. 13, we show the network throughput under various pose times. Longer pose time is beneficial for all types of QoS routes, which is reasonable because the probability of route broken will drop.

5. Conclusions

In this paper, we have proposed a TDMA-based bandwidth reservation protocol for QoS routing in a MANET. Most existing MANET routing protocols do not guarantee bandwidth when searching for routes. Few works have considered the same QoS routing problem, but are under a stronger multi-antenna model or a less stronger CDMA-over-TDMA channel model. Our protocol assumes a simpler (and perhaps more practical) TDMA-based channel model. One single common channel is assumed to be shared by all hosts in the MANET. Hence the result may be applied immediately to current wireless LAN cards. One interesting point is that our protocol can take into account the difficult hidden-terminal and exposed-terminal problems when establishing a route. So more accurate route bandwidth can be calculated and the precious wireless bandwidth can be better utilized. We are currently trying to further optimize the bandwidth utilization from a global view.

6. References

Haas, Z. J. & Pearlman, M. R. (1998). The Zone Routing Protocol (ZRP) for Ad-Hoc Networks, *Internet draft*, August, 1998.

Johnson, D. B.; Maltz, D. A.; Hu, Y.-C. & Jetcheva, J. G. (2001). The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks, *Internet draft*, November, 2001.

Liao, W.-H.; Tseng, Y.-C. & Sheu, J.-P. (2001). GRID: A Fully Location-Aware Routing Protocol for Mobile Ad Hoc Networks, *Telecommunication Systems*, Vol. 18, No. 1-3, pp. 37-60, 2001.

Perkins, C. & Bhagwat, P. (1994). Highly Dynamic Destination-Sequenced Distance-Vector (DSDV) Routing for Mobile Computers, *ACM SIGCOMM Symposium on Communications, Architectures and Protocols*, 1994.

Perkins, C.; Royer, E. M. & Das, S. R. (2002). Ad Hoc On Demand Distance Vector (AODV) Routing, *Internet draft*, January, 2002.

Royer, E. M. & Toh, C.-K. (1999). A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks, *IEEE Personal Communications*, Vol. 6, No. 2, pp. 46-55, April, 1999.

Wu, J. & Li, H. (2001). A Dominating-Set-Based Routing Scheme in Ad Hoc Wireless Networks, *Telecommunication Systems*, Vol. 18, No. 1, pp. 13-36, 2001.

Chen, S. & Nahrstedt, K. (1999). Distributed Quality-of-Service Routing in Ad Hoc Networks, *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, pp. 1488-1505, August, 1999.

Liao, W.-H.; Tseng, Y.-C.; Wang, S.-L. & Sheu, J.-P. (2002). A Multi-Path QoS Routing Protocol in a Wireless Mobile Ad Hoc Network, *Telecommunication Systems*, Vol. 19, No. 3, pp. 329-347, 2002.

Lin, C.-R. (2001). On-Demand QoS Routing in Multihop Mobile Networks, *IEEE INFOCOM*, 2001.

Lin, C.-R. & Liu, J.–S. (1999). QoS Routing in Ad Hoc Wireless Networks, *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, pp. 1426-1438, August, 1999.

Stojmenovic, I.; Russell, M. & Vukojevic, B. (2000). Depth First Search and Location Based Localized Routing and QoS Routing in Wireless Networks, *International Conference on Parallel Processing*, 2000.

Chalmers, D. & Sloman, M. (1999). A Survey of Quality of Service in Mobile Computing Environments, *IEEE Communications Surveys*, Vol. 2, No. 2, pp. 2-10, April, 1999.

Wang, Z. & Crowcroft, J. (1996). Quality-of-Service Routing for Supporting Multimedia Applications, *IEEE Journal on Selected Areas in Communications*, Vol. 14, No. 7, pp. 1228-1234, September,1996.

Sobrinho, J. L. & Krishnakumar A. S. (1999). Quality-of-Service in Ad Hoc Carrier Sense Multiple Access Wireless Networks, *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, pp. 1353-1368, August, 1999.

Ni, S.-Y.; Tseng, Y.-C.; Chen, Y.-S. & Sheu, J.-P. (1999). The Broadcast Storm Problem in a Mobile Ad Hoc Network, *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99)*, 1999.

Bertossi, A. & Bonuccelli, M. (1995). Code Assignment for Hidden Terminal Interference Avoidance in Multihop Radio Networks, *IEEE/ACM Transation on Networks*, Vol. 3, No. 4, pp. 441-449, August, 1995.

Garcia-Luna-Aceves, J. J. & Raju, J. (1997). Distributed Assignment of Codes for Multihop Packet-Radio Networks, *IEEE MILCOM '97*, 1997.

Ju, J.-H. & Li, V. O. K. (1999). TDMA Scheduling Dedsign of Multihop Packet Radio Networks Based on Latin Squares, *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, pp. 1345-1352, August, 1999.

IEEE Std 802.11–1997：Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, *Institute of Electrical and Electronics Engineers, Inc.*, 1997.

Marina, M. K. & Das, S. R. (2001). Performance of Route Caching Strategies in Dynamic Source Routing, *IEEE Wireless Networking and Mobile Computing (WNMC)*, 2001.

11

Link Quality Aware Robust Routing for Mobile Multihop Ad Hoc Networks

Sangman Moh, Moonsoo Kang, and Ilyong Chung
Chosun University
South Korea

1. Introduction

A *mobile ad hoc network* (*MANET*) (Perkins, 2001; Siva Ram Murthy & Manoj, 2004; IETF, 2009) is a collection of mobile nodes without any fixed infrastructure or any form of centralized administration. In other words, it is a temporary network of mobile nodes without existing communication infrastructure such as access points or base stations. In such a network, each node plays a router for multihop routing as well. MANETs can be effectively applied to military battlefields, emergency disaster relief, and other application-specific areas including wireless sensor networks and vehicular ad hoc networks.

In mobile ad hoc networks, interference and noise are two major obstacles in realizing their full potential capability in delivering signals. In wireless links, the signal propagation is affected by path loss, shadowing and multi-path fading, and dynamic interferences generate additional noise from time to time degrading *link quality*. In this study, as an effective and practical metric of link quality, *signal-to-interference plus noise ratio* (*SINR*) is used because it takes interference and noise as well as signal strength into account. Note that SINR is measurable with no additional support at the receiver (Krco & Dupcinov, 2003; Zhao et al., 2005). Furthermore, as nodes are fast moving, poor links are unpredictably increased. Actually, it is shown that the communication quality of mobile ad hoc networks is low and users can experience strong fluctuation in link quality in practical operation environments (Gaertner & Cahill, 2004). In particular, sending real-time multimedia over mobile ad hoc networks is more challenging because it is very sensitive for packet loss and the networks are error prone due to node mobility and weak links (Karlsson et al., 2005). Accordingly, it is very important to include as many high-quality links as possible in a routing path. Also, the dynamic behavior of link quality should be taken into consideration in protocol design.

In the IEEE 802.11 MAC (IEEE, 1999), *broadcast packets* are transmitted at the base data rate of 1 Mbps. It is mainly due to the potential demand that a broadcast packet should cover as large area as possible in the wireless LAN environment. Note here that, given radio hardware and transmit power, the transmission range is affected by the transmit rate. In mobile ad hoc networks, the *route request* (*RREQ*) packet in routing protocols is a broadcast packet. Therefore, if a distant node receiving an RREQ rebroadcasts the RREQ, a long weak link with low data rate can be included in the discovered route. Intuitively, this helps the routing protocols to find out the *minimum hop-count route* from source to destination. Note here that the minimum hop-count route is a routing path with the minimum number of hops from source to destination and sometimes called the shortest path in the viewpoint of graph algorithm. However, such long links are relatively weak and unreliable and increase the possibility that they are broken. That is, the minimum hop-count route does not mean the best route as measured in (De Couto et. al., 2002; De Couto et. al., 2003). Furthermore, as an effort, SINR-based design of optimized link state routing was introduced for scenarios where VoIP (Voice over IP) traffic is carried over a static multihop networks (Kortebi et al., 2007). In our study, in order to find out a robust route for high delivery efficiency and network performance in MANETs, strong links are selected by examining link quality (or SINR) instead of the number of hops.

This paper proposes a *link quality aware routing protocol* for MANETs resulting in robust delivery and high throughput by finding out a robust route with strong links. During route discovery, the strong links are effectively exploited by forwarding the RREQ packet with the highest SINR among the multiple RREQ packets received. In case there are RREQ packets within δ dB ($\delta = 1$ in this study) from the highest SINR, the first-arrived one among them is chosen to cope with the dynamic behavior of SINR. Any node that has received an RREQ receives successive RREQ packets until the predetermined RREQ waiting time expires; afterwards, RREQ packets for the route discovery are ignored. Compared to the conventional protocols such as AODV, in which only the first-arrived RREQ is forwarded and the others are ignored, the proposed scheme may not have the minimum hop-count route but the one with more number of hops (links). However, the found route is a reliable path with high data rate because it consists of strong links, resulting in high performance as well as robust routing. For performance study, in this paper, the *link quality aware AODV* (*LA-AODV*) is implemented in ns-2 (NS-2, 2008; CMU, 2008). For practical system simulation, we introduce a realistic reception model that takes *BER* and *frame error rate* (*FER*) into account instead of the deterministic reception model in the ns-2 network simulator. Note that the deterministic reception model in ns-2 is based on three fixed thresholds such as carrier sense, receive and capture thresholds (NS-2, 2008; CMU, 2008). According to our performance study, it is shown that *packet delivery ratio* is improved by up to 70% and *per-route goodput* is dramatically increased by a factor of up to 12. It is also shown that the acceptable value of the *RREQ waiting time* (T_w) is 1 msec in the simulated environment, which is enough to achieve fairly good performance.

The rest of the paper is organized as follows: As preliminaries for this study, the basic AODV routing protocol and the rate adaptation mechanisms are summarized in the following section. Section 3 presents the proposed link quality aware routing; i.e., the RREQ forwarding algorithm and the robust routing protocol LA-AODV are described, and then the impact of link quality is analyzed. Performance study including reception model, simulation environment, and evaluation results is discussed in Section 4. Finally, conclusions are given in Section 5.

2. Preliminaries

In this section, the ad hoc on-demand distance vector (AODV) routing protocol (Perkins et al., 2003; Belding-Royer & Perkins, 2003), which is a representative routing protocol for MANETs, is briefly overviewed. Then, the rate adaptation mechanisms to exploit as high transmission rate as possible are summarized.

2.1 AODV routing

The AODV routing protocol (Perkins et al., 2003; Belding-Royer & Perkins, 2003) is an on-demand routing protocol based on the DSDV protocol (Perkins & Watson, 1994). The main characteristics of AODV are to use the periodic beaconing for neighbor sensing and sequence numbering procedure of DSDV and a flooding-based route discovery procedure.

In AODV, route discovery works as follows: Whenever a source needs a route to a destination, it first checks whether it has a route in its route cache (routing table). If it does not have a route, it initiates a route discovery by flooding a route request (RREQ) packet for the destination in the network and, then, waits for a route reply (RREP) packet. When an intermediate node receives the first copy of an RREQ, it sets up a reverse path to the source using the previous hop of RREQ as the next hop on the reverse path. In addition, if there is a valid route available for the destination, it unicasts an RREP back to the source via the reverse path; otherwise, it rebroadcasts RREQ. Duplicate copies of RREQ are immediately discarded upon reception at every node. The destination on receiving the first copy of an RREQ forms a reverse path in the same way as intermediate nodes, and it also unicasts an RREP back to the source along the reverse path. As RREP proceeds towards the source, it establishes a forward path to the destination at each hop. Note here that the destination generates RREPs only when its destination sequence number is grater than or equal to the destination sequence number of the RREQ received.

Route maintenance is done by means of route error (RERR) packets. When an intermediate node detects a link failure (*e.g.,* via a link-layer feedback), it generates an RERR. RERR propagates towards all sources having a route via the failed link, and erases all broken routes on the way. A source upon receiving RERR initiates a new route discovery if it still needs the route. Apart from this route maintenance mechanism, AODV also has a timer-based mechanism to purge stale routes.

2.2 Rate adaptation

As a wireless channel is time-varying and location-dependent due to path loss, shadowing and small-scale fading as well as interference, rate adaptation is a powerful way to overcome channel variations (Zhai et al., 2006). For example, IEEE 802.11b standard incorporates physical-layer multi-rate capability, the feasible data rate set of which is 1, 2, 5.5 and 11 Mbps. However, the IEEE 802.11 standards do not specify how to choose the data rate based on varying channel conditions and thus some schemes to select the rate adaptively have been proposed.

The auto rate fallback (ARF) protocol (Kamerman & Monteban, 1997) is the first commercial MAC that utilizes rate adaptation. Each sender attempts to use higher transmission rate after consecutive transmission successes at a given rate and revert to a lower rate after 1 or 2 consecutive failures. A timer is reset and started each time the rate is changed. When either the timer expires or the number of successfully received acknowledgements reaches the threshold of 10, the rate is increased. The first transmission after the rate increase must succeed or the rate is immediately decreased. When two consecutive transmissions fail in a row, the current rate is decreased. However, if the channel conditions change very quickly due to fast multipath fading, ARF cannot adapt effectively. The adaptive ARF (AARF) protocol (Lacage et al., 2004) continuously changes the threshold at runtime to better reflect the channel conditions. When the transmission of the probing frame fails, the data rate is switched back immediately and the threshold is doubled. The threshold is reset to its initial value of 10 when the rate is decreased due to two consecutive failed transmissions. However, AARF still cannot take the frame loss due to collisions over the wireless link into consideration. The loss-differentiating ARF (LD-ARF) protocol (Pang, 2005) effectively adapts to collision losses as well as link error losses. The data rate is reduced only when a loss of data frame is caused by link errors, not by collisions. Note that it is assumed that if the CTS frame is not received, most likely a collision has occurred because RTS and CTS are short and usually transmitted at a base rate of 1 Mbps.

In the receiver based auto rate (RBAR) protocol (Holland et al., 2001), each receiver measures the channel quality (SINR) of the received RTS frame and, then, selects the transmission rate to be used by the upcoming CTS, data, and acknowledgement frames according to the highest achievable value based on the SINR. The rate to use is then sent back to the sender in the CTS frame. Note that the sender chooses a data rate for RTS based on some heuristic or sets it at a base rate of 1 Mbps. To allow all the nodes within the transmission range to correctly update their network allocation vector (NAV), the RTS, CTS, and data frames have to contain information on the size and rate of the data transmission. If a node that heard the RTS frame hears the data frame, it should recalculate the reservation duration and update its NAV correctly. Since the channel quality is evaluated just before data packet transmission, RBAR yields significant throughput gain compared to ARF. In RBAR, only one packet is allowed to transmit each time, which is not efficient especially when the channel condition is good for a long time. To better exploit the duration of high-quality channel condition, the opportunistic auto rate (OSR) protocol (Sadeghi et al., 2002) opportunistically sends multiple back-to-back data packets whenever the channel quality is good. It achieves significant throughput gains compared to RBAR. In the opportunistic packet scheduling and auto rate (OSAR) protocol (Wang et al., 2004), a sender multicasts RTS to a group of candidate receivers simultaneously and, then, a receiver with channel quality better than a certain level replies CTS. If there are more than one candidate receivers with good channel condition, a coordinating rule is applied in a distributed fashion to avoid collision.

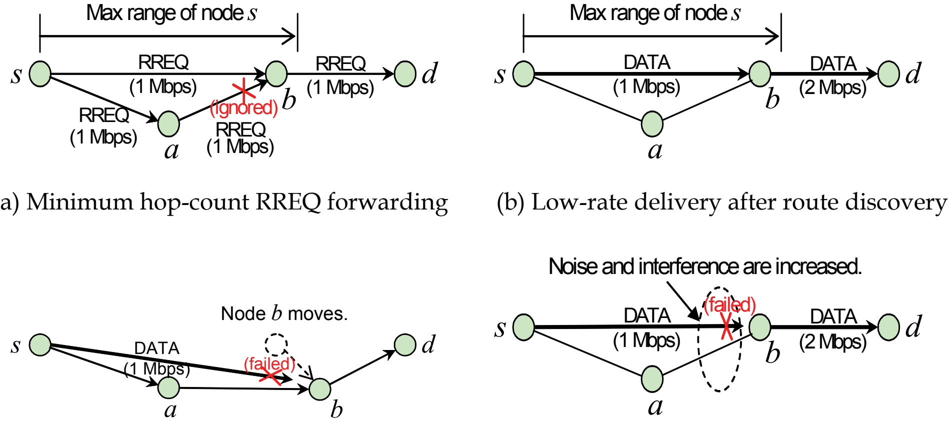
As in (Zhao et al., 2005), we implement a SINR-based rate adaptation scheme in ns-2 (NS-2, 2008; CMU, 2008). The scheme is based on RBAR (Holland et al., 2001), and the data rate of RTS is set at a base rate of 1 Mbps to safely cope with dynamically changing link quality in MANETs. Such a rate adaptation is effectively utilized in our link quality aware routing protocol which will be presented in Section 3.

3. Link quality aware routing

The proposed link quality aware routing protocol, which finds out a robust route with strong links during route discovery, is presented and discussed in this section. The key idea of finding out a robust route is to forward the *RREQ packet* with the *highest SINR* among multiple RREQ packets received. In case there are multiple RREQ packets within δ dB from the highest SINR, the first-arrived one among them is chosen to cope with the dynamic behavior of SINR. The RREQ forwarding algorithm is presented first and then the link quality aware AODV (LA-AODV) is followed. The route reliability and throughput are analyzed in terms of link quality or SINR.

3.1 RREQ forwarding algorithm

In the conventional routing protocols such as AODV, the intermediate nodes forward only the first-arrived RREQ during route discovery in order to find out the minimum hop-count route even though the route does not mean the best route as measured in (De Couto et. al., 2002; De Couto et. al., 2003). This results in a fragile route with long, weak and unreliable links. In this subsection, a new *RREQ forwarding algorithm* is presented to find out a robust and high-performance route.



(a) Minimum hop-count RREQ forwarding

(b) Low-rate delivery after route discovery

(c) Delivery failure after node *b* moves

(d) Delivery failure when noise increases

Fig. 1. Minimum hop-count RREQ forwarding and its possible problems.

Fig. 1 shows the minimum hop-count RREQ forwarding and its possible problems in the conventional routing protocols such as AODV. Since the first-arrived RREQ is forwarded and the others are ignored, node *b* receives the RREQ packet directly come from *s* and forwards it, resulting in a routing path <*s, b, d*> with two hops as shown in Fig. 1(a). The RREQ packet come from node *a* is ignored at node *b* because it arrives later. Once a route is discovered, subsequent data delivery is done through the route as shown in Fig. 1(b), but the throughput is 1 Mbps because the weak link <*s, b*> in the route limits the data rate to the base rate of 1 Mbps. On the other hand, if node *b* moves and exists out of the maximum range of node *s* as shown in Fig. 1(c), it does not receive data packets from node *s* any more, resulting in delivery failure and initiating a new route discovery. The effect of mobility changes the received signal power, which is exponentially decreased as the communication distance increases, and thus affects SINR. Fig. 1(d) shows another example of delivery failure. If interference and noise on the link <*s, b*> are increased due to unstable and dynamic network environment, SINR of the packet transmitted from node *s* becomes less than the threshold (*e.g.*, 10 dB) and, thus, node *b* does not receive the packet successfully even though it does not move. The interference and noise are influenced by unstable and dynamic network environment and unexpectedly changes from time to time, and thus affects SINR. As explained earlier, the weak point of the conventional routing protocols, which is got over in this paper, is the *RREQ forwarding algorithm* in which the intermediate nodes forward the first-arrived RREQ to find out the minimum hop-count route even though the route does not mean the best route as measured in (De Couto et. al., 2002; De Couto et. al., 2003).

In the proposed LA-AODV protocol, the route discovery and maintenance are necessary as in the basic AODV. The main difference between AODV and LA-AODV is RREQ forwarding during route discovery. Fig. 2 represents the proposed *RREQ forwarding algorithm*. The new RREQ forwarding algorithm helps find out a reliable route with strong links. When a node has a packet to send, it needs a route to the destination. If it has no route in its route cache or routing table, it issues route discovery by broadcasting an RREQ packet

```
// RREQ forwarding procedure at every node
/* This algorithm is carried out during route discovery at every node that receives an RREQ packet:
   i.e., if a node receives an RREQ packet, this routine is immediately called and run by the node.
*/
1:  $S = \{R_1\}$;  // keep track of received RREQs (including link quality or SINR).
2:              // subscript i in set element $R_i$ represents the order of receipt
3:  set the timer as $T_w$;  // initialize the timer to $T_w$
4:  while the timer does not reach 0, do {   // repeat lines 4~7 until the timer reaches 0
5:        // receives successive RREQs until the predetermined RREQ waiting time expires
6:        if any successive RREQ arrives, append it into $S$;
7:  }
8:  $k = |S|$;  // number of elements in $S$
9:  if $k = 1$, forward $R_1$;
10:  else{   // if there are two or more RREQs received
11:       sort $S$ in decreasing (non-increasing) order of SINR;
12:       if there are one or more RREQs within $\delta$dB from the highest SINR in $S$ { // $\delta$ =1 in this study
13:              // for coping with the dynamic behavior of SINR
14:              select the first-arrived one among them;
15:              forward the selected one;
16:       }
17:       else    forward the RREQ with the highest SINR;
18:  }
19:  return;  // afterwards, RREQ packets are ignored
```

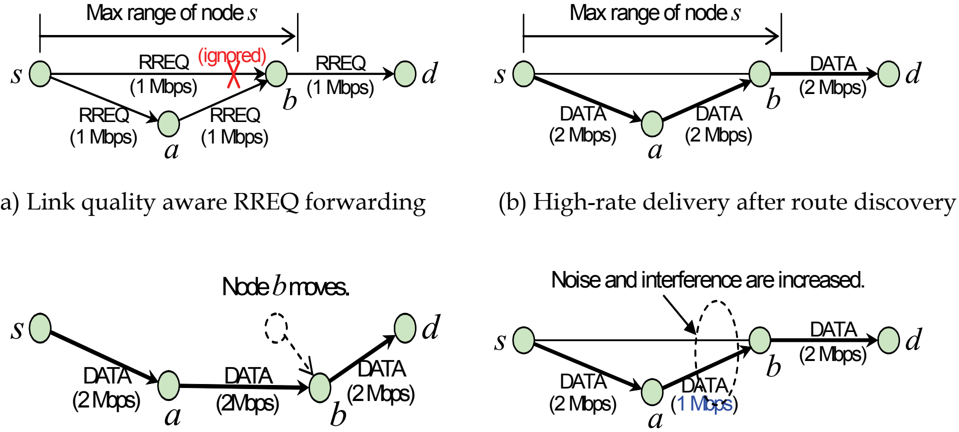
Fig. 2. Proposed RREQ forwarding algorithm.

for the destination. Intermediate nodes forward the RREQ packet with the highest SINR among multiple RREQ packets received for the predetermined *RREQ waiting time* (T_w) after the first RREQ is received. In case there are multiple RREQ packets within δ dB ($\delta = 1$ in this study) from the highest SINR, the first-arrived one among them is chosen to cope with the dynamic behavior of SINR. The other RREQ packets arrived later are ignored if any. Similarly, the destination takes the RREQ packet with the highest SINR for route reply.

3.2 Link quality aware end-to-end routing

Based on the RREQ forwarding algorithm, the *link quality aware AODV* (*LA-AODV*) routing protocol is presented and discussed in this subsection. Since the RREQ forwarding algorithm finds out a robust route with strong links, the proposed LA-AODV results in robust delivery and high performance. Note that the route discovery operation of LA-AODV differs from that of AODV but there is no noticeable difference in the route maintenance. Accordingly, LA-AODV can be easily implemented.

Fig. 3 shows the proposed link quality aware RREQ forwarding and its resulting effects for the same example as in Fig. 1. During route discovery, node *b* forwards the RREQ packet come from node *a* rather than that come from node *s* as shown in Fig. 3(a) because the former has the better link quality (*i.e.,* higher SINR) than the latter. Notice that, in the



(a) Link quality aware RREQ forwarding

(b) High-rate delivery after route discovery

(c) Data delivery after node *b* moves

(d) Data delivery when noise increases

Fig. 3. Link quality aware RREQ forwarding and its resulting effects for the same example as in Fig. 1.

proposed *RREQ forwarding algorithm*, the intermediate nodes forward the RREQ packet with the *highest SINR* among multiple RREQ packets received for the predetermined RREQ waiting time after the first RREQ is received. In case there are RREQ packets within δ dB ($\delta = 1$ in this study) from the highest SINR, the first-arrived one among them is chosen to cope with the dynamic behavior of SINR. Fig. 3(b) shows data delivery after route discovery, in which data is delivered at 2 Mbps along with 3 hops. That is, the throughput of the route is 2 Mbps, which is double of 1 Mbps in the conventional protocols as shown in Fig. 1(b), because strong links <*s, a*> and <*a, b*> instead of the weak link <*s, b*> are exploited in the proposed RREQ forwarding algorithm. Even when node *b* moves as in Fig. 3(c), the data delivery is successful with the same throughput of 2 Mbps without performance degradation. If node *b* moves further away from node *a* or node *d*, the throughput might be reduced but still the route may be alive. Fig. 3(d) shows another example of data delivery in case of unstable and dynamic network environment. If interference and noise are increased resulting in link quality fluctuation, SINR of the packet transmitted from node *a* is reduced but the link <*a, b*> is strong enough to receive the packet without error and, thus, node *b* can still receive the packet successfully at lower data rate (*e.g.,* 1 Mbps). Note here that the transmission data rate is decreased (*i.e.,* from 2 Mbps to 1 Mbps in the figure) because SINR is reduced due to the increased interference and noise on the link <*a, b*>. Conclusively, the proposed approach achieves high throughput as well as robust delivery by exploiting strong links during route discovery.

In the conventional protocols such as AODV, only the first-arrived RREQ is forwarded and the others are ignored. The rationale for such design is that it finds out the shortest path (*i.e.,* the minimum hop-count route) because the first arrival means the smaller number of hops from the source. That is, to discover the minimum hop-count route is the primary goal of the conventional protocols as in the most wired networks. As described in Introduction, however, the minimum hop-count route does not mean the best route as measured in (De Couto et. al., 2002; De Couto et. al., 2003). On the other hand, the proposed approach might not have the minimum hop-count route but the one with more number of hops (links). However, the found route in the proposed LA-AODV is a reliable path with high data rate because it consists of strong links, resulting in high throughput as well as robust routing.

Obviously, a routing path with strong links is more reliable and has higher quality compared to that with weak links. It significantly extends the lifetime of a routing path, reducing *route discovery frequency*. Moreover, a high-quality link transmits packets at high data rate. Therefore, the proposed LA-AODV results in higher *packet delivery ratio* and higher *throughput* as well as more robust routing compared to AODV. In the proposed protocols, the *RREQ waiting time* is a critical design factor because it directly determines the amount of overhead affecting the route discovery time. Even though the overhead of the RREQ waiting time is a minor factor compared to the positive effects of finding out a robust routing path, it should be optimized to eliminate unnecessary operations. In Section 4, some different RREQ waiting time is applied to performance simulation in order to investigate the performance impact of the RREQ waiting time.

Note that LA-AODV is the same as AODV except for that the new RREQ forwarding algorithm presented earlier is used instead of the first-arrived RREQ forwarding used in AODV and DSR during route discovery. Therefore, LA-AODV protocol can be easily implemented by redesigning only the RREQ forwarding module in AODV and tuning some related modules appropriately. Note that the proposed RREQ forwarding algorithm is feasible since SINR is measurable with no additional support at the receiver (Krco & Dupcinov, 2003; Zhao et al., 2005). In this paper, the *link quality-aware AODV* (*LA-AODV*) routing protocol, which is the modified version of AODV (Perkins et al., 2003; Belding-Royer & Perkins, 2003), is implemented in *ns*-2 (NS-2, 2008; CMU, 2008) and its performance is evaluated and compared with the conventional routing protocols of AODV in Section 4.

3.3 Analysis on impact of link quality

For a multi-hop route, the impact of link quality is analyzed in this subsection. The route reliability and throughput are discussed in terms of link quality or SINR. In general, the link quality can be represented by signal strength, signal-to-noise ratio (SNR), or SINR. In our study, SINR is used as the metric of link quality because it takes all the signal strength, interference and noise into account. Note that SINR directly affects bit error rate (BER) which determines the probability that a packet is successfully transferred. Given a modulation method, BER is inversely proportional to SINR. How to calculate SINR and a typical example of SINR-BER curve will be given in Section 4.1.

Given a k-hop route R from source to destination in a mobile ad hoc network, the probability P_R that a packet is successfully delivered along with R can be represented by

$$P_R = \prod_{i=1}^{k} p_i \tag{1}$$

where p_i is the probability that a packet is successfully transferred via the i-th link in R. Note here that the data rate is fixed and the same for all the k links in R. When p_i is relatively low, P_R is quickly decreased as the number of hops in a route increases. Therefore, p_i needs to be as high as possible to provide scalability. In other words, a route with strong links is highly required to obtain a reliable route of high P_R. Note that P_R and p_i are *reliability* of R and the i-th link in R, respectively. P_R is often called *packet delivery ratio*.

On the other hand, the end-to-end throughput λ_R of a k-hop route R is calculated by using geometric mean. Note that geometric mean is used if the product of the observations is a quantity of interest. Therefore, λ_R can be simply given by

$$\lambda_R = (\prod_{i=1}^{k} \lambda_i)^{\frac{1}{k}} \tag{2}$$

where λ_i is the throughput or data rate of the i-th link in R. Note here that the data rate (λ_i) is directly correlated to the link quality (p_i). To attain high end-to-end throughput, every link of a route has to transmit frames at high data rate. To achieve high data rate for a link, the link need to be as strong as possible.

In summary, the reliability and throughput can be significantly improved by exploiting strong links during route discovery. The more strong links are taken, the better reliability and throughput are attained. In this paper, per-route goodput is evaluated via extensive simulation instead of throughput in the next section because goodput is more practical and application oriented than throughput.

4. Performance evaluation

In this section, the performance of the proposed link quality aware AODV (LA-AODV) is evaluated in comparison to the normal AODV using the *ns*-2 network simulator (NS-2, 2008; CMU, 2008). Section 4.1 introduces the realistic reception model we have used in this study and Section 4.2 explains the simulation environment including parameters. Simulation results are discussed in Section 4.3.

4.1 Reception model

The reception model implemented in the *ns*-2 network simulator (NS-2, 2008; CMU, 2008) is based on three fixed thresholds, *i.e., carrier sense threshold* (CSThresh), *receive threshold* (RxThresh) and *capture threshold* (CPThresh). When a frame is received, each node in the proximity calculates the received signal power based on radio propagation model and compares it against CSThresh and RxThresh. If it is smaller than CSThresh, the receiver ignores the signal. If it is in between the two thresholds, the receiver considers the medium busy but do not attempt to decode the signal. If it is higher than RXThresh, the receiver attempts to receive the frame. However, when the node receives another signal during receiving the first signal, their ratio is compared against CPThresh. If one of them is much stronger (*e.g.,* 10 dB higher), it captures the other; otherwise, both frames fail. However, real wireless links are characterized with random and probabilistic behavior.

Even though the abovementioned deterministic reception model is not realistic, it has been used in most simulation studies for simple comparison. For the realistic evaluation of wireless links with probabilistic behavior, however, it is important to simulate a realistic reception model. Our evaluation takes *bit error rate* (*BER*) into consideration in the context of *ns*-2 because BER is a function of SINR and modulation method (Pavon & Choi, 2003). In other words, given a modulation method, BER is inversely proportional to SINR.

Here, we describe how SINR is calculated in *ns*-2 (NS-2, 2008; CMU, 2008). While the receiver receives one signal, other signals may arrive at the receiver resulting in interference. As a result, SINR of the receiving signal, γ, is calculated by

$$\gamma = \frac{P_r}{\sum_{i \neq r} P_i + N} \tag{3}$$

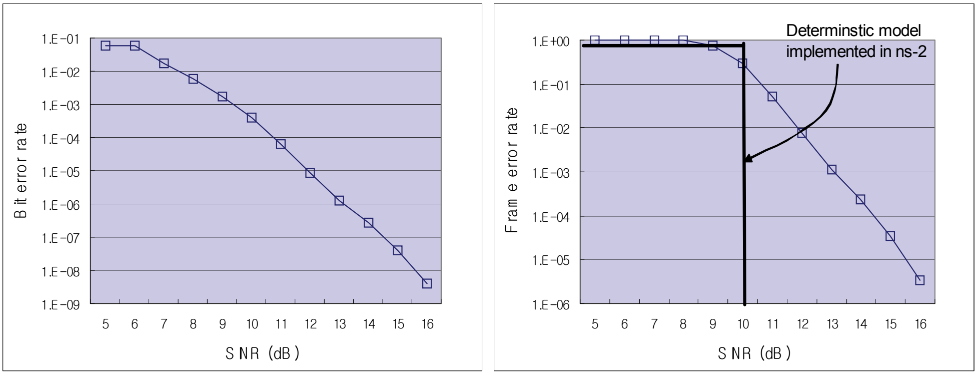
where P_r is the received power (signal strength) of the signal, P_i denotes the individual received power of other signals received by the receiver simultaneously, and N is the effective noise at the receiver. There are two components in the above equation – received power and interference plus noise.

First, the received power at the receiver (P_r) is calculated according to the radio propagation model at the receiver in *ns*-2. In our study, *Ricean fading* model (Punnoose et al., 2000; NS-2, 2009) is used as a radio propagation model. The Ricean fading is a radio propagation anomaly caused by partial cancellation of a radio signal by itself; *i.e.*, the signal arrives at the receiver by two or more different paths and at least one of the paths is changing. It occurs when one of the paths, typically a line of sight signal, is much stronger than others. The Ricean fading model is effectively applied to the environment that, in addition to scattering, there is a strongly dominant signal seen at the receiver usually caused by a line of sight.

Second, noise contains the noise generated by the receiver and the one come from environment. The effective noise level generated by the receiver can be obtained by adding up the noise figure of a network interface card (NIC) onto the thermal noise (IEEE, 1994). We first compute the thermal noise level within the channel bandwidth of 22 MHz in the IEEE 802.11 standard (IEEE, 1999). This bandwidth is 73 dB above -174 dBm/Hz, or -101 dBm. Assuming a system noise figure of 6 dB as in (IEEE, 1994), the effective noise level generated by the receiver is -95 dBm. The environment noise or channel noise is the *additive white Gaussian noise* (*AWGN*) that is modeled as a Gaussian random variable. It is assumed that the environment noise is fixed throughout the whole medium access of a communication. For realistic simulation of noisy and unstable environments, the environment noise can be varied for different medium accesses. On the other hand, interference is the received signal power calculated as described above for other frames received by the receiver simultaneously.

Based on the aforementioned discussions and the product specification of the Intersil HFA3861B radio chip (Intersil, 2007a), we are able to calculate the BER as shown in Fig. 4(a), which models the QPSK modulation with 2 Mbps. Note that the BER-E_b/N_0 curve given in (Intersil, 2007a) is simply converted into the BER-SINR curve since SINR = $E_b/N_0 \times R/B_T$, where E_b is energy required per bit of information, N_0 is interference plus noise in 1 Hz of bandwidth, R is system data rate, and B_T is system bandwidth that is given by $B_T = R$ for QPSK in the Intersil chipset (Intersil, 2007b). In an IEEE 802.11 frame, physical layer convergence protocol (PLCP) preamble, PLCP header and payload (data) may be transmitted at different rate with different modulation method. Hence, BER should be calculated separately for the three parts of a frame.

Once BER is obtained, *frame error rate* (*FER*) can be calculated, which determines the percentage that a frame is received correctly. For example, given α-bit preamble, β-bit PLCP header and γ-bit payload with BER of p_a, p_b and p_c, respectively, FER is obtained by $1 - (1 - p_a)^{\alpha}(1 - p_b)^{\beta}(1 - p_c)^{\gamma}$. For comparison, Fig. 4(b) also shows the FER curve used in unmodified *ns*-2. As discussed earlier in this section, if SINR is larger than CPThresh, *e.g.*, 10 dB as in Fig. 4(b), the frame succeeds (FER = 0.0). Otherwise, it fails (FER = 1.0). Our performance evaluation study modifies *ns*-2 so that FER is not deterministically but probabilistically determined based on SINR, making our evaluation more realistic and convincing.



(a) BER versus SINR (b) FER versus SINR

Fig. 4. BER and FER for QPSK with 2 Mbps in the Intersil HFA3861B radio chip. (The PHY frame size for calculating FER is assumed to be 864 bits, *i.e.*, 144-bit preamble, 48-bit PLCP header and 84-byte payload.)

4.2 Simulation environment

In our simulation study, it is assumed that 50 mobile nodes move over a square area of 300*m* × 1,500*m*. The propagation channel of *Ricean fading* model is assumed with a data rate of 2 Mbps. As mentioned in Section 2.2, the SINR-based rate adaptation scheme based on RBAR (Holland et al., 2001) is modeled and used in *ns*-2 (NS-2, 2008; CMU, 2008), where the data rate of RTS is set at a base rate of 1 Mbps to safely cope with dynamically changing link quality in MANETs. The constant bit rate (CBR) source of 2 packets per second is assumed with UDP-based traffic and the data payload of the packets is 512 bytes long. Mobile nodes are assumed to move randomly according to the *random waypoint model* (Broch et al., 1998), where two parameters of maximum node speed and pause time determine the mobility pattern of the mobile nodes. Each node starts its journey from a randomly selected location to a target location, which is also selected randomly in the simulation area, at a randomly chosen speed (uniformly distributed between 0 and maximum speed). The maximum speed is set as 5 *m/sec* throughout the simulation. When a node reaches the target location, it stays there during the pause time and then repeats the mobility behavior.

As for performance metrics, we evaluate the followings: *Packet delivery ratio* is the ratio of the number of data packets successfully delivered to the destination over the number of data packets sent by the source. *Per-route goodput* is the application level throughput excluding protocol overhead and retransmitted data packets, which is sometimes given by the inverse of the averaged end-to-end data packet delay. *Normalized control overhead* is the ratio of the total number of control packets transmitted for medium access and routing over the number of data packets successfully delivered to the destination, where each hop-wise transmission of a control packet is counted as one transmission.

For measuring the performance metrics, the simulation factors of the environment noise level, the number of sessions, and the pause time are varied in a meaningful range; *i.e.,* the environment noise level of -90 ~ -80 dBm (*i.e.,* -90, -88, -86, -84, -82, and -80dBm) modeled as a Gaussian random variable with the standard deviation of 1 dB, the number of sessions from 2 to 18 (*i.e.*, 2, 6, 10, 14, and 18), and the pause time of 100 ~ 900 *sec* (*i.e.*, 0, 20, 50, 100, 200, 300, 600, and 900 *sec*) are applied. While one simulation factor is varied during a simulation, the others are fixed as follows: the environment noise level of -84 dBm (which represents a relatively harsh environment), the number of sessions of 4, and the pause time of 100 *sec*. Note that the number of sessions is the number of connections. Source-destination pairs are randomly selected. Each run has been executed for 900 *sec* of simulation time.

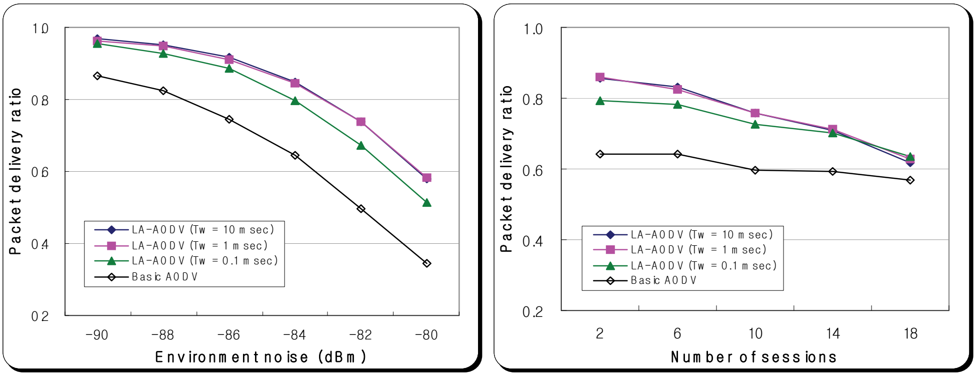
4.3 Simulation results and discussion

4.3.1 Packet delivery ratio.

Fig. 5 shows the packet delivery ratio for varying the environment noise and the number of sessions. It is shown that the proposed LA-AODV outperforms the basic AODV by up to 70 % and 34% for the environment noise and the number of sessions, respectively. Note here that LA-AODV shows almost the same performance for the two different values of *RREQ waiting time* (T_w) of 1 *msec* and 10 *msec*. The two cases with T_w of 1 *msec* and 10 *msec* outperform LA-AODV with T_w of 0.1 *msec*. Therefore, it can be easily inferred that T_w of 1 *msec* is long enough to achieve the most robust delivery in the given environment. As the environment noise increases, PDR is decreased as expected. It is slightly decreased with the increased number of sessions.

4.3.2 Per-route goodput.

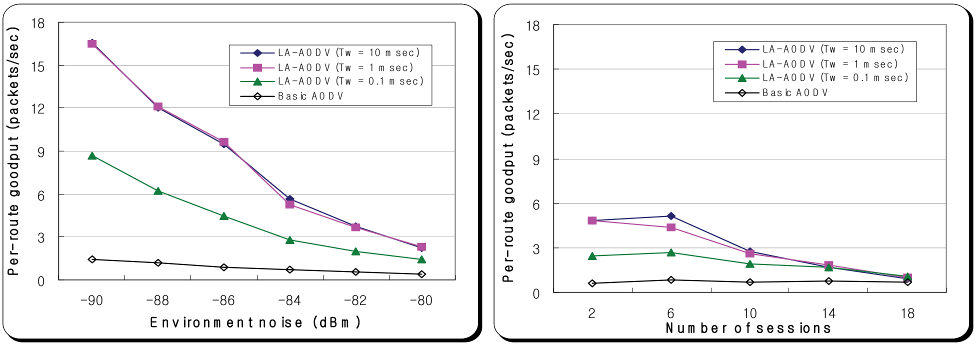
Fig. 6 shows the per-route goodput for varying the environment noise and the number of sessions. It is shown that the proposed LA-AODV outperforms the basic AODV by a factor of up to 12 and 8 for the environment noise and the number of sessions, respectively. As in Fig. 5, LA-AODV shows almost the same performance for the two different values of *RREQ waiting time* (T_w) of 1 *msec* and 10 *msec*, and the two cases with T_w of 1 *msec* and 10 *msec* outperform LA-AODV with T_w of 0.1 *msec*. Hence, T_w of 1 *msec* is long enough to achieve the highest performance in the given environment. As the environment noise increases, the per-route goodput of LA-AODV is rapidly decreased compared with the basic AODV. It is also decreased with the increased number of sessions.



(a) Varying environment noise

(b) Varying the number of sessions

Fig. 5. Packet delivery ratio.

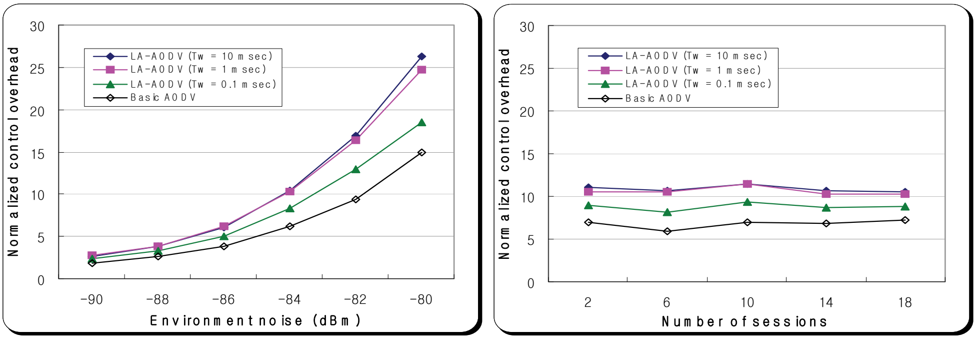


(a) Varying environment noise (b) Varying the number of sessions

Fig. 6. Per-route goodput.

4.3.3 Normalized control overhead.

Fig. 7 shows the normalized control overhead for varying the environment noise and the number of sessions. As can be expected, LA-AODV incurs more control overhead compared to the basic AODV for both the environment noise and the number of sessions. This is a kind of side effect paid to achieve robust delivery and high performance. As the environment noise increases, the normalized overhead is increased as expected. It is almost constant with the increased number of sessions. This mainly due to the fact that, as the number of sessions increases, the number of delivered data packets is also increased while the number of control packets is increased.

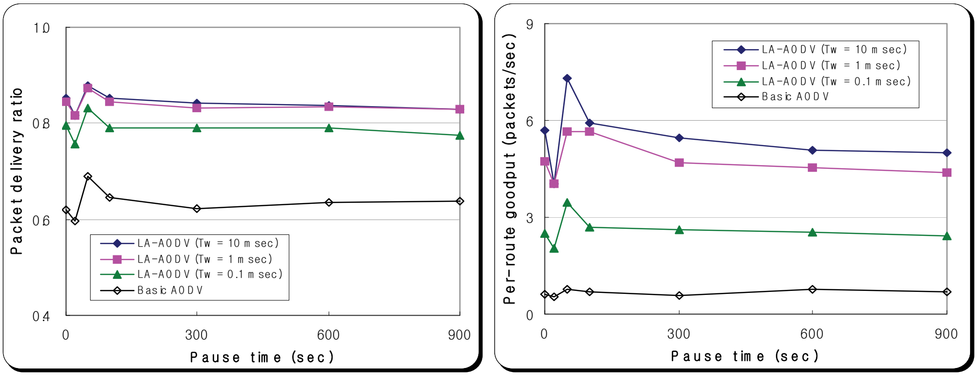


(a) Varying environment noise (b) Varying the number of sessions

Fig. 7. Normalized control overhead.

4.3.4 Impact on node mobility in the harsh environment.

In general, the network performance is highly affected by node mobility in the normal operation environment and it is degraded with increased mobility. Fig. 8 shows the impact on node mobility in the harsh environment with the noise level of -84 dBm. LA-AODV outperforms the basic AODV in terms of packet delivery ratio and per-route goodput for different pause time. However, the packet delivery ratio and per-route goodput are almost constant with increased pause time except for very high mobility. In other words, it is inferred from the results that the node mobility is not a major factor affecting performance in the harsh operation environment.



(a) Packet delivery ratio (b) Per-route goodput

Fig. 8. Effect of varying pause time in the harsh environment.

5. Conclusions

In this paper, the *link quality aware AODV* (*LA-AODV*) has been presented by devising the *RREQ forwarding algorithm*, resulting in robust packet delivery and high network performance. The RREQ forwarding algorithm finds out a reliable path with strong links. During route discovery, the strong links are effectively exploited by forwarding the route request (RREQ) packet with the highest *link quality* or *signal to interference plus noise ratio* (*SINR*) among the multiple RREQ packets received. Some tolerance is applied to the link quality in choosing an RREQ to be forwarded in order for coping with the dynamic behavior of SINR. Compared to the basic AODV, the proposed scheme may not have the minimum hop-count route but the one with more number of hops. However, the discovered route is a reliable path with high data rate because it consists of strong links, resulting in high performance as well as robust routing. The performance study shows that *packet delivery ratio* is improved by up to 70% and *per-route goodput* is dramatically increased by a factor of up to 12. It is also shown that the acceptable value of the *RREQ waiting time* (T_w) is 1 *msec* in the simulated environment, which is enough to achieve fairly good performance.

The proposed mechanism can be easily applied to other routing protocols using *broadcast-based route discovery*. To extend the LA-AODV principle to hierarchical routing protocols and multicast protocols is another future work. Our future work includes the exploration of a new link quality aware routing protocol for MANETs with asymmetric links as well, which should be a very challenging work.

6. Acknowledgement

This research was supported in part by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2010-C1090-1021-

0013). A preliminary version of this work was presented at the 11th IEEE International Conference on High Performance Computing and Communications, June 25-27, 2009 (Moh, 2009).

7. References

Addition to the *NS* Network Simulator to Handle Ricean and Rayleigh Fading. (2009). http://www.ece.cmu.edu/wireless/downloads.html.

Belding-Royer E. M. & Perkins, C. E. (2003). Evolution and Future Directions of the Ad Hoc on-Demand Distance-Vector Routing Protocol, *Ad Hoc Networks* , Vol. 1, pp. 125-150.

Broch, J.; Maltz, D. A.; Johnson, D.; Hu, Y.-C. & Jetcheva, J. (1998). A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols, *Proc. of 4th ACM/IEEE Int. Conf. on Mobile Computing and Networking* (*MobiCom*), pp. 85-97, Oct. 1998.

CMU Monarch Project. (2008). http://www.monarch.cs.cmu.edu/cmu-ns.html.

De Couto, D. S. J.; Aguayo, D.; Chambers, B. A. & Morris, R. (2002). Performance of Multihop Wireless Networks: Shortest Path is Not Enough, *Proc. of 1st Workshop on Hot Topics in Networking* (*HotNets-I*), Oct. 2002.

De Couto, D. S. J.; Aguayo, D.; Bicket, J. & Morris, R. (2003). A High-Throughput Path Metric for Multi-Hop Wireless Routing, *Proc. of 9th ACM Int. Conf. on Mobile Computing and Networking* (*MobiCom'03*), Sep. 2003.

Gaertner, G. & Cahill, V. (2004). Understanding Link Quality in 802.11 Mobile Ad Hoc Networks. *IEEE Internet Computing*, Vol. 8, No. 1, pp. 55-60.

Holland, G.; Vaidya, N. H. & Bahl, P. (2001). A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks, *Proc. of 7th Annual Int. Conf. on Mobile Computing and Networking (MobiCom 2001)*, pp. 236-251, Rome, Italy, July 16-21, 2001.

IEEE P802.11-94/132. (1994). Wireless Access Methods and Physical Layer Specifications: Elaborate Clear-Channel Assessment for Indoor Communication Systems Operating in Uncontrolled UHF & Microwave Bands, July 1994.

IEEE Std 802.11-1999. (1999). Local and Metropolitan Area Network, Specific Requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

Internet Engineering Task Force (IETF), (2009). Mobile Ad Hoc Networks (MANET) Working Group Charter, http://www.ietf.org/html.charters/manet-charter.html.

Intersil. (2007). Intersil HFA3861B chip Specification, http://www.chipdocs.com/pndecoder/datasheets/INTRS/HFA3861B.html.

Intersil. (2007). Intersil Prism WLAN Chipset Application Note: Tutorial on Basic Link Budget Analysis, http://www.sss-mag.com/pdf/an9804.pdf.

Kamerman, A. & Monteban, L. (1997). WaveLAN-II: A High-Performance Wireless LAN for the Unlicensed Band, *Bell Labs Technical Journal*, Vol. 2, Issue 3, pp. 118-133.

Karlsson, J.; Li, H. & Ericksson, J. (2005). Real-Time Video over Wireless Ad-Hoc Networks, *Proc. of 4th Int. Conf. on Computer Communications and Networks* (*ICCCN*), Oct. 2005.

Kortebi, R.; Meddour, D.-E.; Gourhant, Y. & Agoulmine, N. (2007). SINR-Based Routing in Multi-Hop Wireless Networks to Improve VoIP Applications Support, *Proc. of 4th IEEE Consumer Communications and Networking Conference (CCNC 2007)*, pp. 491-496, Jan. 2007.

Krco, S. & Dupcinov, M. (2003). Improved Neighbor Detection Algorithm for AODV Routing Protocol. *IEEE Communication Letters*, Vol. 7, No. 12, pp. 584-586.

Lacage, M.; Manshaei, M. H. & Turletti, T. (2004). IEEE 802.11 Rate Adaptation: A Practical Approach, *Proc. of 7th ACM Int. Symp. on Modeling, Analysis and Simulation of Wireless and Mobile Systems* (*MSWiM 2004*), pp. 126-134, Venezia, Italy, Oct. 4-6, 2004.

Moh, S. (2009). Link Quality Aware Route Discovery for Robust Routing and High Performance in Mobile Ad Hoc Networks, *Proc. of 11th IEEE Int. Conf. on High Performance Computing and Communications*, pp. 281-288, June 25-27, 2009.

The Network Simulator *ns*-2. (2008). http://www.isi.edu/nsnam/ns/.

Pang, Q.; Leung, V. C. M. & Liew, S. C. (2005). A Rate Adaptation Algorithm for IEEE 802.11 WLANs Based on MAC-Layer Loss Differentiation, *Proc. of 2nd Int. Conf. on Broadband Networks*, Vol. 1, pp. 659-667, Oct. 3-7, 2005.

Pavon J. P. & Choi, S. (2003). Link Adaptation Strategy for IEEE 802.11 WLAN via Received Signal Strength Measurement, *Proc. of IEEE int. Conf. on Communications* (*ICC*), Vol. 2, pp. 1108-1113, May 2003.

Punnoose, R. J.; Nikitin, P. V. & Stancil, D. D. (2000). Efficient Simulation of Ricean Fading within a Packet Simulator, *Proc. of 52nd Vehicular Technology Conference* (*VTC2000-Fall*), Vol. 2, pp. 764-767, Boston, Massachusetts, Sep. 24-28, 2000.

Perkins C. E. & Watson, T. J. (1994). Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers, *Proc. of Int. Conf. on Communications Architectures (ACM SIGCOMM'94),* pp. 234-244, Oct. 1994.

Perkins, C. E. (2001). *Ad Hoc Networking*, Addison Wesley, Upper Saddle River, NJ, USA.

Perkins, C. E.; Belding-Royer, E. M. & Das, S. R. (2003). Ad Hoc On-Demand Distance Vector (AODV) Routing, http://www.ietf.org/rfc/rfc3561.txt.

Sadeghi, B.; Kanodia, V.; Sabharwal, A. & Knightly, E. (2002). Opportunistic Media Access for Multirate Ad Hoc Networks, *Proc. of 8th Annual Int. Conf. on Mobile Computing and Networking (MobiCom 2002)*, pp. 24-35, Atlanta, Georgia, Sep. 23-28, 2002.

Siva Ram Murthy, C. & and Manoj, B. S. (2004). *Ad Hoc Wireless Networks*, Prentice Hall, Upper Saddle River, NJ, USA.

Wang, J.; Zhai, H.; Fang, Y. & Yuang, M. C. (2004). Opportunistic Media Access Control and Rate Adaptation for Wireless Ad Hoc Networks, *Proc. of Int. Conf. on Communications (ICC 2004)*, Vol. 1, pp. 154-158, Paris, France, June 20-24, 2004.

Zhai, H.; Wang, J.; Chen, X. & Fang, Y. (2006). Medium Access Control in Mobile Ad Hoc Networks: Challenges and Solutions, *Wireless Communications and Mobile Computing*, Vol. 6, Issue 2, pp. 151-170.

Zhao, S.; Wu, Z.; Acharya, A. & Raychaudhuri, D. (2005). PARMA: A PHY/MAC Aware Routing Metric for Ad-Hoc Wireless Networks with Multi-Rate Radios, *Proc. of 6th Int. Symp. on a World of Wireless Mobile and Multimedia Networks (WoWMoM 2005*), Vol. 1, pp. 286-292, Taormina, Italy, June 13-16, 2005.

12

A Location Prediction Based Routing Protocol and its Extensions for Multicast and Multi-path Routing in Mobile Ad hoc Networks

Natarajan Meghanathan
Jackson State University, Jackson, MS
United States of America

1. Introduction

A mobile ad hoc network (MANET) is a dynamic distributed system of wireless nodes that move independently of each other. MANET routing protocols are either proactive or reactive in nature. Proactive routing protocols determine and maintain routes between any pair of nodes irrespective of their requirement. The reactive on-demand routing protocols determine a route only when required. As the network topology changes dynamically, reactive routing has been preferred over proactive routing (Broch et. al., 1998). We will focus only on the reactive on-demand routing protocols in this chapter.

Based on route selection principles, MANET routing protocols can be classified as minimum-weight based and stability-based (Meghanathan, 2009d). Most of the minimum-weight based protocols aim to minimize the number of hops in a path. Some of the well-known minimum-hop based routing protocols are the Dynamic Source Routing (DSR) protocol (Johnson et. al., 2000), Ad hoc On-demand Distance Vector (AODV) routing protocol (Perkins & Royer, 1999) and the Location-Aided Routing (LAR) protocol (Ko & Vaidya, 2000). Stability-based protocols aim for routes with longer lifetimes in order to reduce the number of route discoveries. The Flow Oriented Routing Protocol, FORP, (Su et. al., 2001) yields the sequence of most stable routes among the stable path routing protocols available in the literature (Meghanathan, 2008). Performance comparison studies reveal that the stable path protocols could incur as low as half the number of route discoveries incurred by the minimum-hop based protocols, but the average hop count of stable paths could be as large as twice the minimum hop count (Meghanathan, 2008). Frequent flooding-based route discoveries incurred by the minimum-hop based protocols significantly consume network bandwidth and congest the network. Stable paths with larger hop count also consume more network bandwidth and reduce frequency reuse. Moreover, nodes that are part of a stable path are used more predominantly compared to the other nodes in the network. In this chapter, we discuss a new MANET routing protocol called "Location Prediction Based Routing" (LPBR) protocol (Meghanathan, 2009a) that simultaneously minimizes the number of route discoveries as well as the hop count of paths used for a source-destination session. We assume all the nodes are position-aware using techniques like Global Positioning Systems (Hofmann-Wellenhof, 2004) and the clocks across all nodes are synchronized.

LPBR works as follows: Whenever a source node has data packets to send to a destination node but does not have a route to that node, it initiates a flooding-based route discovery by broadcasting a Route-Request (RREQ) packet. During this flooding process, each node forwards the RREQ packet exactly once after incorporating its location update vector (LUV) in the RREQ packet. The LUV of a node comprises the node ID, the current X and Y co-ordinates of the nodes, the current velocity and angle of movement with respect to the X-axis. The destination node collects the LUV information of all the nodes in the network from the RREQ packets received through several paths and sends a Route-Reply (RREP) packet to the source on the minimum hop path traversed by a RREQ packet. The source starts sending the data packets on the path learnt (based on the RREP packet) and informs the destination about the time of next packet dispatch through the header of the data packet currently being sent. If an intermediate node could not forward a data packet, it sends a Route-Error (RERR) packet to the source node, which then waits a little while for the destination to inform it of a new route predicted using the LUVs gathered from the latest flooding-based route discovery. If the destination does not receive the data packet within the expected time, it locally constructs the current global topology by predicting the locations of the nodes. Each node is assumed to be currently moving in the same direction and speed as mentioned in its latest LUV. If there is at least one path in the predicted global topology, the destination node sends the source a LPBR-RREP packet on the minimum hop path in the predicted topology. If the predicted path actually exists in reality, the intermediate nodes on the predicted route manage to forward the LPBR-RREP packet to the source. The source uses the route learnt through the latest LPBR-RREP packet to send the data packets. A costly flooding-based route discovery has been thus avoided. If an intermediate node could not forward the LPBR-RREP packet (i.e., the predicted path did not exist in reality), the intermediate node sends a LPBR-RREP-ERROR packet to the destination informing it of the failure to forward the LPBR-RREP packet. The destination discards all the LUVs and the source initiates the next flooding-based route discovery after timing out for the LPBR-RREP packet.

In the second part of the chapter, we discuss two multicast extensions to LPBR (Meghanathan, 2009b), referred to as NR-MLPBR and R-MLPBR. Both the multicast extensions are aimed at minimizing the number of global broadcast tree discoveries as well as the hop count per source-receiver path of the multicast tree. They use a similar idea of letting the receiver nodes to predict a new path based on the locally constructed global topology obtained from the location and mobility information of the nodes learnt through the latest broadcast tree discovery. Receiver nodes running NR-MLPBR (Non-Receiver aware Multicast extensions of LPBR) are not aware of the receivers of the multicast group, whereas each receiver node running R-MLPBR (Receiver-aware Multicast Extension of LPBR) is aware of the identity of the other receivers of the multicast group. NR-MLPBR attempts to predict a minimum hop path to the source, whereas R-MLPBR attempts to predict a path to the source that has the minimum number of non-receiver nodes. If more than one path has the same minimum number of non-receiver nodes, then R-MLPBR breaks the tie among such paths by choosing the path with the minimum number of hops to the source. Thus, R-MLPBR is also designed to reduce the number of links in the multicast tree, in addition to the average hop count per source-receiver path and the number of global broadcast tree discoveries.

In the final part of the chapter, we discuss a node-disjoint multi-path extension to the LPBR protocol, referred to as LPBR-M (Meghanathan, 2009c). It has been earlier observed that for different conditions of network density and node mobility, the number of broadcast route discoveries needed for node-disjoint multi-path routing is not significantly different from the number of route discoveries for link-disjoint multi-path routing (Meghanathan, 2007). Also, there is not much difference in the average hop count of the node-disjoint paths and the link-disjoint paths. On the other hand, node-disjoint paths are preferred for fault tolerance, load balancing and extending the lifetime of the nodes. LPBR-M minimizes the control overhead by reducing the number of broadcast route discoveries as much as possible using multi-path routing. Also, LPBR-M yields an average hop count per multi-path that is almost equal to that of the minimum-hop based multi-path routing protocols.

The rest of the chapter is organized as follows: Section 2 describes the design of the LPBR protocol and Section 3 describes the performance comparison study of LPBR with minimum-hop based and stability-based unicast routing protocols. Section 4 describes the design of the multicast extensions, NR-MLPBR and R-MLPBR and Section 5 describes the performance comparison study of these two multicast extensions of LPBR with that of minimum-hop based and minimum-edge based multicast routing protocols. Section 6 describes the design of the node-disjoint multi-path extension of LPBR, referred to as LPBR-M, and Section 7 presents its simulation performance comparison study with well-known link-disjoint and node-disjoint multi-path routing protocols. Section 8 concludes the chapter. Throughout the chapter, the terms 'node' and 'vertex', 'link' and 'edge', 'message' and 'packet' are used interchangeably. They mean the same.

2. Design of the Unicast Location Prediction Based Routing (LPBR) protocol

2.1 Route discovery to collect Location Update Vectors (LUVs)

When the source has a data packet to send to a destination and is not aware of any route to that node, the source initiates a flooding-based route discovery by sending a Route-Request packet (RREQ) to its neighbors. The source maintains a monotonically increasing sequence number for the flooding-based route discoveries it initiates to reach the destination. Each node on receiving the first RREQ packet (with a sequence number greater than that seen before), will include its location update vector LUV (comprising the node ID, X, Y co-ordinate information, current velocity and angle of movement with respect to the X-axis) in the RREQ packet. The intermediate node also appends its node ID in the "Route record" field of the RREQ packet. The LUV and the RREQ are shown in Figures 1 and 2 respectively.

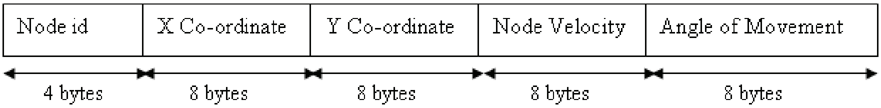


Fig. 1. Location Update Vector Collected from Each Node

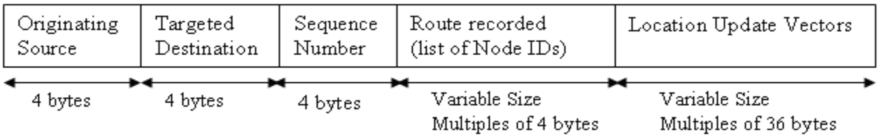
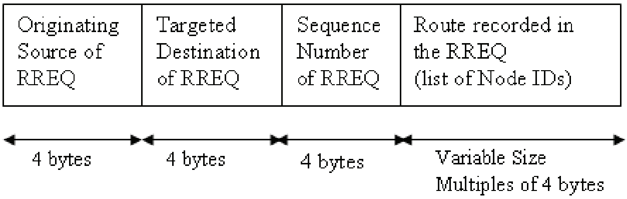

Originating Source
Targeted Destination
Sequence Number
Route recorded (list of Node IDs)
Location Update Vectors
4 bytes
4 bytes
4 bytes
Variable Size Multiples of 4 bytes
Variable Size Multiples of 36 bytes

Fig. 2. Route Request (RREQ) Packet with the LUVs

The destination receives several RREQ packets across different paths and selects the minimum hop path among them using the "Route record" field in these RREQ packets. A Route-Reply (RREP) packet (refer Figure 3) is sent on the discovered minimum hop route to the source. All the nodes receiving the RREP packet will update their routing tables to forward incoming data packets (for the source-destination session) to the node that sent the RREP packet. Note that in order to collect the latest location and mobility information of each node through the LUVs, we intentionally do not let any intermediate node to respond to the source with a RREP for a RREQ packet. RREQ packets reach the destination through several paths and will gather the LUVs from several nodes in the network. We consider the time of receipt of the RREQ packet as the time of obtaining the LUV of a node as we expect no major link failures to happen during the time lapsed in between a node sending its LUV in the RREQ packet and the RREQ reaching the destination.

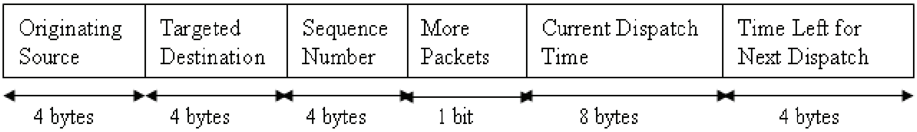


Originating Source of RREQ	Targeted Destination of RREQ	Sequence Number of RREQ	Route recorded in the RREQ (list of Node IDs)
4 bytes	4 bytes	4 bytes	Variable Size Multiples of 4 bytes

Fig. 3. Route Reply (RREP) Packet

2.2 Data packet transmission and route maintenance

The source starts sending data packets to the destination on the route learnt through the RREP packet. In addition to the usual sequence number, source and destination fields, the header of the data packet (refer Figure 4) has three specialized fields: the 'More Packets' (*MP*) field, the 'Current Dispatch Time' (*CDT*) field and the 'Time Left for Next Dispatch' (*TLND*) field. The additional overhead associated with these three header fields amount to only 97 bits per data packet.



Originating Source	Targeted Destination	Sequence Number	More Packets	Current Dispatch Time	Time Left for Next Dispatch
4 bytes	4 bytes	4 bytes	1 bit	8 bytes	4 bytes

Fig. 4. Structure of the Header of the Data Packet

The *CDT* field stores the time as the number of milliseconds lapsed since Jan 1, 1970, 12 AM. If the source has more data to send, it sets the *MP* flag to 1 and the *TLND* field to be the number of milliseconds since the *CDT* of the latest data packet sent. If the source has no more data to send, the *MP* flag is set to 0 and the *TLND* field is left blank. As we assume synchronized clocks across all nodes, the destination calculates the end-to-end delay for the data packet based on the local time of receipt of the data packet and the *CDT* field in the header of the data packet. The destination maintains an average of the end-to-end delay per data packet incurred for the path currently being used to communicate with the source and updates it based on the end-to-end delay suffered by the data packet currently received. If the source has set the *MP* flag, the destination computes the 'Next Expected Packet Arrival

Time' (*NEPAT*) as *CDT* + *TLND* + 2*Average end-to-end delay per data packet. A timer is started for the *NEPAT* value. If a link failure occurs due to two nodes constituting the link drifting away, the upstream node of the broken link informs the source through a Route-Error packet (Figure 5). The source on learning the route failure stops sending data packets and waits for the destination to inform it of any new route through a LPBR-RREP packet.

Node sending the error packet	Sequence number of the data packet that could not be forwarded	Source node of the data packet	Destination node of the data packet	Downstream node with which the link has failed
4 bytes	4 bytes	4 bytes	4 bytes	4 bytes

Fig. 5. Structure of the Route-Error Packet

2.3 Predicting node location using location update vector

If the destination does not receive the data packet within the *NEPAT* time, it will attempt to locally construct the global topology using the location and mobility information of the nodes learnt from the latest flooding-based route discovery. Each node is assumed to continue to move in the same direction with the same speed as mentioned in its latest LUV. Let (X_u^{STIME}, Y_u^{STIME}) be the X and Y co-ordinates of node *u* learnt from its LUV collected at time *STIME*. Let $Angle_u^{STIME}$ and $Velocity_u^{STIME}$ represent the angle of movement with respect to the X-axis and the velocity at which node *u* is moving. We determine the location of node *u* at time *CTIME*, denoted by (X_u^{CTIME}, Y_u^{CTIME}), as follows: Distance traveled by node *u* from time *STIME* to *CTIME* is: $Distance_u^{STIME-CTIME} = (CTIME - STIME + 1) * Velocity_u^{STIME}$. Then, $X_u^{CTIME} = X_u^{STIME} + Offset\text{-}X_u^{CTIME}$ and $Y_u^{CTIME} = Y_u^{STIME} + Offset\text{-}Y_u^{CTIME}$. The offsets in the X and Y-axes depend on the angle of movement and the distance traveled.

$$Offset\text{-}X_u^{CTIME} = Distance_u^{STIME-CTIME} * \cos(Angle_u^{STIME})$$

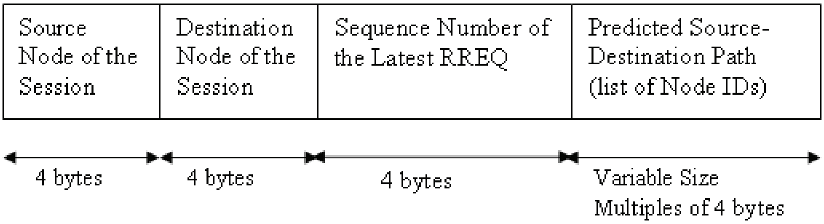
$$Offset\text{-}Y_u^{CTIME} = Distance_u^{STIME-CTIME} * \sin(Angle_u^{STIME})$$

$$\text{where } 0^\circ \le Angle_u^{STIME} \le 360^\circ$$

Let the network boundaries be given by [0, 0], [X_{max}, 0], [X_{max}, Y_{max}] and [0, Y_{max}]. X_u^{CTIME} and Y_u^{CTIME} are bounded by the constraints: $0 \le X_u^{CTIME} \le X_{max}$ and $0 \le Y_u^{CTIME} \le Y_{max}$. If a predicted X and/or Y co-ordinate value falls outside this window, the value is reset to the nearest co-ordinate limit.

2.4 Path prediction and source notification

Based on the predicted locations of each node in the network at time *CTIME*, the destination node locally constructs the global topology. Note that there exists an edge between two nodes in the locally constructed global topology if the predicted distance between the two nodes (computed based on their predicted locations) is less than or equal to the transmission range of the nodes. The destination node *d* then locally runs the Dijkstra's minimum hop path algorithm (Cormen et. al., 2001) with the starting node being the source *s* on the predicted global topology. If at least one *s-d* path exists, the destination sends a LPBR-RREP packet (refer Figure 6) on the minimum hop *s-d* path with the route information included in the packet.



Source Node of the Session	Destination Node of the Session	Sequence Number of the Latest RREQ	Predicted Source-Destination Path (list of Node IDs)
4 bytes	4 bytes	4 bytes	Variable Size Multiples of 4 bytes

Fig. 6. Structure of the LPBR-RREP Packet

Each intermediate node receiving the LPBR-RREP packet updates its routing table to record the incoming interface of the packet as the outgoing interface for any data packet sent from *s* to *d* and then forwards the LPBR-RREP packet to the next node on the path to the source node. If the predicted *s-d* path exists in reality, then the source *s* is most likely to receive the LPBR-RREP packet before the LPBR-RREP-timer expires. The source now sends the data packets on the route learnt through the latest LPBR-RREP packet received from the destination. The Route-Repair Time (*RRT*) is the time that lapsed since the source received the Route-Error packet. An average *RRT* value is maintained at the source as it undergoes several route failures and repairs before the next flooding-based route discovery. The LPBR-RREP-timer (initially set to the route acquisition time) is then set to 1.5*Average *RRT* value, so that we give sufficient time for the destination to learn about the route failure and generate a new LPBR-RREP packet. Nevertheless, this timer value will be still far less than the route acquisition time that would be incurred if the source were to launch a flooding-based route discovery. Hence, our approach will only increase the network throughput.

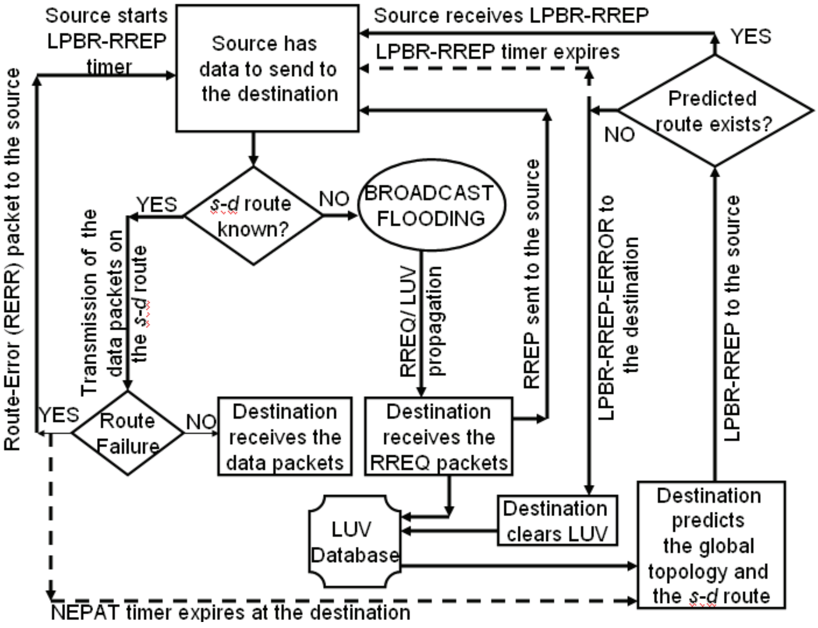

Source starts LPBR-RREP timer
Source has data to send to the destination
Source receives LPBR-RREP
YES
LPBR-RREP timer expires
Predicted route exists?
NO
Route-Error (RERR) packet to the source
YES
s-d route known?
NO
BROADCAST FLOODING
Transmission of the data packets on the s-d route
RREQ/ LUV propagation
RREP sent to the source
LPBR-RREP-ERROR to the destination
LPBR-RREP to the source
YES
Route Failure
NO
Destination receives the data packets
Destination receives the RREQ packets
LUV Database
Destination clears LUV
Destination predicts the global topology and the s-d route
NEPAT timer expires at the destination

Fig. 7. Comprehensive Illustration of the Working of the LPBR Protocol

2.5 Handling prediction failures

If an intermediate node could not successfully forward the LPBR-RREP packet to the next node on the path towards the source, it informs the absence of the route to the destination through a LPBR-RREP-ERROR packet (refer Figure 8). The destination on receiving the LPBR-RREP-ERROR packet discards all the LUVs and does not generate any new LPBR-RREP packet. After the LPBR-RREP-timer expires, the source initiates a new flooding-based route discovery. Figure 7 comprehensively illustrates the working of the LPBR protocol.

Node sending the error packet	Sequence number in the LPBR-RREP packet that could not be forwarded	Source Node of the Data Session	Destination Node of the Data Session	Downstream node with which the link has failed
4 bytes	4 bytes	4 bytes	4 bytes	4 bytes

Fig. 8. Structure of the LPBR-RREP-ERROR Packet

3. Simulation performance study of LPBR

We use ns-2 (version 2.28; Breslau, et. al., 2000) as the simulator for our study. We implemented the LPBR, FORP and LAR protocols and used the implementation of DSR that comes with ns-2. The MAC layer module used is the IEEE 802.11 (Bianchi, 2000) implementation available in ns-2. The network dimension used is a 1000m x 1000m square network. The transmission range of each node is assumed to be 250 m. The number of nodes used is 25 and 75 nodes representing networks of low and high density respectively.

Traffic sources are constant bit rate (CBR). The number of source-destination (*s-d*) sessions used is 15 (indicating low traffic load) and 30 (indicating high traffic load). The starting timings of these *s-d* sessions are uniformly distributed between 1 to 50 seconds. The sessions continue until the end of the simulation time, which is 1000 seconds. Data packets are 512 bytes in size and the packet sending rate is 4 data packets/second. For each node, we made sure that the node does not end up a source for more than two sessions and/ or not as a destination for more than two sessions.

The node mobility model used in all of our simulations is the Random Waypoint model (Bettstetter, 2004), a widely used mobility model in MANET simulation studies. According to this model, each node starts moving from an arbitrary location to a randomly selected destination location at a speed uniformly distributed in the range [0,…,v_{max}]. Once the destination is reached, the node may stop there for a certain time called the pause time and then continue to move by choosing a different target location and a different velocity. The v_{max} values used are 10 m/s, 30 m/s and 50 m/s representing scenarios of low, moderate and high node mobility respectively. Pause time is 0 seconds.

We measure the following performance metrics. Each data point in Figures 9 through 12 is an average of data collected using 5 mobility trace files and 5 sets of randomly selected 15 or 30 *s-d* sessions, depending on the simulation condition.

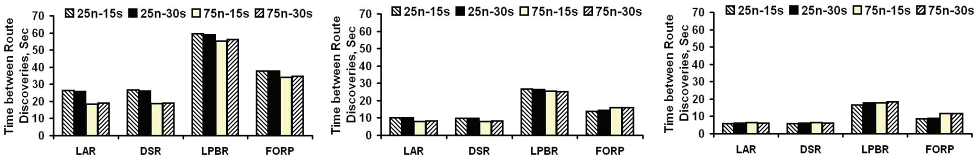
- *Time between successive route discoveries*: It is the time between successive global broadcast flooding based route discoveries per *s-d* session, averaged over all the *s-d* sessions. The larger the time between successive broadcast route discoveries, the lower is the number of route discoveries and vice-versa.

- *Hop count per path*: It is the average hop count per path, time-averaged over all the *s-d* sessions. For example, if we have been using two paths P_1 of hop count 3 and P_2 of hop count 5 for time 10 and 20 seconds respectively, then the time-averaged hop count of P_1 and P_2 is (3*10 + 5*20)/30 = 4.33.
- *End-to-end delay per Data packet*: It is the average of the delay incurred by the data packets that originate at the source and delivered at the destination. The delay incurred by a data packet includes all the possible delays – the buffering delay due to the route acquisition latency, the queuing delay at the interface queue to access the medium, the transmission delay, propagation delay, and the retransmission delays due to the MAC layer collisions.
- *Packet delivery ratio*: It is the ratio of the data packets delivered to the destination to the data packets originated at the source, computed over all the *s-d* sessions.
- *Control messages received*: It is the sum of the route discovery control messages (like RREQ, RREP, Route-Error in the case of DSR, LAR and FORP; RREQ, RREP, Route-Error, LPBR-RREP, and LPBR-RREP-Error messages in the case of LPBR) received by the nodes in the network, computed over all the *s-d* sessions of a simulation run. Note that most of the control messages for route discovery are broadcast in nature. The sum of the energy lost at all the receivers of a broadcast message is far greater than the energy lost by the transmitter of the broadcast message. Hence, we measure the control message overhead as the sum of the number of control messages received at all the nodes in the network across all the *s-d* sessions of a simulation run.

3.1 Time between successive route discoveries

LPBR incurs the largest time between successive route discoveries as observed in Figure 9. Note that the values for the time between successive route discoveries for LAR and DSR are low, but are almost the same. This is because the minimum hop *s-d* routes obtained by LAR are almost similar to that of DSR. Only the means by which the minimum hop routes are acquired are different. At least 80% of the LAR route discoveries are successful when made within the Request Zone. At most only 20% of the LAR route discoveries have to be made through global flooding. Note that the time between successive route discoveries is almost independent of the offered data traffic load because the source-destination sessions are independent of each other.

The stable path based FORP incurs larger time between successive route discoveries (i.e. a reduced number of route discoveries) when compared with the minimum hop based DSR and LAR. FORP routes are more stable in networks of high density compared to networks of low density. This is because as we increase the number of nodes in the neighborhood, there are more chances of finding stable links that will exist for a longer time. On the other hand, minimum hop based routes are more stable in networks of low density compared to networks of high density. The instability of the minimum hop paths could be attributed to the larger physical distance of the constituent hops on the path (Edge effect; Lim et. al., 2002). The physical distance of each hop in a minimum-hop path is on average close to 80% of the transmission range of the nodes. Thus, the constituent nodes of a hop are more likely to drift away quickly. In the case of LPBR, even though its routes are also likely to be prone to Edge effect because of incurring a hop count that is only at most 10% more than that of DSR for any simulation condition tested, the effectiveness and accuracy of the location prediction and the route prediction approaches helps to avoid the flooding-based route discoveries as much as possible.

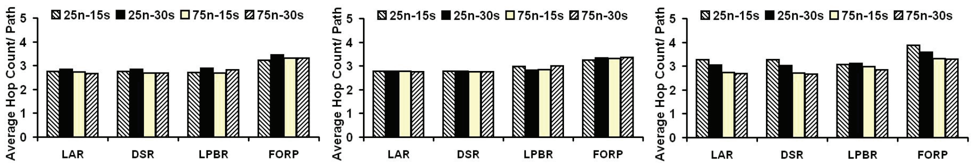


v_{max} = 10 m/s v_{max} = 30 m/s v_{max} = 50 m/s

Fig. 9. Average Time between Successive Route Discoveries per Source-Destination Session

3.2 Hop count per path

In Figure 10, we observe that the hop count incurred by LPBR routes is almost close to that incurred by DSR and LAR routes. On the other hand, the hop count of LPBR routes is significantly smaller compared to that incurred by FORP. This indicates the effectiveness of the route prediction approach adopted in LPBR. The hop count of the routes predicted upon a route failure is close to being the minimum in the network at that instant of time. FORP routes have a larger hop count in networks of higher node mobility. When the network topology changes dynamically, a link is predicted to have a higher lifetime, if the constituent nodes of a link are approaching towards each other or traveling parallel to each other. As node mobility increases, stable links can be predicted to exist only if the distance separating the constituent nodes of the link is far smaller than the transmission range of the nodes. The hop count of FORP routes in networks of moderate and high network mobility is 5 to 15% more than that of FORP routes in networks of low mobility.



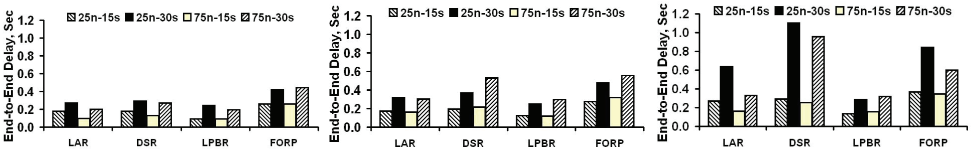
v_{max} = 10 m/s v_{max} = 30 m/s v_{max} = 50 m/s

Fig. 10. Average Hop Count per Source-Destination (*s-d*) Path

As the routing protocols simulated in this chapter do not take the queue size into consideration while determining the routes, the hop count of the routes for each protocol is independent of the offered data traffic load. Similarly, the hop count of the routes chosen by LPBR, DSR and LAR are almost independent of the maximum velocity of the nodes. But, the hop count of FORP routes is somewhat influenced by the dynamics of node mobility.

3.3 End to end delay per data packet

Figure 11 illustrates the end-to-end delay per data packet for the routing protocols. LPBR incurs the lowest end-to-end delay per data packet for all of the simulation conditions. For networks of high density, the end-to-end delay per data packet for both LAR and LPBR are close to each other and their delay values are significantly smaller than those incurred by the other routing protocols. In high-density networks, all nodes are not heavily loaded and the hop count of LAR paths are lower than that obtained in networks of low density. Most of the LAR route discoveries are done within the Request Zone and the route discovery control overhead is only 60% of that incurred due to regular flooding. Hence, the data packets sent using LAR suffer lower delays. The end-to-end delays per data packet for both LAR and LPBR in networks of high density are only within 5% of each other's value.



v_{max} = 10 m/s v_{max} = 30 m/s v_{max} = 50 m/s

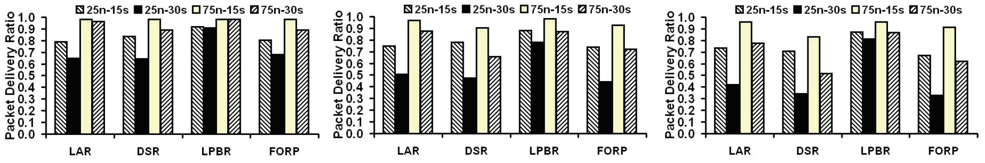
Fig. 11. Average End-to-End Delay per Data Packet for a Source-Destination (*s-d*) Session

FORP suffers a higher end-to-end delay per packet than that of LPBR and LAR. This is attributed to the paths of larger hop count chosen by FORP to reduce the number of route transitions. In the case of DSR, the routing protocol incurs a higher end-to-end delay per data packet, compared to LPBR and LAR. The route-acquisition delay for DSR is lower than that of LPBR for the 25 nodes and 15 *s-d* pairs scenario (by a factor of 20 to 30%). On the other hand, as we increase the network density and/or the offered data traffic load, the route acquisition delay of DSR tremendously increases. DSR is not scalable as we increase node mobility and the number of source-destination sessions.

3.4 Packet delivery ratio

Figure 12 illustrates the packet delivery ratio achieved with the routing protocols simulated in this chapter. LPBR achieves the highest packet delivery ratio among all the protocols in all the simulation conditions tested. This indicates the effectiveness of the location prediction approach and the route prediction technique adopted by LPBR. As LPBR undergoes the minimal number of route discoveries, there is not much route discovery control overhead traffic that blocks the data packets from going through the queues of the nodes. Among the four sets of simulation conditions tested (50 nodes with 30 *s-d* pairs, 50 nodes with 15 *s-d* pairs, 25 nodes with 30 *s-d* pairs, 25 nodes with 30 *s-d* pairs), the 25 nodes with 30 *s-d* pairs scenario generates the maximum amount of data traffic load per node in the network because the routes between the 30 *s-d* pairs have to be handled by only 25 nodes in the network and most of the nodes are also either source and/or destination of at least one *s-d* session. Though the packet delivery ratio of LPBR drops by 10 to 25% for this scenario, LPBR still incurs the highest packet delivery ratio among all the routing protocols.

For a given offered traffic load and node mobility, DSR incurs a larger packet delivery ratio for low density networks; whereas, LAR and FORP incur a relatively larger packet delivery ratio than DSR for high density networks. This could be attributed to the relative instability of DSR routes in high-density networks vis-à-vis low-density networks. As a result, DSR incurs larger control traffic route discovery overhead in high density networks, leading to the dropping of data packets at the queues of the nodes. LAR incurs relatively the least amount of route discovering control traffic overhead in high-density networks, as most of the broadcasts within the Request Zone are successful in discovering the routes from the source to the destination and the amount of control traffic generated by broadcasting within the Request Zone is only 60% of that generated due to global flooding.

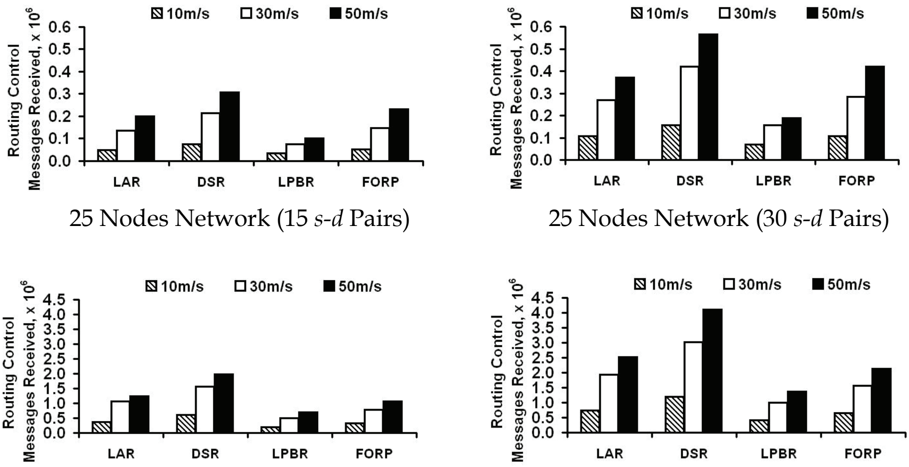


v_{max} = 10 m/s v_{max} = 30 m/s v_{max} = 50 m/s

Fig. 12. Average Packet Delivery Ratio for a Source-Destination (*s-d*) Session

3.5 Control message received

We measure the control message overhead as the number of control messages received by the nodes in the network, rather than the number of control messages transmitted (because the control traffic is broadcast in nature). For example, if a node has 10 neighbors and it broadcasts a packet to its neighborhood, then there is just one transmission, but there are 10 receptions. All neighbors of the node lose energy to receive the packet. Broadcast transmissions of control packets are not preceded by MAC layer (Request-To-Send–Clear-To-Send) RTS-CTS mechanisms (Bianchi, 2000) to reserve the medium. So, a receiving node has no option other than to receive the entire control packet and then dump it if it is redundant or not useful. Figure 13 illustrates the number of route discovery control messages received across all nodes in the network for all the sessions of a particular simulation condition, averaged over several runs of the same condition. LPBR incurs the least route discovery control message overhead when compared to DSR, LAR and FORP.



25 Nodes Network (15 *s-d* Pairs) 25 Nodes Network (30 *s-d* Pairs)

75 Nodes Network (15 *s-d* Pairs) 75 Nodes Network (30 *s-d* Pairs)

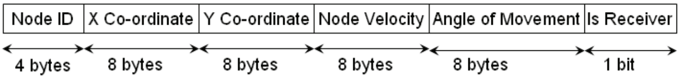
Fig. 13. Routing Control Overhead – summed over all Source Destination (*s-d*) Sessions

4. Multicast extensions of LPBR

The objective of the multicast extensions to LPBR (referred to as NR-MLPBR and R-MLPBR) is to simultaneously minimize the number of global broadcast tree discoveries as well as the hop count per source-receiver path. The Non-Receiver aware Multicast extension to LPBR (NR-MLPBR) precisely does this and it does not assume the knowledge of the receiver nodes of the multicast group at every receiver node. The Receiver-aware multicast extension of LPBR (R-MLPBR) assumes that each receiver node knows the identities of the other receiver nodes in the multicast group. This enables R-MLPBR to also reduce the number of links in the multicast tree in addition to reducing the number of global broadcast tree discoveries and the hop count per source-receiver path. Each receiver node running R-MLPBR learns the identity information of peer receiver nodes through the broadcast tree discovery procedure. Both the multicast extensions assume the periodic exchange of beacons in the neighborhood. This is essential for nodes to learn about the moving away of the downstream nodes in the multicast tree. The following sections describe the working of the two multicast extensions in detail. Unless otherwise stated specifically, the description holds good for the both NR-MLPBR and R-LPBR. A multicast group comprises of nodes that wish to receive data packets from an arbitrary source, which is not part of the group.

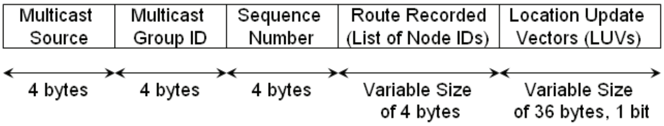
4.1 Broadcast of multicast tree request messages

Whenever a source node has data packets to send to a multicast group and is not aware of a multicast tree to the group, the source initiates a broadcast tree discovery procedure by broadcasting a Multicast Tree Request Message (MTRM) to its neighbors. Each node, including the receiver nodes of the multicast group, on receiving the first MTRM of the current broadcast process (i.e., a MTRM with a sequence number greater than those seen before), includes its Location Update Vector, LUV in the MTRM packet. The LUV of a node comprises the following: node ID, X, Y co-ordinate information, Is Receiver flag, Current velocity and Angle of movement with respect to the X-axis. The *Is Receiver* flag in the LUV, if set, indicates that the node is a receiving node of the multicast group. The node ID is also appended on the "Route record" field of the MTRM packet. The structure of the LUV and the MTRM is shown in Figures 14 and 15 respectively.



Node ID	X Co-ordinate	Y Co-ordinate	Node Velocity	Angle of Movement	Is Receiver
4 bytes	8 bytes	8 bytes	8 bytes	8 bytes	1 bit

Fig. 14. Location Update Vector (LUV) per Node



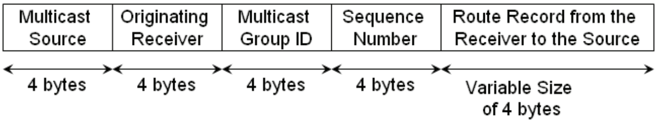
Multicast Source	Multicast Group ID	Sequence Number	Route Recorded (List of Node IDs)	Location Update Vectors (LUVs)
4 bytes	4 bytes	4 bytes	Variable Size of 4 bytes	Variable Size of 36 bytes, 1 bit

Fig. 15. Structure of the Multicast Tree Request Message

4.2 Construction of the multicast tree

Paths constituting the multicast tree are independently chosen at each receiver node. A receiver node gathers several MTRMs obtained across different paths and selects the minimum hop path among them by looking at the "Route Record" field in these MTRMs. A Multicast Tree Establishment Message (MTEM) is sent on the discovered minimum hop route to the source. The MTEM originating from a receiver node has the list of node IDs corresponding to the nodes that are on the minimum hop path from the receiver node to the source (which is basically the reverse of the route recorded in the MTRM). The structure of the MTEM packet is shown in Figure 16.

An intermediate node upon receiving the MTEM packet checks its multicast routing table whether there exist an entry for the <Multicast Source, Multicast Group ID> in the table. If an entry exists, the intermediate node merely adds the tuple <One-hop sender of the MTEM, Originating Receiver node of the MTEM> to the list of <Downstream node, Receiver node> tuples for the multicast tree entry and does not forward the MTEM further. The set of downstream nodes are part of the multicast tree rooted at the source node for the multicast group. If a <Multicast Source, Multicast Group ID> entry does not exist in the multicast table, the intermediate node creates an entry and initializes it with the <One-hop sender of the MTEM, Originating Receiver node of the MTEM> tuple. For each MTEM received, the source adds the neighbor node that sent the MTEM and the corresponding Originating Receiver node to the list of <Downstream node, Receiver node> tuples for the group.



Multicast Source	Originating Receiver	Multicast Group ID	Sequence Number	Route Record from the Receiver to the Source
4 bytes	4 bytes	4 bytes	4 bytes	Variable Size of 4 bytes

Fig. 16. Structure of Multicast Tree Establishment Message

4.3 Multicast tree acquisition and data transmission

After receiving the MTEMs from all the receivers within the Tree Acquisition Time (*TAT*), the source starts sending the data packets on the multicast tree. The *TAT* is based on the maximum possible diameter of the network (an input parameter in our simulations). The diameter of a network is the maximum of the hop count of the minimum hop paths between any two nodes in the network. The *TAT* is dynamically set at a node based on the time it took to receive the first MTEM for a broadcast tree discovery procedure. The structure of the header of the multicast data packet is shown in Figure 17. In addition to regular fields like Multicast Source, Multicast Group ID and Sequence Number, the header of the multicast data packet includes three specialized fields: the 'More Packets' (*MP*) field, the 'Current Dispatch Time' (*CDT*) field and the 'Time Left for Next Dispatch' (*TNLD*) field. The *CDT* field stores the time as the number of milliseconds lapsed since Jan 1, 1970, 12 AM. These additional overhead (relative to that of the other ad hoc multicast protocols) associated with the header of each data packet amounts to only 12 more bytes per data packet.

The source sets the *CDT* field in all the data packets sent. If the source has any more data to send, it sets the *MP* flag to 1 and sets the appropriate value for the *TLND* field, which indicates the number of milliseconds since the *CDT*. If the source does not have any more data to send, it will set the *MP* flag to 0 and leaves the *TLND* field blank. As we assume the clocks across all nodes are synchronized, a receiver will be able to calculate the end-to-end delay for the data packet based on the time the packet reaches the node and the *CDT* field in the header of the data packet. An average end-to-end delay per data packet is maintained at the receiver for the current path to the source. If the source has set the MP flag, the receiver computes the 'Next Expected Packet Arrival Time' (*NEPAT*), as the *CDT* field + *TLND* field + 2*Average end-to-end delay per data packet. A timer is started for the *NEPAT* value.

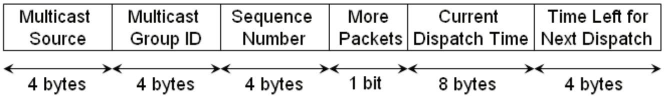


Fig. 17. Structure of the Header of the Multicast Data Packet

4.4 Multicast tree maintenance

If an intermediate node notices that its link with a downstream node has failed (i.e., the two nodes have moved away and are no longer neighbors), the intermediate node generates and sends a Multicast Path Error Message (MPEM) to the source of the multicast group entry. The MPEM has information about the receiver nodes affected (obtained from the multicast routing table) because of the link failure with the downstream node. Figure 18 shows the structure of an MPEM. The intermediate node removes the tuple(s) corresponding to the downstream node(s) and the affected receiver node(s). After these deletions, if no more <Downstream node, Receiver node> tuple exists for a <Source node, Multicast group ID> entry, the intermediate node removes the entire row for this entry from the routing table.

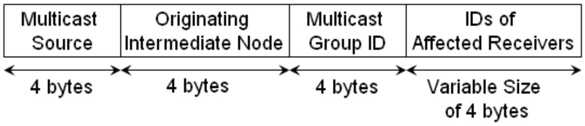


Fig. 18. Structure of a Multicast Path Establishment Message (MPEM)

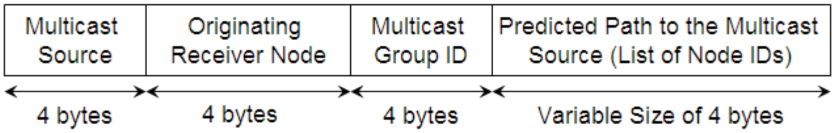
The source, upon receiving the MPEM, will wait to receive a Multicast Predicted Path Message (MPPM) from each of the affected receivers, within a MPPM-timer maintained for each receiver. The source estimates a Tree-Repair Time (*TRT*) for each receiver as the time that lapsed between the reception of the MPEM from an intermediate node and the MPPM from the affected receiver. An average value for the TRT per receiver is maintained at the source as it undergoes several path failures and repairs before the next global broadcast based tree discovery. The MPPM-timer (initially set to the time it took for the source to receive the MTEM from the receiver) for a receiver will be then set to 1.5* Average *TRT* value, so that we give sufficient time for the destination to learn about the route failure and generate a new MPPM. Nevertheless, this timer will be still far less than the tree acquisition time that would be incurred if the source were to launch a global broadcast tree discovery. Hence, our approach will only increase the network throughput and does not decrease it.

4.5 Prediction of node location using the Location Update Vectors (LUVs)

If a receiver does not receive the data packet within the *NEPAT* time, it will attempt to locally construct the global topology using the location and mobility information of the nodes learnt from the latest broadcast tree discovery. The procedure to predict the location of a node at a time instant *CTIME* based on the LUV gathered from node *u* at time *STIME* is the same as that explained in Section 2.3. The two multicast extensions, NR-MLPBR and R-MLPBR, differ from each other on the nature of the paths predicted at the receiver.

4.6 NR-MLPBR: Multicast path prediction

The receiver node locally runs the Dijkstra's minimum hop path algorithm (Cormen, 2001) on the predicted global topology. If at least one path exists from the source node to the receiver node in the generated topology, the algorithm returns the minimum hop path among them. The receiver node then sends a MPPM (structure shown in Figure 19) on the discovered path with the route information included in the message.



Multicast Source	Originating Receiver Node	Multicast Group ID	Predicted Path to the Multicast Source (List of Node IDs)
4 bytes	4 bytes	4 bytes	Variable Size of 4 bytes

Fig. 19. Structure of the Multicast Predicted Path Message (MPPM)

4.7 R-MLPBR: Multicast path prediction

The receiver node uses the LUV obtained from each of the intermediate nodes during the latest global tree broadcast discovery process to learn about the identification (IDs) of its peer receiver nodes that are part of the multicast group. If there existed a direct path to the source on the predicted topology, the receiver node chooses that path as the predicted path towards the source. Otherwise, the receiver node determines a set of node-disjoint paths on the predicted global topology. The node-disjoint paths to the source are ranked depending on the number of non-receiver nodes that act as intermediate nodes on the path. The path that has the least number of non-receiver nodes as intermediate nodes is preferred. The reason is a path that has the least number of non-receiver nodes is more likely to be a minimum hop path and if a receiver node lies on that path, the number of newly added links to the tree would also be reduced. R-MLPBR thus aims to discover paths with the minimum hop count and at the same time attempts to conserve bandwidth by reducing the number of links that get newly added to the tree as a result of using the predicted path. The MPPM is hence sent on the predicted path that has minimum number of non-receiver nodes. If two or more paths has the same minimum number of non-receiver nodes, R-MLPBR breaks the tie by choosing the path with the minimum hop count to the source.

Note that R-MLPBR chooses the path with the minimum number of non-receiver nodes, rather than the path with the maximum number of receiver nodes, as the latter design has the possibility of yielding paths with significantly larger hop count from the source to the receiver node without any guarantee on the possible reduction in the number of links. Our choice of choosing the path with the minimum number of non-receiver nodes helps to maintain the hop count per source-receiver path close to the minimum hop count and at the same time does helps to reduce the number of links in the tree to a certain extent.

4.8 Propagation of the multicast predicted path message towards the source

An intermediate node on receiving the MPPM adds the tuple <One-hop sender of the MPPM, Originating Receiver node of the MPPM> to the list of <Downstream node, Receiver node> tuples for the multicast tree entry corresponding to the source node and the multicast group to which the MPPM belongs to. The MPPM is then forwarded to the next downstream node on the path towards the source. If the source receives the MPPM from the appropriate receiver before the MPPM-timer expires, it indicates that the predicted path does exist in reality. A costly global broadcast tree discovery has been thus avoided. If an intermediate node could not successfully forward the MPPM to the next node on the path towards the source, it informs the receiver node of the absence of the route through a

MPPM-Error packet. The receiver node on receiving the MPPM-Error packet discards all the LUVs and does not generate any new MPPM. After the MPPM-timer expires, the multicast source initiates a new global broadcast-based tree discovery procedure.

5. Simulation performance study of NR-MLPBR and R-MLPBR

The network dimension used is a 1000m x 1000m square network. The transmission range of each node is assumed to be 250m. The number of nodes used in the network is 25 and 75 nodes representing networks of low and high density with an average distribution of 5 and 15 neighbors per node respectively. Initially, nodes are uniformly randomly distributed in the network. We compare the performance of NR-MLPBR and R-MLPBR with that of the minimum-hop based Multicast Extension of Ad hoc On-demand Distance Vector (MAODV) routing protocol (Royer & Perkins, 1999) and the minimum-link based Bandwidth Efficient Multicast Routing Protocol (BEMRP) (Ozaki, et. al., 2001). We implemented all of these four multicast routing protocols in ns-2. The broadcast tree discovery strategy employed is the default flooding approach. The node mobility model used is the Random Waypoint model with each node starts moving from an arbitrary location to a randomly selected destination location at a speed uniformly distributed in the range [0,…,v_{max}]. The v_{max} values used are 10 m/s, 30 m/s and 50 m/s representing scenarios of low, moderate and high node mobility respectively. Pause time is 0 seconds. Simulations are conducted with a multicast group size of 2, 4 (small size), 8, 12 (moderate size) and 24 (larger size) receiver nodes. For each group size, we generated 5 lists of receiver nodes and simulations were conducted with each of them. Traffic sources are constant bit rate (CBR). Data packets are 512 bytes in size and the packet sending rate is 4 data packets/second. The multicast session continues until the end of the simulation time, which is 1000 seconds.

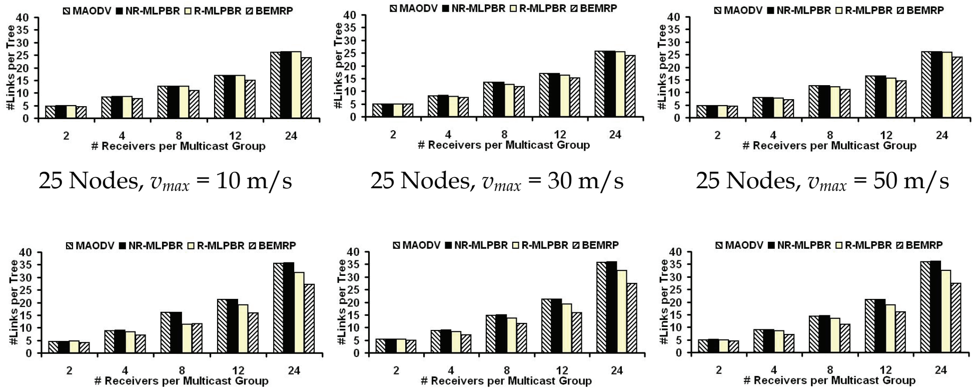
The performance metrics studied through this simulation are the following:

- Number of Links per Tree: This is the time averaged number of links in the multicast trees discovered and computed over the entire multicast session.
- Hop Count per Source-Receiver Path: This is the time averaged hop count of the paths from the source to each receiver of the multicast group and computed over the entire multicast session.
- Time between Successive Broadcast Tree Discoveries: This is the time between two successive broadcast tree discoveries, averaged over the entire multicast session. The larger the time between successive broadcast tree discoveries, the lower is the number of broadcast tree discoveries. This metric is a measure of the lifetime of the multicast trees discovered and also the effectiveness of the path prediction approach followed in NR-MLPBR and R-MLPBR.

The performance results for each metric displayed in Figures 20 through 22 are an average of the results obtained from simulations conducted with 5 sets of multicast groups and 5 sets of mobility profiles for each group size, node velocity and network density values. The multicast source in each case was selected randomly among the nodes in the network and the source is not part of the multicast group. The nodes that are part of the multicast group are merely the receivers.

5.1 Number of links per multicast tree

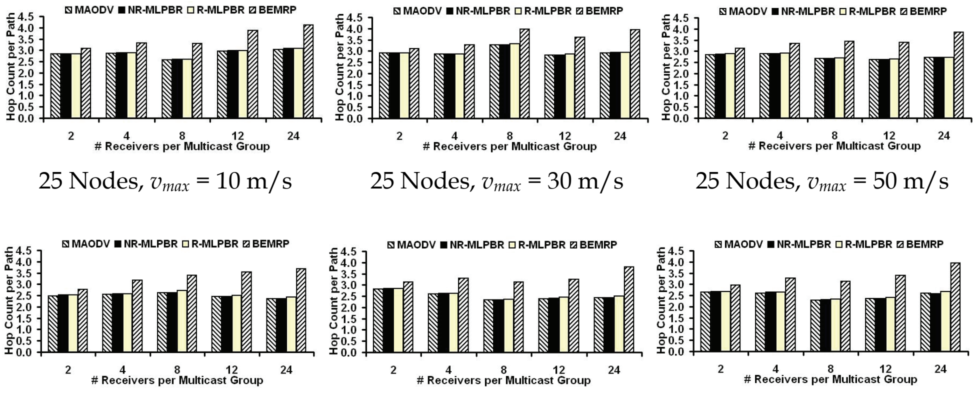
R-MLPBR manages to significantly reduce the number of links (Figure 20) vis-à-vis the MAODV and NR-MLPBR protocols without yielding to a higher hop count per source-receiver path. R-MLPBR is the first multicast routing protocol that yields trees with the reduced number of links and at the same time, with a reduced hop count (close to the minimum) per source-receiver path. However, R-MLPBR cannot discover trees that have minimum number of links as well as the minimum hop count per source-receiver path. BEMRP discovers trees that have a reduced number of links for all the operating scenarios. However, this leads to larger hop count per source-receiver paths for BEMRP (Figure 21).



25 Nodes, v_{max} = 10 m/s 25 Nodes, v_{max} = 30 m/s 25 Nodes, v_{max} = 50 m/s

75 Nodes, v_{max} = 10 m/s 75 Nodes, v_{max} = 30 m/s 75 Nodes, v_{max} = 50 m/s

Fig. 20. Average Number of Links per Multicast Tree



25 Nodes, v_{max} = 10 m/s 25 Nodes, v_{max} = 30 m/s 25 Nodes, v_{max} = 50 m/s

75 Nodes, v_{max} = 10 m/s 75 Nodes, v_{max} = 30 m/s 75 Nodes, v_{max} = 50 m/s

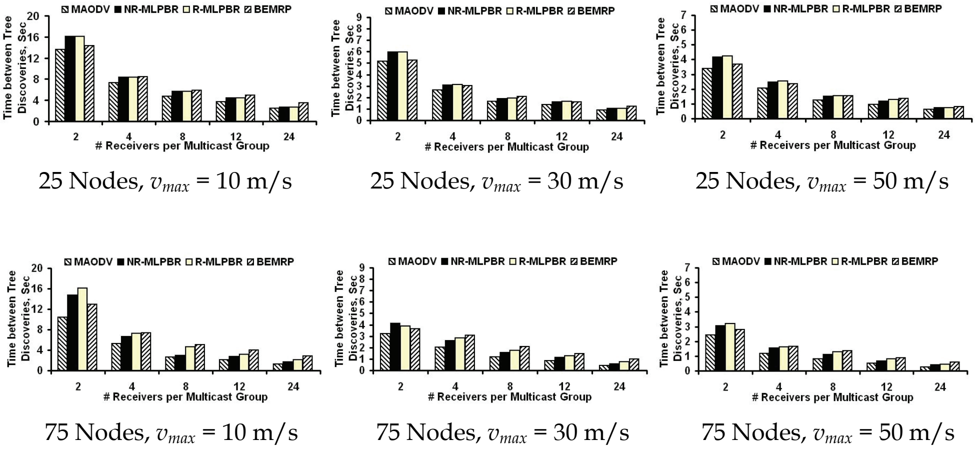
Fig. 21. Average Hop Count per Source-Receiver Path for a Multicast Session

5.2 Hop count per source-receiver path

All the three multicast routing protocols – MAODV, NR-MLPBR and R-MLPBR, incur almost the same average hop count per source-receiver path (refer Figure 21) and it is considerably lower than that incurred for BEMRP. The hop count per source-receiver path is an important metric and it is often indicative of the end-to-end delay per multicast packet from the source to a specific receiver. BEMRP incurs a significantly larger hop count per source-receiver path and this can be attributed to the nature of this multicast routing protocol to look for trees with a reduced number of links. When multiple receiver nodes have to be connected to the source through a reduced set of links, the hop count per source-receiver path is bound to increase. The hop count per source-receiver path increases significantly as we increase the multicast group size.

5.3 Time between successive broadcast tree discoveries

The time between successive broadcast tree discoveries (Figure 22) is a measure of the stability of the multicast trees and the effectiveness of the location prediction and path prediction approach of the two multicast extensions. For a given node density and node mobility, both NR-MLPBR and R-MLPBR incur relatively larger time between successive broadcast tree discoveries for smaller and medium sized multicast groups. MAODV tends to be more unstable as the multicast group size is increased, owing to the minimum hop nature of the paths discovered and absence of any path prediction approach. For larger multicast groups, the multicast trees discovered using BEMRP are relatively more stable by virtue of the protocol's tendency to strictly minimize only the number of links in the tree.



25 Nodes, v_{max} = 10 m/s

25 Nodes, v_{max} = 30 m/s

25 Nodes, v_{max} = 50 m/s

75 Nodes, v_{max} = 10 m/s

75 Nodes, v_{max} = 30 m/s

75 Nodes, v_{max} = 50 m/s

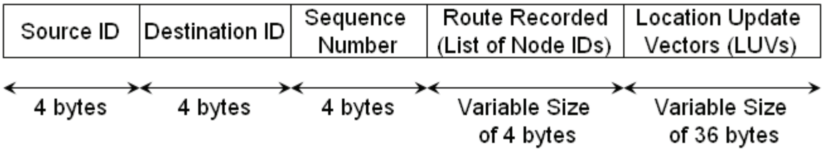
Fig. 22. Average Time between Successive Broadcast Tree Discoveries

6. Node-disjoint multi-path extension of LPBR (LPBR-M)

We define a multi-path between a source-destination (*s-d*) pair as the set of multiple paths between the source *s* and destination *d*. We now propose a multi-path extension for LPBR to discover node-disjoint multi-paths such that both the number of global broadcast multi-path discoveries as well as the hop count per *s-d* multi-path (average of the hop count of all the multiple node-disjoint paths of a multi-path) is simultaneously minimized. We assume that the clocks across all nodes are at least loosely synchronized. This is essential to ensure proper timeouts at the nodes for failure to receive a certain control message.

6.1 Broadcast of route request messages

Whenever a source node has data packets to send to a destination and is not aware of any path to the latter, the source initiates a broadcast route discovery procedure by broadcasting a Multi-path Route Request (MP-RREQ) message to its neighbors. Each node, except the destination, on receiving the first MP-RREQ of the current broadcast process (i.e., a MP-RREQ with a sequence number greater than those seen before), includes its Location Update Vector, LUV, in the MP-RREQ message. The LUV of a node (same as that in Figure 1) comprises the following: Node ID, X, Y co-ordinate information, Current velocity and Angle of movement with respect to the X-axis. The Node ID is also appended in the "Route Record" field of the MP-RREQ message (refer Figure 23).

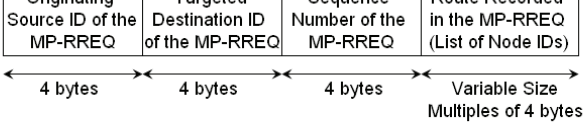


Source ID	Destination ID	Sequence Number	Route Recorded (List of Node IDs)	Location Update Vectors (LUVs)
4 bytes	4 bytes	4 bytes	Variable Size of 4 bytes	Variable Size of 36 bytes

Fig. 23. Multi-path Route Request (MP-RREQ) Message

6.2 Generation of the route reply messages

When the destination receives a MP-RREQ message, it extracts the path traversed by the message (sequence of Node IDs in the Route Record) and the LUVs of the nodes (including the source) that forwarded the message. The destination stores the paths learnt in a set, *RREQ-Path-Set*, maintained in the increasing order of their hop count. Ties between paths with the same hop count are broken in the order of the time of arrival of their corresponding MP-RREQ messages at the destination. The LUVs are stored in a LUV-Database maintained for the latest broadcast route discovery procedure initiated by the source. The destination runs a local path selection heuristic to extract the set of node-disjoint paths, *RREQ-ND-Set*, from the *RREQ-Path-Set*. The heuristic makes sure that except the source and the destination nodes, a node can serve as an intermediate node in at most only one path in the *RREQ-ND-Set*. The *RREQ-ND-Set* is initialized and updated with the paths extracted from the *RREQ-Path-Set* satisfying this criterion. In other words, a path *P* in the *RREQ-Path-Set* is added to the *RREQ-ND-Set* only if none of the intermediate nodes in *P* are already part of any of the paths in the *RREQ-ND-Set*. Once the *RREQ-ND-Set* is built, the destination sends a Multi-path Route Reply (MP-RREP) message for every path in the *RREQ-ND-Set*. An intermediate node receiving the MP-RREP message (refer Figure 24) updates its routing table by adding the neighbor that sent the message as the next hop on the path from the source to the destination. The MP-RREP message is then forwarded to the next node towards the source as indicated in the Route Record field of the message.



Originating Source ID of the MP-RREQ	Targeted Destination ID of the MP-RREQ	Sequence Number of the MP-RREQ	Route Recorded in the MP-RREQ (List of Node IDs)
4 bytes	4 bytes	4 bytes	Variable Size Multiples of 4 bytes

Fig. 24. Multi-path Route Reply (MP-RREP) Message

6.3 Multi-path acquisition time and data transmission

After receiving the MP-RREP messages from the destination within a certain time called the Multi-path Acquisition Time (*MP-AT*), the source stores the paths learnt in a set of node-disjoint paths, *NDP-Set*. The *MP-AT* is based on the maximum possible diameter of the network (an input parameter in our simulations). The diameter of the network is the maximum of the hop count of the minimum hop paths between any two nodes in the network. The *MP-AT* is dynamically set at a node depending on the time it took to receive the first MP-RREP for a broadcast discovery process.

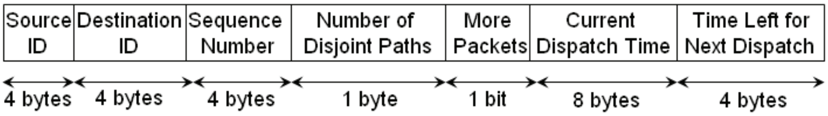


Fig. 25. Structure of the Data Packet

For data transmission, the source uses the path with the minimum hop count among the paths in the *NDP-Set*. In addition to the regular fields of source and destination IDs and the sequence number, the header of the data packet (refer Figure 25) includes four specialized fields: the ‘Number of Disjoint Paths’ field that indicates the number of active node-disjoint paths currently being stored in the *NDP-Set* of the source, the ‘More Packets’ (*MP*) field, the ‘Current Dispatch Time’ (*CDT*) field and the ‘Time Left for Next Dispatch’ (*TNLD*) field. The *CDT* field stores the time as the number of milliseconds lapsed since Jan 1, 1970, 12 AM. These additional overhead (relative to the other routing protocols) associated with the header is only 13 more bytes per data packet.

The source sets the *CDT* field in all the data packets sent. In addition, if the source has any more data to send, it sets the *MP* flag to 1 and sets the appropriate value for the *TLND* field, which indicates the number of milliseconds since the *CDT*. If the source does not have any more data to send, it will set the *MP* flag to 0 and leaves the *TLND* field blank. As we assume the clocks across all nodes are at least loosely synchronized, the destination uses the *CDT* field in the header of the data packet and the time of arrival of the packet to update the average end-to-end delay per data packet for the set of multi-paths every time after receiving a new data packet on one of these paths. If the MP flag is set, the destination computes the ‘Next Expected Packet Arrival Time’ (*NEPAT*), which is *CDT* field + *TLND* field + 2**NDP-Set Size**Average end-to-end delay per packet. A timer is started for the *NEPAT* value. To let the destination to wait until the source manages to successfully route a packet along a path in the *NDP-Set*, the *NEPAT* time takes the *NDP-Set Size* into account.

6.4 Multi-path maintenance

If an intermediate node could not forward the data packet due to a broken link, the upstream node of the broken link informs about the broken route to the source node through a Multi-path-Route-Error (MP-RERR) message, structure shown in Figure 26. The source node on learning the route failure will remove the failed path from its *NDP-Set* and attempt to send data packet on the next minimum-hop path in the *NDP-Set*. If this path is actually available in the network at that time instant, the data packet will successfully propagate its way to the destination. Otherwise, the source receives a MP-RERR message on the broken path, removes the failed path from the *NDP-Set* and attempts to route the data packet on the next minimum hop path in the *NDP-Set*. This procedure is repeated until the source does not receive a MP-RERR message or runs out of an available path in the *NDP-Set*. In the former case, the data packet successfully reaches the destination and the source continues to transmit data packets as scheduled. In the latter case, the source is not able to successfully transmit the data packet to the destination.

Node originating the MP-RERR message	Source ID of the Data packet dropped	Destination ID of the Data packet dropped	Sequence Number of the Data packet dropped	Downstream Node with which the link failed
4 bytes	4 bytes	4 bytes	4 bytes	4 bytes

Fig. 26. Multi-path Route Error (MP-RERR) Message

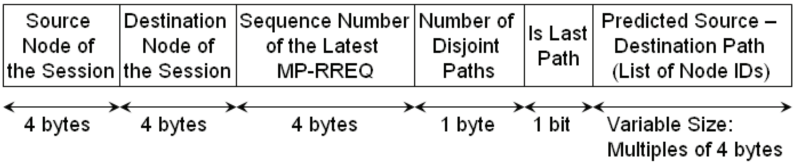
Before initiating another broadcast route discovery procedure, the source will wait for the destination node to inform it of a new set of node-disjoint routes through a sequence of MP-LPBR-RREP messages. The source will run a *MP-LPBR-RREP-timer* and wait to receive at least one MP-LPBR-RREP message from the destination. For the failure of the first set of node-disjoint paths, the value of this timer would be set to the multi-path acquisition time (the time it took to get the first MP-RREP message from the destination since the inception of route discovery), so that we give sufficient time for the destination to learn about the route failure and generate a new sequence of MP-LPBR-RREP messages. For subsequent route-repairs, the *MP-LPBR-RREP-timer* will be set based on the time it takes to get the first MP-LPBR-RREP message from the destination.

6.5 LPBR-M: Multi-path prediction

If a destination node does not receive the data packet within the *NEPAT* time, it will attempt to locally construct the global topology using the location and mobility information of the nodes learnt from the latest broadcast tree discovery. The procedure to predict the location of a node (say node *u*) at a time instant *CTIME* based on the LUV gathered from node *u* at time *STIME* is the same as that explained in Section 2.3. The destination locally runs the algorithm for determining the set of node-disjoint paths (Meghanathan, 2007) on the predicted global topology. The algorithm is explained as follows: Let *G* (*V*, *E*) be the graph representing the predicted global topology, where *V* is the set of vertices and *E* is the set of edges in the predicted network graph. Let P_N denote the set of node-disjoint *s*-*d* paths between source *s* and destination *d*. To start with, we run the O($|V|^2$) Dijkstra algorithm (Cormen, 2001) on *G* to determine the minimum hop *s*-*d* path. If there is at least one *s*-*d* path in *G*, we include the minimum hop *s*-*d* path *p i*n the set P_N. We then remove all the intermediate nodes (nodes other than source *s* and destination *d*) that were part of the minimum-hop *s*-*d* path *p* in the original graph G to obtain the modified graph *G′* (*V′*, *E′*). We then determine the minimum-hop *s*-*d* path in *G′* (*V′*, *E′*), add it to the set P_N and remove the intermediate nodes that were part of this *s*-*d* path to get a new updated *G′* (*V′*, *E′*). We repeat this procedure until there exists no more *s*-*d* paths in the network. The set P_N contains the node-disjoint *s*-*d* paths in the original network graph *G*. Note that when we remove a node from a network graph, we also remove all the links associated with the node.

6.6 MP-LPBR-RREP message propagation and handling prediction failure

The destination *d* sends a MP-LPBR-RREP message (refer Figure 27) to the source *s* on each of the predicted node-disjoint paths. Each intermediate node receiving the MP-LPBR-RREP message updates its routing table to record the incoming interface of the message as the outgoing interface for any new data packets received from *s* to *d*. The MP-LPBR-RREP message has a "Number of Disjoint Paths' field to indicate the total number of paths predicted and a 'Is Last Path' Boolean field that indicates whether or not the reported path is the last among the set of node-disjoint paths predicted. If the source *s* receives at least one MP-LPBR-RREP message before the *MP-LPBR-RREP-timer* expires, it indicates that the corresponding predicted *s-d* path on which the message propagated through does exists in reality. The source creates a new instance of the *NDP-Set* to store all the newly learnt node-disjoint *s-d* routes and sends data on the minimum hop path among them.



Source Node of the Session	Destination Node of the Session	Sequence Number of the Latest MP-RREQ	Number of Disjoint Paths	Is Last Path	Predicted Source – Destination Path (List of Node IDs)
4 bytes	4 bytes	4 bytes	1 byte	1 bit	Variable Size: Multiples of 4 bytes

Fig. 27. Structure of the MP-LPBR-RREP Message

The source node estimates the Route-Repair Time (*RRT*) as the time that lapsed between the reception of the last MP-RERR message from an intermediate node and the first MP-LPBR-RREP message from the destination. An average value of the *RRT* is maintained at the source as it undergoes several route failures and repairs before the next broadcast route discovery. The *MP-LPBR-RREP-timer* (initially set to the multi-path acquisition time) will be then set to 1.25*Average *RRT* value, so that we give sufficient time for the destination to learn about the route failure and generate a sequence of MP-LPBR-RREP messages.

If an intermediate node attempting to forward a MP-LPBR-RREP message of the destination could not successfully forward the message to the next node on the path towards the source, the intermediate node informs the absence of the route through a MP-LPBR-RREP-RERR message sent back to the destination. If the destination receives MP-LPBR-RREP-RERR messages for all the MP-LPBR-RREP messages initiated or the *NEPAT* time has expired, then the node discards all the LUVs and does not generate any new MP-LPBR-RREP message. The destination waits for the source to initiate a broadcast route discovery. After the *MP-LPBR-RREP-timer* expires, the source initiates a new broadcast route discovery.

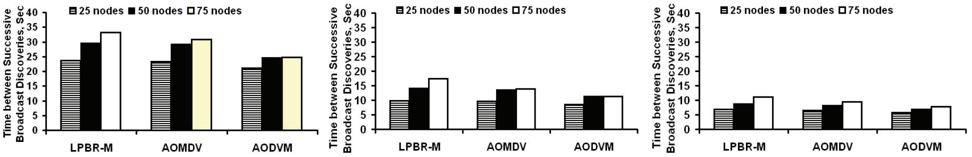
7. Simulation performance study of LPBR-M

We study the performance of LPBR-M through extensive simulations and also compare its performance with that of the link-disjoint path based AOMDV (Marina & Das, 2001) and the node-disjoint path based AODVM (Ye et. al., 2003) routing protocols. We implemented all these three multi-path routing protocols in ns-2. We use a 1000m x 1000m square network. The transmission range per node is 250m. The number of nodes used in the network is 25, 50 and 75 nodes representing networks of low, medium and high density with an average distribution of 5, 10 and 15 neighbors per node respectively. For each combination of network density and node mobility, simulations are conducted with 15 source-destination (*s-d*) pairs. Traffic sources are constant bit rate (CBR). Data packets are 512 bytes in size and the packet sending rate is 4 data packets/second. Simulation time is 1000 seconds. The node mobility model used is the Random Waypoint model (Bettstetter, 2004). During every direction change, the velocity of a node is uniformly and randomly chosen from the range [0,…,v_{max}] and the values of v_{max} used are 10, 30 and 50 m/s, representing node mobility levels of low, moderate and high respectively. The Medium-Access Control (MAC) layer model used is the IEEE 802.11 model (Bianchi, 2000) involving Request-to-Send (RTS) and Clear-to-Send (CTS) message exchange for coordinating channel access.

The performance metrics studied are the following:

- Time between Successive Broadcast Multi-path Route Discoveries: This is the time between two successive broadcast multi-path route discoveries, averaged for all the *s-d* sessions over the simulation time. We use a set of multi-paths as long as at least one path in the set exists, in increasing order of their hop count. We opt for a broadcast route discovery when all paths in a multi-path set fails. Hence, this metric is a measure of the lifetime of the multi-path set and a larger value is preferred for a routing protocol.
- Control Message Overhead: This is the ratio of the total number of control messages (MP-RREQ, MP-RREP, MP-LPBR-RREP and MP-LPBR-RREP-RERR) received at every node to that of the total number of data packets delivered at a destination, averaged over all the *s-d* sessions for the entire simulation time. In a typical broadcast operation, the total amount of energy spent to receive a control message at all the nodes in a neighborhood is greater than the amount of energy spent to transmit the message.
- Average Hop Count of all Disjoint-paths used: This is the time-averaged hop count of the disjoint paths determined and used by each of the multi-path routing protocols.

Each data point for the performance metrics in Figures 28 and 29 is an average of the results obtained from simulations conducted with 5 sets of mobility profiles of the nodes and 15 randomly picked *s-d* pairs, for each combination of node mobility and density.



v_{max} = 10 m/s v_{max} = 30 m/s v_{max} = 50 m/s

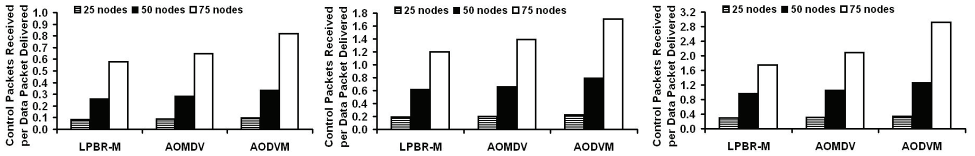
Fig. 28. Time between Successive Broadcast Multi-path Route Discoveries

7.1 Time between successive multi-path route discoveries

LPBR-M yields the longest time between successive broadcast multi-path route discoveries (refer Figure 28). Thus, the set of node-disjoint paths discovered and predicted by LPBR-M are relatively more stable than the set of link-disjoint and node-disjoint paths discovered by the AOMDV and AODVM routing protocols respectively. As we increase node mobility, the difference in the time between successive multi-path route discoveries incurred for AOMDV and AODVM vis-à-vis LPBR-M increases. Also, for a given level of node mobility, as we increase the network density, the time between successive route discoveries for LPBR-M increases relatively faster compared to those incurred for AOMDV and AODV-M.

7.2 Control message overhead

For a given level of node mobility and network density, LPBR-M incurs the lowest control message overhead (refer Figure 29). For a given level of node mobility, AOMDV and AODVM respectively incur 4%-16% and 14%-34% more control message overhead than LPBR-M when flooding is used. In networks of moderate node mobility, the control message overhead incurred by the three multi-path routing protocols while using flooding is 2.1 (high density) to 3.4 (low density) times more than that incurred in networks of low node mobility. In networks of high node mobility, the control message incurred by the three multi-path routing protocols while using flooding is 3.0 (high density) to 3.7 (low density) times more than that incurred in networks of low node mobility.



v_{max} = 10 m/s v_{max} = 30 m/s v_{max} = 50 m/s

Fig. 29. Control Message Overhead for LPBR-M, AOMDV and AODVM

7.3 Average hop count per multi-path

For a given routing protocol and network density, the average hop count of the disjoint-paths used is almost the same, irrespective of the level of node mobility. As we add more nodes in the network, the hop count of the paths tends to decrease as the source manages to reach the destination through relatively lesser number of intermediate nodes. With increase in network density, there are several candidates to act as intermediate nodes on a path. The average hop count of the paths in high and moderate density networks is 6%-10% less than the average hop count of the paths in networks of low density. The average hop count for all the three multi-path routing protocols is almost the same.

8. Conclusions

This chapter discusses the design of a location prediction based routing protocol (LPBR) and its extensions for multicast and multi-path routing in mobile ad hoc networks (MANETs). The aim of each category of the LPBR protocols is to simultaneously minimize the number of times the underlying communication structures (single path, tree or multi-paths) are discovered through a global broadcast discovery as well as the hop count of the paths and/or the number of links that are part of these communication structures. Simulation performance results indicate that the number of broadcast route discoveries incurred with LPBR is significantly lower than that incurred with the best stable path routing protocol (FORP) known in the literature and at the same time, the hop count per path is only at most 12% more than that of the most commonly used minimum-hop based routing protocol (DSR). The time between successive LPBR route discoveries can be as large as 50-100% and 120-220% more than that incurred with FORP and DSR respectively. The receiver-aware multicast extension of LPBR (R-MLPBR) manages to significantly reduce the number of multicast tree discoveries with very minimal increase (as large as only 20%) in the hop count per source-receiver path and the number of links per multicast tree. The non receiver-aware multicast extension of LPBR (NR-MLPBR) determines multicast trees that have hop count very close to that of the minimum-hop based MAODV protocol, albeit with a reduced number of broadcast tree discoveries. The node-disjoint multi-path extension of LPBR (LPBR-M) reduces the number of multi-path broadcast route discoveries to as large as 44% compared to AOMDV and AODVM and at the same time, incurs a hop count that is very much the same as these two multi-path routing protocols.

All of these performance results indicate the effectiveness of the location prediction approach in LPBR. The rationale behind the success in re-discovering routes and trees (using location prediction) without often going through a broadcast discovery process is that two nodes that form a link in the actual network may not exactly be positioned at the location predicted; but the predicted locations are close enough to include the a link in the global topology locally predicted at the destination node. Another notable characteristic of LPBR and its extensions is that the location information of the nodes is not periodically disseminated offline through a location service mechanism; instead, the location information is disseminated along with the route discovery control messages. As there exist no single unicast single path or multi-path/ multicast routing protocol that can simultaneously minimize the number of route discoveries as well as the hop count per path and/or the number of links per tree, LPBR and its multicast and multi-path extensions are a valuable addition to the MANET literature.

9. Acknowledgments

The research on the multicast and multi-path extensions of the Location Prediction Based Routing Protocol was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-08-2-0061. The views and conclusions in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

10. References

Bettstetter, C.; Hartenstein, H. & Perez-Costa, X. (2004). Stochastic Properties of the Random Way Point Mobility Model. *Wireless Networks*, Vol. 10, No. 5, (September 2004), pp. 555-567, ISSN: 10220038.

Bianchi, G. (2000). Performance Analysis of the IEEE 802.11 Distributed Coordination Function. *IEEE Journal of Selected Areas in Communications*, Vol. 18, No. 3 (March 2000), pp. 535-547, ISSN: 07338716.

Breslau, L.; Estrin, D.; Fall, K.; Floyd, S.; Heidemann, J.; Helmy, A.; Huang, P.; McCanne, S.; Varadhan, K.; Xu, Y.; Yu, H. (2000). Advances in Network Simulation. *IEEE Computer*, Vol. 33, No. 5 (May 2000), pp. 59-67, ISSN: 00189162.

Broch, J.; Maltz, D. A.; Johnson, D. B.; Hu, Y. C. & Jetcheva, J. (1998). A Performance Comparison of Multi-hop Wireless Ad Hoc Routing Protocols, *Proceedings of the 5th ACM Annual Conference on Mobile Computing and Networking (MOBICOM),* pp. 85-97, ISBN: 158113035X, October 1998, Dallas, TX, USA.

Cormen, T. H.; Leiserson, C. E.; Rivest, R. L. & Stein, C. (2001). *Introduction to Algorithms*, 2nd Edition, MIT Press, ISBN: 0262032937.

Hofmann-Wellenhof, B.; Lichtenegger, H. & Collins, J. (2004). *Global Positioning System*, 5th Edition, Springer, ISBN: 978-3211835340.

Johnson, D. B.; Maltz, D. A. & Broch, J. (2001). DSR: The Dynamic Source Routing Protocol for Multi-hop Wireless Ad hoc Networks, In: Ad hoc Networking, Charles E. Perkins, (Ed.), 139 – 172, Addison Wesley, ISBN: 0201309769.

Ko, Y.-B. & Vaidya, N. H. (2000). Location-Aided Routing (LAR) in Mobile Ad hoc Networks. *Wireless Networks*, Vol. 6, No. 4 (July 2000), pp. 307-321, ISSN: 10220038.

Lim, G.; Shin, K.; Lee, S.; Yoon, H.; & Ma, J. S. (2002). Link Stability and Route Lifetime in Ad hoc Wireless Networks, *Proceedings of the International Conference on Parallel*

Processing Workshops, pp. 116-123, ISBN: 0769516777, August 2002, Vancouver, Canada.

Marina, M. K. & Das, S. R. (2001). On-demand Multi-path Distance Vector Routing in Ad hoc Networks, *Proceedings of the IEEE International Conference on Network Protocols*, pp. 14-23, ISBN: 0769514294, November 2001, Riverside, CA, USA.

Meghanathan, N. (2007). Stability and Hop Count of Node-Disjoint and Link-Disjoint Multi-path Routes in Ad Hoc Networks, *Proceedings of the 3rd IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, p. 42, ISBN: 0769528899, October 2007, White Plains, NY, USA.

Meghanathan, N. (2008). Exploring the Stability-Energy Consumption-Delay-Network Lifetime Tradeoff of Mobile Ad hoc Network Routing Protocols. *Journal of Networks*, Vol. 3, No. 2, (February 2008), pp. 17-28, ISSN: 17962056.

Meghanathan, N. (2009a). A Location Prediction Based Reactive Routing Protocol to Minimize the Number of Route Discoveries and Hop Count per Path in Mobile Ad hoc Networks. *The Computer Journal*, Vol. 52, No. 4, (July 2009), pp. 461–482, ISSN: 00104620.

Meghanathan, N. (2009b). Multicast Extensions to the Location Prediction Based Routing Protocol for Mobile Ad hoc Networks. *ISAST Transactions on Computers and Intelligent Systems*, Vol. 1, No. 1, (August 2009), pp. 56–65, ISSN: 17982448.

Meghanathan, N. (2009c). A Node-Disjoint Multi-path Extension of the Location Prediction Based Routing Protocol for Mobile Ad hoc Networks, *Proceedings of the 3rd International Conference on Signal Processing and Communication Systems,* ISBN: 9781424444731, September 2009, Omaha, NE, USA.

Meghanathan, N. (2009d). Survey and Taxonomy of Unicast Routing Protocols for Mobile Ad hoc Networks. *The International Journal on Applications of Graph Theory in Wireless Ad hoc Networks and Sensor Networks*, Vol. 1, No. 1, (December 2009), pp. 1 – 21, ISSN: 09757260.

Ozaki, T.; Kim, J.-B. & Suda, T. (2001). Bandwidth-Efficient Multicast Routing for Multihop, Ad hoc Wireless Networks, *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Vol. 2, pp. 1182-1192, ISBN: 0780370163, August 2002, Anchorage, AK, USA.

Perkins, C. E. & Royer, E. M. (1999). Ad hoc On-demand Distance Vector Routing, *Proceedings of the 2nd Annual IEEE International Workshop on Mobile Computing Systems and Applications*, pp. 90–100, ISBN: 0769500250, February 1999, New Orleans, LA, USA.

Royer, E. M. & Perkins, C. E. (1999). Multicast Operation of the Ad hoc On-demand Distance Vector Routing Protocol, *Proceedings of the 5th ACM Annual Conference on Mobile Computing and Networking (MOBICOM),* pp. 207-218, ISBN: 1581131429, August 1999, Seattle, WA, USA.

Su, W.; Lee, S.-J. & Gerla, M. (2001). Mobility Prediction and Routing in Ad hoc Wireless Networks. *International Journal of Network Management*, Vol. 11, No. 1, (Jan-Feb. 2001), pp. 3-30, ISSN: 10991190.

Ye, Z.; Krishnamurthy, S. V. & Tripathi, S. K. (2003). A Framework for Reliable Routing in Mobile Ad hoc Networks, *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Vol. 1, pp. 270-280, ISBN: 0780377524, March-April 2003, San Francisco, CA, USA.

13

An Adaptive Broadcasting Scheme in Mobile Ad Hoc Networks

Dimitrios Liarokapis, and Ali Shahrabi
Glasgow Caledonian University
United Kingdom

1. Introduction

Mobile and static nodes in battlefields or within the vicinity of disaster areas may not depend on fixed infrastructure for communication. To rapidly provide the required communication between the nodes in such environments, a Mobile Ad hoc Network (MANET) is the only available platform. With no fixed infrastructure, the efficient use of MANETs resources is highly crucial for the successful communication between mobile nodes. In situations where both the transmitting and the receiving nodes are placed within the transmission range of each other, communication is possible through a single-hop connection. In all other scenarios where the nodes are distanced, the exchange of packets is possible as long as a multi-hop path is available between them. Despite the unique characteristics of MANETs, they share many attributes and operations with other traditional networks. DNS lookups, exchange of control packets for management purposes and routing discovery requests are some examples of common operations, which all require broadcasting pieces of information across the network. However, due to lack of a centralised administrative and hardware, some modification is required to adopt broadcast operation for MANET environment.

The most straightforward broadcast mechanism used in MANETs is Simple Flooding (SF). The algorithmic procedure followed in SF is very simple, thus making its implementation and integration inside more complex operations fairly undiscomforting. In SF, upon reception of a broadcast packet the receiver will check whether or not this is a duplicate packet. If it is a new packet it will immediately retransmit it to all of its neighbouring nodes. Simply flooding the entire network may be the fastest and easiest way for a node to broadcast information over the network but it has been found to be a very unreliable and resource inefficient mechanism leading to the Broadcast Storm Problem (Ni et al., 1999) especially in highly populated and dense networks.

Over the past few years many studies (Leng et al., 2004), (Zhu et al., 2004), (Qayyum et al., 2002), (Hsu et al., 2005), (Purtoosi et al., 2006), (Barrit et al., 2006), (Bauer et al., 2005) have proposed novel broadcast mechanisms to alleviate the effects of SF. Early works were focused on developing schemes where the rebroadcast decision is made based on fixed and pre-determined threshold values. Probability-Based (PB), Counter-Based (CB) and Distance-Based (DB) are three schemes which have been proposed based on the concept of introducing a threshold value. PB bases it rebroadcasting decision on a fixed probability value, CB decides it by counting the number of received duplicate packets and finally the rebroadcasting in DB is based on the distance between sender and receiver (Ni et al., 1999). All of these schemes were found to considerably improve the performance of the broadcast operation in various network topologies but they also introduced a new dependency. The threshold value to be selected in order to reach optimum overall network performance highly depends on traffic load volume and node population. The degree of dependency is such that in certain network topologies SF performs better than these schemes (Williams et al., 2002).

The development of threshold-based adaptive broadcast schemes has consequently been considered to alleviate these dependencies. According to their algorithmic procedures, these schemes adaptively adjust the threshold value to be used depending on local information with regards to the density of the network within the transmission range of the sender (number of one-hop neighbours) or within a broader network area (number of two- hop neighbours) or even within the entire topology. Hence, all the schemes can be categorized based on the mechanism used in order to implement adaptivity. The most commonly used mechanism implies all nodes to periodically exchange HELLO packets with their neighbouring nodes in order to calculate density (Ryu et al., 2004), (Lee et al., 2006), (Chen et al., 2002), (Colagrosso 2007), (Chen et al., 2003), (Kyasanur et al., 2006), (Tseng et al., 2003). Alternatively, the other group of adaptive broadcast schemes utilise a positioning system, e.g. GPS, resulting in the construction of a network map for every mobile node, calculating in that way a very precise value for the density of the network (Deng et al., 2006). These schemes either introduce more overhead traffic to the network or demand the existence of expensive and fairly unreliable positioning systems.

In this chapter, a novel Distance-Based Adaptive (DibA) scheme is proposed. Based on the Distance-Based broadcast scheme, DibA implements adaptivity by dynamically adjusting the distance threshold value for every rebroadcast operation independently. Knowledge on local network densities is created on demand, without relying on HELLO packets or GPS systems, thus making DibA highly reliable avoiding at the same time the introduction of extra overhead traffic.

The remainder of this chapter is organised as follows. In Section 2 we overview related work. DibA as an adaptive broadcast mechanism is introduced in Section 3. In Section 4 the process of building a highly diverse network topology, where the performance of adaptive schemes can be evaluated appropriately is explained. The performance study is presented in Section 5. Finally, we make concluding remarks in Section 6.

2. Related works

In this section the Distance-Based scheme will be presented in detail, as our proposed scheme enhances this algorithm in order to make it locally adaptive. We will also discuss the general characteristics of other adaptive schemes and their methods.

2.1 Distance-Based scheme

DB is a broadcast mechanism that uses the distance between sender and receiver to make the decision whether to rebroadcast or not (Ni et al., 1999). The power of the received signal is a parameter that can be used to calculate the distance. GPS can also be used for that purpose. The specific algorithm for DB is presented in Fig. 1.

```
Algorithm: DB
Input: broadcast message (msg)
Output: decides whether to rebroadcast or not

S1. When a broadcast message, msg, is heard for the first time, initialize $d_{min}$ to the distance of the broadcasting node. If $d_{min} < D$ (where $D$ is the distance threshold), proceed to S5. In S2, if msg is heard again, interrupt the waiting and perform S4.
S2. Wait for a random number of slots. Then submit msg for transmission and wait until the transmission actually starts.
S3. The message is on the air. The procedure exits.
S4. Update $d_{min}$ if the distance to the host from which msg is heard is smaller. If $d_{min} < D$, proceed to S5. Otherwise, resume the waiting in S2.
S5. Cancel the transmission of msg if it was submitted in S2. The host is inhibited from rebroadcasting the message. Then exit.
```

Fig. 1. DB Algorithm

The distance threshold used in DB, is a parameter valued by default. This value is fixed and does not change unless there is administrative intervention. This is the major drawback of DB, as a static threshold may be appropriate for a network of specific density under particular circumstances. It could potentially cause poor network performance when the density or other network conditions greatly differ (Williams et al., 2002).

2.2 Adaptive schemes

Over the past few years, a growing number of studies have been trying to develop adaptive versions of DB. In order to achieve this, having the instantaneous knowledge of network configuration (in particular, the number of mobile nodes placed within the transmission range of each sender) is required. Currently, there are only two methods used to determine the local density for every individual node.

The first mechanism makes use of a positioning system such as Global Positioning Systems (GPS) (Deng et al., 2006). Mobile nodes periodically exchange messages including their exact coordinates. When a mobile node receives these coordinates it can calculate the distance from its current position and decides if the transmitting node is placed inside the transmission radius. In case that is true, the node increases its neighbours counter and therefore it can determine the level of network density locally. The use of expensive positioning systems, such as GPS, is the limitation of this approach.

According to the second mechanism (Ryu et al., 2004), (lee et al., 2006), (Chen et al., 2002), (Colagrosso 2007), (Chen et al., 2003), (Kyasanur et al., 2006), (tseng et al., 2003) the mobile nodes need to periodically send HELLO packets to all their neighbouring nodes and consequently count the number of responses they receive to measure the local density. It is obvious that this approach introduces a significant amount of overhead traffic in the network that could negatively affect the overall network performance, especially in cases where the network is highly populated and already overwhelmed with other types of traffic. In addition, one also needs to decide on the frequency of this procedure to take place. It should be remembered that although an increase in performance is the net result of introducing overhead (i.e. HELLO packets) and reducing overhead (i.e. fewer rebroadcasting), a frequent transmission of HELLO packets in static networks only increases the amount of overhead.

Although both supporting mechanisms exploit adaptivity, they also have significant drawbacks that could produce additional constraints. In the next section, we propose a novel broadcast algorithm which is neither relying on any positioning system nor introducing overhead traffic.

3. Distance-based Adaptive scheme (DibA)

To perform adaptively without introducing any further constraints and in order to decide whether or not to rebroadcast a message, any broadcast scheme requires to provide information about the local density of network for every node.

In our approach, we make use of Step 2 (S2) of the DB original algorithm, presented in Fig. 1, and make minor changes to Step 4 (S4). According to DB in S2, the receiving mobile node needs to wait for a random number of slots and remains in listening mode for duplicate broadcast packets. During that period of time, upon reception of a duplicate packet, it calculates the new distance and compares it with the distance threshold *D*.

We take advantage of this waiting period and calculate the number of duplicate packets received, using a simple counter which is updated in S4. The number of identical packets arriving at the mobile node is closely connected to the number of neighbouring nodes. Each time the value of the counter increases, the distance threshold is tuned according to a specific pattern.

The increase or decrease of the distance threshold is closely related to the potential additional coverage area that could be achieved when the broadcast packet is transmitted. If a large extra area is predicted to be covered by rebroadcasting of a packet, the distance threshold should be set to a low value. That is the case when the counter value is low. On the contrary, if the predicted coverage area is small, the distance threshold should be adjusted to a high value. This is also the case when the counter value is high. It is obvious that counter value, distance threshold and extra coverage area greatly affect one another in that order.

The DibA algorithm makes use of a scaled if statement for the adjustment of the distance. This should lead to an exponential increase of the distance threshold depending on the counter value. An example of the scaled if statement is as follows.

```
if(count = 1)
        D = 50m;
else if(count < 4)
        D = 125m;
else
        D =200m;
```

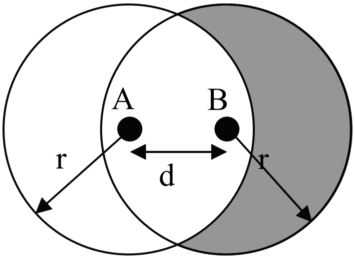


Fig. 2. Extra Coverage Area Analysis

In order to justify the reason why this pattern is used, we need to take into consideration the redundant rebroadcast analysis performed in (Ni et al., 1999). Consider the scenario in Fig. 2. Node A sends a broadcast packet and node B decides to rebroadcast it. Let S_A and S_B denote the circle areas covered by the transmission ranges of nodes A and B respectively. The gray area represents the additional area that will be covered by B's rebroadcast named S_{B-A}. We can derive that:

$$\left|S_{B-A}\right| = \pi r^2 - INTC(d)$$

where *INCT(d)* is the intersection area of the two circles centred at two points distanced by d.

$$INTC(d) = 4\int_{d/2}^{r} \sqrt{r^2 - x^2}\,dx$$

The extra coverage area gets the maximum value when r = d and is equal to:

$$\pi r^2 - INTC(r) = r^2\left(\frac{\pi}{3} + \frac{\sqrt{3}}{2}\right) \approx 0.61\pi r^2$$

Thus, B's rebroadcast can cover an extra area of 61% of the area covered by the previous transmission. The average extra coverage area can be obtained by integrating the above value over the circle of radius *x* centred at A for *x* in [0, r]:

$$\int_0^r \frac{2\pi x \cdot \left[\pi r^2 - INTC(x)\right]}{\pi r^2}dx \approx 0.41\pi r^2$$

A rebroadcast can cover an additional of 41% area in average. Following the same pattern, the extra area covered can be calculated depending on the number of transmissions heard for the broadcast packet. The result is shown in the graph of Fig. 3.

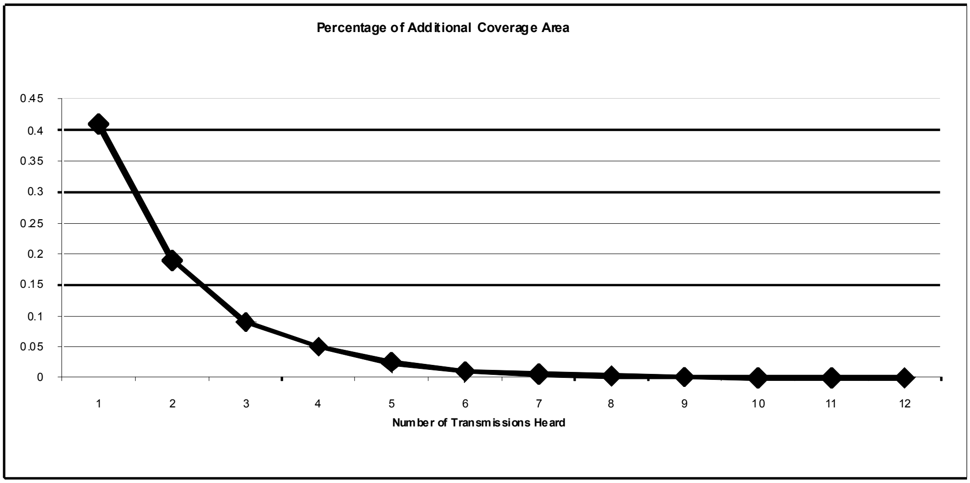


Fig. 3. Analysis of Redundant Rebroadcasts

```
Algorithm: DibA
Input: broadcast message (msg)
Output: decides whether to rebroadcast msg or not

S1. When a broadcast message msg is heard for the first time, initialize $d_{min}$ to the
    distance of the broadcasting node and the count to 1. If $d_{min} < D$ (where $D$ is the
    distance threshold), proceed to S5. In S2, if msg is heard again, interrupt the waiting,
    increase count by 1 and perform S4.
S2. Wait for a random number of slots. Then submit msg for transmission and wait
    until the transmission actually starts.
S3. The message is on the air. The procedure exits.
S4. Update $d_{min}$ if the distance to the host from which msg is heard is smaller.
     If count is less than $c_1$
     then D ← $D_1$
     else if count is less than $c_2$
     then D ← $D_2$
     else …
     ……...
     else if count is greater than $c_n$
     then D ← $D_n$
    If $d_{min} < D$, proceed to S5. Otherwise, resume the waiting in S2.
S5. Cancel the transmission of msg if it was submitted in S2. The host is inhibited from
    rebroadcasting message. Then exit.
```

Fig. 4. DibA Algorithm

The value of the distance threshold could change multiple times during the waiting period and every time a duplicate broadcast packet is received, the distance between sender and receiver is compared with the current value of the threshold. The details of DibA algorithm are presented in Fig. 4, where D is the distance threshold, *count* is the counter described above, D_1, D_2 … D_n are the predetermined threshold values and c_1, c_2 … c_n are predetermined counter values.

DibA's primary goal is not to calculate accurately the number of neighbouring nodes, but to decide upon the density level of the network locally inside the transmission radius. This feature gives an extra advantage to our approach in comparison to other adaptive schemes.

Let us consider part of a network topology as shown in Fig. 5. This is an extremely diverse topology as in the right part of the network only 1 node is placed. The left part of the network covers 12 nodes. All nodes have the same transmission range TR. The black node (BN) sends a broadcast message that will be received by all of its neighbouring nodes. In this example, the only neighbour of BN is the grey node (GN).

When we use one of the already existing adaptive schemes, GN will try to calculate the exact number of nodes inside the transmission radius. Either using GPS or HELLO packets, the end result of the calculation will be very close to 12, the total of all white nodes (WN) and BN. As a result, GN will decide that the network is very dense locally and tune the distance threshold to be high, in order to rebroadcast only if it is placed at the edge of BN's transmission range. In case that the distance between BN and GN is not large enough to exceed the tuned distance threshold (Fig. 6), GN will not rebroadcast. None of the WNs will receive the broadcast packet.

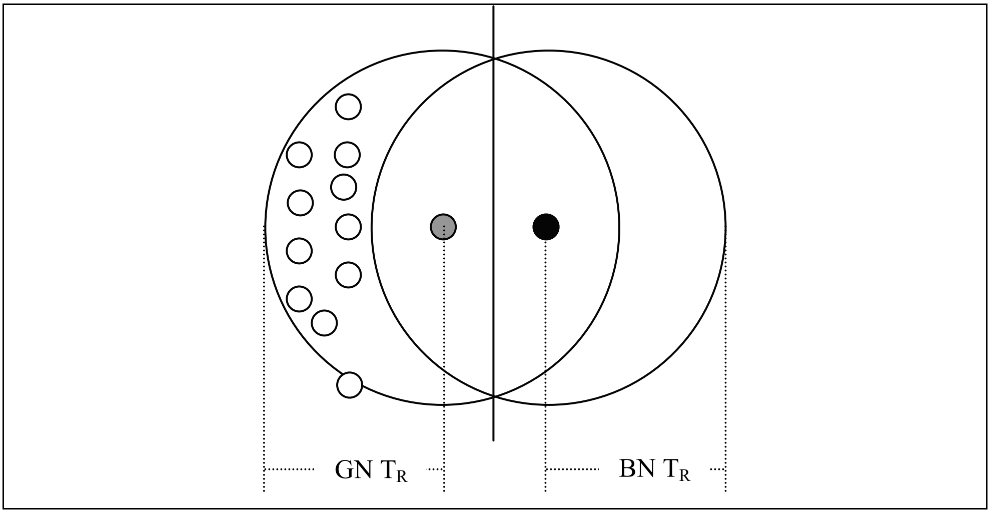


Fig. 5. Diverse Network Topology

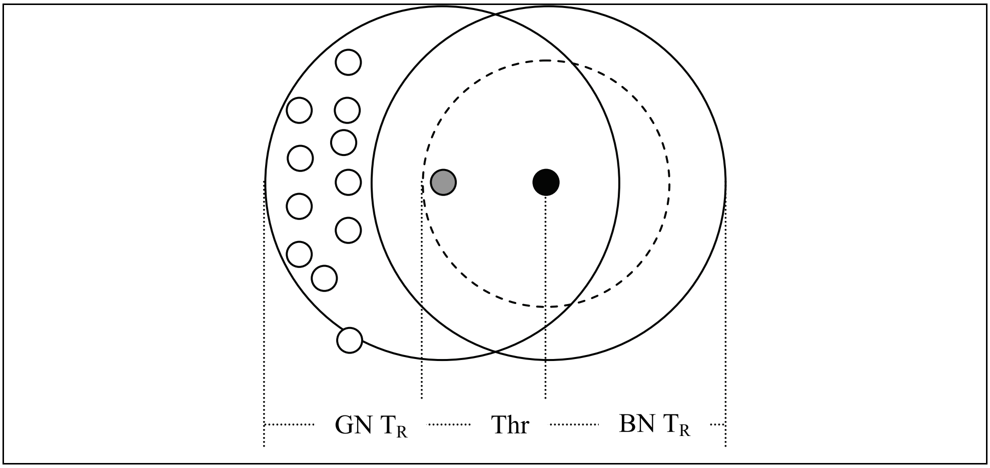


Fig. 6. Existing Adaptive Schemes

In case that DibA is used as the broadcast scheme, after reception, GN will wait for a random period of time counting duplicate packets. As BN is the only neighbour that has broadcasted the packet, GN exits the listening mode with the counter value of 1. It then assigns a very low value for the distance threshold. Now, it is highly possible at this point, as the threshold is very low, that GN is placed outside the dotted circle, as shown in Fig. 7. As a result, GN will rebroadcast the packet and all WNs will receive it.

In this example, we have shown that knowing the exact number of neighbouring nodes is not always ideal when trying to decide upon the appropriate value for the distance threshold. DibA measures the level of local density, depending on duplicate receptions and not on the knowledge about the amount of neighbours. Thus, it is highly reliable for both normal and extremely diverse network topologies.

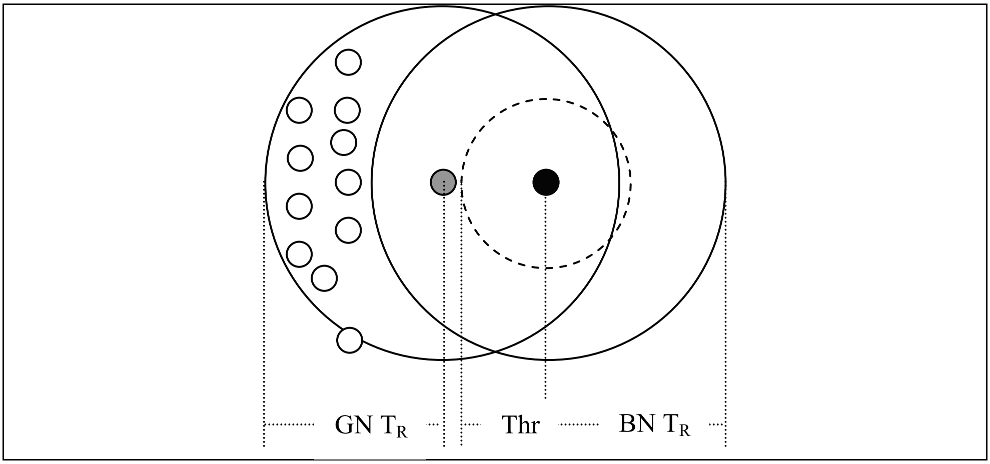


Fig. 7. DibA

4. Building a diverse network topology

Most of studies (Ni et al., 1999), (Leng et al., 2004), (Zhu et al., 2004), (Qayyum et al., 2002), (Hsu et al., 2005), (Purtoosi et al., 2006), (Barrit et al., 2006), (Ryu et al., 2004), (Lee et al., 2006), (Chen et al., 2002), (Colagrosso 2007), (Chen et al., 2003), (kyasanur et al., 2006), (Tseng et al., 2003) are relying on a simple network topology consisted of nodes distributed nearly evenly in an area when studying the performance of a broadcast scheme. However, the performance of any adaptive scheme is more appropriately demonstrated when tested on a diverse network topology, where part or parts of the network significantly differ in mobile nodes population volumes. In this section, we present the implementation of an automatic mechanism that can be used to create this kind of topologies.

The simulation tool that we use for our experiments is NS-2.30. NS-2 offers a single tool for creating mobility files using the setdest command. The user has the options to select the length and width of the topology, the number of nodes, pause time, maximum and minimum speed and simulation time. Unfortunately, setdest does not provide options to create more complex scenarios. However, the mobility files generated are of a simple text format, which gives us the opportunity to manually intervene inside the files and make appropriate changes.

The structure of the mobility file is as follows. Every node is assigned with its initial X, Y, Z coordinates in a command line. For example:

at 0.0 (time) node(0) 2.345 4.123 0.0

After all nodes are assigned initial coordinates, setdest randomly selects the time point where each node will change its direction and speed in order to reach a specific (X, Y, Z) point inside the topology. An example of such a command line is:

at 3.4567 (time) node (0) 4.899 13.756 10.392

Where the first parameter after “node(0)” (4.899) is the X coordinate for the reaching point, the second parameter (13.756) is the Y coordinate for the reaching point and the third parameter (10.392) is the speed of the mobile node. We have not included other parameters that are of no significance for the movement of the nodes in our examples.

We will explain how our mechanism works using a simple example. Let us consider the case where we want to create the topology presented in Fig. 8.

The nodes need to move inside their own half of the network. The fact that there is limitation of movement using borders helps to keep a balanced percentage of

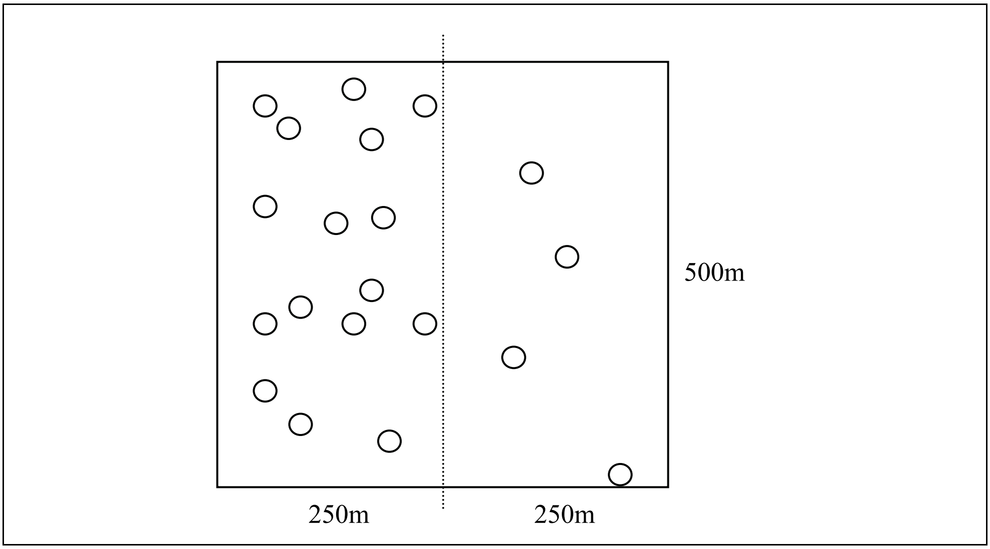


Fig. 8. A Sample Diverse Topology

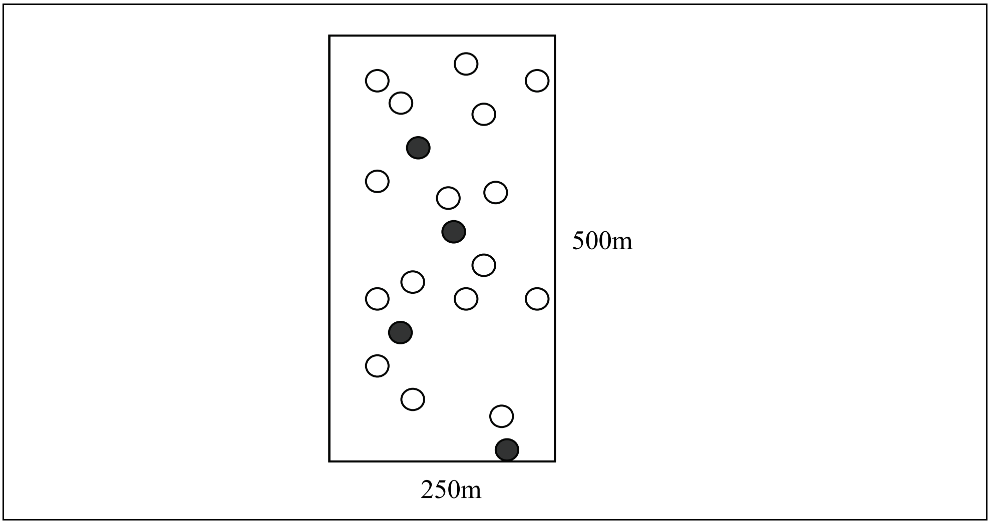


Fig. 9. Base Topology

differentiation. Simulation results are not affected, as the traffic generated is not unicast or multicast but broadcast. Our main goal is for all the mobile nodes to receive the broadcast packet.

In the above topology, 20% of the mobile nodes (4/20) are placed inside the right part of the network and 80% of them (16/20) are placed inside the left part. In order to create this topology, we need to start from a base topology as presented in Fig. 9.

The volume of diversity is then specified by selecting an appropriate percentage of the mobile nodes, which in our example is 20%. These are the black nodes of Fig. 9. We developed a simple software tool that scans the mobility file for all the command lines that either initialize or change the movement of all 4 black nodes. The value of X in these command lines is then increased by 250m, in order to migrate the black nodes over to a topology of identical length and width that touches the base topology vertically. Fig. 10 shows the migration process.

The movement of the black nodes initially is limited with regards to the X coordinate between 0m and 250m. Thus, after the modification of the mobility file, these nodes are restrained to move inside the right half of the topology, with the X value varying between 250m and 500m.

As a result of the process described above we get the end result of Fig. 8.

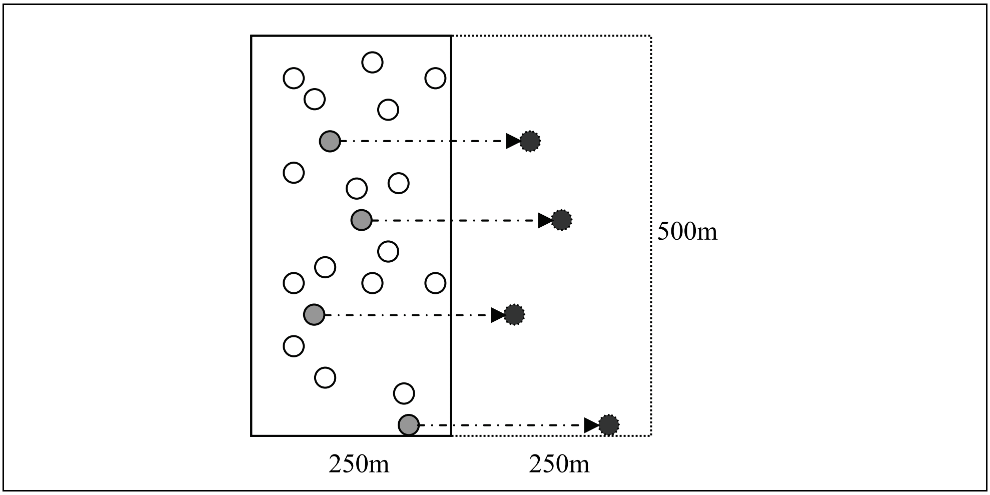


Fig. 10. Migration Process

5. Performance analysis

We implemented Distance-Based Adaptive scheme (DibA) and Distance-Based scheme (DB) using the network simulator NS2.30. We have used the NS2 code for DB provided by (Barrit et al., 2006), (Williams et al., 2002).

5.1 Simulation set-up and parameters

Node mobility is simulated using mobility files that are generated by the NS2 mobility generation feature setdest. Our experiments make use of both normal and diverse topologies,

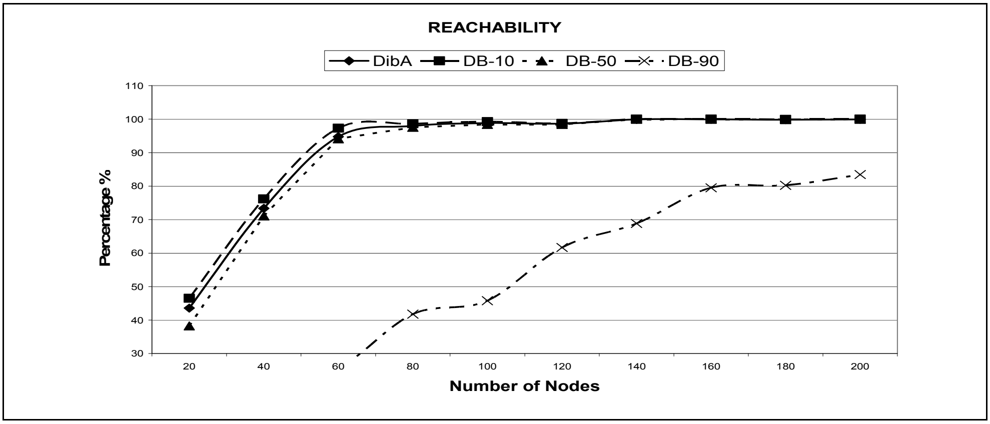


Fig. 11. Reachability – Normal – BGR 5p/s

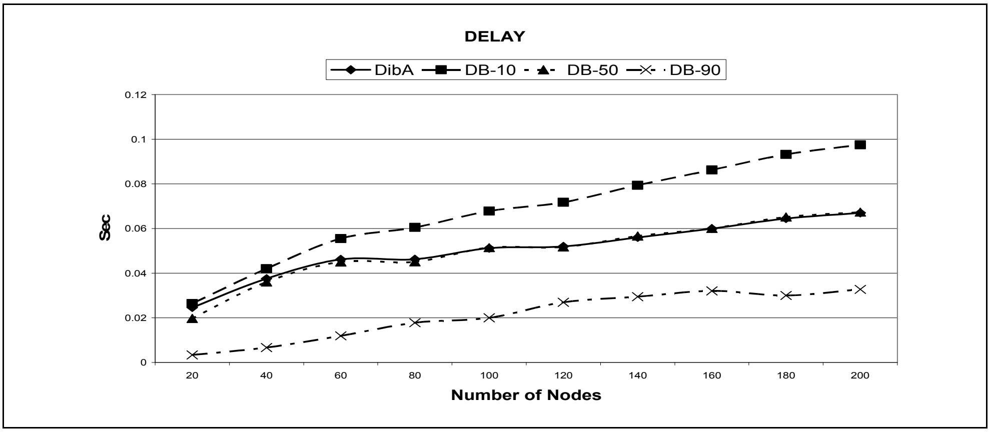


Fig. 12. Delay – Normal – BGR 5p/s

in order to cover the majority of possible scenarios. The network area is of fixed size 500x500m2. The mobility files are created with zero pause time. Mobile nodes move with maximum speed of 5m/sec. Each simulation has duration of 100secs and all mobile nodes use a transmission range of 100m.

Each scenario is restricted to the transmission of broadcast traffic only. This is a common strategy, especially when using very high broadcast generation rates (BGR). Combining normal traffic with broadcast traffic is a step further for our work with the implementation currently taking place. In order to avoid anomalies, we run three simulations for every scenario using three different mobility files. Our research has found no work until this point, where more than 3 or 4 repetitions are used. The final results are created as an average of the three simulations.

Experiments where performed using 3 different distance thresholds for DB of 10m, 50m and 90m, in order to cover the two extremes and an intermediate value. DibA tunes the distance threshold to one of the 3 thresholds mentioned above, depending on the local level of density. The number of nodes has a starting value of 20 and reaches a maximum of 200 nodes with a step of 20 (20, 40, 60, …, 200).

We first divide our simulations into two groups according to the broadcast generation rate. BGR is set to 5packets/sec and 60packets/sec. Furthermore, we also divide the simulations depending on whether a normal or a diverse topology is used.

The following performance metrics are considered:

- Reachability – The percentage of nodes that successfully receive the broadcast message.
- Delay – The time elapsed from the initiation of the broadcast process until no more rebroadcasts take place.
- Average number of Packets transmitted per node (APT) – This is a self explained performance metric which is closely related to energy efficiency.

5.2 Simulation results

Fig. 11, 12 and 13 present the performance of the 4 schemes, when normal scenarios are used and BGR is set to 5packets/sec.

Fig. 11 shows that DB-90 performs very poorly due to the high threshold value, whereas all the other schemes perform almost identical. Although DB-90 appears to be very fast in Fig. 12, that is because of the very low level of reachability. DB-10 is the slowest, despite the fact that has similar reachability with DB-50 and DibA. The latter two again perform in a similar way. Fig. 13 shows that DB-10 uses a significantly higher number of transmissions in order to achieve the same level of reachability with DB-50 and DibA. Thus, it is the least energy efficient.

Fig. 14, 15 and 16 show how the 4 schemes perform when the topology is diverse and the broadcast generation rate is low.

Fig. 14 reflects the performance of all schemes in terms of reachability. Although DibA, DB-10 and DB-50 perform almost identical when the network is dense (120 nodes or more), for sparse topologies DB-10 is slightly better than DibA and in turn that is better than DB-50. DB-90 again performs poorly. DB-10's slightly better performance for reachability, proves to be extremely costly, as it is much slower than the rest and APT is almost double than the

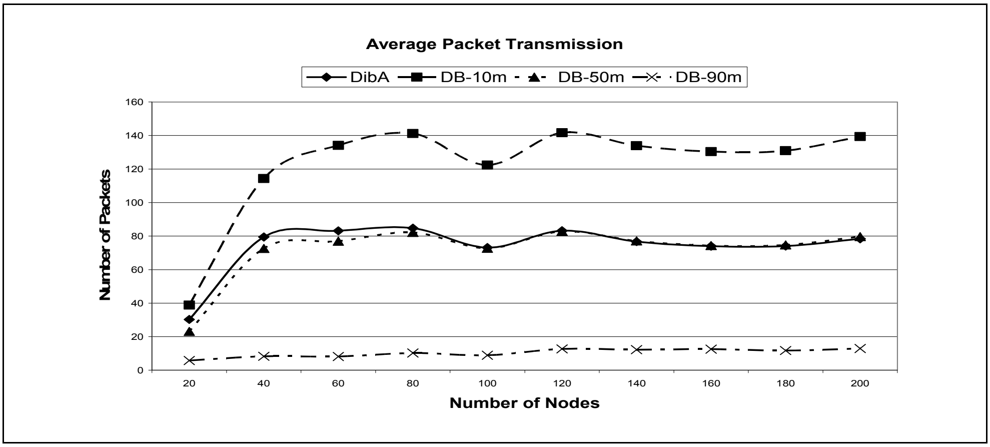


Fig. 13. APT – Normal – BGR 5p/s

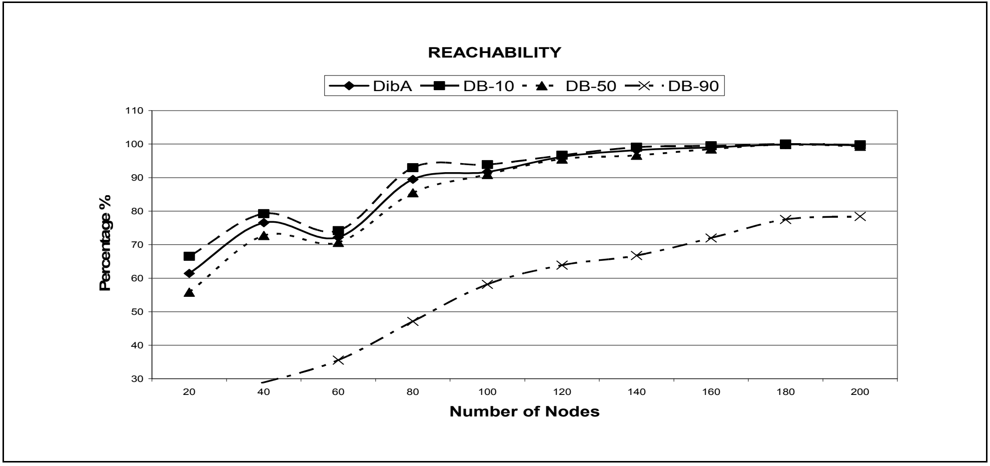


Fig. 14. Reachability – Diverse – BGR 5p/s

following scheme. Energy efficiency is very poor in these conditions. DibA appears to be better than DB-50 for sparse topologies and similar when density increases. Better reachability usually comes with more latency and more APT. For DibA and DB-50 this is reflected in Fig. 15 and 16.

Fig. 17, 18 and 19 present the performance of the 4 schemes when normal scenarios are used and BGR is set to 60packets/sec.

Fig. 17 shows that for sparse networks (up to 60 nodes) DibA and DB-10 have the same performance with DB-50 being slightly worse. For very dense networks, DibA is now performing better than the rest. DB-90 is completely outperformed. Despite the fact that DB-10 has lower reachability when compared to DibA, Fig. 18 and 19 show that it is disproportionally slower and energy inefficient. DB-50 shows slightly better performance for delay and APT, but that is due to its lower reachability.

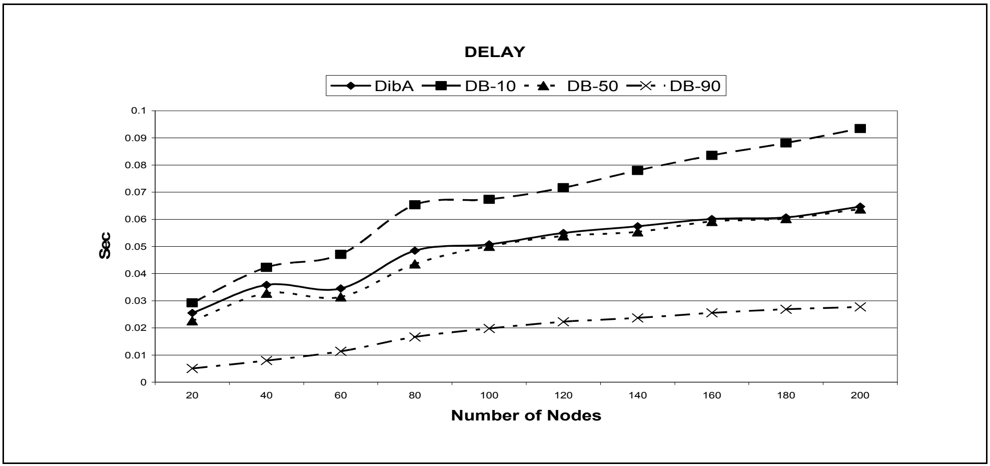


Fig. 15. Delay – Diverse – BGR 5p/s

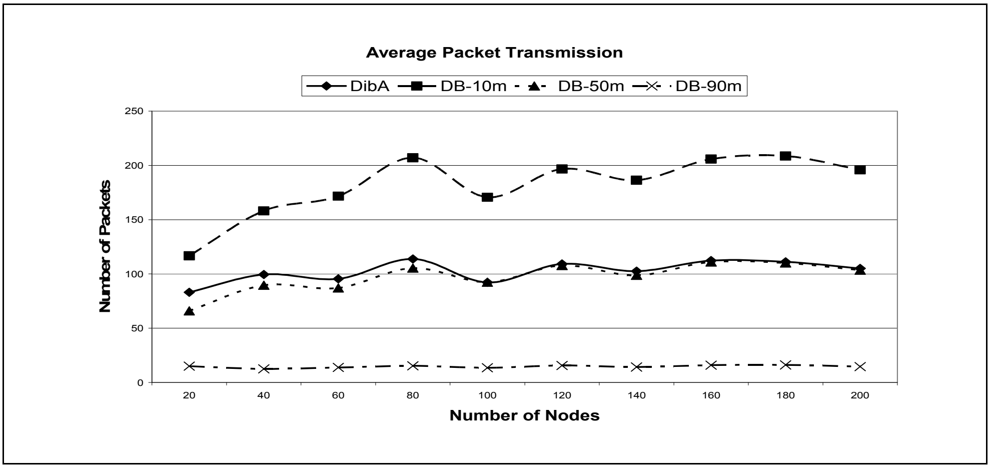


Fig. 16. APT – Diverse – BGR 5p/s

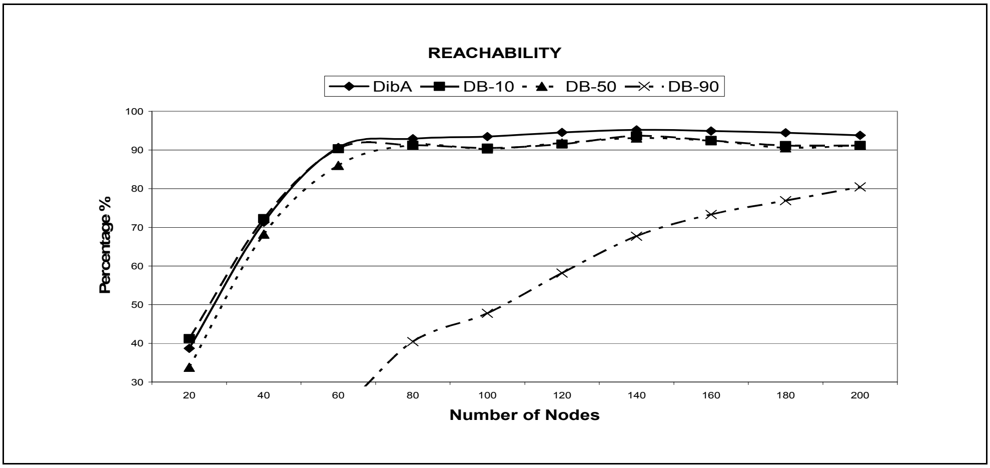


Fig. 17. Reachability – Normal – BGR 60p/s

Fig. 20, 21 and 22 show how the 4 schemes perform when the topology is diverse and the broadcast generation rate is extremely high.

In this group of experiments we have used a very high broadcast generation rate and extremely diverse network topologies. The results, in terms of reachability, are reflected in Fig. 20. DB-10 is better for sparse networks, but as density increases, it is found to finish last for dense networks of 200 nodes. DB-50 proves to be more stable, but at no point does it perform better than all the rest. The results for DB-90, prove that even the use of a very low distance threshold is the appropriate selection when both density and traffic rate are set to high values. DibA appears to be the most reliable option. Fig. 21 and 22 show that DB-10 is neither fast nor energy efficient. DB-50 performs well but being faster and more energy efficient is the result of its low reachability levels.

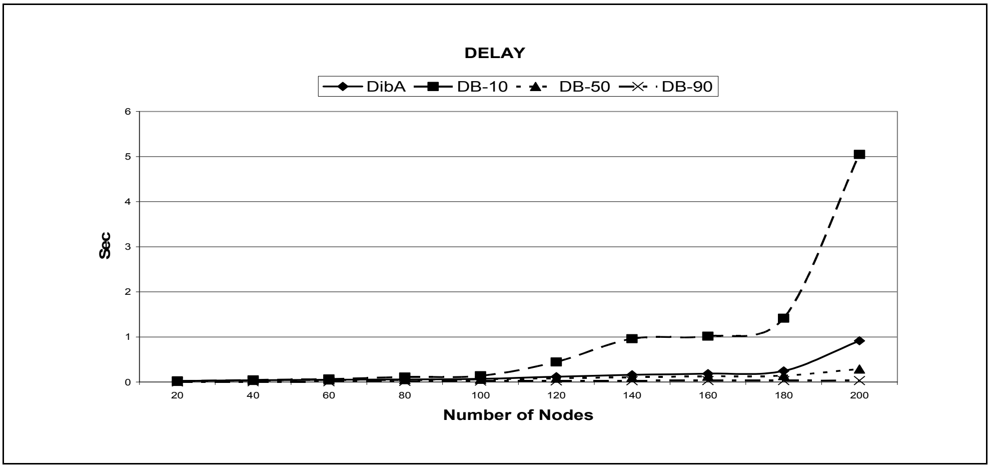


Fig. 18. Delay – Normal – BGR 60p/s

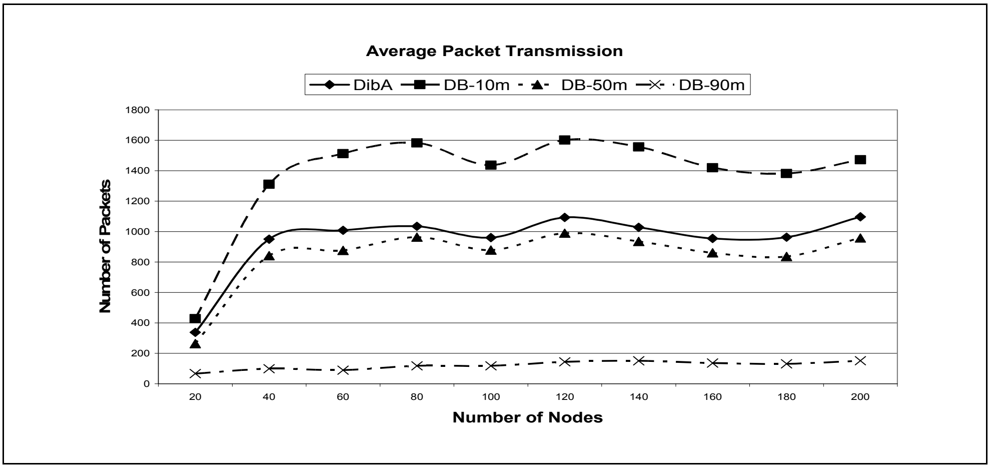


Fig. 19. APT – Normal – BGR 60p/s

6. Conclusions

In this chapter we have shown how the Distance-Based broadcast scheme can potentially alleviate the effects of Simple Flooding by controlling the amount of replicated messages. We have also demonstrated how the selection of a single, fixed and pre-determined distance threshold is not appropriate for all scenarios and does not satisfy the needs of highly dynamic network topologies despite the fact that the Broadcast Storm Problem was overcome when SF was substituted by DB. When the network is sparsely populated and the broadcast generation rate is also low, a small distance threshold needs to be selected in order for the broadcast operation not to "die out" due to lack of pathways to remote or isolated mobile nodes inside the topology. Respectively, a large distance threshold is

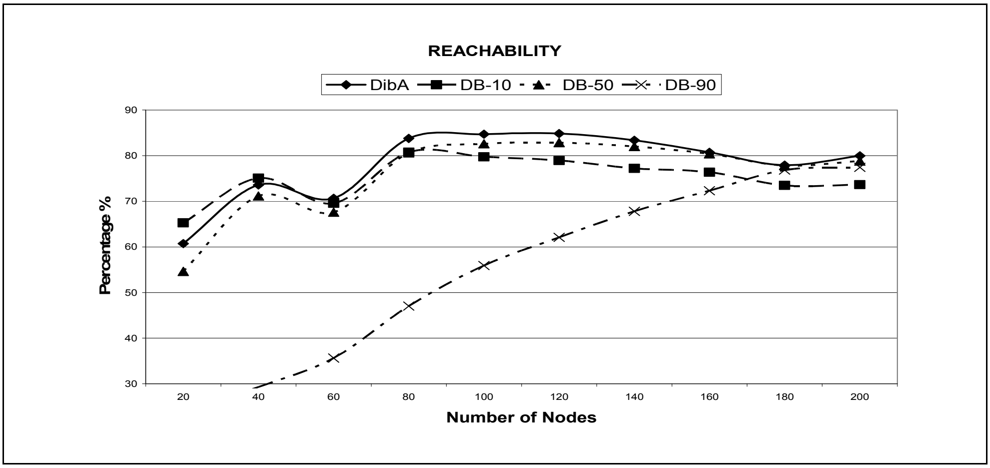


Fig. 20. Reachability – Diverse – BGR 60p/s

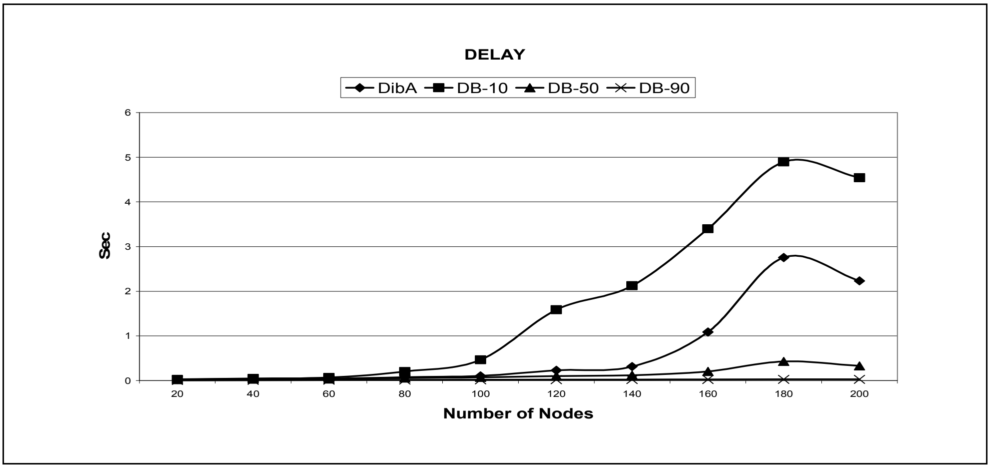


Fig. 21. Delay – Diverse – BGR 60p/s

required for very dense topologies with high broadcast generation rates as it requires fewer broadcast relays to take place decreasing in that way the volumes of contention and collisions. Consequently, adaptive schemes were proposed that adjust the thresholds used depending on the local density of the network.

The adaptive schemes proposed so far introduce further overhead. Alternatively, schemes that make use of positioning systems require the existence of such expensive and unreliable hardware, e.g. GPS. In this chapter, we have presented a new scheme, called DibA, utilising duplicate packets to adjust the distance threshold accordingly. A small threshold is individually set for every node when the duplicate packet counter is small in order to force more nodes to rebroadcast. Respectively, a large distance threshold is set when the counter is increased to a high value aiming in fewer broadcast relays.

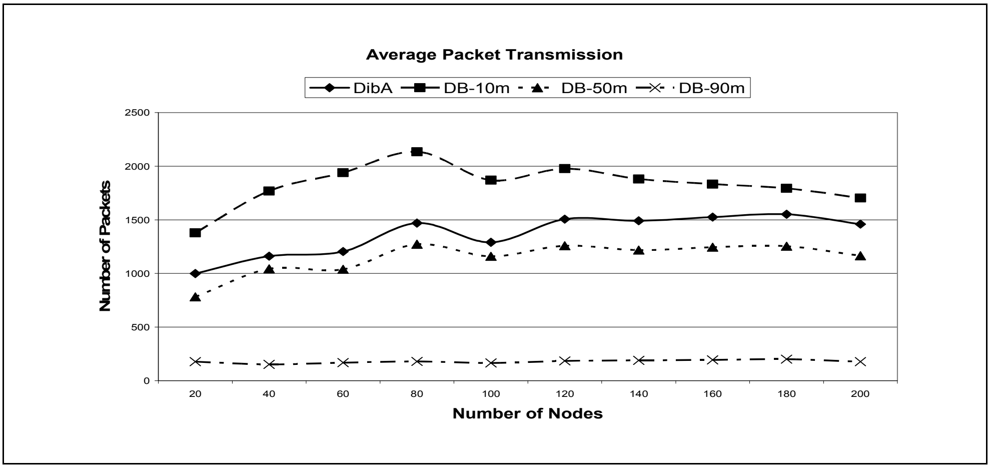


Fig. 22. APT – Diverse – BGR 60p/s

Our performance study compared DibA against DB with three different distance thresholds of 10m, 50m and 90m under common and diverse operational conditions throughout simulation. In order for our study to cover all extremes, we have also made use of highly diverse network topologies that include both sparse and dense areas. These scenarios have aided in demonstrating the superiority of DibA over DB more appropriately. In all scenarios, the simulation results clearly showed that DibA outperforms DB for various topologies and broadcast generation rates. Furthermore, DibA is also more reliable and power efficient than DB as the number of broadcast relays does not linearly grow up with the network density.

7. References

D. Cavin, Y. Sasson & A. Schiper. (2002). On the Accuracy of MANET Simulators, *Proceedings of POMC*, pp. 38-43, Toulouse France, 2002, ACM, New York, USA.

K. Viswanath & K Obraczka. (2006). Modeling the performance of flooding in wireless multi-hop ad hoc networks, *Computer Communications*, Vol. 29, No. 8, pp. 949-956.

S. Ni, Y. Tseng, Y. Chen & J. Sheu. (1999). The broadcast storm problem in a mobile ad hoc network, *Proceedings of MOBICOM*, pp. 151-162, Seatttle, Washington, USA, 1999, ACM, New York, USA.

S. Leng, L. Zhang, L. Wu Yu & C. Heng Tan. (2004). An efficient broadcast relay scheme for MANETs, *Computer Communications*, Vol. 28, No. 5, pp. 467-476, Butterworth-Heinemann, Newton, MA, USA.

C. Zhu, M. J. Lee & T. Saadawi. (2004). A Border-aware broadcast scheme for Wireless Ad Hoc Networks, *Proceedings of CCNC*, pp. 134-139, 2004, IEEE.

A. Qayyum, L. Viennot & A. Laouiti. (2002). Multipoint relaying for flooding broadcast messages in mobile wireless networks, *Proceedings of HICSS*, pp. 298, 2002, IEEE Computer Society, Washington, DC, USA.

C. Hsu, C. Chen & H. Wang. (2005). DISCOUNT: A Hybrid Probability-Based Broadcast Scheme for Wireless Ad Hoc Networks, *Proceedings of VTC-FALL*, pp. 2706-2710, Dallas, USA.

R. Purtoosi, H. Taheri, A. Mohammadi & F. Foroozan. (2006). Improving broadcast performance by traffic isolation in wireless ad hoc networks. *International Journal of Communication Systems*, Vol. 19, No. 9, pp. 1029–1043.

B. J. Barritt, B. Malakooti & Z. Guo. (2006). Intelligent Multiple-Criteria Broadcasting in Mobile Ad-hoc Networks, *Proceedings of IEEE LCN / P2MNet*, pp. 761-768, Tampa, Florida, USA.

N. Bauer, M. Colagrosso & T. Camp. (2005). Efficient implementations of all-to-all broadcasting in mobile ad hoc networks, *Pervasive and Mobile Computing*, Vol. 1, No. 3, pp. 311–342.

B. Williams & T. Camp. (2002). Comparison of broadcast techniques for mobile ad hoc networks, *Proceedings of MOBIHOC*, pp. 194-205, Lausanne, Switzerland.

Young-Ching Deng, Ching-Chi Shue & Ferng-Ching lin. (2006). An Adaptive Medium Access Control Protocol for Reliable Broadcast and Unicast in Ad Hoc Networks, *IEICE – Transactions on Information & Systems*, pp. 527-535.

Jung-Pil Ryu, Min-Su Kim, Sung-Ho Hwang & Ki-Jun Han. (2004). An Adaptive Probabilistic Broadcast Scheme for Ad-Hoc Networks, In: *High Speed Networks and Multimedia Communications*, pp. 646-654, Springer Berlin / Heidelberg, Toulouse, France, 2004.

Sung-Hee Lee & Young-Bae Ko. (2006). An Efficient Neighbor Knowledge Based Broadcasting for Mobile Ad Hoc Networks, In: *Computational Science - ICCS*, pp. 1097-1100, Reading, UK.

Xiaohu Chen, Michalis Faloutsos & Srikanth Krishnamurthy. (2002). Distance ADaptive (DAD) Broadcasting for Ad Hoc Networks, *Proceedings of MILCOM*, pp. 878-882.

Michael D. Colagrosso. (2007). Intelligent Broadcasting in Mobile Ad Hoc Networks: Three Classes of Adaptive Protocols, *Proceedings of EURASIP*, pp. 25-25.

Xiaohu Chen, Michalis Faloutsos & Srikanth V. Krishnamurthy. (2003). Power Adaptive Broadcasting with Local Information in Ad hoc networks, *Proceedings of ICNP*, pp. 168-178.

Pradeep Kyasanur, Romit Roy Choudhury & Indranil Gupta. (2006). Smart Gossip: An Adaptive Gossip-based Broadcasting Service for Sensor Networks, *Proceedings of MAAS*, pp. 91-100, Vancouver, BC, Canada.

Yu-Chee Tseng, Sze-Yao Ni & En-Yu Shih. (2001). Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Network, *Proceedings of Distributed Computing Systems*, pp. 481-488.

14

Predictive RSS with Fuzzy Logic based Vertical Handoff Decision Scheme for Seamless Ubiquitous Access

Sunisa Kunarak and Raungrong Suleesathira
King Mongkut's University of Technology Thonburi
Thailand

1. Introduction

Currently, adverse wireless and mobile networks including Worldwide Interoperability for Microwave Access (WiMAX), Wireless Local Area Network (WLAN), Third Generation (3G) mobile communications such as Universal Mobile Telecommunications Systems (UMTS), Wideband Code Division Multiple Access (WCDMA) and Bluetooth as shown in Fig. 1, have emerged and continuously developed to achieve high-speed transmission. The network characteristics are summarized in Table 1. However, no one network can provide all types of desired services, e.g. wide coverage, high bandwidth and low access costs. For example, WLAN provides high data rates within limited coverage areas, e.g. hotel, airport, campus and other hotspots whereas UMTS provides lower data rates over a larger coverage area.

Therefore, one of the challenges in the next generation of wireless communications (McNair & Fang, 2004) ; (Frattasi et al., 2006) ; (Boudriga et al., 2008) is the integration of existing and future wireless technologies and supporting transparent and seamless vertical handoffs without degrading quality of services (QoS) between these heterogeneous networks (Kassar et al., 2008) ; (Haibo et al., 2009). This will need a multi-interfaced terminal which can change connections during inter-network movement. Received Signal Strength (RSS) based handoff scheme is commonly used to initiate a handover (Pollini, 1996) ; (Pahlavan et al., 2000) ; (Majlesi & Khalaj, 2002). In heterogeneous wireless networks, RSS is not however sufficient for a vertical handoff decision because the RSS of different networks cannot be compared directly, and moreover, RSS cannot reflect network conditions adequately. In order to develop vertical handoff decisions, new metrics such as service types, monetary cost, network conditions, mobile terminal conditions and user preference should be used in conjunction with RSS measurement. In policy-based approaches, multi-criteria are needed not only for decision when the handover occurs but also determine which network should be chosen for user choice and intervention (Nkansah-Gyekye & Agbinya, 2008) ; (Stevens-Navarro et al., 2008) ; (Sun et al., 2008) ; (Nay & Zhou, 2009) ; (Haibo et al., 2009).

In (Song & Jamalipour, 2008), a merit function is proposed to evaluate network performance based on user preferences and adopted to find the best possible network for users. However, the counter to ensure the conditions in handoff policy consistently true is fixed which is not adjusted to the mobile computing and network environment. The approach proposed in (Chang & Chen, 2008) determines the optimal target network in two phases, i.e., RSS prediction and Markov decision process (MDP). Predicting RSS can minimize the dropping probability but time complexity of this MDP-based predictive RSS approach depends on the number of WLANs and WMANs (Wireless Metropolitan Area Networks). Moreover, there is no dwell timer to check the condition of RSS comparison in order to avoid ping-pong effect. Besides RSS, mobile station velocity and movement pattern are important factors for handoff decision procedure (Lee et al., 2009). The movement-aware vertical handoff algorithm avoids unnecessary handovers by adjusting the dwell time adaptively and predicting the residual time in the target cell. Its algorithm has to estimate velocity and direction of the mobile in the first step which is the location update procedure. If the mobile movement is irregular suddenly, the estimation would not be precise and result in an error decision. Due to a robust mathematical framework for dealing with impression and uncertainty problem, fuzzy logic based network selection algorithms have been proposed (Majlesi & Khalaj, 2002) ; (Tansu & Salamah, 2006) ; (Stoyanova & Mähönen, 2007) ; (Xia et al., 2008) ; (Haibo et al., 2009). Fuzzy logic theory based quantitative decision algorithm (FQDA) (Sivanandam et al., 2007) has an advantage over traditional fuzzy logic algorithm which there is no need to establish a database to store rule bases. The vertical handoff algorithm presented in (Xia et al., 2008) also used FQDA for the optimized network selection but it considers only network conditions. Vertical handoff scheme should balances against user satisfaction and network efficiency for different types of service applications.

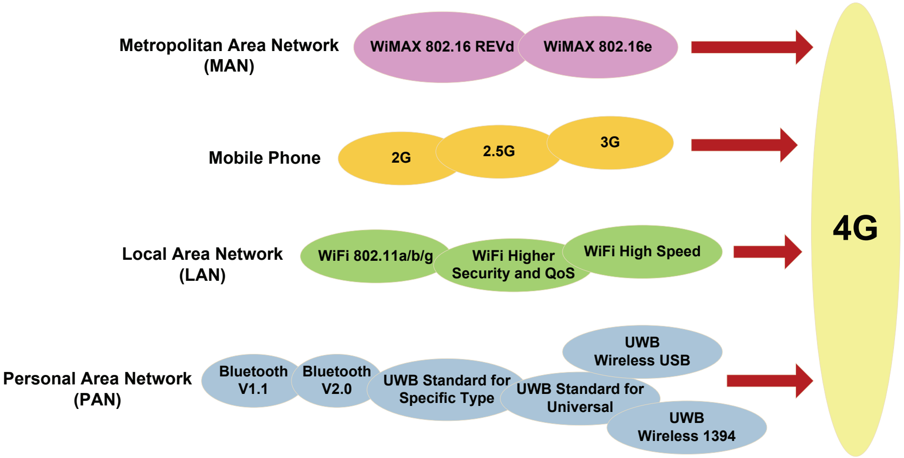


Fig. 1. Evolution of wireless and mobile networks toward 4G ubiquitous access

In this paper, we presented a vertical handoff scheme satisfying between user requirement and network conditions and avoiding unnecessary handoffs as well. The upper and lower bounds of dwell time depend on the service types i.e. real time and nonreal time services. The policy is to minimize handoff delay for real time service and prolong staying time for nonreal time services if the mobile node stays in WLAN/WiMAX while handoff from UMTS to the WLAN/WiMAX is the last time the signal strength reaches the acceptable level. Back propagation neural network is used to predict RSS. RSS of current serving network and predictive RSS of target networks are used to consider whether the handoff should be triggered. In the network selection procedure, the merit function is adopted to find candidate networks satisfying preference of a user. FQDA using five handoff metrics, RSS, bandwidth, number of users, power consumption and monetary cost as the input can determine the optimal target network.

The remainder of paper is organized as follows. Section 2 explains the RSS prediction using back propagation neural network. Merit function and dwell time are described in section 3 and 4, respectively. In section 5, network selection using FQDA is presented. Vertical handoff scheme is proposed in section 6. Section 7 presents and discusses the simulation results. The conclusion is finally given in section 8.

Network Characteristic	**IEEE 802.11g WiFi**	**IEEE 802.16e Mobile WiMAX**	**3G UMTS/WCDMA**
Coverage	100-300 (m)	1.6-5 (km)	3-10 (km)
Bandwidth	54 Mbps	30 Mbps (10 MHz BW)	1.8-14.4 Mbps (HSDPA+HSUPA)
Frequency	2.4 GHz	2-6 GHz (licence)	1920-1980 MHz (uplink) 2110-2170 MHz (downlink)
Channel Bandwidth	5 MHz	5, 7, 10 MHz	5 MHz
Number of Channels	13	Depending on Country	12
Number of user/channel	1	Many (100, ...)	Many (order of magnitude: 25; data rate decreases)

– *HSDPA:* High Speed Downlink Packet Access

– *HSUPA:* High Speed Uplink Packet Access

Table 1. Network Characteristics

2. Received signal strength prediction using back-propagation neural network

Although the RSS with hysteresis and threshold approach can reduce the number of unnecessary handoffs, this approach results in a low data rate and high dropping probability since the mobile node receives too weak RSS from the serving network at the handoff point. Given the future values of the RSS of each neighbor base stations, the handoff process would perform before the RSS becomes weak. Consequently, prediction technique based scheme with hysteresis is beneficial in avoiding unnecessary handoff, minimizing the handoff dropping probability as well as obtaining higher data rate. We use the back-propagation training algorithm for a two-layer network as in Fig. 2 to predict the future RSS. The input and output of the hidden layer are denoted as z_i and y_j, respectively while the output of the network is denoted as o_k for $i = 1,2,...,I, j = 1,2,...,J$ and $k = 1,2,...,K$. These input and output values can be arranged in a vector notation as $\mathbf{z} = [\mathrm{z}_1,\mathrm{z}_2,...,\mathrm{z}_\mathrm{I}]^\mathrm{t}$, $\mathbf{y} = [\mathrm{y}_1,\mathrm{y}_2,...,\mathrm{y}_\mathrm{J}]^\mathrm{t}$ and $\mathbf{o} = [\mathrm{o}_1,\mathrm{o}_2,...,\mathrm{o}_\mathrm{K}]^\mathrm{t}$. The weight v_{ji} connects the i^{th} input with the input to the j^{th} hidden node and the weight w_{kj} connects the output of the j^{th} neuron with the input to the k^{th} neuron. Given P training pairs of inputs and outputs $\{(\mathbf{z}_1,\mathbf{d}_1),(\mathbf{z}_2,\mathbf{d}_2),...,(\mathbf{z}_\mathrm{P},\mathbf{d}_\mathrm{P})\}$ the weights are updated after each sample pair as follow (Zurada, 1992) ; (Haykin, 2009):

1. For $p = 1$, submit training pattern $\mathbf{z}_\mathrm{p}$ and compute layer responses

$$y_j = f\left(\sum_{i=1}^{I} v_{ji} z_i\right) \tag{1}$$

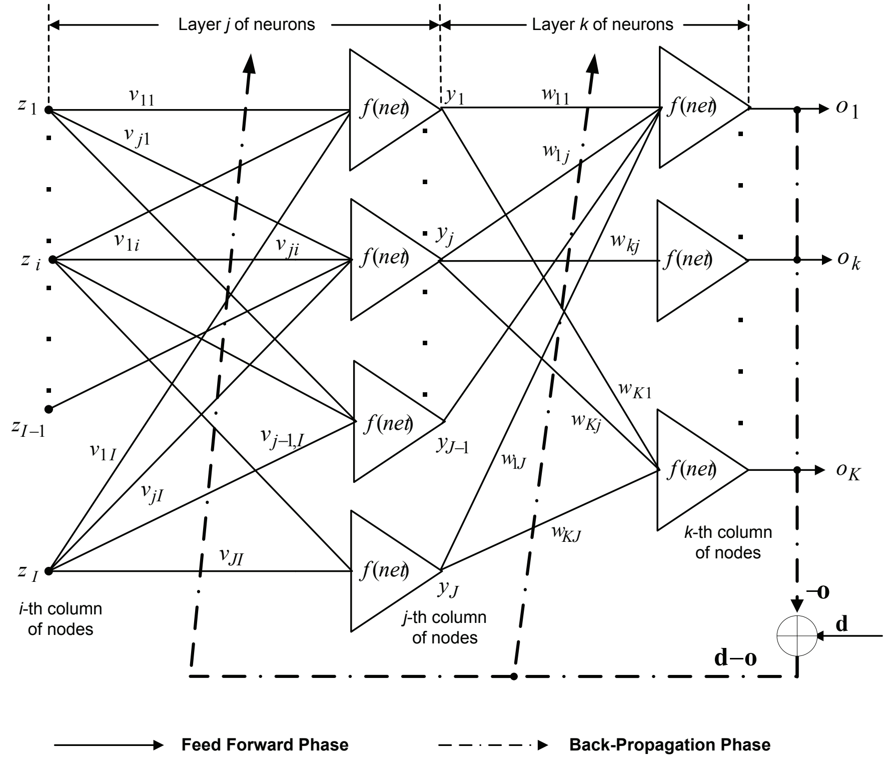


Fig. 2. Two-neuron layer network

$$o_k = f\left(\sum_{j=1}^{J} w_{kj} y_j\right) \tag{2}$$

when $f(net) \underline{\underline{\Delta}} \frac{2}{1+\exp(-\lambda net)} - 1$ and $\lambda > 0$.

2. Calculate errors

$$\delta_{ok} = \frac{1}{2}\left(d_k - o_k\right)\left(1 - o_k^2\right) \tag{3}$$

$$\delta_{yj} = \frac{1}{2}\left(1 - y_j^2\right) \sum_{k=1}^{K} \delta_{ok} w_{kj}. \tag{4}$$

3. Adjust the output layer weights and hidden layer weights using the delta learning rule

$$w_{kj} \leftarrow w_{kj} + \eta \delta_{ok} y_j$$
$$v_{ji} \leftarrow v_{ji} + \eta \delta_{yj} z_i \quad ; \qquad \eta > 0.$$

4. Increase $p = p + 1$ and if $p < P$ then perform step 1 until $p = P$.

The learning procedure stops when the cumulative final error in the entire training set,

$$E = \sum_{p=1}^{P} \frac{1}{2} \left\| \mathbf{d}_p - \mathbf{o}_p \right\|^2,$$

below the upper bound E_{max} is obtained otherwise initiate the new training cycle.

We implemented the back-propagation algorithm to predict the RSS in UMTS by using four input nodes, four hidden nodes and one output node. As shown in Fig. 3, the predictive RSS matches the actual RSS values very well.

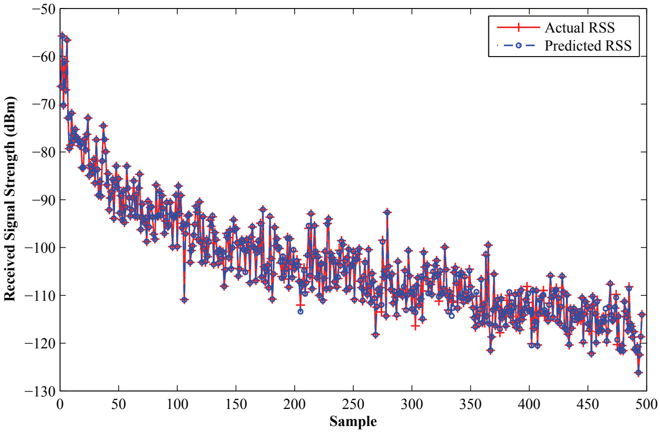


Fig. 3. Prediction results by using back-propagation neural network

3. Merit function

Merit function is a measurement of the benefit obtained by handing over to a particular network. It is calculated for each network available in the vicinity of the user. The neighbor networks with higher value than the serving network become candidate networks. The merit function for wireless network n is calculated as (Song & Jamalipour, 2008)

$$F_n = E_n \sum_i w_i \ln(p'_{n,i}) \tag{5}$$

where $p_{n,i}$ is the i^{th} QoS factor in network $n, p'_{n,i} = p_{n,i}$ if the increase of $p_{n,i}$ contributes the merit value to network n, while $p'_{n,i} = \frac{1}{p_{n,i}}$ if the decrease of $p_{n,i}$ contributes the merit value, w_i is the weight assigned to the i^{th} QoS factor with $\sum_i w_i = 1$, E_n is the elimination factor of network n. The value of E_n is either 0 or 1 decided by QoS requirements based on user preference and service applications. For example, $E_n = 0$ if the data rate supported by a network is lower than that required by the current service, otherwise $E_n = 1$. Suppose that the current service is real time video, the UMTS should be deleted from the candidates by the eliminate factor i.e. $E_n = 0$ due to very high bandwidth unprovided. The considered QoS parameters consist of bandwidth (BW), delay (D) and monetary cost (C) given in the merit function as

$$F_n = E_n \left(w_{BW} \cdot \ln p_{n,BW} + w_D \cdot \ln \frac{1}{p_{n,D}} + w_C \cdot \ln \frac{1}{p_{n,C}} \right). \tag{6}$$

4. Dwell time

The traditional handover decision policy based on RSS, hysteresis and threshold can cause a serious ping-pong effect if the mobile node moves around the overlap area. To alleviate sequential handovers evoked too frequently, the conditions for handoff decision must continue to be true until the timer expires in order to determine when the handover occurs (Pahlavan et al., 2000) ; (Kassar et al., 2008). The duration of dwell timer can be adjusted according to the movement of mobile node and perceived QoS from each neighbor network. If the merit of target network is much better than the current serving network, the dwell timer is shortened, and if movement direction is irregular, i.e. ping-pong effect, the dwell timer is extended. This leads the dwell timer defined as

$$t_d = \min[ubound(t_d), \delta] \qquad \text{s.t.} \qquad \delta = \max\left(lbound(t_d), (1 + \tilde{p}\tilde{p}_t) \cdot \frac{F_{\text{cur}}}{F_{\text{targ}}} \cdot \hat{t}_d \right) \tag{7}$$

where $ubound(t_d)$ and $lbound(t_d)$ denotes the upper and lower bounds of dwell timer (t_d), $\hat{t}_d$ is the default value of the dwell time, pp_t is the ping-pong flag at time t which is set to 1 if direction change between time t and $t-1$ more than 90 degree, otherwise $pp_t = 0$. $\tilde{p}\tilde{p}_t$ is an average ping-pong flag until time t given by (Lee et al., 2009)

$$\tilde{p}\tilde{p}_t = \begin{cases} 1 & avg(pp_t) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

$$avg(pp_t) = \sum_{i=1}^{t} \alpha(1-\alpha) pp_{t-i+1} \tag{9}$$

where $0 < \alpha \leq 1$ is an exponential smoothing factor. Note that we use the random waypoint mobility model (Haykin, 2009) to determine the location and movement of mobile node which enables us to calculate the mobile directions.

5. Fuzzy logic using quantitative decision algorithm based network selection

In this paper, fuzzy logic using quantitative decision algorithm (FQDA) is used as an handoff decision criteria to choose which network to hand over among different available access networks. These criteria can be classified as a multi-criteria strategy regarding to network, terminal, user preference and services. The FQDA has three procedures: fuzzification, quantitative evaluation, and quantitative decision (Sivanandam et al., 2007) ; (Xia et al., 2008).

5.1 Fuzzification

The membership function shown in Fig. 4 has the fuzzy set: very low, low, medium, high and very high. The constants M_{min},M_2,M_3,M_4,M_{max} can be specified with different values according to the specific characteristics of the network being considered. Using five handoff metrics, received signal strength (RSS), bandwidth (BW), number of users (NU), power consumption (P) and monetary cost (C), the presentation of a member function is

$$\mu_{\mathrm{QoS}} = [\mu_{VL}, \mu_{L}, \mu_{M}, \mu_{H}, \mu_{VH}]_{\mathrm{QoS}} \tag{10}$$

where QoS represents the fuzzy variables including RSS, BW, NU, P and C. For example, when the input value in a certain candidate network is $RSS = P$, that network has a membership degree of the fuzzy variable RSS is $[0,0,0,0.18,0.62]_{RSS}$.

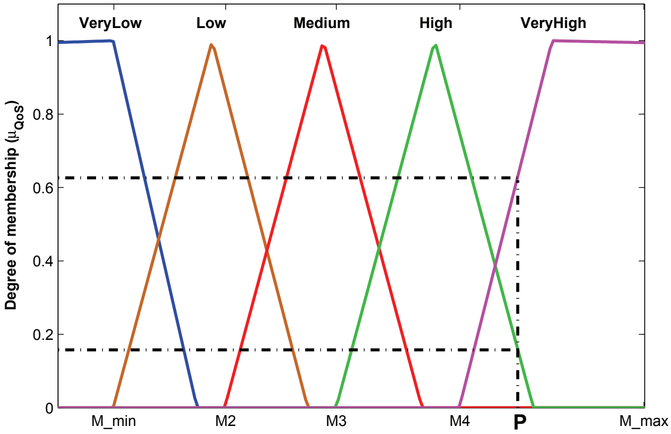


Fig. 4. Membership function

5.2 Quantitative evaluation

Quantitative evaluation is denoted as $Q_{\mathrm{QoS}} = [Q_{VL}, Q_{L}, Q_{M}, Q_{H}, Q_{VH}]_{\mathrm{QoS}}$ which can be specified with different values according to the specific characteristics of the network. We assign $Q_{\mathrm{QoS}} = [0, 0.25, 0.5, 0.75, 1]_{\mathrm{QoS}}$ when QoS is RSS and BW, and $Q_{\mathrm{QoS}} = [1, 0.75, 0.5, 0.25, 0]_{\mathrm{QoS}}$ when QoS is NU, P and C. The quantitative evaluation value (QEV) of each QoS metric for a candidate network n is a sum of evaluated membership degree calculated as

$$QEV_{n,\mathrm{QoS}} = Q_{\mathrm{QoS}}\mu^{T}_{\mathrm{QoS}}. \tag{11}$$

5.3 Quantitative decision

In order to balance against user satisfaction and network efficiency, each $QEV_{n,\mathrm{QoS}}$ should be weighted to reflect the important of the QoS factor. The quantitative decision value (QDV) of network n is therefore defined as

$$QDV_n = W_{RSS}QEV_{n,RSS} + W_{BW}QEV_{n,BW} + W_{NU}QEV_{n,NU+}W_{P}QEV_{n,P} + W_{C}QEV_{n,C}. \tag{12}$$

For each QoS metric, the weight can be calculated by

$$W_{\mathrm{QoS}} = \frac{\phi_{\mathrm{QoS}}}{\Phi} \tag{13}$$

where ϕ_{QoS} is a function of mean and variance of $QEV_{n,\text{QoS}}$. The mean, m_{QoS}, and variance, σ_{QoS}, of the $QEV_{n,QoS}$ are estimated as follows:

$$m_{\text{QoS}} = \frac{1}{N}\sum_{n=1}^{N} QEV_{n,\text{QoS}} \tag{14}$$

$$\sigma_{\text{QoS}} = \frac{1}{N-1}\sqrt{\sum_{n=1}^{N}\left(QEV_{n,\text{QoS}} - m_{\text{QoS}}\right)} \tag{15}$$

where N is the number of networks. The function ϕ_{QoS} depending on the mean and variance of $QEV_{n,QoS}$ is given as

$$\phi_{\text{QoS}} = \exp(-m_{\text{QoS}} + \sigma_{\text{QoS}}). \tag{16}$$

Once we have ϕ_{QoS} for all QoS merit, we can calculate Φ which is $\Phi = \phi_{RSS} + \phi_{BW} + \phi_{NU} + \phi_P + \phi_C$. To select the most optimal network from those available in the candidate list, the network with largest QDV becomes the handoff target network.

6. Vertical handoff scheme with predictive RSS and fuzzy logic

The proposed vertical handoff algorithm consists of two steps: network QoS monitoring to decide whether handoff procedure is triggered, network selecting to determine which access network should be chosen. Handoff trigger is related to the measured and predicted RSS whereas network selection is related user preference. The service application types, real time and nonreal time, are used in conjunction with duration of signal strength measurements. We proposed two vertical handoff algorithms which one is for the mobile node located in UMTS and another one is for that is located in WLAN/WiMAX as shown in Fig. 5.

In Fig. 5, when a mobile node is in UMTS, the Predictive RSS (PRSS) is first used to help the mobile know whether it is moving toward the target network during dwell time by comparing the PRSS with the maximum threshold $(\text{RSS}_{\text{max\_th\_WL/WM}})$. It is beneficial to handoff if the residence time (t_{res}) in the target network is more than the delay cause by the handoff procedure. Therefore, the condition $t_{res} > (t_{hd} + t_{mu})$ should be also satisfied where t_{hd} and t_{mu} are handoff delay time and make up time, respectively. The residence time in the target network can be predicted by using mobile node velocity and the range to the target network boundary. The lower and upper bounds to calculate the dwell time are chosen based on the handoff policy which is to attempt to prolong the time staying in WLAN/WiMAX for nonreal time services. In addition to take into account both PRSS and residence time, handoff to the target network has to be performed providing the RSS of current serving BS lower than the threshold $(\text{RSS}_{\text{th\_UMTS}})$ in order to prevent the call from being dropped. In network selection procedure, candidate networks are found by comparing merit values of the target networks satisfying the mentioned conditions. If the PRSS is not larger than the high threshold and the RSS of the current serving network is less than the threshold, the available network having $F_{\text{targ}} > 0$ is selected into the list. However, if the PRSS is larger, $F_{\text{targ}} > F_{\text{cur}}$ is the condition to assure that its performance is continuously better than the current one. The network in the list with the largest QDV is the selected networks.

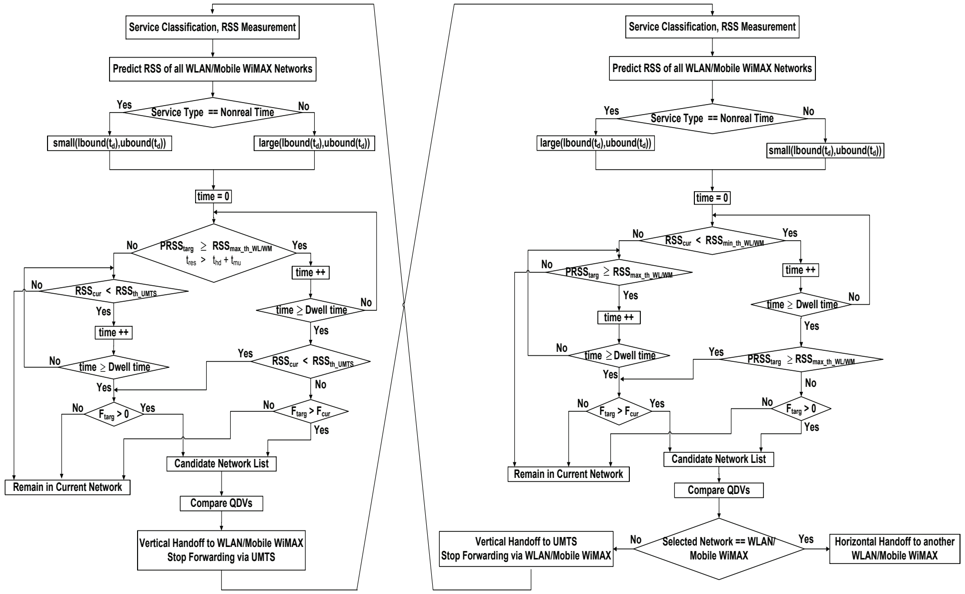

Service Classification, RSS Measurement
Predict RSS of all WLAN/Mobile WiMAX Networks
Yes
Service Type == Nonreal Time
No
small(lbound(t_d),ubound(t_d))
large(lbound(t_d),ubound(t_d))
time = 0
No
PRSS_targ ≥ RSS_max_th_WL/WM
t_res > t_hd + t_mu
Yes
time ++
No
RSS_cur < RSS_th_UMTS
Yes
time ≥ Dwell time
No
time ++
Yes
Yes
RSS_cur < RSS_th_UMTS
No
time ≥ Dwell time
Yes
No
F_targ > 0
Yes
No
F_targ > F_cur
Yes
Candidate Network List
Remain in Current Network
Compare QDVs
Vertical Handoff to WLAN/Mobile WiMAX
Stop Forwarding via UMTS
Service Classification, RSS Measurement
Predict RSS of all WLAN/Mobile WiMAX Networks
Yes
Service Type == Nonreal Time
No
large(lbound(t_d),ubound(t_d))
small(lbound(t_d),ubound(t_d))
time = 0
No
RSS_cur < RSS_min_th_WL/WM
Yes
No
PRSS_targ ≥ RSS_max_th_WL/WM
time ++
Yes
time ≥ Dwell time
No
time ++
Yes
No
time ≥ Dwell time
Yes
Yes
PRSS_targ ≥ RSS_max_th_WL/WM
No
No
F_targ > F_cur
Yes
No
F_targ > 0
Yes
Candidate Network List
Remain in Current Network
Compare QDVs
Vertical Handoff to UMTS
Stop Forwarding via WLAN/Mobile WiMAX
No
Selected Network == WLAN/Mobile WiMAX
Yes
Horizontal Handoff to another WLAN/Mobile WiMAX

Fig. 5. Handoff decision algorithm when a mobile node is in UMTS or in WLAN/WiMAX

When a mobile node stays in WLAN/WiMAX, it starts working if the RSS of the current serving network is less than the minimum threshold ($\mathrm{RSS}_{\mathrm{min\_th\_WL/WM}}$). The lower and upper bounds for the dwell time calculation are short for real time applications to reduce the handoff delay otherwise it is longer for nonreal time applications. Then the mobile node checks whether the PRSS of each target network is stronger than the maximum threshold ($\mathrm{RSS}_{\mathrm{max\_th\_WL/WM}}$). The target networks are candidates if their merit values excel the current one or greater than zero. Finally, fuzzy logic is used to find the largest QDV network as the handoffed network. If there is no selected network, it handoffs to UMTS. The network selection order is $WLAN/WiMAX > 3G$ due to lower cost and better QoS.

7. Simulation results

This section evaluates the performance of the proposed handoff decision mechanism (i.e., denoted by PRSS+FQDA) by simulating heterogeneous wireless networks where UMTS, WLAN and WiMAX overlay as shown in Fig. 6. The channel propagation model used for the RSS received by a mobile node is different in different networks. Given the distance between a mobile node and a base station is $d(meters)$, the $RSS(d)$ in UMTS is

$$RSS(d) = P_t - PL(d) \tag{17}$$

where P_t is the transmit power, and $PL(d)$ is the path loss at distance d which is defined as (Bing et al., 2003)

$$PL(d)_{\mathrm{dB}} = S + 10n\log(d) + \chi_\sigma \tag{18}$$

where S denotes the path loss constant, n denotes the path loss exponent and χ_σ represents the shadow effects which is a zero-mean Gaussian distributed random variable (in dB) with standard deviation σ (also in dB). We use $S = 5$ and $n = 3.5$.
In WiMAX, the path loss at distance d is formulated by (Betancur et al., 2006)

$$PL(d)_{\text{dB}} = 20\log\left(\frac{4\pi d_0}{\lambda}\right) + 10n\log\left(\frac{d}{d_0}\right) + \chi_\sigma \tag{19}$$

where the first term represents the free space path loss at the reference distance d_0, λ is the wavelength. We set n to 4 and a carrier frequency to 3.5 GHz.

In WLAN, the RSS received by the mobile node is computed based on the propagation model as (Chang & Chen, 2008)

$$RSS(d)_{\text{dBm}} = 10\log\left(\frac{100}{(39.37d)^\gamma}\right) \tag{20}$$

where γ denotes the environmental factor of transmissions which is set to 2.8. Several simulation parameters are summarized in Table 2.

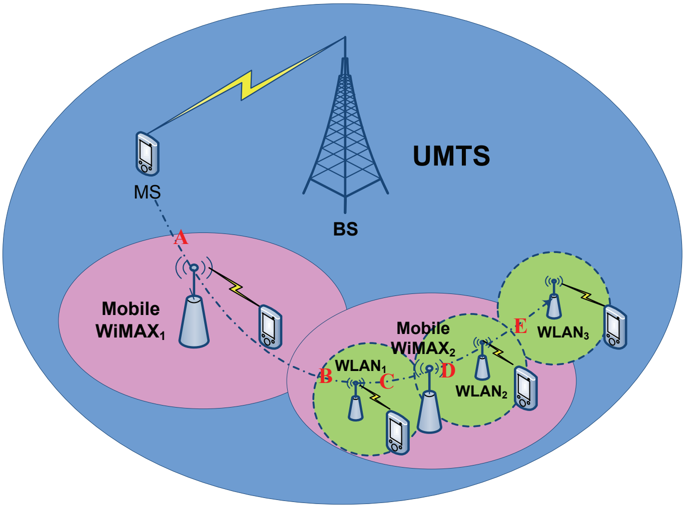


Fig. 6. Mobile model in heterogeneous networks integrating with WLAN, Mobile WiMAX and UMTS

7.1 Network selection performance

In the first simulation, the mobility of a mobile node is fixed according to the path from A to E as seen in Fig. 6. The user speed is 10 m/s and using a 64 kbps service. The calculated FQDA where the handoffs occurred at location A, B, C, D and E are shown in Table 3. The selected network at each handoff location has the largest QDV. The results indicate that the proposed PRSS+FQDA approach can trigger whether handoff is needed. If it is needed, it can choose the optimal network as a target network as well.

Simulation Parameters	**Values**
Cell radius of UMTS/WLAN/Mobile WiMAX	3000/100/1500 (m)
Transmission Power of UMTS/WLAN/Mobile WiMAX	1/0.1/0.5 (w)
$RSS_{min\_th\_WL}/RSS_{max\_th\_WL}$	-85/-72 (dBm)
$RSS_{min\_th\_WM}/RSS_{max\_th\_WM}$	-102/-85 (dBm)
RSS_{th_UMTS}	-119 dBm
small $(lbound(t_d), ubound(t_d))$/large $(lbound(t_d), ubound(t_d))$	(1,5)/(4,20)
Default value of dwell time $(\hat{t}_d)$	2s
Handoff delay time (t_{hd})/ Handoff make up time (t_{mu})	1/1 (s)
Bandwidth of UMTS/WLAN/Mobile WiMAX	7.2/11/15 (Mbps)
$w_{BW}/w_D/w_C$ for 64/96/128 kbps service	(0.3/0.5/0.2)/(0.4/0.3/0.3)/ (0.5/0.3/0.2)
Arrival rate (Poisson distribution)	3 s
Average holding time (Exponential distribution)	1 s
Monetary cost of UMTS/WLAN/Mobile WiMAX	0.8/0.4/0.6
Power consumption of UMTS/WLAN/Mobile WiMAX	0.8/0.2/0.4
Number of UMTS/WLAN/Mobile WiMAX	12/13/15
User capacity/channel of UMTS/WLAN/Mobile WiMAX	50/1/100
Velocity (random waypoint mobility)	1-30 m/s

Table 2. Simulation Parameters

7.2 Handoff decision performance

In this subsection, we present some simulation results to show the performance of the proposed PRSS+FQDA approach by comparing number of handoffs, handoff call dropping probability (P_h) and Grade of Services (GoS). The GoS metric is given by (Tansu & Salamah, 2006);(Chang & Chen, 2008)

$$GoS = P_n + kP_h \tag{21}$$

where P_n is a new call blocking probability and k is the penalty. The impact of the handoff dropping is over the new call blocking since dropping connections results in the revenue loss more than blocking new connections. The recommended range of k is 5 to 20 which $k = 10$ in this simulation.

The proposed PRSS+FQDA approach is compared to 1) the predicted RSS based approach with two thresholds as an interval of hysteresis threshold (HT) (denoted by PRSS+HT) (Pollini, 1996);(McNair & Fang, 2004);(Kassar et al., 2008), 2) the neural network based approach using the Self-Organizing Maps (SOM) (Stoyanova & Mähönen, 2007). The benefit of handoff decision making process using a SOM algorithm is an adaptive inherent organizing technique but it does not guarantee finding the weight vector, corresponding to the network with the best parameter at a time. For training the winner-take-all learning in Fig. 7, the 30-dimentional input vector is generated as

$$\mathbf{x} = \{QEV_1^{RSS} \quad QEV_1^{BW} \quad QEV_1^{NU} \quad QEV_1^{P} \quad QEV_1^{MC}, ..., \quad QEV_6^{RSS}, ..., \quad QEV_6^{MC} \ \} \tag{22}$$

where six networks are in the scenario. The output node satisfying the following condition is the winner

Location	Networks	QDV	Target
A	$Mobile\ WiMAX_1$	0.793	$Mobile\ WiMAX_1$
	$UMTS$	0.219	
B	$MobileWiMAX_2$	0.569	$Mobile\ WiMAX_2$
	$WLAN_1$	0.517	
	$UMTS$	0.339	
C	$Mobile\ WiMAX_2$	0.561	$WLAN_1$
	$WLAN_1$	0.748	
	$UMTS$	0.275	
D	$Mobile\ WiMAX_2$	0.436	$WLAN_2$
	$WLAN_2$	0.459	
	$UMTS$	0.282	
E	$Mobile\ WiMAX_2$	0.391	$WLAN_3$
	$WLAN_3$	0.674	
	$UMTS$	0.317	

Table 3. QDVs of Candidate Networks

$$\|\mathbf{x} - \hat{\mathbf{w}}_n\| = \min_{i=1,\ldots,6} \|\mathbf{x} - \hat{\mathbf{w}}_i\| \tag{23}$$

where the index n denotes the wining neuron number and $\mathbf{w}_n = [w_{n1} \ \ w_{n2} \ \ ,..., \ \ w_{n30}]$ is the weight vector to the n^{th} neuron. Weight adjustment in the k^{th} step of the winner uses the learning rule as (Zurada, 1992)

$$\mathbf{w}_n^{k+1} = \mathbf{w}_n^k + \alpha^k(\mathbf{x} - \mathbf{w}_n^k) \tag{24}$$

$$\mathbf{w}_i^{k+1} = \mathbf{w}_i^k \qquad \text{for} \qquad i \neq n \tag{25}$$

where α^k is a learning constant at the k^{th} step.

In the simulation, an area in which there are three WLANs, two WiMAX and a UMTS is considered as shown in Fig. 6. We first evaluate the performance under number of users ranging from 100-2100, as seen in Figs. 8-10. Figure 8 illustrates that the proposed PRSS+FQDA approach yields the fewest number of vertical handoffs in comparison to the PRSS+HT and SOM approaches. Meanwhile, the numbers of vertical handoffs of all approaches increase when the number of users increases. The number of vertical handoffs using PRSS+FQDA is gently increases as the number of users increases, but that of PRSS+HT and SOM obviously increase. In Fig. 9, the dropping probability of PRSS+FQDA is fewest since it determines the optimal network regarding to the network condition whether it satisfies the preference of users and has a strong RSS as well. Accordingly, this yield the fewest GoS using the PRSS+FQDA approach as shown in Fig. 10.

The performance metrics under different arrival rates ranging from 6 to 16 are demonstrated in Figs. 11-13. The simulation results shown in Figs. 11-13 reveal that the proposed PRSS+FQDA approach outperforms the PRSS+HT and SOM approaches in terms of the number of vertical handoffs, handoff call dropping probability and GoS. In Fig. 11, the number of handoffs increases gradually as the mean arrival rate increases while PRSS+HT and SOM quite increase. Figure 12 shows the dropping probability comparison. The PRSS+FQDA

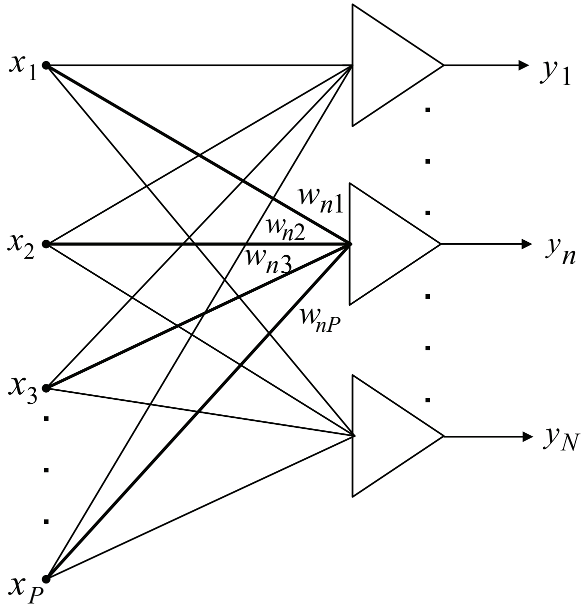


Fig. 7. SOM neural network

scheme yields the lower probability than other schemes which results in a fewest GoS as shown in Fig. 13.

In Figs. 14-16, we presented the results of the proposed PRSS+FQDA approach for the handoff numbers, handoff call dropping probability and GoS under various mobile velocities ranging from 5 to 30 m/s comparing to the other three vertical handoff algorithms, namely the PRSS+HT and SOM approaches. In Fig. 14, PRSS+FQDA yields the fewest vertical handoffs under various velocities but PRSS+HT yields the most vertical handoffs. As the velocity increases, the numbers of vertical handoffs of all approaches also increase. However, the impact of velocity to PRSS+FQDA is less than PRSS+HT and SOM. The handoff call dropping probability of the different approaches are investigated in Fig. 15. PRSS+FQDA has the lowest dropping probabilities and gently increases as the velocity increases while the other three methods obviously increase. Finally, the GoS versus mobile velocity of all approaches are shown in Fig. 16. The proposed PRSS+FQDA approach achieves low GoS although the mobile is moving in high speed. PRSS+HT and SOM generate higher GoS and proportionally vary to the velocity.

8. Conclusions

This paper has proposed a predictive RSS and fuzzy logic based network selection for vertical handoff in heterogeneous wireless networks. The RSS predicted by back propagation neural network is beneficial to avoid dropping calls if it predictes a mobile is moving away from the monitored wireless network. In additional to the RSS metric, the residence time in the target network is predicted which is taken into account for handoff trigger. The prediction period is calculated by the adaptive dwell time. For nonreal time service, the handoff policy is to attempt to use services of WLAN/WiMAX as long as possible. Meanwhile, the handoff

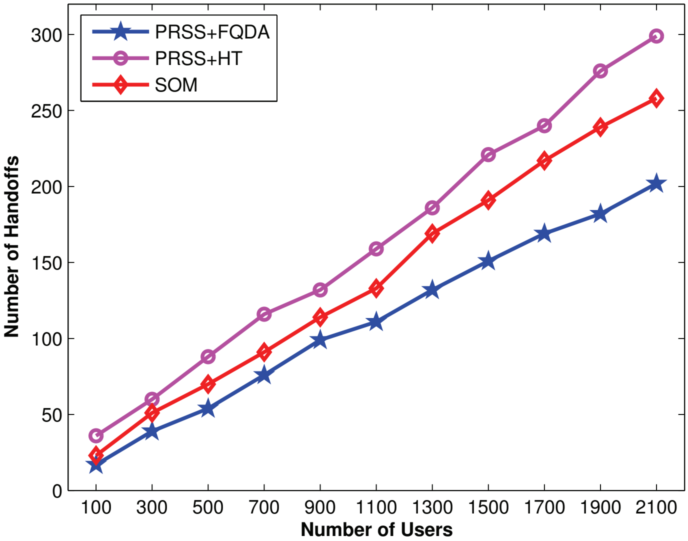


Fig. 8. Number of handoffs versus numbers of users (Arrival rate = 3 sec)

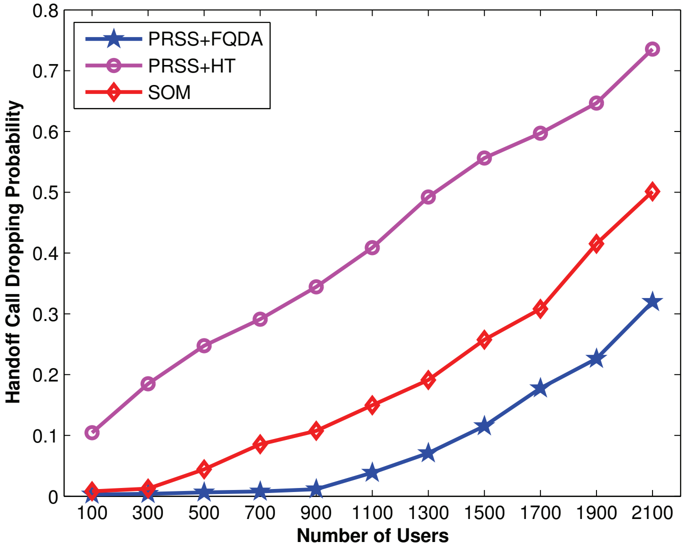


Fig. 9. Handoff call dropping probability versus numbers of users (Arrival rate = 3 sec)

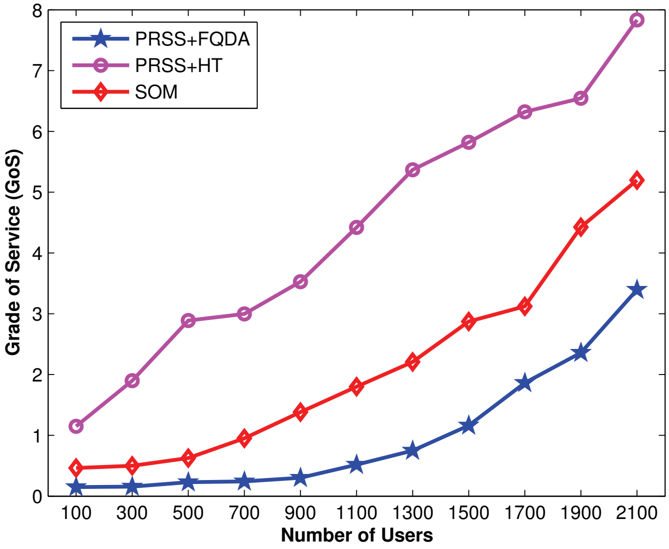


Fig. 10. GoS versus numbers of users (Arrival rate = 3 sec)

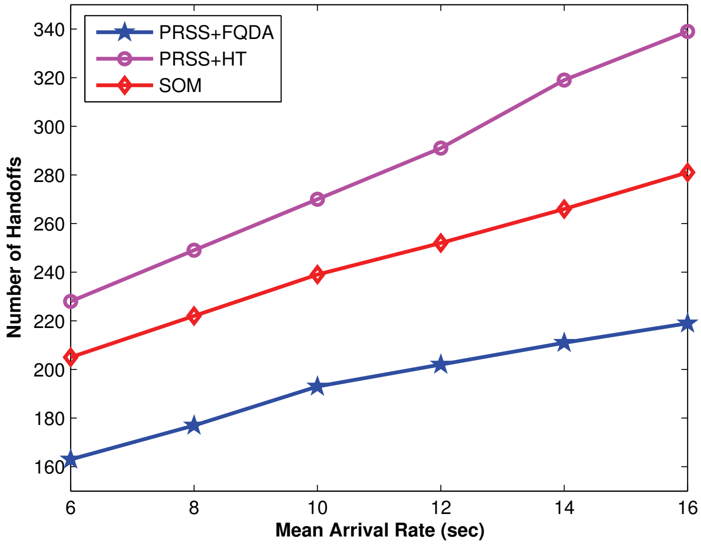


Fig. 11. Number of handoffs versus mean arrival rates (Number of users = 1,500)

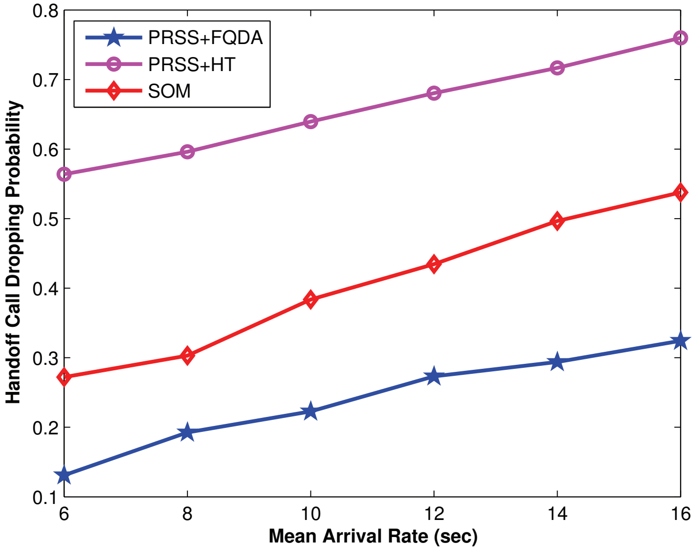


Fig. 12. Handoff call dropping probability versus mean arrival rates (Number of users = 1,500)

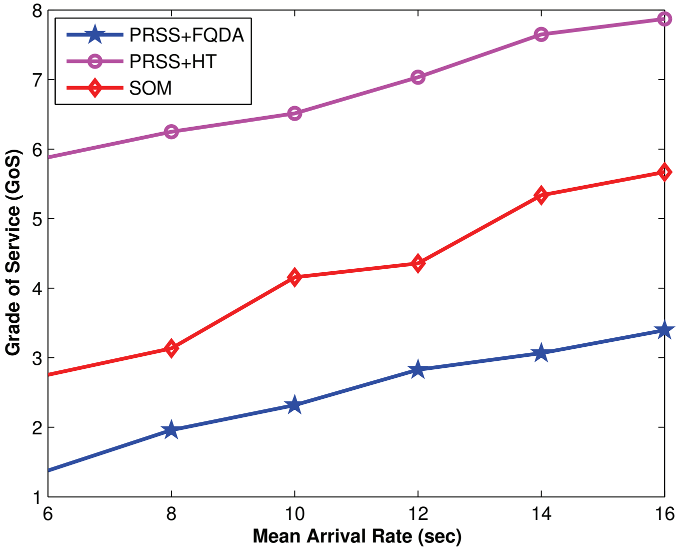


Fig. 13. GoS versus mean arrival rates (Number of users = 1,500)

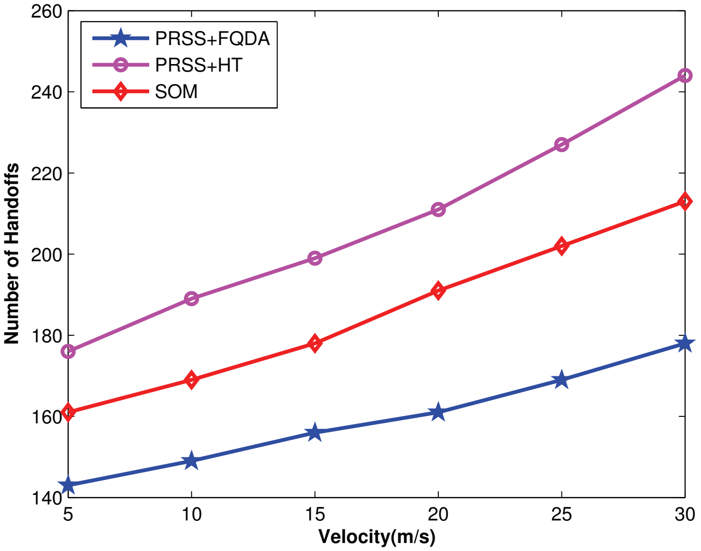


Fig. 14. Number of handoffs versus mobile velocity (Number of users = 1,500 and Arrival rate = 3sec)

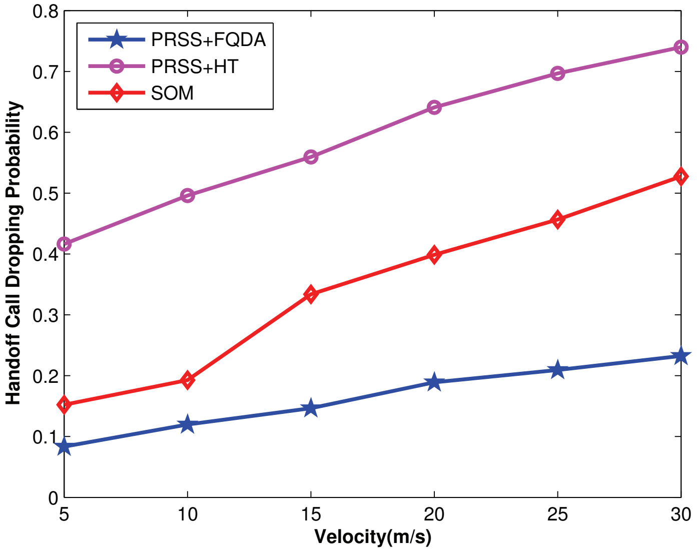


Fig. 15. Handoff call dropping probability versus mobile velocity (Number of users = 1,500 and Arrival rate = 3sec)

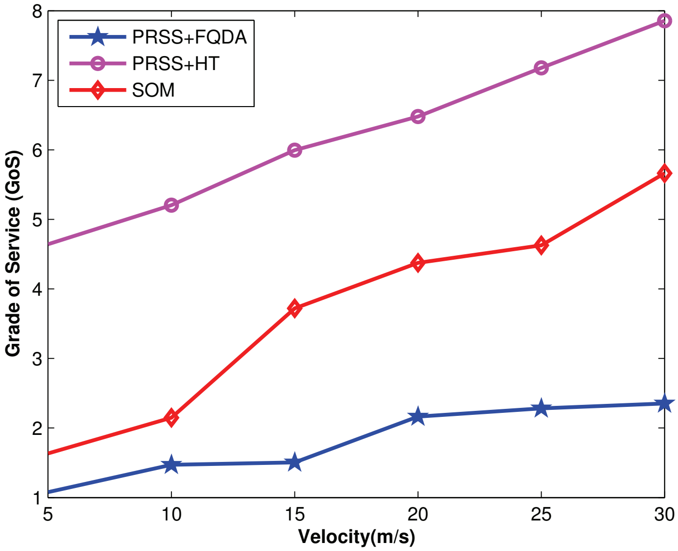


Fig. 16. GoS versus mobile velocity (Number of users = 1,500 and Arrival rate = 3sec)

policy of real time service is to have small delay. Merit function evaluating network conditions and user preference is used as the handoff criteria to determine candidate networks. Fuzzy logic using quantitative decision algorithm makes a final decision to select the optimal target network with the largest QDV. The proposed approach outperforms other approaches in number of vertical handoffs and call dropping probability and GoS.

9. Acknowledgments

This work is supported in part of by Telecommunications Research Industrial and Development Institute (Tridi), National Telecommunications Commission Fund under Grant No. PHD/004/2008.

10. References

Betancur, L., Hincapié, R. & Bustamante, R. (2006). Wimax channel-phy model in network simulator 2, *Workshop on ns-2: the IP Network Simulator Proceeding*, Italy, pp. 1–8.

Bing, H., He, C. & Jiang, L. (2003). Performance analysis of vertical handover in a umts-wlan integrated network, *14th IEEE International Symposium on Personal, Indoor and Mobile Radio Communication Proceedings*, China, pp. 187–191.

Boudriga, N., Obaidat, M. S. & Zarai, F. (2008). Intelligent network functionalities in wireless 4G networks: Integration scheme and simulation analysis, *Computer Communications Journal* Vol. 31: 3752–3759.

Chang, B. & Chen, J. (2008). Cross-layer-based adaptive vertical handoff with predictive rss in heterogeneous wireless networks, *IEEE Transactions on Vehicular Technology* Vol. 57: 3679–3692.

Frattasi, S., Fathi, H., Fitzek, F. H. P. & Prasad, R. (2006). Defining 4G technology from the user's perspective, *IEEE Network* Vol. 20(No. 1): 35–41.

Haibo, X., Hui, T. & Ping, Z. (2009). A novel terminal-controlled handover scheme in heterogeneous wireless networks, *Computers and Electrical Engineering* Vol. 10: 1–11.

Haykin, S. (2009). *Neural networks and learning machines*, Prentice Hall Publishing, USA.

Kassar, M., Kervella, B. & Pujolle, G. (2008). An overview of vertical handover decision strategies in heterogeneous wireless networks, *Computer Communications Journal* Vol. 31: 2607–2620.

Lee, W., Kim, E., Kim, J., Lee, I. & Lee, C. (2009). Movement-aware vertical handoff of wlan and mobile wimax for seamless ubiquitous access, *IEEE Transactions on Consumer Electronics* Vol. 53: 1268–1275.

Majlesi, A. & Khalaj, B. H. (2002). An adaptive fuzzy logic based handoff algorithm for interworking between wlans and mobile networks, *13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Portugal, pp. 2446–2451.

McNair, J. & Fang, Z. (2004). Vertical handoffs in fourth-generation multinetwork environments, *IEEE Wireless Communications* Vol. 11(No. 1): 8–15.

Nay, P. & Zhou, C. (2009). Vertical handoff decision algorithm for integrated umts and leo satellite networks, *International Conference on Communications and Mobile Computing*, China, pp. 180–184.

Nkansah-Gyekye, Y. & Agbinya, J. I. (2008). A vertical handoff decision algorithm for next generation wireless networks, *3rd IEEE International Conference on Broadband Communications, Information Technology and Biomedical Applications*, South Africa, pp. 358–364.

Pahlavan, K., Krishnamurthy, P., Hatami, A., Ylianttila, M., Makela, J.-P., Pichna, R. & Vallstrm, J. (2000). Handoff in hybrid mobile data networks, *IEEE Personal Communications* Vol. 7: 34–37.

Pollini, G. P. (1996). Trends in handover design, *IEEE Communications Magazine* Vol. 34: 82–90.

Sivanandam, S. N., Sumathi, S. & Deepa, S. (2007). *Introduction to Fuzzy Logic using MATLAB*, Springer Publishing, USA.

Song, Q. & Jamalipour, A. (2008). A quality of service negotiation-based vertical handoff decision scheme in heterogeneous wireless systems, *European Journal of Operational Research* Vol. 191: 1059–1074.

Stevens-Navarro, E., Lin, Y. & Wong, V. (2008). An MDP-based vertical handoff decision algorithm for heterogeneous wireless networks, *IEEE Transactions on Vehicular Technology* Vol. 57: 1243–1254.

Stoyanova, M. & Mähönen, P. (2007). Algorithmic approaches for vertical handoff in heterogeneous wireless environment, *IEEE Wireless Communications and Networking Conference*, Hong Kong, pp. 3780–3785.

Sun, C., Stevens-Navarro, E. & Wong, V. W. S. (2008). A constrained MDP-based vertical handoff decision algorithm for 4G wireless networks, *IEEE International Conference on Communications*, China, pp. 2169 – 2174.

Tansu, F. & Salamah, M. (2006). On the vertical handoff decision for wireless overlay networks, *7th IEEE International Symposium on Computer Networks*, France, pp. 111–115.

Xia, L., Ling-ge, J., Chen, H. & Hong-wei, L. (2008). An intelligent vertical handoff algorithm in heterogeneous wireless networks, *International Conference on Neural Networks and Signal Processing*, China, pp. 550–555.

Zurada, J. M. (1992). *Introduction to Artificial Neural Systems*, Addison-Wesley Publishing, USA.

15

Energy Issues and Energy aware Routing in Wireless Ad-hoc Networks

Marco Fotino and Floriano De Rango
University of Calabria
Italy

1. Introduction

The problem of energy efficiency in MANETs can be addressed at different layers. In recent years, many researchers have focused on the optimization of energy consumption of mobile nodes, from different points of view. Some of the proposed solutions try to adjust the transmission power of wireless nodes, other proposals tend to efficiently manage a sleep state for the nodes (these solutions range from pure MAC-layer solutions (as the power management of 802.11) to solutions combining MAC and routing functionality). Finally, there are many proposals which try to define an energy efficient routing protocol, capable of routing data over the network and of saving the battery power of mobile nodes. Such proposals are often completely new, while others aim to add energy-aware functionalities to existing protocols (like AODV, DSR and OLSR).

The aim of energy-aware routing protocols is to reduce energy consumption in transmission of packets between a source and a destination, to avoid routing of packets through nodes with low residual energy, to optimize flooding of routing information over the network and to avoid interference and medium collisions. Many energy efficient routing protocol proposals were originally studied for sensor networks, where the limited energy of nodes is a strong constraint; in MANET, however, the requirements are different: a node has generally more hardware resources (capable of better performance, but consuming more energy) and the protocol must preserve the resources of every node in the network (not only a subset of them, because each node can be, at any time, source or destination of data). A single node failure in sensor networks is usually unimportant if it does not lead to a loss of sensing and communication coverage; ad-hoc networks, instead, are oriented towards personal communication and the loss of connectivity to any node is significant.

In the routing protocol design of mobile nodes, many issues need to be considered in order to offer many important properties such as scalability, QoS support, security, low power consumption and so on. In this chapter we focus on the energy issues facing some important aspects going from the energy model definition for the computation of the energy consumption to energy-aware metrics definition and routing protocol design. If a network composed of mobile nodes communicatiing using a wireless radio and where each node can communicate with each other using the other mobile nodes as relay nodes is applied in a communication system, many challenging design issues need to be addressed. MANET technology became, in the last years, more commercial in comparison with the past where it was used for military purpose and this implies more additional features to offer to the end-user with particular reference to quality of service, security and to node lifetime duration. In this chapter energy saving techniques at network layer and the routing strategies that allow a better energy expenditure and load distribution in order to prolong the network lifetime are considered. After defining a simple energy consumption model to use as reference for the protocol performance evaluation and after introducing some well-known energy based metric, some routing protocols belonging to different families of routing strategies are briefly presented. In particular we refer to proactive routing protocols with particular reference to OLSR, reactive routing with reference to AODV, DSR and LEAR, hybrid routing with reference to GAF, and scalable routing strategies based on the concept of clustering or topological hierarchy.

2. Wireless ad-hoc networks

MANET is a special type of wireless network in which a collection of mobile network interfaces may form a temporary network without aid of any established infrastructure or centralized administration. Ad Hoc wireless network has applications in emergency search-and-rescue operations, decision making in the battlefield, data acquisition operations in hostile terrain, etc. It is featured by dynamic topology (insfrastructrureless), multi-hop communication, limited resources (bandwidth, CPU, battery, etc.) and limited security. These characteristics put special challenges in routing protocol design. The one of the most important objectives of MANET routing protocol is to maximize energy efficiency, since nodes in MANET depend on limited energy resources.

The primary objectives of MANET routing protocols are to maximize network throughput, to maximize network lifetime, and to maximize delay. The network throughput is usually measured by packet delivery ratio while the most significant contribution to energy consumption is measured by routing overhead which is the number or size of routing control packets. A major challenge that a routing protocol designed for ad hoc wireless networks faces is resource constraints. Devices used in the ad hoc wireless networks in most cases require portability and hence they also have size and weight constraints along with the restrictions on the power source. Increasing the battery power may make the nodes bulky and less portable. The energy efficiency remains an important design consideration for these networks. Therefore ad hoc routing protocol must optimally balance these conflicting aspects.

To achieve the desired behavior, some proposals make use of clustering or maintain multiple paths to destinations (in order to share the routing load among different nodes).

The majority of energy efficient routing protocols for MANET try to reduce energy consumption by means of an energy efficient routing metric, used in routing table computation instead of the minimum-hop metric. This way, a routing protocol can easily introduce energy efficiency in its packet forwarding. These protocols try either to route data through the path with maximum energy bottleneck, or to minimize the end-to-end transmission energy for packets, or a weighted combination of both.

The energy optimization of a routing protocol, however, can exploit also other network layer mechanisms, like control information forwarding. In OLSR, for example, the MPR selection mechanism can be varied in an energy-aware way: MPRs can be selected by their residual energy, rather than by their 2-hop neighborhood coverage. Some works applied both techniques (MPR selection criteria modification and path determination algorithm modification) to increase the energy efficiency of OLSR protocol.

3. Issues in MANETs: Energy, scalability and quality of services

Due to the fact that bandwidth is scarce in MANET nodes and that the population in a MANET is increasing the scalability issue for wireless multi-hop routing protocols is mostly concerned with excessive routing message overhead caused by the increase of network population and mobility. Routing table size is also a concern in MANETs because large routing tables imply large control packet size hence large link overhead. Routing protocols generally use either distance-vector or link-state routing algorithms and only in the last years also geographical routing protocols that make use of node location/position have been investigated (De Rango et al., 2006). However, scalability issues in terms of overhead and, consequently number of nodes operating in the network, are strongly related also to energy consumption because higher number of control packets overhead imply more energy consumption spent in transmission, reception and overhearing. This means that trying to design a more scalable protocols can offer more benefits also to the energy saving of mobile nodes in a MANET.

When we consider the design of an energy efficient routing protocols not always this means that the routing strategies are also scalable because the protocols can reduce the energy consumption under just some specific operative conditions such as lower mobility, light traffic load or low number of nodes. This means that the design of an energy-efficient routing protocols should consider also scalability issue in order to apply it in wider scenarios and to be sure that the protocol performance do not degrade too much when some project parameters are changing. At this purpose, more advanced techniques that try to exploit the clustering formation among nodes, the nodes position, zone location or the hierarchical topology structure have been considered and some of these techniques are referred in this chapter. Moreover, another important issue should be considered in the routing strategies applied to MANETs. It is the QoS in terms of many metrics definition such as minimum bandwidth availability, maximum end-to-end delay, minimum delay jitter, path stability and so on. Often, in literature, these QoS issues are not related to energy consumption but in the protocol design some connection between QoS support and energy consumption exist. In particular, the selection of the lowest energy path among a couple of nodes can led to the selection of a longer route with higher end-to-end delay (De Rango, Guerriero, 2006; De Rango, 2011). Moreover, the possibility to offer higher bandwidth to a connection and consequently higher data rate imply often to deplete the battery charge of a node more quickly. In this view, also QoS aware routing protocols should take into account also the energy issues related to the rationale of the forwarding scheme, route maintenance and path discovery. In the rest of the chapter, some of the most famous approaches related to the energy aware routing protocols are presented with particular reference to proactive, reactive, hybrid, cluster-based, hierarchical and position based routing protocols.

4. Energy consumption model

A wireless network interface can be in one of the following four states: Transmit, Receive, Idle or Sleep. Each state represents a different level of energy consumption.

- Transmit: node is transmitting a frame with transmission power P_{tx};
- Receive: node is receiving a frame with reception power P_{rx}. That energy is consumed even if the frame is discarded by the node (because it was intended for another destination, or it was not correctly decoded);

- Idle (listening): even when no messages are being transmitted over the medium, the nodes stay idle and keep listening the medium with P_{idle};
- Sleep: when the radio is turned off and the node is not capable of detecting signals. No communication is possible. The node uses P_{sleep} that is largely smaller than any other power.

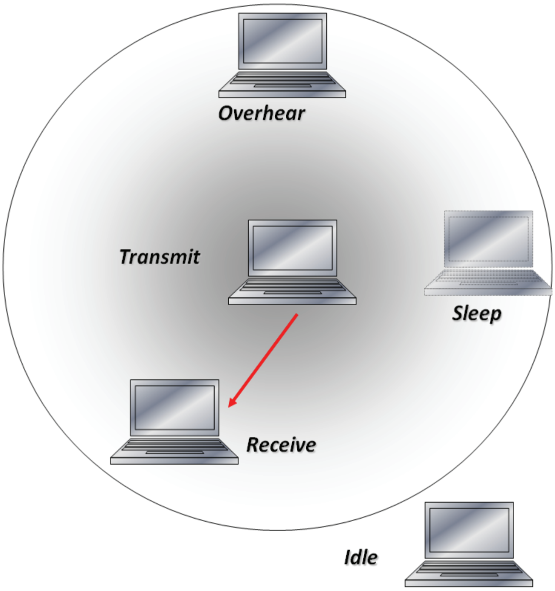


Fig. 1. Energy consumption in a wireless network

In Table 1, typical values of consumption for a wireless interface (measured for a Lucent Silver Wavelan PC Card) are reported.

State	*Power value*
Transmit	P_{tx} = 1.3W
Receive	P_{rx} = 0.9W
Idle	P_{idle} = 0.74W
Sleep	P_{sleep} = 0.047W

Table 1. Power value in each radio state

The energy dissipated in transmitting (E_{tx}) or receiving (E_{rx}) one packet can be calculated as:

$$\begin{aligned} E_{tx} &= P_{tx} \times Duration \\ E_{rx} &= P_{rx} \times Duration \end{aligned} \tag{1}$$

where *Duration* denote the transmission duration of the packet.
When a transmitter transmits a packet to the next hop, because of the shared nature of wireless medium, all its neighbors receive this packet even it is intended to only one of them. Moreover, each node situated between transmitter range and interference range

receives this packet but it cannot decode it. These two problems generate loss of energy. So to compute the energy dissipated by one transmission, we must take into account these losses as follows (Allard et al., 2006):

$$\cos t_{tx}(i) = E_{tx} + n \times E_{rx} \tag{2}$$

where n represents the number of non-sleeping nodes belonging to the interference zone of the transmitter i.

5. Energy aware metrics

The majority of energy efficient routing protocols for MANET try to reduce energy consumption by means of an energy efficient routing metric, used in routing table computation instead of the minimum-hop metric. This way, a routing protocol can easily introduce energy efficiency in its packet forwarding. These protocols try either to route data through the path with maximum energy bottleneck, or to minimize the end-to-end transmission energy for packets, or a weighted combination of both.

A first approach for energy-efficient routing is known as MTPR (Minimum Transmission Power Routing; Toh, 2001). That mechanism uses a simple energy metric, represented by the total energy consumed to forward the information along the route. This way, MTPR reduces the overall transmission power consumed per packet, but it does not directly affect the lifetime of each node (because it does not take into account the available energy of network nodes). However, minimizing transmission energy only differs from shortest-hop routing if nodes can adjust transmission power levels, so that multiple short hops are more advantageous, from an energy point of view, than a single long hop (Kunz, 2008). In 802.11 we do not have access to this capability, so that, in a fixed transmission power context, this metric corresponds to a Shortest Path routing.

Another routing metric, minimizing a function of the remaining battery power of the nodes in a path, is called MBCR (Minimum Battery Cost Routing; Toh, 2001). The proposed battery cost function is

$$f_i(t) = 1 / c_i(t) \tag{3}$$

where $c_i(t)$ is the battery capacity of node n_i at time t. The less capacity a node has, the more reluctant it is to forward packets.

If only the summation of battery costs on a route is considered, a route containing nodes with little remaining battery capacity may still be selected. MMBCR (Minimum Maximum Battery Cost Routing; Toh, 2001), defines the route cost as

$$R(r_j) = \max_{\forall n_i \in r_j} f_i(t) \tag{4}$$

The desired route r_O is obtained so that

$$R(r_O) = \max_{r_j \in r_*} R(r_j) \tag{5}$$

where r_* is the set of all possible routes.

Since MMBCR considers the weakest and crucial node over the path, a route with the best condition among paths impacted by each crucial node over each path is selected. CMMBCR metric (Conditional MMBCR; Toh, 2001) attempts to perform a hybrid approach between MTPR and MMBCR, using the former as long as all nodes in a route have sufficient remaining energy (over a threshold) and the latter when all routes to the destination have at least a node with less energy than the threshold.

Power saving mechanisms based only on the remaining power cannot be used to establish the best route between source and destination nodes. If a node is willing to accept all route requests only because it currently has enough residual battery capacity, too much traffic load will be injected through that node. In this sense, the actual drain rate of power consumption of the node will tend to be high, resulting in an unfair sharp reduction of battery power. To address the above problem, the MDR (Minimum Drain Rate; Kim et al., 2003) mechanism can be utilized with a cost function that takes into account the drain rate index (DR) and the residual battery power (RBP) to measure the energy dissipation rate in a given node.

In the MDR mechanism, the ratio

$$f_i(t) = \frac{RBP_i(t)}{DR_i(t)} \tag{6}$$

at node n_i, calculated at time t, indicates when the remaining battery of node n_i will be exhausted, i.e., how long node n_i can keep up with routing operations with current traffic conditions. Therefore, the maximum lifetime of a given path r_j is determined by the minimum value of $f_i(t)$ over the path. Finally, the MDR mechanism is based on selecting the route r_O, contained in the set of all possible routes between the source and the destination r_*, having the highest maximum lifetime value.

Since the drain rate is calculated at regular time intervals, its measure is affected by isolated consumption peaks (both positive or negative). To avoid the use of incorrect values of drain rate during these peaks, an α parameter can be introduced. This parameter makes the drain rate value between adjacent intervals smoother, acting in the following manner: after calculating the drain rate sample at interval t, $DR_{sample}(i)$, MDR uses a value of drain rate of

$$DR(i) = (1-\alpha)\cdot DR_{sample}(i) + \alpha \cdot DR(i-1) \tag{7}$$

MDR suffers from the same problem as MMBCR, ignoring the total transmission power consumed by a single path: this way, it could even lead to a higher overall energy consumption in the network. To prevent this issue, MDR can be introduced in a hybrid way, as a CMDR (Conditional MDR) metric: as far as all nodes in a route have sufficient remaining lifetime (over a threshold), a simple MTPR approach is used.

Other works (like Misra & Banerjee, 2002) use a larger number of variables in the cost function of the algorithms, for example by taking into account not only the residual energy and the transmission power, but also the energy cost of possible packet retransmissions. Similarly to the MDR metric, an important aspect for the design of energy aware routing protocols is highlighted: the estimation of future energy consumption. The energy that is expected to be used in order to successfully send a packet across a given link is estimated by a cost function that comprises both a node-specific parameter (battery power B_i of node i) and a link-specific parameter (packet transmission energy $E_{i,j}$). The cost of the reliable communication across the link (between nodes i and j) is defined as

$$C_{i,j} = \frac{B_i}{E_{i,j}} \tag{8}$$

The expected transmission energy is defined by the power needed to transmit a packet over the link between nodes i and j ($T_{i,j}$) and the link's packet error probability ($p_{i,j}$):

$$E_{i,j} = \frac{T_{i,j}}{(1-p_{i,j})^L} \tag{9}$$

The main reason for adopting the above is that link characteristics can significantly affect energy consumption and can lead to excessive retransmissions of packets. The cost of choosing a particular link is defined as the maximum number of packets that can be transmitted by the transmitting node over that specific link. It is also assumed that there is complete absence of any other cross traffic at that node. The maximum lifetime of a given path is determined by the weakest intermediate node.

Another approach (Chiasserini & Rao, 2000) make use of the available battery capacity by means of battery-sensitive routing. That approach studies the lifetime of the battery and proposes an algorithm based on two processes, namely, recovery (reimbursement) and discharging loss (over-consumed power). These processes are experienced when either no traffic or new traffic is transmitted. This study led to the design of a cost function that penalizes the discharging loss event and prioritizes routes with "well recovered" nodes. Thus, battery recovery can take place and a node's maximum battery capacity can be attained. The selection function is a minimum function over the cost functions of all routes.

6. Energy saving techniques at routing layer

The problem of energy efficiency in MANETs can be addressed at different layers. In recent years, many researchers have focused on the optimization of energy consumption of mobile nodes, from different points of view. Some of the proposed solutions try to adjust the transmission power of wireless nodes (Cardei et al., 2004; Ingelrest et al., 2006). Other proposals tend to efficiently manage a sleep state for the nodes: these solutions range from pure MAC-layer solutions (as the power management of 802.11) to solutions combining MAC and routing functionality (Xu et al., 2001). Finally, there are many proposals which try to define an energy efficient routing protocol, capable of routing data over the network and of saving the battery power of mobile nodes (Toh, 2001; Jones et al., 2001; Lindsey et al., 2001; Wan et al., 2001; Kim et al., 2003; Jinet al., 2005; Taddia et al., 2005). Such proposals are often completely new, while others aim to add energy-aware functionalities to existing protocols, like AODV (Senouci & Naimi, 2005; Jung et al., 2005), DSR (Garcia et al., 2003; Luo et al., 2003) and OLSR (Ghanem et al., 2005; Benslimane et al., 2006; Guo & Malakooti, 2007).

The aim of energy-aware routing protocols is to reduce energy consumption in transmission of packets between a source and a destination, to avoid routing of packets through nodes with low residual energy, to optimize flooding of routing information over the network and to avoid interference and medium collisions.

Some routing protocols organize wireless nodes into clusters, such as Leach (Heinzelman et al., 2000). In (Xia & Vlajic, 2007) the conditions under which such protocols are energy efficient are established and the optimal radius of a cluster is determined.

Existing energy efficient routing protocols can be first distinguished by the number of paths maintained to a destination: a single path or multiple paths.

Multipath routing protocols (Shah & Rabaey, 2002; Ganesan et al., 2001) have the advantage of sharing load of any flow on several paths, leading to a lower consumption on the nodes of the selected paths. It has been shown in (Srinivas & Modiano, 2003) that two paths with different links are generally sufficient.

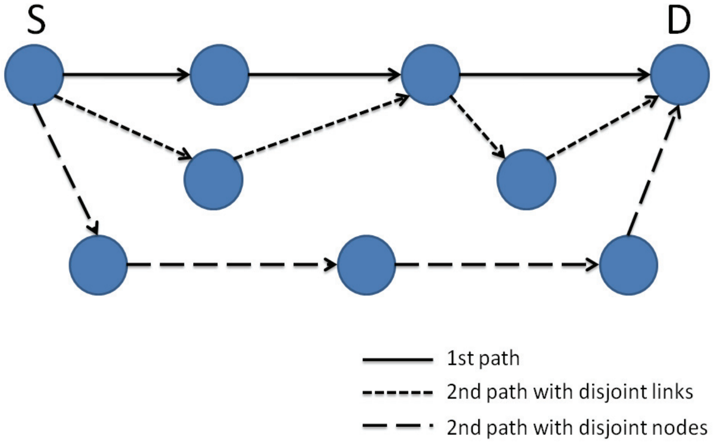


Fig. 2. Multipath Routing

We can distinguish three families of energy efficient routing protocols:

- the protocols selecting the path consuming the minimum energy. The advantage is that each transmission of a packet from its source to its destination minimizes the energy consumed. We can cite for example (Senouci & Naimi, 2005) and a more sophisticated protocol (Kwon & Shroff, 2006) where the selected path minimizes the additional energy dissipated by the routing of the new flow, taking into account the SINR and the energy lost in interferences. However, such protocols use always the same nodes (those minimizing the energy consumed) without any consideration on their residual energy. Consequently, these nodes will exhaust their battery more quickly than the others and the network lifetime is not maximized.
- the protocols selecting the path visiting the nodes with the highest residual energy, such as (Hassanein, 2006). Each flow is ensured to have enough energy on the selected path: depleted nodes are avoided. However, the path selected does not minimize the energy needed to transmit a flow packet from its source to its destination. Hence, the network lifetime may not be maximized.
- the hybrid protocols selecting the path with the minimum cost, where the cost takes into account the residual energy of each visited node (and possibly its neighbors) and the energy consumption of a packet on this path. These protocols avoid the problems encountered by the protocols of the two previous categories by weighing the factors used in the cost computation. We can cite for instance (Shresta, 2006).

Routing protocols for mobile ad hoc networks have different features. Regarding the way to exchange routing information, the main difference is between reactive and proactive routing protocols. A reactive (or on-demand) routing protocol determines routes only when there is any data to send. If a route is unknown the source node initiates a search to find one and it is primarily interested in finding any route to a destination, not necessarily the optimal route. A proactive routing protocol, instead, attempts to maintain routes to all destinations at all time, regardless of whether they are needed. To support this, the routing protocol propagates information updates about the network's topology or connectivity through the network. From the node organization point of view, there can be a hierarchical routing system (some routers form a sort of backbone) or a flat address space (where the routers are peers of all others).

6.1 Proactive energy-aware routing

With table-driven routing protocols, each node attempts to maintain consistent, up-to-date routing information to every other node in the network. This is done in response to changes in the network by having each node update its routing table and propagate the updates to its neighboring nodes. Thus, it is *proactive* in the sense that when a packet needs to be forwarded the route is already known and can be immediately used. As is the case for wired networks, the routing table is constructed using either *link-state* or *distance vector* algorithms containing a list of all the destinations, the next hop, and the number of hops to each destination. Many routing protocols including *Destination-Sequenced Distance Vector (DSDV)* and *Fisheye State Routing (FSR)* protocol belong to this category, and they differ in the number of routing tables manipulated and the methods used to exchange and maintain routing tables.

The energy optimization of a proactive routing protocol can exploit various network layer mechanisms, like control information forwarding. In OLSR, for example, the MPR selection mechanism can be varied in an energy-aware way. As suggested in RFC 3626, MPRs can be selected by their residual energy, rather than by their 2-hop neighborhood coverage (Ghanem et al., 2005). Some works applied both techniques (MPR selection criteria modification and path determination algorithm modification) to increase the energy efficiency of OLSR protocol (Benslimane et al, 2006; De Rango et al., 2008; Kunz, 2008).

Another mechanism that allows energy saving in OLSR protocol (without changing its behavior) is the Overhearing Exclusion (De Rango et al., 2008). Turning off the device when a unicast message exchange happens in the node's neighborhood, can save a large amount of energy. This can be achieved using the signaling mechanisms of the lower layers (i.e. the RTS/CTS exchange performed by IEEE 802.11 to avoid collisions), and does not affect protocol performance. In fact, OLSR does not take any advantage from unicast network information directed to other nodes (while other protocols, such as DSR, have mechanisms to do so).

6.2 Reactive energy-aware routing

With on-demand driven routing, routes are discovered only when a source node desires them. *Route discovery* and *route maintenance* are two main procedures: The route discovery process involves sending route-request packets from a source to its neighbor nodes, which then forward the request to their neighbors, and so on. Once the route-request reaches the destination node, it responds by unicasting a route-reply packet back to the source node via the neighbor from which it first received the route-request. When the route-request reaches an intermediate node that has a sufficiently up-to-date route, it stops forwarding and sends a route-reply message back to the source. Once the route is established, some form of route maintenance process maintains it in each node's internal data structure called a route-cache until the destination becomes inaccessible along the route. Note that each node learns the routing paths as time passes not only as a source or an intermediate node but also as an overhearing neighbor node. In contrast to table-driven routing protocols, not all up-to-date routes are maintained at every node. *Dynamic Source Routing* (DSR) and *Ad-Hoc On-Demand Distance Vector* (AODV) are examples of on-demand driven protocols.

In generic on-demand (also known as reactive) ad-hoc algorithms, all nodes participate in the phase of path searching, while the final decision is made in the source or destination node. The Local Energy-Aware Routing (LEAR; Woo et al., 2001) algorithm grants each node in the network permission to decide whether to participate in route searching: this way, the decision process is spread among all nodes in the network. That algorithm uses the energy profile of the nodes as a main criterion for the routing decision. The residual energy of each node defines the reluctance or willingness of that node to reply to route requests and forward data traffic. When energy E_i in a node i is lower than a given threshold Th

$$E_i < Th \tag{10}$$

the node does not forward the route request control message, but simply drops it. Thus, it will not participate in the selection and forwarding phase.

The technique of spreading the responsibility from the source/destination nodes to the intermediate nodes avoids the needing for a periodic exchange of control information, thus leading to reduced bandwidth and energy consumption. This technique has been commonly used to improve the performance of the routing protocols in many recent approaches.

6.3 Hybrid energy-aware routing

The work in (Xu et al., 2001) introduces a new way of optimizing the energy consumption in a wireless network, independently from the routing protocol adopted by the nodes. Assuming that all the devices in the network are equipped with a GPS (Global Positioning System) receiver, that work introduces the Geographical Adaptive Fidelity (GAF) for ad-hoc wireless networks. GAF conserves energy by identifying nodes that are equivalent from a routing perspective and then turning off unnecessary nodes, keeping a constant level of routing fidelity. GAF moderates this policy using application- and system-level information; nodes that source or sink data remain on and intermediate nodes monitor and balance energy use. Simulations of GAF suggest that network lifetime increases proportionally to node density. Power consumption in current wireless networks is idle-time dominated, so GAF focus on turning the radio off as much as possible.

6.4 Comparative performance evaluation from an energetic point of view

Many energy efficient routing protocol proposals were originally studied for sensor networks, where the limited energy of nodes is a strong constraint; in MANET, however, the requirements are different: a node has generally more hardware resources (capable of better performance, but consuming more energy) and the protocol must preserve the resources of every node in the network (not only a subset of them, because each node can be, at any time, source or destination of data). A single node failure in sensor networks is usually unimportant if it does not lead to a loss of sensing and communication coverage; ad-hoc networks, instead, are oriented towards personal communication and the loss of connectivity to any node is significant.

The lifetime of a network is usually defined according to the following criteria (Vassileva & Barcelo-Arroyo, 2008):

- the time until the first node burns out its entire battery budget;

- the time until a certain proportion of the nodes fails;
- the time until network partitioning occurs.

A single node failure represents a serious problem in ad-hoc networks, because its occurrence can lead to the network partitioning. In contrast, a single node failure in sensor networks is usually unimportant if it does not lead to a loss of sensing and communication coverage. Ad hoc networks are oriented towards personal communications and the loss of connectivity to any node is significant. Consider, for example, a disaster recovery scenario. In such case, it is important that the rescuers do not lose connectivity with any other member of their team, and the connectivity among rescuers should be maintained as long as possible, or at least the duration of the rescue operation. Network partitioning interrupts communication sessions and can be caused by node movement or by node failure due to energy depletion. While the former cannot be controlled by the routing protocol, the latter can be avoided through appropriate routing decisions.

Operational lifetime can be defined as the time until network partitioning occurs due to battery outage. In order to achieve the objective of maintaining connectivity as long as possible, the distribution of network tasks among its nodes should be equal so that they all decrease power at the same rate and eventually run out of energy at approximately the same time. The network must be designed to achieve the simultaneous failure of the nodes (due to a lack of energy), so that communication requirements are met. This leads to consider as the operational lifetime of such networks their relative lifetime, rather than the absolute lifetime of their devices. The useful lifetime of ad-hoc networks can be significantly lower than the network's devices lifetime, but from an engineering and application perspective the former time span is much more interesting and meaningful. For instance, a case could be envisaged in which some nodes have fully charged batteries but are unable to establish successful communications because they belong to disconnected parts of the network or must communicate with nodes that are turned off due to a lack of energy. In such a scenario, the absolute lifetime of a network will be longer compared to the useful life span, but this is not of practical interest.

Many works have been presented in literature to give a measure of the energy consumption of various routing solutions in a wide range of scenarios, exploring the behavior of different protocols (especially OLSR and DSR) and trying to highlight the strength and weakness points of each of them (De Rango et al., 2008; Fotino et al., 2007; McCabe et al., 2005; Zhao & Tong, 2005). These researches are a good starting point for every energy-aware routing proposal for MANETs.

To achieve the desired behavior, some proposals make use of clustering (Heinzelman et al., 2000) or maintain multiple paths to destinations (in order to share the routing load among different nodes; Srinivas & Modiano, 2003).

7. Scalable energy-aware routing

In a hierarchical network, the network elements are partitioned into several groups, called clusters. In each cluster, there is a master node that manages all the other nodes (slave nodes) within the cluster. The depth of the network can vary from a single tier to multiple tiers. However, most hierarchical networks are two-tier networks.

Two-tier mobile ad hoc networks require sophisticated algorithms to perform clustering based on limited resources, such as the energy of each node, to communicate with each other. The cluster area of a node is related to the transmission power. Therefore, a larger cluster area requires more energy. The energy required by a two-tier mobile ad hoc network varies with the clustering configuration (the master node selection of slave nodes) because the transmission power of each node must be set to satisfy the minimum power level at the receiving node.

Therefore, there exists an optimum clustering configuration that minimizes the call drop rate and the energy required for the still snapshot of the network. However, the optimum clustering configuration cannot be calculated quickly. A heuristic clustering scheme resulting in energy conservation for the network that can be implemented and executed in a limited time is needed for real-time clustering.

In (Ryu et al., 2001) the authors propose two distributed heuristic clustering schemes for energy conservation in two-tiered mobile ad hoc networks. The proposed schemes can be implemented and executed in real time. The mean transmission power and the call drop rate for the proposed schemes approximate optimum results. Hence, the proposed schemes are suitable for periodic or event-driven cluster reconfiguration. The proposed double-phase scheme is useful when energy conservation and call completion are more important than computing power and the speed of the scheme. In the opposite case, the proposed single-phase scheme can be adopted.

8. Implementation issues in energy management functionalities

Aiming to extend the time until network partitioning, routing protocol designers often try to optimize the use of battery power, in order to maximize the life of a node. However, extending nodes' lifetime could not be the better way to increase the connectivity between all of the nodes in the network.

The min-max algorithms are implemented to overcome the problem that arises when the total energy cost of routes is used as an argument for the selection of a route, that is, when nodes with low residual energy are excluded. However, if these protocols are analyzed in terms of a network's operational lifetime, the problem of extending the network's lifespan for as long as possible persists. Simulation results (like in Cao et al., 2007) show that protocols that implement min-max algorithms or the energy drain rate have lower values for the standard deviation of the remaining energy in comparison with algorithms that use transmission power as a metric. Furthermore, the distribution of the energy of the nodes along the path is not even in any of the protocols. If in the cost function it is taken into account only the specific energy state of the nodes without considering the overall distribution of the energy along the routes, optimal results will not be obtained when the operational lifetime of a network is being examined.

The energy-aware protocols usually implement only energy-wise metrics. An improvement on this general approach is the inclusion of the speed with which the battery is burned. The energy drain rate is helpful in stopping a node from powering down. It does so by deviating traffic when a certain threshold is reached. The load at each node and in its neighbors is an indicator of the energy to be consumed for transmitting packets by a particular node. Moreover, it accounts for the shared nature of the radio as a medium. The network tasks in which each node is involved are a main item in the battery budget. When this item is considered along with the current energy state of a node, it can regulate the speed of energy consumption.

Additional metrics should be considered, such as the fact that when neighboring nodes are engaged in transmitting packets, they are competing for the wireless medium.

Retransmissions that may possibly take place (Misra & Banerjee, 2002) should also be taken into consideration. The resulting collisions and retransmissions are energy-consuming and cannot simply be represented by the residual energy metric.

9. Conclusions

Since the majority of the devices for personal mobile communication are powered by batteries, the study of energy efficiency in wireless networks raised as a primary constraint for MANETs. In the last few years, a number of researchers have focused their attention on this issue. While the energy consumption problem has been widely considered in wireless sensor networks, mobile ad-hoc networks present a completely different set of constraints to take into account. This work tries to briefly survey the state of the art about energy efficient routing approaches for ad-hoc networks.

The main proposals in literature are presented and the main approaches adopted for ad-hoc energy consumption reduction are explained. The works presented are categorized by the nature of their behavior: proactive, reactive and hybrid ones. In many cases, it is difficult to compare them directly since each method has a different goal with different assumptions and employs different means to achieve the goal. Moreover, the energy-aware protocols' performance are often compared with classical (non energy-aware) protocols, making difficult to compare the different proposed solutions among them.

The primary goal of this work is to highlight all the energy-aware approaches to date, putting in evidence their strength and weakness points. The needing for an efficient, energy-aware routing scheme for the devices of a mobile ad-hoc network is rising very fast, with the growing diffusion of devices for personal communications.

10. References

Allard, G.; Minet, P.; Nguyen, D. Q. & Shresta N. (2006). Evaluation of the energy consumption in MANET, *Adhoc-Now 2006*, Ottawa, Canada, August 2006

Benslimane, A. ; El Khoury, R.; El Azouzi, R. & Pierre, S. (2006). Energy Power-Aware Routing in OLSR Protocol, *Proc. First International Mobile Computer and Wireless Communication Conference*, September 2006, pp. 14-19

Bheemalingaiah M., Naidu M.M., Rao D.S. (2009). Energy aware Clustered based Multipath Routing in Mobile Ad Hoc Networks, I.J.Communications, Network and System Sciences, 2009, 2, 91-168

Cao, L.; Dahlberg, T. & Wang, Y. (2007). Performance evaluation of energy efficient ad hoc routing protocols, *Proc. IPCCC*, IEEE , 2007, pp. 306-313

Cardei, M. ; Wu, J. & Yang S. (2004). Topology Control in Ad hoc Wireless Networks with Hitch-hiking, *IEEE SECON 2004*, October 2004

Chiasserini, C.-F. & Rao, R. R. (2000). Routing protocols to maximize battery efficiency, Proc. *MILCOM*, IEEE, 2000, pp. 496-500

De Rango, F.; Fotino, M. & Marano, S. (2008). EE-OLSR: Energy Efficient OLSR Routing Protocol for Mobile Ad-hoc Networks, *Milcom'08*

De Rango F., Guerriero F., Marano S., Bruno E. (2006). A Multiobjective Approach for Energy Consumption and Link Stability Issues in Ad Hoc Networks, IEEE Communications Letters, Vol.10, N°1, Jan.2006, pp.28-30.

De Rango F., Gerla M., Marano S. (2006). A Scalable Routing Scheme with Group Motion Support in Large and Dense Wireless Ad Hoc Networks, Elsevier Computers and Electrical Engineering Journal, Vol.32, Issue 1-3, May 2006, pp.224-240.

De Rango F., Cano J.-C., Fotino M., Calafate C., Manzoni P., Marano S. (2008). OLSR vs DSR: A comparative analysis of proactive and reactive mechanisms from an energetic point of view in Wireless Ad HocNetworks, Computer Communication Journal, Elsevier, Oct.2008, pp. 3843-3854.

De Rango F., Guerriero F., Fazio P.(2011). Link-Stability and Energy aware Routing Protocol in Distributed Wireless Networks," to be published on IEEE Transaction on Parallel and Distributed Systems.

Fotino, M. ; Gozzi, A. ; Cano, J.-C. ; Calafate, C. ; De Rango, F. ; Marano, S. & Manzoni, P. (2007). Evaluating Energy-aware Behavior of Proactive and Reactive Routing Protocols for Mobile Ad Hoc Networks, *10th International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'07)*, 16-18 July, San Diego, CA, USA

Ganesan, D.; Govindan, R.; Shenker, S. & Estrin, D. (2001). Highly-resilient, energy-efficient multipath routing in wireless sensor networks, *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, volume 1, no 2, 2001

Garcia, J.-E. ; Kallel, A. ; Kyamakya, K. ; Jobman, K. ; Cano, J. C. & Manzoni, P. (2003). A novel DSR-based energy-efficient routing algorithm for mobile ad-hoc networks, *58th IEEE Vehicular Technology Conference (VTC Fall. 2003)*, Orlando, Florida, USA, 6–9 October 2003, pp. 2849–2854

Ghanem, N.; Boumerdassi, S. & Renault, E. (2005). New energy saving mechanisms for mobile ad-hoc networks using OLSR. *Proc. 2nd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, Oct. 2005, pp. 273 – 274

Guo, Z. & Malakooti, B. (2007). Energy Aware Proactive MANET Routing with Prediction on Energy Consumption, *Proc. 2007 International Conference on Wireless Algorithms, Systems and Applications*, August 2007, pp. 287-293

Hassanein, H.; Luo, J. (2006). Reliable Energy Aware Routing In Wireless Sensor networks, *IEEE DSSNS 2006*, April 2006

Heinzelman, W. ; Chandrakasan, A. & Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks, *HICSS 2000*, Maui, Hawaii, USA, vol. 2, January 2000

Ingelrest, F. ; Simplot-Ryl, D. & Stojmenovic, I. (2006). Optimal Transmission Radius for Energy Efficient Broadcasting Protocols in Ad Hoc Networks, *IEEE Transactions on Parallel and Distributed Systems*, June 2006

Jin, X. ; Cai, W. & Zhang, Y. (2005). A RED based minimum energy routing algorithm for wireless ad-hoc networks, *Proc. 2005 International Conference on Wireless Communications, Networking and Mobile Computing*, Sept. 2005, pp. 757 – 761

Jones, C. E. ; Sivalingam, K. M. ; Agrawal, P. & Chen, J. C. (2001). A survey of energy efficient network protocols for wireless networks, *Wireless Network Journal*, 17 (4) (2001) 343–358

Jung, S. ; Hundewale, N. & Zelikovsky, A. (2005). Energy efficiency of load balancing in MANET routing protocols, *Proc. Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, 2005 and *First ACIS International Workshop on Self-Assembling Wireless Networks*, May 2005, pp. 476 – 483

Kim, D. ; Garcia-Luna-Aceves, J. J. ; Obraczka, K. ; Cano, J. C. & Manzoni, P. (2003). Routing Mechanisms for Mobile Ad Hoc Networks Based on the Energy Drain Rate", *IEEE Transactions on Mobile Computing*, Vol. 2, No. 2, Jan. 2003, pp. 161-173

Kunz, T. (2008). Energy-Efficient Manet Routing: Ideal vs Realistic Performance, 2008

Kwon, S. & Shroff, N. B. (2006). Energy-Efficient Interference-Based Routing for Multi-hop Wireless Networks, *IEEE INFOCOM 2006*, Barcelona, Spain, April 2006

Lindsey, S. ; Sivalingam, K. & Raghavendra, C. S. (2001). Power optimization in routing protocols for wireless and mobile networks, *Handbook of Wireless Networks and Mobile Computing*, Wiley, 2001

Luo, Y. ; Wang, J. & Chen, S. (2006). An energy-efficient DSR routing protocol based on mobility prediction, *Proc. International Symposium on a World of Wireless, Mobile and Multimedia Networks*, June 2006

McCabe, A. ; Fredin, M. ; Cullen, A. & Axelsson L. (2005). A power consumption study of DSR and OLSR, *IEEE Military Communications Conference (MILCOM'05)*, 17–20 October 2005, pp.1954–1960

Misra, A. & Banerjee S. (2002). MRPC : Maximizing network lifetime for reliable routing in wireless environments, *Proc. IEEE WCNC*, 2002, pp. 800 – 806

Ryu, J.-H.; Song, S. & Cho, D.-H. (2001). New Clustering Schemes for Energy Conservation in Two-Tiered Mobile Ad-Hoc Networks, *Proc. IEEE ICC'01*, vo1. 3, June 2001, pp. 862–66.

Senouci, S. & Naimi, M. (2005). New routing for balanced energy consumption in mobile ad hoc networks, *Proc. 2nd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, Oct. 2005, pp. 238 – 241

Shah, R.C. & Rabaey, J.M. (2002). Energy Aware Routing for Low Energy Ad Hoc Sensor Networks, *IEEE WCNC 2002*, Volume 1, pp. 17-21, March 2002

Shresta, N. (2006). Reception Awareness for Energy Conservation in Ad Hoc Networks, *PhD Thesis*, Macquarie University Sydney, Australia, November 2006

Srinivas, A. & Modiano, E. (2003). Minimum Energy Disjoint Path Routing in Wireless Ad-Hoc Networks, *MOBICOM'03*, September 2003

Taddia, C. ; Giovanardi, A. ; Mazzini, G. & Zorzi, M. (2005). Energy efficient unicast routing protocols over 802.11b, *IEEE Global Telecommunications Conference (GLOBECOM'05)*, 28 November to 2 December 2005

Toh, C.-K. (2001). Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks, *IEEE Communication Magazine*

Vassileva, N. & Barcelo-Arroyo, F. (2008). A Survey of Routing Protocols for Maximizing the Lifetime of Ad Hoc Wireless Networks, *International Journal of Software Engineering and Its Applications*, Vol. 2, No. 3, July, 2008

Wan, P.-J. ; Calinescu, G. ; Li, X.-Y. & Frieder, O. (2001). Minimum-energy broadcast routing in static ad hoc wireless networks, *IEEE INFOCOM*, Anchorage, AK, 2001

Woo, K. ; Yu, C. ; Lee, D. ; Youn, H. & Lee, B. (2001) Non-blocking, localized routing algorithm for balanced energy consumption in mobile ad hoc networks, *Proc. MSCOTS*, ACM/IEEE, 2001, pp.117-124

Xia, D. & Vlajic, N. (2007). Near-optimal node clustering in wireless sensor networks for environment monitoring, *AINA 2007*, Niagara Falls, Ontario, Canada, May 2007

Xu, Y.; Heidemann, J. & Estrin, D. (2001). Geography-informed Energy Conservation for Ad Hoc Routing, *Proc. ACM Mobile Computer and Networking Conference*, July 2001, pp. 70-84

Zhao, Q. & Tong, L. (2005). Energy efficiency of large-scale wireless networks : proactive versus reactive networking, *IEEE Journal on Selected Areas in Communications*, 23 (5) (2005) 1100–1113

Part 3

Routing in Ad Hoc Networks

16

Routing in Mobile Ad Hoc Networks

Fenglien Lee
University of Guam
Guam 96923,
USA

1. Introduction

A mobile ad hoc network (MANET), sometimes called a mobile mesh network, is a self- configuring network of mobile devices connected by wireless links. In other words, a MANET is a collection of communication nodes that wish to communicate with each other, but has no fixed infrastructure and no predetermined topology of wireless links. Each node in a MANET is free to move independently in any direction, and will therefore change its links to other devices frequently. Individual nodes are responsible for dynamically discovering other nodes that they can directly communicate with. Due to the limitation of signal transmission range in each node, not all nodes can directly communicate with each other. Each node must forward traffic unrelated to its own use, and therefore be a router.

The primary challenge in building a MANET is equipping each device to continuously maintain the information required to properly route traffic. Therefore, nodes are required to relay packets on behalf of other nodes in order to deliver data across the network. A significant feature of ad hoc networks is that changes in connectivity and link characteristics are introduced due to node mobility and power control practices.

Ad hoc networks can be built around any wireless technology, including infrared, radio frequency (RF), global positioning system (GPS), and so on. Usually, each node is equipped with a transmitter and a receiver to communicate with other nodes [Lee2009] [Wiki2010a].

1.1 Routing in a MANET

The absence of fixed infrastructure in a MANET poses several types of challenges. The biggest challenge among them is routing. Routing is the process of selecting paths in a network along which to send data packets. An ad hoc routing protocol is a convention, or standard, that controls how nodes decide which way to route packets between computing devices in a mobile ad-hoc network.

In ad hoc networks, nodes do not start out familiar with the topology of their networks; instead, they have to discover it. The basic idea is that a new node may announce its presence and should listen for announcements broadcast by its neighbors. Each node learns about nearby nodes and how to reach them, and may announce that it can reach them too. The routing process usually directs forwarding on the basis of routing tables which maintain a record of the routes to various network destinations. Thus, constructing routing tables, which are held in the router's memory, is very important for efficient routing [Lee2009][Wiki2010b].

1.2 Routing protocols for MANET

The growth of laptops and 802.11/Wi-Fi wireless networking has made MANETs a popular research topic since the 1990s. Many academic papers evaluate protocols and abilities assuming varying degrees of mobility within a bounded space, usually with all nodes within a few hops of each other and usually with nodes sending data at a constant rate. Different protocols are then evaluated based on the packet drop rate, average routing load, average end-to-end-delay, and other measures. The proposed solutions for routing protocols could be grouped in three categories: proactive (or table-driven), reactive (or on-demand), and hybrid protocols. Even the reactive protocols have become the main stream for MANET routing. In this chapter, we introduce some popular routing protocols in each of the three categories and for IPv6 networks [Lee2009][Wiki2010a][Wiki2010c].

1.3 Applications for MANET

Ad hoc networks are suited for use in situations where infrastructure is either not available or not trusted, such as a communication network for military soldiers in a field, a mobile network of laptop computers in a conference or campus setting, temporary offices in a campaign headquarters, wireless sensor networks for biological research, mobile social networks such as Facebook, MySpace and Twitter, and mobile mesh networks for Wi-Fi devices [Lee2009].

2. Proactive routing protocols

Every proactive routing protocol usually needs to maintain accurate information in their routing tables. It attempts to continuously evaluate all of the routes within a network. This means the protocol maintains fresh lists of destinations and their routes by periodically distributing routing tables throughout the network. So that when a packet needs to be forwarded, a route is already known and can be used immediately. Once the routing tables are setup, then data (packets) transmissions will be as fast and easy as in the tradition wired networks.

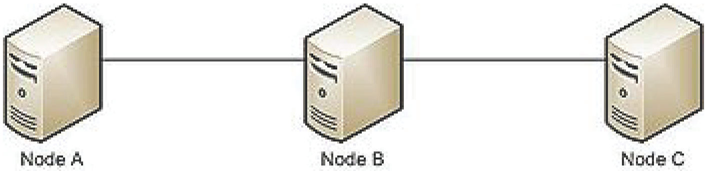
Unfortunately, it is a big overhead to maintain routing tables in the mobile ad hoc network environment. Therefore, the proactive routing protocols have the following common disadvantages:

1. Respective amount of data for maintaining routing information.
2. Slow reaction on restructuring network and failures of individual nodes.

Proactive routing protocols became less popular after more and more reactive routing protocols were introduced. In this section, we introduce three popular proactive routing protocols – DSDV, WRP and OLSR. Besides the three popular protocols, there are many other proactive routing protocols for MNAET, such as CGSR, HSR, MMRP and so on [Wiki2010c][Sholander2002].

2.1 Destination-Sequenced Distance Vector (DSDV)

Destination-Sequenced Distance-Vector Routing (DSDV) is a table-driven routing scheme for ad hoc mobile networks based on the Bellman-Ford algorithm. It was developed by C. Perkins and P. Bhagwat in 1994. The main contribution of the algorithm was to solve the routing loop problem. Each entry in the routing table contains a sequence number. If a link presents the sequence numbers are even generally, otherwise an odd number is used. The number is generated by the destination, and the emitter needs to send out the next update with this number. Routing information is distributed between nodes by sending *full dumps* infrequently and smaller incremental updates more frequently.



For example the routing table of Node A in the above network is

Destination	Next Hop	Number of Hops	Sequence Number	Install Time
A	A	0	A46	001000
B	B	1	B36	001200
C	B	2	C28	001500

Naturally the table contains description of all possible paths reachable by node A, along with the next hop, number of hops, sequence number and install time.

Selection of Route

If a router receives new information, then it uses the latest sequence number. If the sequence number is the same as the one already in the table, the route with the better metric is used. Stale entries are those entries that have not been updated for a while. Such entries as well as the routes using those nodes as next hops are deleted. Then new destination comes. This is how it works.

Influence

Since no formal specification of this algorithm is present, there is no commercial implementation of this algorithm. But some other protocols have used similar techniques. The best-known sequenced distance vector protocol is AODV, which, by virtue of being a reactive protocol, can use simpler sequencing heuristics. Besides, Babel is a distance-vector routing protocol for IPv4 and IPv6 with fast convergence properties. It was designed to make DSDV more robust, more efficient and more widely applicable for both wireless mesh networks and classical wired networks while staying within the framework of proactive protocols [Abohansen2009].

Advantages

DSDV was one of the early algorithms available. It is quite suitable for creating ad hoc networks with small number of nodes.

Disadvantages

DSDV requires a regular update of its routing tables, which uses up battery power and a small amount of bandwidth even when the network is idle. Also, whenever the topology of the network changes, a new sequence number is necessary before the network re-converges; thus, DSDV is not suitable for highly dynamic networks [Wiki2010d][Perkins94].

2.2 Wireless Routing Protocol (WRP)

The Wireless Routing Protocol (WRP) is a proactive unicast routing protocol for MANETs. WRP uses an enhanced version of the distance-vector routing protocol, which uses the Bellman-Ford algorithm to calculate paths. Because of the mobile nature of the nodes within the MANET, the protocol introduces mechanisms which reduce route loops and ensure reliable message exchanges.

The wireless routing protocol (WRP), similar to DSDV, inherits the properties of the distributed Bellman-Ford algorithm. To solve the count-to-infinity problem and to enable faster convergence, it employs a unique method of maintaining information regarding the shortest path to every destination node and the penultimate hop node on the path to every destination node in the network. Since WRP, like DSDV, maintains an up-to-date view of the network, every node has a readily available route to every destination node in the network. It differs from DSDV in table maintenance and in the update procedures. While DSDV maintains only one topology table, WRP uses a set of tables to maintain more accurate information. The tables that are maintained by a node are the following: distance table (DT), routing table (RT), link cost table (LCT), and a message retransmission list (MRL).

Distance Table

The DT contains the network view of the neighbors of a node. It contains a matrix where each element contains the distance and the penultimate node reported by a neighbor for a particular destination.

Routing Table

The RT contains the up-to-date view of the network for all known destinations. It keeps the shortest distance, the predecessor node (penultimate node), the successor node (the next node to reach the destination), and a flag indicating the status of the path. The path status may be a simple path (correct), or a loop (error), or the destination node not marked (null, invalid route). Note, storing the previous and successive nodes assists in detecting loops and avoiding the counting-to-infinity problem - a shortcoming of Distance Vector Routing.

Link Cost Table

The LCT contains the cost (e.g., the number of hops to reach the destination) of relaying messages through each link. The cost of a broken link is infinity. It also contains the number of update periods (intervals between two successive periodic updates) passed since the last successful update was received from that link. This is used to detect link breaks.

The LCT maintains the cost of the link to its nearest neighbors (nodes within direct transmission range), and the number of timeouts since successfully receiving a message from the neighbor. Nodes periodically exchange routing tables with their neighbors via update messages, or whenever the link cost table changes.

Message Retransmission List

The MRL contains an entry for every update message that is to be retransmitted and maintains a counter for each entry. This counter is decremented after every retransmission of an update message. Each update message contains a list of updates. A node also marks each node in the RT that has to acknowledge the update message it transmitted. Once the counter reaches zero, the entries in the update message for which no acknowledgments have been received are to be retransmitted and the update message is deleted. Thus, a node detects a link break by the number of update periods missed since the last successful transmission. After receiving an update message, a node not only updates the distance for transmission neighbors but also checks the other neighbors' distance, hence convergence is much faster than DSDV. The MRL maintains a list of which neighbors are yet to acknowledge an update message, so they can be retransmitted if necessary. If there is no change in the routing table, a node is required to transmit a "hello" message to affirm its connectivity. When an update message is received, a node updates its distance table and reassesses the best route paths. It also carries out a consistency check with its neighbors, to help eliminate loops and speed up convergence.

Advantages

WRP has the same advantage as that of DSDV. In addition, it has faster convergence and involves fewer table updates.

Disadvantages

The complexity of maintenance of multiple tables demands a larger memory and greater processing power from nodes in the wireless ad hoc network. At high mobility, the control overhead involved in updating table entries is almost the same as that of DSDV and hence is not suitable for a highly dynamic and for a very large ad hoc wireless network as it suffers from limited scalability [Wiki2010e][Murthy1996].

2.3 Optimized Link State Routing (OLSR)

The Optimized Link State Routing Protocol (OLSR) is an IP routing protocol optimized for mobile ad-hoc networks, which can also be used on other wireless ad-hoc networks. OLSR is a proactive link-state routing protocol, which uses Hello and Topology Control (TC) messages to discover and then disseminate link state information throughout the mobile ad-hoc network. Individual nodes use this topology information to compute next hop destinations for all nodes in the network using shortest hop forwarding paths.

Features Specific to OLSR

Link-state routing protocols such as OSPF and IS-IS elect a *designated router* on every link to perform flooding of topology information. In wireless ad-hoc networks, there is different notion of a link, packets can go out the same interface; hence, a different approach is needed in order to optimize the flooding process. Using Hello messages the OLSR protocol at each node discovers 2-hop neighbor information and performs a distributed election of a set of *multipoint relays* (MPRs). Nodes select MPRs such that there is a path to each of its 2-hop neighbors via a node selected as an MPR. These MPR nodes then forward TC messages that contain the MPR selectors. This functioning of MPRs makes OLSR unique from other link state routing protocols in a few different ways: The forwarding path for TC messages is not shared among all nodes but varies depending on the source, only a subset of nodes source link state information, not all links of a node are advertised but only those that represent MPR selections.

Since link-state routing requires the topology database to be synchronized across the network, OSPF (Open Shortest Path First) and IS-IS (Intermediate System to Intermediate System) perform topology flooding using a reliable algorithm. Such an algorithm is very difficult to design for ad-hoc wireless networks, so OLSR doesn't bother with reliability; it simply floods topology data often enough to make sure that the database does not remain unsynchronized for extended periods of time.

Messages Used in OLSR

OLSR uses the "Hello" messages to find its one hop neighbors and its two hop neighbors through their responses. The sender can then select its multipoint relays (MPR)

OLSR uses the "Hello" messages to find its one hop neighbors and its two hop neighbors through their responses. The sender can then select its multipoint relays (MPR)

based on the one hop node that offers the best routes to the two hop nodes. Each node has also an MPR selector set, which enumerates nodes that have selected it as an MPR node. OLSR uses Topology Control (TC) messages along with MPR forwarding to disseminate neighbor information throughout the network. Host and Network Association (HNA) messages are used by OLSR to disseminate network route advertisements in the same way TC messages advertise host routes. Below are the formats of Topology and Hello Control messages.

1. Topology Control Message

<table>
<tr><td colspan="4">0 (bits 0-9)</td><td colspan="10">1 (bits 10-19)</td><td colspan="4">2 (bits 20-29)</td><td colspan="2">3</td></tr>
<tr><td>0</td><td>1</td><td>…….</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>…….</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="10">ANSN</td><td colspan="10">Reserved</td></tr>
<tr><td colspan="20">Advertised Neighbor Main Address</td></tr>
<tr><td colspan="20">Advertised Neighbor Main Address</td></tr>
</table>

Note: Each row has 32 bits.

2. Hello Control Message

<table>
<tr><td colspan="4">0 (bits 0-9)</td><td colspan="10">1 (bits 10-19)</td><td colspan="7">2 (bits 20-29)</td><td colspan="2">3</td></tr>
<tr><td>0</td><td>1</td><td>…….</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>….</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="10">Reserved</td><td colspan="8">Htime</td><td colspan="5">Willingness</td></tr>
<tr><td colspan="5">Link Code</td><td colspan="5">Reserved</td><td colspan="13">Link Message Size</td></tr>
<tr><td colspan="23">Neighbor Interface Address</td></tr>
<tr><td colspan="23">Neighbor Interface Address</td></tr>
<tr><td colspan="23">….
….</td></tr>
<tr><td colspan="5">Link Code</td><td colspan="5">Reserved</td><td colspan="13">Link Message Size</td></tr>
<tr><td colspan="23">Neighbor Interface Address</td></tr>
<tr><td colspan="23">Neighbor Interface Address</td></tr>
</table>

Advantages

Being a proactive protocol, routes to all destinations within the network are known and maintained before use. Having the routes available within the standard routing table can be useful for some systems and network applications as there is no route discovery delay associated with finding a new route. The routing overhead generated, while generally greater than that of a reactive protocol, does not increase with the number of routes being used.

Default and network routes can be injected into the system by HNA (Host and Network Association) messages allowing for connection to the internet or other networks within the OLSR MANET cloud. Network routes using reactive protocols do not currently execute well. Timeout values and validity information is contained within the messages conveying information allowing for differing timer values to be used at differing nodes.

Disadvantages

The original definition of OLSR does not include any provisions for sensing of link quality; it simply assumes that a link is up if a number of hello packets have been received recently. This assumes that links are bi-modal (either working or failed), which is not necessarily the case on wireless networks, where links often exhibit intermediate rates of packet loss.

Implementations such as the open source OLSRD (OLSR Daemon, commonly used on Linux-based mesh routers) have been extended (as of v. 0.4.8) with link quality sensing.

Being a proactive protocol, OLSR uses power and network resources in order to propagate data about possibly unused routes. While this is not a problem for wired access points, and laptops, it makes OLSR unsuitable for sensor networks that try to sleep most of the time. For small scale wired access points with low CPU power, the open source OLSRD project showed that large scale mesh networks can run with OLSRD on thousands of nodes with very little CPU power on 200 MHz embedded devices.

Being a link-state protocol, OLSR requires a reasonably large amount of bandwidth and CPU power to compute optimal paths in the network. In the typical networks where OLSR is used (which rarely exceed a few hundreds of nodes), this does not appear to be a problem.

By only using MPRs to flood topology information, OLSR removes some of the redundancy of the flooding process, which may be a problem in networks with moderate to large packet loss rates - however the MPR mechanism is self-pruning (which means that in case of packet losses, some nodes that would not have retransmitted a packet may do so).

OLSR Version 2

OLSRv2 is currently being developed within the IETF. It maintains many of the key features of the original including MPR selection and dissemination. Key differences are the flexibility and modular design using shared components: packet format, and neighborhood discovery protocol (NHDP). These components are being designed to be common among next generation IETF MANET protocols. Differences in the handling of multiple address and interface enabled nodes is also present between OLSR and OLSRv2 [Abohansen2009] [Wiki2010f][Clausen2003].

3. Reactive routing protocols

In bandwidth-starved and power-starved environments, it is interesting to keep the network silent when there is no traffic to be routed. Reactive routing protocols do not maintain routes, but build them on demand. A reactive protocol finds a route on demand by flooding the network with Route Request packets. These protocols have the following advantages:

1. No big overhead for global routing table maintenance as in proactive protocols.
2. Quick reaction for network restructure and node failure.

Even reactive protocols have become the main stream for MANET routing, they still have the following main disadvantages:

1. High latency time in route finding.
2. Excessive flooding can lead to network clogging.

There are many reactive routing protocols for MANET. We only introduce three popular (AODV, DSR and DYMO) and one new (ODCR) protocols in this section [Wiki2010c].

3.1 Ad hoc On-demand Distance Vector (AODV)

Ad hoc On-Demand Distance Vector (AODV) Routing is a routing protocol for mobile ad hoc networks (MANETs) and other wireless ad-hoc networks. It is jointly developed in Nokia Research Center, University of California, Santa Barbara and University of Cincinnati by C. Perkins, E. Belding-Royer and S. Das. AODV is capable of both unicast and multicast routing. It is a reactive routing protocol, meaning that it establishes a route to a destination only on demand. In contrast, the most common routing protocols of the Internet are proactive, meaning they find routing paths independently of the usage of the paths. AODV is, as the name indicates, a distance-vector routing protocol. AODV avoids the *counting-to-infinity* problem of other distance-vector protocols by using sequence numbers on route updates, a technique pioneered by DSDV.

In AODV, the network is silent until a connection is needed. At that point the network node that needs a connection broadcasts a request for connection. Other AODV nodes forward this message, and record the node that they heard it from, creating an explosion of temporary routes back to the needy node. When a node receives such a message and already has a route to the desired node, it sends a message backwards through a temporary route to the requesting node. The needy node then begins using the route that has the least number of hops through other nodes. Unused entries in the routing tables are recycled after a time. When a link fails, a routing error is passed back to a transmitting node, and the process repeats.

Much of the complexity of the protocol is to lower the number of messages to conserve the capacity of the network. For example, each request for a route has a sequence number. Nodes use this sequence number so that they do not repeat route requests that they have already passed on. Another such feature is that the route requests have a "time to live" number that limits how many times they can be retransmitted. The third feature is that if a route request fails, another route request may not be sent until twice as much time has passed as the timeout of the previous route request.

Technical Description

The AODV Routing protocol uses an on-demand approach for finding routes, that is, a route is established only when it is required by a source node for transmitting data packets. It employs destination sequence numbers to identify the most recent path. The major difference between AODV and Dynamic Source Routing (DSR) is that DSR uses source routing in which a data packet carries the complete path to be traversed; however, in AODV, the source node and the intermediate nodes store the next-hop information corresponding to each flow for data packet transmission.

In an on-demand routing protocol, the source node floods the *RouteRequest* packet in the network when a route is not available for the desired destination. It may obtain multiple routes to different destinations from a single *RouteRequest*. The major difference between AODV and other on-demand routing protocols is that it uses a *destination sequence number* (DestSeqNum) to determine an up-to- date path to the destination. A node updates its path information only if the *DestSeqNum* of the current packet received is greater than the last *DestSeqNum* stored at the node.

A RouteRequest carries the source identifier (SrcID), the destination identifier (DestID), the source sequence number (SrcSeqNum), the destination sequence number (DestSeqNum), the broadcast identifier (BcastID), and the *time to live* (TTL) field. DestSeqNum indicates the freshness of the route that is accepted by the source. When an intermediate node receives a RouteRequest, it either forwards it or prepares a RouteReply if it has a valid route to the destination. The validity of a route at the intermediate node is determined by comparing the sequence number at the intermediate node with the destination sequence number in the RouteRequest packet. If a RouteRequest is received multiple times, which is indicated by the BcastID-SrcID pair, the duplicate copies are discarded. All intermediate nodes having valid routes to the destination, or the destination node itself, are allowed to send RouteReply packets to the source.

Every intermediate node, while forwarding a RouteRequest, enters the previous node address and its BcastID. A timer is used to delete this entry in case a RouteReply is not received before the timer expires. This helps in storing an active path at the intermediate node as AODV does not employ source routing of data packets. When a node receives a RouteReply packet, information about the previous node from which the packet was received is also stored in order to forward the data packet to this next node as the next hop toward the destination.

Advantages

The main advantage of this protocol is that routes are established on demand and destination sequence numbers are used to find the latest route to the destination. The connection setup delay is lower. It creates no extra traffic for communication along existing links. Also, distance vector routing is simple, and doesn't require much memory or calculation.

Disadvantages

AODV requires more time to establish a connection, and the initial communication to establish a route is heavier than some other approaches. Also, intermediate nodes can lead to inconsistent routes if the source sequence number is very old and the intermediate nodes have a higher but not the latest destination sequence number, thereby having stale entries. Also multiple RouteReply packets in response to a single RouteRequest packet can lead to heavy control overhead. Another disadvantage of AODV is that the periodic beaconing leads to unnecessary bandwidth consumption [Wiki2010g] [Perkins2003].

3.2 Dynamic Source Routing

Dynamic Source Routing (DSR) is a routing protocol for wireless mesh networks. It is similar to AODV in that it forms a route on-demand when a transmitting computer requests one. However, it uses source routing instead of relying on the routing table at each intermediate device. Many successive refinements have been made to DSR, including DSRFLOW.

Determining source routes requires accumulating the address of each device between the source and destination during route discovery. The accumulated path information is cached by nodes processing the route discovery packets. The learned paths are used to route packets. To accomplish source routing, the routed packets contain the address of each device the packet will traverse. This may result in high overhead for long paths or large addresses, like IPv6 (Internet Protocol version 6). To avoid using source routing, DSR optionally defines a flow id option that allows packets to be forwarded on a hop-by-hop basis.

This protocol is truly based on source routing whereby all the routing information is maintained (continually updated) at mobile nodes. It has only two major phases which are Route Discovery and Route Maintenance. Route Reply would only be generated if the message has reached the intended destination node (route record which is initially contained in Route Request would be inserted into the Route Reply).

To return the Route Reply, the destination node must have a route to the source node. If the route is in the Destination Node's route cache, the route would be used. Otherwise, the node will reverse the route based on the route record in the Route Reply message header (this requires that all links are symmetric). In the event of fatal transmission, the Route Maintenance Phase is initiated whereby the Route Error packets are generated at a node.

The erroneous hop will be removed from the node's route cache, all routes containing the hop are truncated at that point. Again, the Route Discovery Phase is initiated to determine the most viable route.

Dynamic source routing protocol (DSR) is an on-demand protocol designed to restrict the bandwidth consumed by control packets in ad hoc wireless networks by eliminating the periodic table-update messages required in the table-driven approach. The major difference between this and the other on-demand routing protocols is that it is beacon-less and hence does not require periodic hello packet (beacon) transmissions, which are used by a node to inform its neighbors of its presence. The basic approach of this protocol (and all other on-demand routing protocols) during the route construction phase is to establish a route by flooding RouteRequest packets in the network. The destination node, on receiving a RouteRequest packet, responds by sending a RouteReply packet back to the source, which carries the route traversed by the RouteRequest packet received.

Consider a source node that does not have a route to the destination. When it has data packets to be sent to that destination, it initiates a RouteRequest packet. This RouteRequest is flooded throughout the network. Each node, upon receiving a RouteRequest packet, rebroadcasts the packet to its neighbors if it has not forwarded it already, provided that the node is not the destination node and that the packet's *time to live* (TTL) counter has not been exceeded. Each RouteRequest carries a sequence number generated by the source node and the path it has traversed.

A node, upon receiving a RouteRequest packet, checks the sequence number on the packet before forwarding it. The packet is forwarded only if it is not a duplicate RouteRequest. The sequence number on the packet is used to prevent loop formations and to avoid multiple transmissions of the same RouteRequest by an intermediate node that receives it through multiple paths. Thus, all nodes except the destination forward a RouteRequest packet during the route construction phase. A destination node, after receiving the first RouteRequest packet, replies to the source node through the reverse path the RouteRequest packet had traversed.

Nodes can also learn about the neighboring routes traversed by data packets if operated in the promiscuous mode (the mode of operation in which a node can receive the packets that are neither broadcast nor addressed to itself). This route cache is also used during the route construction phase. If an intermediate node receiving a RouteRequest with a route to the destination node in its route cache, then it replies to the source node by sending a RouteReply with the entire route information from the source node to the destination node.

Advantages

This protocol uses a reactive approach which eliminates the need to periodically flood the network with table update messages which are required in a table-driven approach. In a reactive (on-demand) approach such as this, a route is established only when it is required and hence the need to find routes to all other nodes in the network as required by the table-driven approach is eliminated. The intermediate nodes also utilize the route cache information efficiently to reduce the control overhead.

Disadvantages

The disadvantage of this protocol is that the route maintenance mechanism does not locally repair a broken link. Stale route cache information could also result in inconsistencies during the route reconstruction phase. The connection setup delay is higher than in table-driven protocols. Even though the protocol performs well in static and low-mobility environments, the performance degrades rapidly with increasing mobility. Also, considerable routing overhead is involved due to the source-routing mechanism employed in DSR. This routing overhead is directly proportional to the path length [Wiki2010h][Au-Yong2006][Johnson1994][Johnson2001].

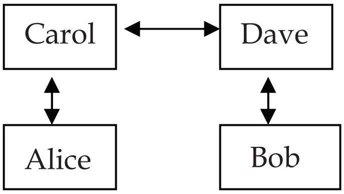
3.3 Dynamic MANET On-Demand Routing (DYMO)

The DYMO routing protocol is a successor to the popular Ad hoc On-Demand Distance Vector (AODV) routing protocol and shares many of its benefits. It is, however, slightly easier to implement and designed with future enhancements in mind. DYMO can work as both a proactive and as a reactive routing protocol, i.e. routes can be discovered just when they are needed. In any way, to discover new routes the following two steps take place:

1. A special "Route Request" (RREQ) messages is broadcast through the MANET. Each RREQ keeps an ordered list of all nodes it passed through, so every host receiving an RREQ message can immediately record a route back to the origin of this message.
2. When an RREQ message arrives at its destination, a "Routing Reply" (RREP) message will immediately get passed back to the origin, indicating that a route to the destination was found. On its way back to the source, an RREP message can simply backtrace the way the RREQ message took and simultaneously allow all hosts it passes to record a complementary route back to where it came from.

So as soon as the RREP message reaches its destination, a two-way route was successfully recorded by all intermediate hosts, and exchange of data packets can commence.

Example



Step 1.

- Alice wants to exchange data with Bob
- Alice does not know a route to Bob yet, so it broadcasts a new RREQ for a route to Bob containing only information about itself.

Step 2.

- Carol receives Alice's RREQ, remembers the contained information about how to reach Alice (directly), then appends information about itself and re-broadcasts the packets.

Step 3.

- Dave receives Carol's RREQ, remembers the contained information about how to reach Carol (directly) and Alice (via Carol), then appends information about itself and re-broadcasts the packet.
- At the same time, Alice also receives Carol's RREQ. Closer examination of the contained information reveals that even the very first information block - how to reach itself, Alice - is of no use. It thus discards the RREQ and does not re- broadcast it as Dave did.

Step 4.

- Bob receives Dave's RREQ and remembers the contained information about how to reach Dave (directly), Carol (via Dave) and Alice (also via Dave). Realizing that he is the target of the RREQ he creates an RREP containing information about itself. He marks the RREP bound for Alice and - knowing that Dave can somehow reach Alice - sends it to Dave.
- Again, at the same time, Carol also receives Dave's RREQ, but - following the same logic as Alice before - ignores it.

Step 5.

- Dave receives the RREP to Alice sent by Bob, remembers the information on how to reach Bob (directly), appends information about itself and - knowing that Alice can be reached via Carol, sends it to Carol.

Step 6.

- Carol receives the RREP to Alice sent by Dave, remembers the contained information on how to reach Dave (directly) and Bob (via Dave), then appends information about itself and - knowing that Alice can be reached directly, sends it to Alice.

Step 7.

- Alice receives the RREP sent to her by Carol and remembers all information on how to reach Carol (directly), Dave (via Carol) and - most importantly - Bob (also via Carol). Now knowing how to reach Bob she can finally send her data packet for him to Carol.

Step 8.

- Carol receives the data packet for Bob from Alice. Because she knows Dave can reach Bob she forwards the packet to him.

Step 9.

- Dave receives the data packet for Bob. Because he knows Bob can be directly reached by him, he forwards the packet to him.

Step 10.

- Bob receives the data packet. Still knowing how to reach Alice, he could now respond with one of his own, and the process repeats until communications are complete or the network changes (e.g. Carol leaves or Eileen joins), where it may be necessary to search the network again for a route [Wiki2010i] [Chakeres2008].

3.4 On-Demand Cache Routing protocol

This protocol presents an efficient algorithm for route discovery, route management and mobility handling for on-demand routing. It is called as "on-demand cache routing" (ODCR) algorithm since it applies caches in each node to improve the routing performance.

In the MANET, each node equips L-1 (level 1 or primary) and L-2 (level 2 or secondary) caches. Usually, the size of L-1 cache is about 64 to 256 KB and L-2 cache is about 256 KB to 2MB). For memory address mapping, they use 2-, 4- or 8-way set associative scheme. Each data entry in a cache is called a "cache line". Most caches use the least-recently-used (LRU) policy for cache line replacement. All cache lines can be searched in parallel in a few processor cycles. This is an important reason why many routing protocols adopted cache for route management. This cache is called as "route cache" because it stores the routing information in the network.

For the initial settings of a MANET, this protocol assumes (1) the communication media among nodes (e.g. laptop computers) is RF; (2) each node has an identification (ID) number; (3) each node keeps an ID list in its own cache (see Figure 1); (4) the wireless links in the network are symmetric (i.e. bi-directional transmission); and (5) the network is scalable and heterogeneous. This means the number of nodes in the network is changeable anytime, and the processor architecture, transmission radius and battery life of each node can be different. In this section, we only present the main algorithm (ODCR). For detail operations of sub-algorithms RDA and MHA mentioned in Algorithm ODCR below, please refer to [Lee2009].

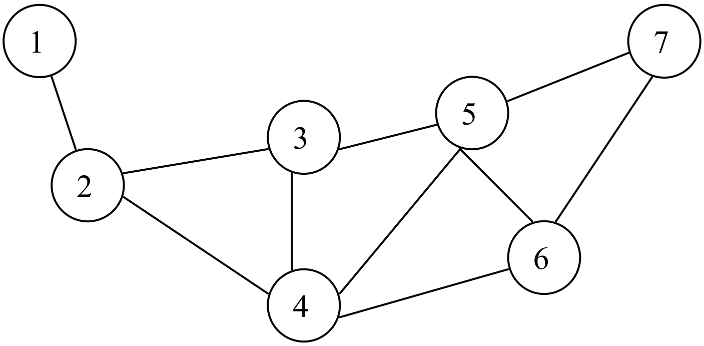


Fig. 1. A simple MANET, where 1, 2, 3, 4, 5, 6 and 7 are node IDs and solid edges are wireless links within the RF transmission radius of each node. For example, node 5 can transmits packets to nodes 3, 4, 6 and 7. In this MANET, each node has an ID list (1, 2, 3, 4, 5, 6, 7).

Algorithm: On-Demand Cache Routing (ODCR)
Inputs: Node identifications (IDs) in the MANET.
Outputs: Transmitted data packets on the network.
Begin

1. If any node in the network wants to send a data packet, at first it has to search the best route (usually the least hop-count route) from its cache. If the route does not exist, go to Step 2. Otherwise (i.e. the route exists) go to Step 3.
2. The source node looks up the destination node in its ID list (as in Figure 1). Then it executes the Route Discovery Algorithm (RDA) to create the best route to its destination node in the network. For instance, the best route from node 1 to node 6 is {1,2,4,6}.
3. The source node attaches its ID, destination node ID and the packet number to eachdata packet, and sends the packet to the destination node along the best route.
4. Each intermediate node uses the best route to the destination node in its cache toforward the data packet to the next or destination node.

5. If any node leaves from, joins to, or moves around the network, it has to execute the Mobility Handling Algorithm (MHA) to notify other nodes about this change and to update their own route information in their caches.
6. Repeat Steps 1 to 5 until the whole network is terminated.

End of On-Demand Cache Routing.

In conclusion, this protocol proposed an efficient on-demand routing algorithm, called ODCR, for route discovery and management, and mobility handling. The ODCR algorithm applied the content-addressable and LRU replacement features in L-1 and L-2 caches for route table creation, updating, and maintenance. The ODCR algorithm with duel-level route caches solved most problems in on-demand routing, such as route tables in "slow" main memory, long connection setup delay, broken link repairing, huge routing overhead for long routes, lengthy data packet in source routing, sending beacons ("hello packets") periodically, control overhead for complicated IDs in data packets, to setup TTL (time-to-live) in a packet or a route path, and to update the stale routes in the route table or cache frequently.

The simulation results showed that the ODCR algorithm outperforms AODV, DSR (Dynamic Source Routing) and CSOR (Cache Scheme in On-Demand Routing) in packet delivery rate, average end-to-end delay and average routing load [Lee2009].

4. Hybrid routing protocols

This type of protocols combines the advantages of proactive and reactive routings. The routing is initially established with some proactively prospected routes and then serves the demand from additionally activated nodes through reactive flooding. The choice for one or the other method requires predetermination for typical cases. The main disadvantages of such algorithms are:

1. Advantage depends on amount of nodes activated.
2. Reaction to traffic demand depends on gradient of traffic volume [Wiki2010j].

4.1 Zone Routing Protocol

Zone Routing Protocol (ZRP) was the first hybrid routing protocol with both a proactive and a reactive routing component. ZRP was first introduced by Haas in 1997. ZRP is proposed to reduce the control overhead of proactive routing protocols and decrease the latency caused by routing discover in reactive routing protocols. ZRP defines a zone around each node consisting of its k-neighborhood (e.g. k=3). That is, in ZRP, all nodes within k-hop distance from node belong to the routing zone of node.

ZRP is formed by two sub-protocols, a proactive routing protocol: Intra-zone Routing Protocol (IARP), is used inside routing zones and a reactive routing protocol: Inter-zone Routing Protocol (IERP), is used between routing zones, respectively. A route to a destination within the local zone can be established from the proactively cached routing table of the source by IARP. Therefore, if the source and destination is in the same zone, the packet can be delivered immediately. Most of the existing proactive routing algorithms can be used as the IARP for ZRP.

For routes beyond the local zone, route discovery happens reactively. The source node sends a route requests to its border nodes, containing its own address, the destination address and a unique sequence number. Border nodes are nodes which are exactly the maximum number of hops to the defined local zone away from the source. The border nodes check their local zone for the destination. If the requested node is not a member of this local zone, the node adds its own address to the route request packet and forwards the packet to its border nodes. If the destination is a member of the local zone of the node, it sends a route reply on the reverse path back to the source. The source node uses the path saved in the route reply packet to send data packets to the destination [Wiki2010k] [Haas2002].

4.2 Order One Network Protocol

The Order One MANET Routing Protocol (OORP) is an algorithm for computer communicating by digital radio in a mesh network to find each other, and send messages to each other along a reasonably efficient path. It was designed for, and promoted as working with wireless mesh networks. OORP can handle hundreds of nodes, where most other protocols handle less than a hundred. OORP uses hierarchical algorithms to minimize the total amount of transmissions needed for routing. Routing overhead is only about 1% to 5% of node to node bandwidth in any network and does not grow as the network size grows.

The basic idea is that a network organizes itself into a tree. Nodes meet at the root of the tree to establish an initial route. The route then moves away from the root by cutting corners, as ant-trails do. When there are no more corners to cut, a nearly optimum route exists. This route is continuously maintained. Each process can be performed with localized minimal communication, and very small router tables. OORP requires about 200K of memory. A simulated network with 500 nodes transmitting at 200 bytes/second organized itself in about 20 seconds. As of 2004, OORP was patented or had other significant intellectual property restrictions.

Assumptions

Each computer or "node" of the network has a unique name. At least one network link and a computer with some capacity hold a list of neighbors.

Organizing a Tree

The network nodes form a hierarchy by having each node select a parent. The parent is a neighbor node that is the next best step to the most other nodes. This method creates a hierarchy around nodes that are more likely to be present, and which have more capacity, and which are closer to the topological center of the network. The memory limitations of a small node are reflected in its small routing table, which automatically prevents it from being a preferred central node. At the top, one or two nodes are unable to find nodes better-connected than themselves, and therefore become parents of the entire network. The hierarchy-formation algorithm does not need a complex routing algorithm or large amounts of communication.

Routing

All nodes push a route to themselves to the root of the tree. A node wanting a connection can therefore push a request to the root of the tree, and always find a route. The commercial protocol uses Dijkstra's algorithm to continuously optimize and maintain the route. As the network moves and changes, the path is continually adjusted.

Advantages

Assuming that some nodes in the network have enough memory to know of all nodes in the network, there is no practical limitation to network size. Since the control bandwidth is defined to be less than 5% regardless of network size, the amount of control bandwidth required is not supposed to increase as network size grows. The system can use nodes with small amounts of memory.

The network has a reliable, low-overhead way to establish that a node is not in the network. This is a valuable property in ad-hoc mesh networks. Most routing protocols scale either by reducing proactive link-state routing information or reactively driving routing by connection requests. OORP mixes the proactive and reactive methods. Properly configured, an OORP net can theoretically scale to 100,000's of nodes and can often achieve reasonable performance even though it limits routing bandwidth to 5%.

Disadvantages

Central nodes have an extra burden because they need to have enough memory to store information about all nodes in the network. At some number of nodes, the network will therefore cease to scale. If all the nodes in the network are low capacity nodes the network may be overwhelmed with change. This may limit the maximum scale. However, in real world networks, the farther away from the edge nodes the more the bandwidth grows.

These critiques may have no practical effect. For example, consider a low bandwidth 9.6Kbit/second radio. If the protocol was configured to send one packet of 180 bytes every 5 seconds, it would consume 3% of overall network bandwidth. This one packet can contain up to 80 route updates. Thus even in very low bandwidth network the protocol is still able to spread a lot of route information. Given a 10Mbit connection, 3% of the bandwidth is 4,100 to 16,000 route updates per second. Since the protocol only sends route updates for changes, it is rarely overwhelmed.

The other disadvantage is that public proposals for OORP do not include security or authentication. Security and authentication may provided by the integrator of the protocol. Typical security measures include encryption or signing the protocol packets and incrementing counters to prevent replay attacks [Wiki2010l][Orderone2010].

4.3 Global On-Demand Routing protocol

The Global On-Demand Routing (GOR) is a clever hybrid routing protocol for the MANET. To simplify simulations in GOR, it assumes (1) all nodes are homogeneous; (2) the transmission range of each node is k; and (3) each node has an ID and a pair of positive x and y coordinates to represent its location in the network. The main algorithm for the GOR protocol is described below. For detail operations of sub-algorithms DFA and NRA in GOR protocol, please refer to [Lee2007].

Algorithm GOR Protocol

Inputs: The ID and (x, y) coordinates of each node.

Outputs: Destination nodes receive data packets from sources nodes.

Begin

1. Select a center or near-center node in the initial network as the root node (RN).
2. The RN runs the Double-Flooding Algorithm (DFA) to create the location table (LT), sorts the LT by IDs in ascending order, and broadcasts the LT to each node in the network.

3. Each node uses the LT to generate its own distance table (DT) concurrently. Then, each node marks any distance that is longer than the transmission range k in the DT as "∞" (infinity).
4. Each node calls the Dijkstra's Algorithm to generate the one-to-all shortest-path table (SPT) concurrently (see Figure 2 below).
5. If a new node joined to the network, an existing node moved out of the transmission range of its any neighbor nodes, or an existing node left from the network, then it calls the Node-Reorganization Algorithm (NRA) to ask other nodes to update (or mark as "new" nodes if any) their own LT for these changes consequently.
6. If any node wants to send packets via or to the above joined or moved nodes, it has to (1) use the updated LT in Step 5 to update its DT (or mark the "∞" distances if any); (2) run the Dijkstra's algorithm again to update its SPT; (3) reset all nodes in the LT to "old" nodes; and (4) follows the paths in the new SPT to send packets to its destination nodes.
7. If network topology changed again, repeat steps 5 and 6 until the whole network dismissed.

End of GOR Protocol.

Figure 2 below shows some shortest paths within the transmission range k for node 1. In this figure, the shortest path between nodes 1 and 6 is (1, 3, 6) not (1, 6) because node 6 locates outside the circular transmission range k of node 1. Note we have marked all "∞" distances in steps 3 and 6 respectively in the main algorithm (Algorithm GOR Protocol).

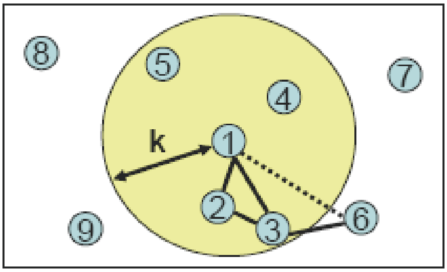


Fig. 2. Sample shortest paths in a MANET.

This algorithm proposed a hybrid global on-demand routing (called GOR) protocol for mobile ad hoc networks. This protocol does not update the routing tables immediately if any node changed its status in the network, such as movement, addition or deletion. Instead, it only handles a node whose move changed the MANET topology or whose move distance is greater than the transmission range k. This critical strategy prevents other nodes from updating the routing tables frequently and hence reducing unnecessary computation and node-reorganization overheads dramatically.

The GOR protocol not only keeps the advantages of proactive and reactive protocols, but also improves the sub-optimal routing overhead and memory consuming problems in local hybrid protocols. Because this protocol retains high packet delivery rate and low end-to-end delay as the DSDV and WRP protocols, and low routing load as the AODV and DSR protocols [Lee2007].

5. MANET routing protocols for IPv6

It is possible that all the IP version 4 (IPv4) addresses will be allocated in next decade. The transition from IP version 4 to IP version 6 (IPv6) will become an important issue in computer networks and Internet in recent years. Therefore, in this section, we introduce IPv6, mobile IPv6, and two popular MANET routing protocols, OLSR and AODV, for IPv6 networks.

5.1 Introduction to IPv6 and mobile IPv6

Internet is built upon a protocol suite called TCP/IP. This abbreviation stands for Transmission Control Protocol, and Internet Protocol. When your computer communicates with the Internet a unique IP address is used to transfer and receive information. Yesterdays IP standard is called IPv4. Each IPv4 address contains 32 binary bits. That is the total address in IPv4 is 2^32 only. Sadly most ISPs and services still only deliver this ancient technology standardized in September 1981. So far, most of IPv4 addresses are already tied up and the Internet is simply running out of IPs. The address shortage problem is aggravated by the fact that portions of the IP address space have not been efficiently allocated.

IPv6 (Internet Protocol version 6) gives citizens the opportunity to become real Internet participants. IPv4 makes citizens into passive consumers who are only able to connect to compartmentalized networks run by companies or governments. This is why the establishment does not want IPv6. Each IPv6 address contains 128 binary bits. This means there are 2^128 unique addresses in IPv6. This huge amount of IP addresses may be able to serve the Internet till the end of this century [Linux2010a].

Mobile IPv6 is the implementation of the IP mobility (Mobile IP) methods and protocols on an Internet Protocol version 6 (IPv6) network. Like its IPv4 counterpart, it is designed to permit IP devices to roam between different networks without losing IP connectivity by maintaining a permanent Internet Protocol (IP) address. Mobile IPv6 is described in RFC3775.

The key benefit of Mobile IPv6 is that even though the mobile node changes locations and addresses, the existing connections through which the mobile node is communicating are maintained. To accomplish this, connections to mobile nodes are made with a specific address that is always assigned to the mobile node, and through which the mobile node is always reachable. Mobile IPv6 provides Transport layer connection survivability when a node moves from one link to another by performing address maintenance for mobile nodes at the Internet layer [Wiki2010m].

5.2 OLSR for IPv6 networks

In this section, we summarize the proposed issues and necessary changes to adapt OLSR to IPv6 from the paper “OLSR for IPv6 Networks” by Laouiti, etc [Laouiti2004]. In order to present a complete IPv6 solution for OLSR, there are several issues to address:

1. Addressing: IPv6 introduce several changes, some more conceptual than others. Changes include the diffusion of data packets and existing multiple addresses of Interfaces.

2. Protocol changes: The OLSR specification gives the protocol format message for IPv4 packets, but some additional changes are proposed.
3. Neighbor discovery: It is described how the neighbor discovery mechanism of IPv6 still operates properly.
4. Autoconfiguration: It is loosely related to addressing, the ability for an IPv6 node to self-configure its addressed yields numerous challenges and had been the subject of elaborate research as seen previously.

IPv6 Ad Hoc Addressing Issues

Several changes are required due to various novelties introduced by IPv6 itself.

1. Interface Addresses: The chosen solution in this paper is to consider a MANET as a single site-local network, and to use site-local prefix with a fixed 16 bits subnet called OLSR_SUBNET. Then, an OLSR node will perform link-local address autoconfiguration, and upon success, will automatically configure for each of its OLSR interfaces. The site-local address with that subnet (FEC0:0:0:OLSR_ SUBNET::/64) will run the OLSR protocol using it.
2. OSLR Diffusion Addresses: In order to reach all the nodes present on the link to get the same effect as in IPv4, this paper proposed that a multicast address ALL_OLSR_NODES is used for the destination address. The ALL_OLSR_NODES could be taken as ALL_LINK_NODES (FF01::1). Also since a node has several interface addresses, the paper proposed that the site-local addresses are used as source addresses.

Diffusing Non-OLSR Packets

Since MANETs are multi-hop routing networks, in order to flood packets to all nodes, retransmissions are usually needed. With OLSR, packets are retransmitted hop by hop to the direct neighborhood by using MPRs (multipoint relays). In the other hand, for any applications, a direct multicast on the local "link" is performed and such packets are never routed. For instance, it is also in the case for most of IPv6 messages for neighbor discovery and autoconfiguration. This relies on the assumption that being on the same network is equivalent to being on same link, an assumption which doesn't hold in MANET networks. As a result, in a multi-hop network, by default, this kind of messages will not be delivered to all nodes. This paper proposed two solutions to diffuse non-OLSR packets to all nodes:

1. Encapsulate the packets in specific OLSR messages, and use the MPR flooding.
2. Use of a new multicast address called ALL-MANET_NODES, instead of the ALL_LINK_NODES.

Changes to the OLSR Routing Protocol

1. OLSR Packet format: The essential change needed for the existing OLSR packet format is to replace the IPv4 addresses with the IPv6 addresses in all messages, as highlighted in the OLSR specification [Clausen2003].
2. Multiple Interface Addresses: In IPv6, an interface can have several addresses. This paper proposed an OLSR node, for each interface, will have:
   - A link-local address: This address is usually obtained by autoconfiguration. It is temporary used as the source address for OLSR packets before autoconfiguration is completed.

- A site-local address: This is derived from the link-local address, in the fixed subnet OLSR_SUBNET for site-local prefix. This address is permanently used as the source for all OLSR packets, once autoconfiguration is completed.
- Any number (possibly zero) of additional global or site local unicast addresses, which are automatically or manually configured.

Neighbor Discovery

In IPv6, nodes (hosts and routers) use Neighbor Discovery [Narten1998] to determine the MAC addresses for neighbors on attached links and to quickly purge invalid cache values. Hosts also use Neighbor Discovery to find neighboring routers that are willing to forward packets on their behalf. Finally, nodes use the protocol to actively keep track of which neighbors are reachable and which are not, and to detect changed MAC addresses.

Routing table in the OLSR indicates the next hop for each reachable destination in the network. This next hop is one of the direct neighbors. This means that the neighbor solicitation for address resolution will work without any modification. In OLSR, gateways declare themselves to the entire network periodically. The neighbor discovery is adapted to OLSR. Consequently it is not necessary to do any modification to the classical procedure.

Autoconfiguration

IPv6 Stateless Address Autoconfiguration is based on several steps: after the creation of a link local address, the node must check whether the address is already in use by another interface of another node, somewhere in the network. In wired network, this means that all the links of the attached interfaces of the node are probed. If the address is not unique the process is interrupted, otherwise the autoconfiguration was successful and the address may be safely used.

In a MANET, the nodes on the links of the attached interfaces would include only the nodes with an interface within radio reach of the transmitter and not all the participating nodes. Hence, the uniqueness of the address is not guaranteed if the classical DAD (Duplicate Address Detection) procedure is applied. This paper proposed an algorithm, following the philosophy of the IPv6 DAD, to perform autoconfiguration in an OLSR network. The algorithm includes reactive probing (i.e. sending a request to the whole network and waiting for a possible answer), proactive checking (i.e. checking periodically for duplicate addresses) and collision resolution (i.e. what should be done upon detection of duplicate addresses) [Laouiti2004][Linux2010b].

5.3 Ad hoc On-demand Distance Vector routing for IPv6 (AODV6)

The operation of AODV for IPv6 is intended to mirror the operation of AODV for IPv4, with changes necessary to allow for transmission of 128-bit addresses in IPv6 instead of the traditional 32-bit addresses in IPv4.

Route Request (RREQ) Message Format

The format of the IPv6 Route Request message (RREQ) contains the same fields with the same functions as the RREQ message defined for IP version 4, except as follows:

1. Destination IP Address: The 128-bit IPv6 address of destination for which a route is desired.

2. Source IP Address: The 128-bit IPv6 address of the node which originated the Route Request.

Note, the order of the fields has been changed to enable alignment along the 128-bit boundaries.

Route Reply (RREP) Message Format

The format of the IPv6 Route Reply message (RREP) contains the same fields with the same functions as the RREP message defined for IP version 4, except as follows:

1. Prefix Size: The Prefix Size is 7 bits instead of 5, to account for the 128-bit IPv6 address space.
2. Destination Sequence Number: The destination sequence number associated to the route.
3. Destination IP Address: The 128-bit IP address of the destination for which a route is supplied.
4. Source IP Address: The 128-bit IP address of the source node which issued the RREQ for which the route is supplied.

Note, the order of the fields has been changed for better alignment.

Route Error Message Format

The format of the Route Error (RERR) message is identical to the format for the IPv4 RERR message except that the IP addresses are 128 bits, not 32 bits.

Route Reply Acknowledgment (RREP-ACK) Message Format

The RREP-ACK message is used to acknowledge receipt of an RREP message. It is used in cases where the link over which the RREP message is sent may be unreliable. It is identical in format to the RREP-ACK message for IPv4.

AODV for IPv6 Operation

The handling of AODV for IPv6 messages analogous to the operation of AODV for IPv4, except that the RREQ, RREP, RERR, and RREP-ACK messages described above are to be used instead; these messages have the formats appropriate for use with 128-bit IPv6 addresses [Perkins2000].

6. Conclusion

In this chapter, we introduced the general concepts of mobile ad hoc networks (MANET), routing in a MANET, and routing protocols for MANETs. For routing protocols, we summarized the key concepts of some popular proactive, reactive and hybrid protocols. We also introduced two popular MANET routing protocols for IPv6 networks, because more and more networks will adopt IPv6 addresses in the near future.

Each protocol introduced in this chapter has its own advantage and disadvantages in different MANET settings or environments. Therefore, it is hard to say which one is the best among them. So far, AODV is the most popular one for both IPv4 and IPv6 networks because it has more advantages than other protocols and it has been implemented successfully. In fact, the ODCR or the GOR algorithm could be a better choice.

7. References

[Abohasan2009] Abolhasan, Hagelstein and Wang, "Real-world Performance of Current Proactive Multi- hop Mesh Protocols", *Proceedings of IEEE APCC2009*, Shanghai, China, 10/2009.

[Au-Yong2006] Au-Yong, "Comparison of On-Demand Mobile Ad Hoc Network Routing Protocols under On/Off Source Traffic Effect", *Proceedings of NCS2006*, Chiang-Mai, Thailand, 3/2006.

[Chakeres2008] Chakeres and Perkins, "Dynamic MANET On-demand (DYMO) Routing", in: *Mobile Ad Hoc Networks Working Groups (draft-ietf-manet-dymo-12)*, available from: http://ianchak.com/dymo/draft-ietf-manet-dymo-12.txt, 2/2008.

[Clausen2003] Clausen and Jacquet, "Optimized Link State Routing Protocols (OLSR)", in: *Network Working Group – Request for Comments 3626*, available from: http://tools.ietf.org/html/rfc3626, 10/2003.

[Haas2002] Haas, Pearlman and Samar, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks" in: *Internet Draft (draft-ietf-manet-zone-zrp-04.txt)*, available from: http://people.ece.cornell.edu/~haas/wnl/Publications/draft-ietf-manet-zone-zrp-04.Txt, 7/2002.

[Johnson] Johnson, Maltz and Broch, "DSR: The Dynamic Source Routing Protocol for Multi-HopWireless Ad Hoc networks", in: *Ad Hoc Networking*, publisher: Edison Wesley, 2001.

[Johnson1994] David Johnson, "Routing in Ad Hoc Networks for Mobile Hosts", *Proceedings of IEEE WMCSA1994*, Santa Cruz, CA, 12/1994.

[Laouiti2004] Laouiti, Boudjit, Minet and Adjih, "OLSR for IPv6 Networks", *Proceedings of Med-Hoc-Net-2004*, available from
http://www2.ece.ohio-state.edu/ medhoc04, 7/2004.

[Lee2007] Lee, Kimm and Reinhart, "A Global On-Demand Routing Protocol for Mobile Ad Hoc Networks", *Proceedings of IEEE NCA2007*, Boston, MA, 7/2007.

[Lee2009] Lee, Swanson and Liu, "An Efficient On-Demand Cache Routing Algorithm for Mobile Ad Hoc Networks", *Proceedings of IEEE ICCSIT2009*, Bejing, China, 8/2009.

[Linux2010a] Linux Reviews, "Why You Want IPv6 (Background – The IP Shortage)", available from: http://en.linuxreviews.org/Why_you_want_IPv6#Background:_ The_IP_shortage, 8/2010.

[Linux2010b] Linux Reviews, "Linux Optimized Link State Routing Protocol (OLSR) IPv6 HOWTO", available from: http://linuxreviews.org/howtos/networking/ OLSR-IPv6-HOWTO/en/index.html, 8/2010.

[Murthy1996] Murthy and Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks", *Mobile Networks and Applications*, Volume 1, Issue 2, pp183-197, 1996.

[Narten1998] Narten, Noedmark and Simpson, , "Neighbor Discovery for IP Version 6 (IPv6)", in: *Network Working Group – Request for Comments 2461*, 12/1998, available from:
http://tools.ietf.org/html/rfc2461

[Orderone2010] Orderone Networks 2010, "Mesh Network Routing Protocol", available from:
http://www.orderonenetworks.com/

[Perkins1994] Perkins and Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", *Proceedings of ACM SIGCOMM94*, pp234-244, London, 8/1994.

[Perkins2000] Perkins, Royer and Das, "Ad hoc On-Demand Distance Vector (AODV) Routing for IP version 6", in: *Mobile Ad Hoc Networking Working Group, Internet Draft*, available from:
http://members.shaw.ca/aodv6-sfu/aodv-ipv6-ietf-1.txt, 11/2000.

[Perkins2003] Perkins, Belding-Royer and Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", in: *Network Working Group – Request for Comments 3561*, available from:
http://tools.ietf.org/html/rfc3561, 7/2003.

[Sholander2002] Sholander, Yankopolus, Coccoli and Tabrizi, "Experimental Comparison of Hybrid and Proactive MANET Routing Protocols", *Proceedings of MILCOM2002*, Anaheim, CA, 10/2002.

[Wiki2010a] Wikipedia, "Mobile Ad Hoc Networks", available from: http://en.wikipedia.org/wiki/Mobile_ad_hoc_network, 8/2010.

[wiki2010b] Wikipedia, "Routing", available from: http://en.wikipedia.org/wiki/ Routing, 8/2010.

[Wiki2010c] Wikipedia, "List of Ad Hoc Routing Protocols", available from: http://en.wikipedia.org/wiki/List_of_ad-hoc_routing_protocols, 8/2009.

[Wiki2010d] Wikipedia, "Destination-Sequenced Distance-Vector Routing" available from: http://en.wikipedia.org/wiki/DSDV, 8/2010.

[Wiki2010e] Wikipedia, "Wireless Routing Protocols", available from: http://en.wikipedia.org/wiki/Wireless_Routing_Protocol, 8/2010.

[Wiki2010f] Wikipedia, "Optimized Link State Routing Protocols", available from: http://en.wikipedia.org/wiki/Optimized_Link_State_Routing_Protocol, 8/2010.

[Wiki2010g] Wikipedia, "Ad Hoc On-Demand Distance Vector Routing", available from: http://en.wikipedia.org/wiki/Ad-hoc_On-demand_Distance_Vector, 8/2010.

[Wiki2010h] Wikipedia, "Dynamic Source Routing", available from: http://en.wikipedia.org/wiki/Dynamic_Source_Routing, 8/2010.

[Wiki2010i] Wikipedia, "DYMO", available from: http://en.wikipedia.org/wiki/ DYMO, 8/2010.

[Wiki2010j] Wikipedia, "Hybrid (both pro-active and reactive) Routing:, available from: http://en.wikipedia.org/wiki/List_of_ad-hoc_routing_protocols#Hybrid_.28both_ pro- active_and_reactive.29_routing, 8/2010.

[Wiki2010k] Wikipedia, "Zone Routing Protocol", available from: http://en.wikipedia.org/wiki/Zone_Routing_Protocol, 8/2010.

[Wiki2010l] Wikipedia, "Order One Network Protocol", available from: http://en.wikipedia.org/wiki/Order_One_Network_Protocol, 8/2010.

[Wiki2010m] Wikipedia, “Mobile IP”, available from: http://en.wikipedia.org/wiki/Mobile_IPv6, 8/2010.

17

Fault-Tolerant Routing in Mobile *Ad Hoc* Networks

B. John Oommen[1,2] and Luis Rueda[3]
[1]*School of Computer Science, Carleton University, Ottawa;*
[2]*University of Agder, in Grimstad,*
[3]*School of Computer Science, University of Windsor,*
401 Sunset Avenue, Windsor, Ontario, N9B 3P4,
[1,3]*Canada*
[2]*Norway*

1. Introduction

Mobile *Ad Hoc* Networks (MANETs) are characterized by the cooperative engagement of mobile nodes that constitute networks possessing continuously-changing infrastructures, the absence of centralized network managers, access points, fixed base stations, a backbone network for controlling the network management functions, and the absence of designated routers for making routing decisions. All the nodes in MANETs participate in the routing process by acting as routers for one another. However, for the transmission of data from one node to another, such networks normally require several hops because of the limited wireless transmission range associated with the operation of the mobile nodes [2,7,9].

The above-mentioned characteristics of MANETs, particularly those arising due to the mobility of the nodes, and the continuously-changing network infrastructure, pose several challenges. Due to the continuously changing infrastructure, the routes that were once considered to be the "best" may no longer remain as the "best" at a later time instant. Therefore, one needs to continuously re-compute the routes, implying that in such networks, there is no permanent convergence to a fixed set of routes. Thus, any routing protocol that needs to operate in MANET network environments should take these issues into consideration [2].

Designing routing protocols poses further challenges when one needs to design routing schemes in the presence of adversarial environments in MANET networks. This is the primary focus of this chapter. More specifically, we discuss fault-tolerant routing schemes when the network contains malfunctioning nodes. To motivate this, we observe that most existing MANET protocols were postulated considering scenarios in which all the mobile nodes in the *ad hoc* network function properly and in an idealistic manner. However, adversarial environments are common in MANET environments, and misbehaving nodes degrade the performance of these routing protocols [11]. The need for fault-tolerant routing protocols was identified to address routing in adversarial environments in the presence of faulty nodes by exploring redundancies in the networks [10,11].

Despite the challenges that we mention above, it is worthwhile to note a few applications of MANETs which have made them popular. One of the popular application domains of

MANETs is communications in moving battlefields [7]. Other applications may be found in rural regions where building up fixed wired or wireless infrastructures can be costly and/or difficult.

Although our primary discussion centers around fault-tolerant routing in MANETs, since this chapter is intended to be of a survey nature, we shall first briefly include an overview of the field and the corresponding routing protocols.

2. Routing protocols for MANETs

Routing in MANETs is currently a challenging and interesting problem studied by the community primarily due to the dynamic nature of the infrastructure present in MANETs, e.g., due to nodes joining and leaving the network. For routing, the transmission of data from one node to another is *direct,* if the source and destination nodes are neighbors, i.e., if they are within the wireless range of each other. On the other hand, the transmission is *indirect*, if the source and destination nodes are not within their range of operation [7]. In such a case, routing is achieved through a series of multiple hops, with intermediate nodes between the source and the destination nodes serving the purpose of routers for relaying the information in between. The dynamic nature of the topology of MANETs due to the constant migration of nodes renders routing considerations difficult. The following characteristics of MANETs make their routing further challenging [7]:

1. The terrain in which the mobile nodes operate in MANETs may pose to be hostile with hazardous conditions that can lead to the frequent failure of the nodes and their mutual links.
2. The medium of transmission of information in MANETs is wireless. Wireless media are relatively unreliable, insecure, and quite susceptible to different kinds of errors and unwanted noise.
3. MANETs operate with battery-powered nodes, which are normally low powered, and resource constrained. If the region of operation of the nodes is in a hostile terrain, the frequent recharging of the nodes may not always be feasible. Consequently, all routing algorithms should be energy-efficient, of low complexity, and should be capable of operating under limited bandwidth.

The different types of errors that can occur in MANETs are the following [7]:

1. Transmission errors
2. Node failures
3. Link failures
4. Route breakages
5. Packet loss due to congested nodes/links.

The currently available MANET routing protocols can be classified into two categories [7]: (i) Unipath routing protocols, and (ii) Multipath routing protocols, explained below.

2.1 Unipath routing protocols

In unipath routing protocols, the transmission of messages between a source-destination pair of nodes takes places through a unique path. All the unipath routing protocols may be classified to be either table-based or on-demand [7]. Table-based protocols are characterized by their ability to maintain routing tables that store information about routes from one node in the network to the others. Obviously, this requires that the nodes in the network maintain the table up-to-date by exchanging routing information between the participating nodes. Although, in general, table-based protocols may be easy to implement, the major limitation associated with these protocols is that due to the highly-mobile and dynamic nature of *ad hoc* networks, maintaining the routing information in these tables is a very challenging task [7].

On-demand routing protocols, on the other hand, alleviate the above problems, and make routing more scalable to highly dynamic and large networks. As the name suggests, on-demand routing protocols are characterized by the computation of routes on an "as-required" basis. In on-demand routing protocols, there is initially a *route discovery* phase in which a route is found between two nodes. The *route discovery* phase is normally followed by a *route maintenance* phase in which a broken link in a route is repaired, or a new route is found [7,9].

Various unipath routing protocols have been proposed in the literature (e.g., [5,9]). Of these, the *Ad Hoc On-Demand Distance Vector* (*AODV*) routing protocol [9], and the *Dynamic Source Routing* (*DSR*) protocol [5] are the most popular ones. In the interest of completeness, we briefly discuss these protocols below, with sufficient details so as to introduce the context for the fault-tolerant routing problem discussed later in this chapter.

2.1.1 The AODV routing protocol

As the name suggests, AODV is classified as a unipath on-demand distance vector routing protocol. It, therefore, functions by using both a *route discovery* phase and a *route maintenance* phase by incorporating multihop routing in the intermediate nodes between the source and destination. In the AODV, every mobile node functions as a specialized router. Routing tables are maintained in the intermediate nodes, with routing information being obtained on an "as-required" basis with no (or little) assumption on the presence of periodic advertisements by the nodes [7,9]. The AODV has been shown to be scalable with the increase in the number of mobile nodes in a MANET. It is characterized by its ability to provide loop-free route information in which broken links are resolved by repairing existing links or introducing new ones. Since there is no assumption on the presence of periodic advertisements by the nodes, there is little requirement on the amount of bandwidth that should be available to the mobile nodes as compared to protocols that require the presence of advertisements. Finally, it is worth mentioning that the AODV works under the assumption that the links are symmetric, and that the communication can be synchronous, implying that both nodes on either side of a link are capable of talking to each other [9].

Perkins and Royer [9] observed that, normally, there are nodes and paths in a network that are not frequently active. Not only do those nodes seldom maintain any routing information, but rather, they also seldom participate in the periodic advertisements of routing information. Furthermore, one should observe that two nodes need to share routing information only when they need to communicate with each other, or whenever one of them is acting as an intermediate node to relay information destined to reach *another* node in the network. Determining the local connectivity between the mobile nodes can be achieved in a number of ways. One of the most common of these is by transmitting local, and not system-wide, so-called "*Hello*" messages. This will assist the routing tables maintained by the nodes in the neighborhood to be updated quickly, and the response time to be optimized for local movements, thereby providing fast responses to establish new routes.

AODV has primarily two phases of operation: (1) the *route discovery* phase, and (2) the *route maintenance* phase [9]. When one node needs to communicate with another node for which there is no routing information in *its* table, the *route discovery* phase is triggered. The source specifies the destination node to which information needs to be transmitted, and floods the network with a so-called *Route Request* (RREQ) packet. The latter contains the information about the source address, the source sequence number, the broadcast identification number (which is incremented every time the source node starts a new *route discovery* request), the destination address, the destination sequence number, and the hop count. Any of the nodes that receives the request checks to see if it is identified as the destination node by the RREQ packet, or if it can serve as an intermediate node to transmit information to another node in the network. If that is the case, that node generates a unicast *Route Reply Packet* (RREP) that is sent back along the reverse path in which the RREQ packet was originally sent by the source node. Once the source receives the RREP packet, it then knows where and how to transmit the packet. If none of the above cases hold true, i.e., the node that received the packet is neither the destination node, nor can it serve as an intermediate node to the destination node, it broadcasts the RREQ packet again. Obviously, by doing so, multiple copies of a RREQ packet may be received by the nodes in the network, and any such superfluous multiple copies are discarded [7,9].

The *route maintenance* phase is triggered whenever a broken link is detected by any node, and when that node attempts to forward a packet to the next hop. In the *route maintenance* phase, once the next hop is found to be unreachable, the upstream node sends an unsolicited RREP packet possessing a new sequence number that is greater than the previously-known sequence number by unity. It also sends a hop count of "∞" to all the neighboring upstream nodes, which, in turn, replay that information to their active neighbors, until all active source nodes are notified [9].

Once the notification of a broken link is received, the source node could initiate a so-called *discovery process*. The latter is initiated only by that node which determines that there is a need for the identification of a route to the destination node. The source node then makes a decision about whether or not it wants to rebuild an alternative route to the destination node (by virtue of the broken link). If it does, a RREQ packet is sent out with a destination sequence number that is greater than the previously-known sequence number by unity [7,9].

To summarize, the AODV scheme sends broadcast discovery messages only when required, distinguishes between neighborhood detection and general topology maintenance, and selectively disseminates information about changes to local connectivity only to those nodes that might need the topology/connectivity change information [9].

2.1.2 The DSR protocol

Like the AODV, the DSR is a unicast dynamic on-demand routing protocol. It is a source routing protocol, where the source explicitly provides a packet with the complete information of the route to follow, which is subsequently used by the intermediate nodes to forward the packet to the correct destination node [7].

The DSR only routes packets between hosts that want to communicate with one another. Like the AODV, the DSR also has a *route discovery* phase and a *route maintenance* phase. When two nodes need to communicate with each other, the sender node determines a route. This is done based on the information stored in its cache, or based on the results of a route discovery phase, depending on whether or not the information about the destination node is already available to the source node [5].

In all brevity, the transmission of a packet from a source node to a destination node obeys the following mechanism. The DSR requires that the sender determines and stores in the packet's header the *source route*, where the address of each host in the network is explicitly provided until it can reach the intended destination node. The source finds out the complete route to the destination from a *route cache* that stores the routing information to different nodes in the network. If such an entry is found, the sender uses this route to send the packet. On the other hand, if such an entry is not found, a *route discovery* exercise, similar to the one discussed for the AODV protocol is initiated by the source route. After the next destination is successfully identified, the packet is then sent to the first hop in the identified sequence of nodes by the source. The first hop node first determines whether it is the final destination. If it is, the packet is considered to be delivered. If not, the next hop is scanned from the sequence of identified nodes to the destination, and the packet is forwarded to the next identified hop. The process continues until the packet is considered to be delivered [5].

As in the AODV, a *route maintenance* exercise may be initiated whenever a broken link is detected. This is a scenario that could occur because any of the nodes along a route fails or is powered down. In such a case, an error message is relayed back to the source node with the information associated with the particular link that failed. Each of the intermediate nodes (including the source node) that receives *this* error message deletes all the routes containing that link from its route cache. A *route discovery* phase may then be initiated subsequently to find new routes [5,7].

The DSR is characterized by its ability to quickly adapt itself to routing changes in environments in which there are frequent and rapidly-occurring host movements. One of the important aspects of the DSR is that there is no requirement for periodic route advertisements, as is frequently required in many routing protocols. This reduces the overall overhead on the network bandwidth, especially because most mobile nodes in *ad hoc* networks are operated over battery power, and there are often situations in *such* networks when there are no periodic routing advertisements taking place [5]. The DSR has hence become popular as a suitable protocol for *ad hoc* networks.

2.2 Multipath routing protocols

Multipath routing protocols proposed in the literature (see, for example, [6,8,16]) are of different types, some of which are based on the foundational principles behind the AODV and DSR protocols. However, all multipath routing protocols share a common characteristic, i.e., they discover *multiple* routes between a pair of source-destination nodes. Multipath routing protocols take advantage of the inherent redundancy observed in networks to find multiple routes from one source node to a destination node. This becomes advantageous for *ad hoc* networks because they are characterized to be very dynamic, and unpredictable in nature [7].

In multipath routing, multiple redundant packets are sent along different paths between a pair of source-destination nodes. This redundancy increases the reliability in the transmission of the information [17], implying that there is a much greater chance (than in unipath routing) that at least one of the paths will be able to successfully deliver the packet. This further ensures its success as a fault-tolerant routing algorithm which provides route resilience when there are route failures in the network. However, the disadvantage of multipath routing is that when redundant packets are sent through different routes, they introduce an unnecessary overhead in the network's capacity [7,18]. This is disadvantageous especially when we take into account the fact that energy-efficiency is an important concern in wireless *ad hoc* networks [18], because most mobile nodes in such environments are battery powered, and are, thus, resource constrained.

Some of the multipath routing algorithms are also capable of providing load balancing in the network by carefully selecting a mechanism to split traffic along different routes to avoid overloading any single route. This is often quite advantageous in wireless network environments because while, sometimes, it might be difficult to guarantee the reservation of a *large* portion of the bandwidth through a single path, it might be possible to reserve *small* portions of the bandwidth over multiple routes through many paths taken together [7].

The multipath routing algorithms, in general, involve three phases: *route discovery*, *route maintenance*, and *traffic allocation*. The overall *route discovery* and *route maintenance* strategies in multipath routing are similar to those in unipath routing, except that in a multipath routing protocol, multiple routes are discovered or maintained between a pair of source-destination nodes [7].

Two important issues arise in multipath routing, which are the number of paths that would be considered to be optimal, and the selection mechanism of the paths. Nelakuditi and Zhang [8] published an interesting paper that addresses these issues, because the performance of a multipath routing scheme is dependent on the number and the quality of the chosen multiple paths. They proposed a hybrid approach that uses the idea of exchanging link state metrics to identify a set of "good" paths. Without delving deeper into their approach, we review below some of the commonly-used approaches for the selection of multiple paths.

The multiple paths discovered in multipath routing may take different forms categorized as being *node disjoint*, *link disjoint*, or *non-disjoint* routes. In node disjoint routes, there are no overlapping nodes or links. In link disjoint routes, there are no overlapping links, while in non-disjoint routes one permits overlapping nodes or links. The advantage of having disjoint routes is that they provide greater fault-tolerance, in the sense that if one of the nodes/links fail, it is quite unlikely *that* the failure will affect any of the other routes. *Route maintenance* in multipath routing is similar to the one done in unipath routing, except that the protocol requires a decision to be made as to when a *route discovery* phase needs to be triggered, i.e., when a broken link is identified. This is because triggering a *route discovery* every time a failure is identified introduces more traffic, and results in a degraded network performance. On the other hand, if one waits for all the disjoint routes between a pair of source-destination nodes to fail before invoking a *route discovery*, it might result in an unreasonable amount of delay [7].

3. Fault-tolerant MANETs

Due to the mobility of the nodes and the associated rapidly-changing topologies, the reliability of the correct transmission of messages is an important concern for MANETs. Hence, we now consider strategies that would guarantee the delivery of packets in adversarial environments, and in the presence of node/link failures.

The well-known MANET routing algorithms listed above (e.g., DSR, multipath routing etc.) are unsuitable as fault-tolerant routing algorithms for MANETs. Since the DSR chooses the shortest path route for packet transmission in adversarial environments, it can be shown that it will achieve a low packet delivery rate. On the other hand, multipath routing algorithms are strong in their fault-tolerance ability, because they send multiple copies of packets through all possible (disjoint) routes between a pair of source-destination nodes. However, the disadvantage with multipath routing algorithms is that they introduce an unnecessary amount of overhead on the network. Without a mechanism that "tolerates" route failures due to malfunctioning nodes (while making routing decisions), the performance of *ad hoc* network protocols will necessarily be poor, and the routing decisions made by those protocols would be erroneous.

Xue and Nahrstedt [10,11] confirmed that devising a fault-tolerant routing algorithm for *ad hoc* networks is inherently hard. This is because the problem itself is NP-complete due to the unavailability of "correct" path information in these environments. In [10], they designed an efficient algorithm, called the *End-to-End Fault Tolerant Routing* (*E2FT*) *Algorithm*, which is capable of significantly lowering the packet overhead, while guaranteeing a certain packet delivery rate. Following the work of Xue and Nahrstedt [10,11], Oommen and Misra [15] proposed a weak-estimation learning based fault-tolerant routing protocol for MANETs. Very recently, Misra et al. [20] also proposed a low overhead ant-swarm inspired routing protocol for MANETs. This chapter is primarily based on the paper published by Oommen and Misra [15], and most of the discussions and results presented here can also be found in [15].

The algorithms that attempt to solve the fault-tolerant routing problem do so by:

1. Either "flooding" the network with multiple redundant packets along different paths between a pair of source-destination nodes (thus, increasing the probability of a successful transfer);
2. Following a dynamic on-demand routing protocol, where the source *explicitly* provides, *a priori*, the transmitted packet with the complete information of the route to be followed, and hence minimizing the number of multiple redundant packets being transmitted; or
3. Seeking a "happy" medium between the latter strategies, namely, by estimating the potential profitability of maintaining selected paths.

The strategy which is presented by Oommen and Misra [15] is a combination of all these three philosophies [15]. The rationale for this strategy can be catalogued as follows:

1. First of all, this strategy opts to retain *certain* multiple redundant paths, and hence follows the basic principles of the multipath families;
2. Secondly, the strategy simultaneously seeks a solution that minimizes the "flooding", and hence pursuing the *dynamic* source-routing philosophy;
3. Finally, the strategy is akin to the one proposed in [10,11], except that it attempts to explicitly consider the nature of the random variables encountered. Observe that since the nodes are *mobile*, these random variables are, by definition, non-stationary. Thus, rather than using traditional maximum likelihood estimates, we argue that it is expedient to utilize *weak* estimates, namely those that converge in *distribution* as opposed to those that converge *with probability one*. We achieve this by invoking novel *weak* estimation methods that are built on the principles of stochastic learning – as explained in [12,13].

To the best of our knowledge, a scheme which collectively uses all these principles is novel to the work of Oommen and Misra [15]. Indeed, more particularly, we are not aware of any reported method which utilizes non-traditional estimates to achieve the ranking of all possible paths. *These are the novel contributions of this chapter*.

4. Problem model

The problem model that was considered by Oommen and Misra [15] is similar to that used by Xue and Nahrstedt [10], with a few differences introduced in order to simulate more realistic MANET scenarios. Their study, however, considers non-stationary environments, as discussed later in this section. We consider a graph G = (V, E) consisting of |V| mobile nodes, and |E| bi-directional links connecting different nodes. If there are n mobile nodes in a path, the length of any path p is denoted by L(p), in which $p = \{v_1, v_2, ..., v_n\}$, where $v_1, v_2, ..., v_n \in V$, and where every pair $(v_i, v_{i+1}) \in E, i \in \{1,2,...,n-1\}$. The multipath routes between a pair of source-destination nodes is denoted by $\pi = \{p_1, p_2, ..., p_m\}$, where m is the number of paths between any pair of source-destination nodes. In such a model, $L(\pi) = \sum_{i=1}^{m} L(p_i)$ is used to represent the length of the multipath route.

The *packet delivery probability of a path* is represented as $\gamma(p) = \prod_{i=1}^{m} \gamma(v_i)$. If there are m paths in a multipath route between a pair of source-destination nodes, the packet *delivery probability of a multipath route,* $\gamma(\pi)$, determines the probability that when multiple copies of the packets are sent along all the m paths between the source-destination pair, at least one copy is received. Clearly, $\gamma(\pi)$ is calculated as $\gamma(\pi) = 1 - \prod_{i=1}^{m} (1 - \gamma(p_i))$.

The problem that is addressed in the subsequent portions of this chapter consists of determining a mechanism for fault-tolerant routing that would route packets through mobile nodes in the above environment (i.e., in the presence of faulty nodes) by providing a certain packet delivery rate guarantee, and at the same time, by attempting to route "the least" number of duplicate packets through multiple routes between a pair of source-destination nodes. The reader should note that "blind" multipath routing algorithms are capable of achieving a high packet delivery rate guarantee, because they utilize the benefits of network redundancy. However, their disadvantage is that they route duplicate packets through the multipath routes to provide such a high packet delivery guarantee. Therefore, a solution was sought that would provide a certain "optimum" packet delivery rate guarantee, and that would, simultaneously, reduce the "overhead" routing that could burden the network by virtue of the packet duplication mechanisms adapted by the existing "blind" multipath routing algorithms.

Another *objective* of the work was to propose an algorithm that would be efficient in *non-stationary* environments, i.e., environments in which the fault probability of a mobile node increases as it moves away from the center of the network in which it is supposed to operate. In other words, we would enforce the constraint that as a node moves away from the center of the region of operation, the likelihood of it dropping packets also increases. This is an enhancement of the work by Oommen and Misra [15] over the work by Xue and Nahrstedt [10].

In the interest of brevity, our present survey of the E2FT algorithm is necessarily brief. The algorithm involves two major phases: A *route estimation* phase and a *route selection* phase. The *route estimation* phase is used to estimate the packet delivery probability of all the routes at the disposal of the algorithm at any time instant. As opposed to this, the *route selection* phase is used to select those routes that are confirmed to have satisfied a certain optimization constraint, and to drop those routes from further consideration that are estimated to be unnecessary among all the available multipath routes between a pair of source-destination nodes.

In the *route estimation* phase, the number of packets sent depends on the level of accuracy desired as per the estimation process. Note that a superior estimation is achieved by sending a larger number of packets, compensated by a tradeoff of the overall high network overhead. The accuracy of the estimation is achieved progressively through iterations.

The *route selection* algorithm works as follows. At the beginning, since no estimation results are available, all paths between a pair of source-destination nodes are selected to route the packets. By using a suitable estimation criterion, when the associated estimates of the paths are guaranteed to be accurate enough, the paths are reviewed to either be *confirmed* as one of the routes that "wins" the selection process and be permanently used for routing all future requests, or be *dropped* from further routing considerations.

5. Weak estimation-based fault tolerant routing

As mentioned earlier, the *objective* of the weak-estimation based fault tolerant routing solution proposed in [15] was to minimize the overhead by sending the least possible number of redundant packets, while guaranteeing a certain rate for the delivery of packets. We again emphasize in this chapter as well that there is a tradeoff between the rate of delivery of packets and the overhead. It is possible to achieve a very high packet delivery rate if the number of packets sent is not a concern (e.g., by using the multipath routing scheme). On the other hand, it is possible to achieve a very low overhead, if we do not care about the number of packets that are successfully delivered (e.g., by using the DSR scheme). Thus, attempting to increase one will decrease another and *vice versa*. What is challenging is to see how we can achieve a "balance" between the two. In other words, we need an algorithm that will be able to minimize the overhead by guaranteeing a certain level of efficiency of the packet delivery process. To achieve our objective, we propose a stochastic learning-based weak estimation fault-tolerant routing scheme.

5.1 Weak estimation learning

In statistical problems involving random variables, the quality, reliability, and accuracy of the estimation are important considerations. Traditionally, there have been different estimation schemes proposed in the literature, which can broadly be classified as either belonging to the *Maximum Likelihood Estimator* (*MLE*) class of algorithms [3,4], or as belonging to the *Bayesian family* of algorithms [1,3]. Although the above estimation schemes have been proved to be quite efficient, they work under the premise that the underlying distribution in the environment is *stationary*, i.e., the estimated parameter does not vary with time. In this context, the first two authors of this chapter studied this problem [12,13], and proposed a novel estimation scheme for learning in non-stationary environments[1]. They considered the case when the parameter associated with Bernoulli trials, which lead to binomially distributed outcomes of random variables, changed with time.

In the fault-tolerant routing solution presented in [15], we had used this efficient procedure for the estimation of the packet delivery probability through available paths. It is called the *Stochastic Learning Weak Estimator* (*SLWE*) scheme[2] [12,13], and is based on the principles of the stochastic learning paradigm. It uses a learning parameter, λ, which does not influence the mean of the final estimate. On the other hand, the variance of the final distribution, and the speed of convergence decrease with the increase in the value of this learning parameter. We discuss below the weak estimation scheme.

[1] The theory of these estimates is presented here, briefly, and without the fine details of the respective proofs. They are found in [12].

[2] The term "weak" used in the SLWE estimator scheme refers to the weak convergence of the random variable with respect to the first and second moments only.

Let us consider a binomially distributed random variable, X, as follows:

$$X=\begin{cases}0 & \text{with probability } s_0\\ 1 & \text{with probability } s_1\end{cases} \tag{1}$$

such that $s_0 + s_1 = 1$, where $S = [s_0, s_1]^T$

At any time, t, let X assume the value x(t). In order to estimate s_0 and s_1, SLWE keeps track of the running estimate $p_i(t)$ of s_i at time t, where i = 0,1. In such a setting, the value of p_0 is updated using the following *multiplicative* scheme:

$$p_0(t+1) = \begin{cases}\lambda \times p_0(t) & \text{if } x(t)=1\\ 1-\lambda \times p_1(t) & \text{if } x(t) = 0\end{cases} \tag{2}$$

where λ is a constant (0<λ<1), called the learning parameter, and $p_1(t+1) = 1 - p_0(t+1)$

We now present below some of the interesting results [12] concerning the SLWE.

Theorem 1: *Let X be a binomially distributed random variable, and $P(n)$ be the estimate of S at time 'n'. Then, $E[P(\infty)] = S$.*

Proof. Based on the updating scheme specified by Eq. (2), the conditional expected value of $p_1(n+1)$ given P can be seen to be:

$$E[p_1(n+1) \mid P] = \lambda s_2 p_1 + s_1 - \lambda s_1 + \lambda s_1 p_1 \tag{3}$$

$$= (1-\lambda)s_1 + \lambda p_1(s_1 + s_2) \tag{4}$$

$$= (1-\lambda)s_1 + \lambda p_1. \tag{5}$$

Taking expectations a second time, we can write (5) as:

$$E[p_1(n+1)] = (1-\lambda)s_1 + \lambda E[p_1(n)]. \tag{6}$$

As $n \to \infty$, $E[p_1(n)]$ converges to a limit because the coefficient of the linear difference equation is λ, where $0 < \lambda < 1$. Futhermore, if it converges to $E[p_1(\infty)]$, we can solve for $E[p_1(\infty)]$ from (6) as:

$$E[p_1(\infty)](1-\lambda) = (1-\lambda)s_1, \tag{7}$$

implying that $E[p_1(\infty)] = s_1$. Similarly, $E[p_2(\infty)] = s_2$, and the result follows. ■

The next results which we shall prove indicate that $E[P(n+1)]$ is related to $E[P(n)]$ by means of a stochastic matrix. We derive the explicit stochastic dependence, and allude to the resultant properties by virtue of the stochastic nature of the matrix. This leads us to two results, namely that of the mean of the limiting distribution of the vector $P(n)$, and that which concerns its rate of convergence. It turns out that while the former is independent of the learning parameter, λ, the latter is determined *only* by λ. The reader will observe that the results we have derived are asymptotic. In other words, the mean of $P(n)$ is shown to converge exactly to the mean of S. The implications of the "asymptotic" nature of the results will be clarified presently.

Theorem 2: *If the components of* $P(n+1)$ *are obtained from the components of* $P(n)$ *as per Eq. (2),* $E\left[P(n+1)\right] = \mathbf{M}^T E\left[P(n)\right]$*, where* $\mathbf{M}$ *is a stochastic matrix. Thus, the limiting value of the expectation of* $P(.)$ *converges to* S*, and the rate of convergence of* P *to* S *is fully determined by* λ*.*
Proof. Consider Eq. (6). Since $p_1 + p_2 = 1$, we can write:

$$E\left[p_1(n+1) \mid P\right] = (1-\lambda)s_1(p_1+p_2) + \lambda E\left[p_1(n)\right] \tag{8}$$

$$E\left[p_2(n+1) \mid P\right] = (1-\lambda)s_2(p_1+p_2) + \lambda E\left[p_2(n)\right]. \tag{9}$$

Substituting the above equalities, simplifying and taking expectations again leads to the following vectorial form:

$$E\left[P(n+1)\right] = \mathbf{M}^T E\left[P(n)\right], \tag{10}$$

where

$$\mathbf{M} = \begin{bmatrix} (1-\lambda)s_1 + \lambda & (1-\lambda)s_2 \\ (1-\lambda)s_1 & (1-\lambda)s_2 + \lambda \end{bmatrix} = (1-\lambda)\begin{bmatrix} s_1 & s_2 \\ s_1 & s_2 \end{bmatrix} + \lambda \mathbf{I}, \tag{11}$$

is a stochastic matrix. Since, as $n \to \infty$, both $E\left[P(n+1)\right]$ and $E\left[P(n)\right]$ converge to $E\left[P(\infty)\right]$, it follows that:

$$E\left[P(\infty)\right] = \mathbf{M}^T E\left[P(\infty)\right]. \tag{12}$$

Using Eq. (11), we now show that:

$$E\left[P(\infty)\right] = S, \tag{13}$$

as follows:

$$E\left[p_1(\infty)\right] = (1-\lambda)s_1\left\{ E\left[p_1(\infty)\right] + E\left[p_2(\infty)\right] \right\} + \lambda E\left[p_1(\infty)\right] \tag{14}$$

$$= (1-\lambda)s_1 + \lambda E\left[p_1(\infty)\right] \tag{15}$$

$$\Rightarrow E\left[p_1(\infty)\right](1-\lambda) = s_1(1-\lambda). \tag{16}$$

which implies that $E\left[p_1(\infty)\right] = s_1$.
An exact parallel argument leads to the result that $E\left[p_2(\infty)\right] = s_2$, whence the first result of the theorem is proved. Observing that $(\mathbf{M} - \lambda\mathbf{I})$ has the common factor $(1-\lambda)$, it follows that the convergence of P to S, which, in general, is determined by the eigenvalues of $\mathbf{M}$, is *fully determined* by λ. Hence the theorem. ■

From the analysis given above, we can derive the explicit expression for the asymptotic variance of the SLWE. We show that a small value of λ leads to fast convergence and a large variance. As opposed to this, a large value of λ implies slow convergence and a small variance.

Theorem 3: *Let* X *be a binomially distributed random variable governed by the distribution* S*, and* $P(n)$ *be the estimate of* S *at time '* n *' obtained by Eq. (2). Then, the algebraic expression for the variance of* $P(\infty)$ *is fully determined by* λ*.*

Proof. Using the same notation as above, the square of p_1 at time ' $n+1$ ' is given by:

$$p_1^2(n+1) = \lambda^2 p_1^2 \qquad w.p.s_2 \tag{17}$$

$$= 1 - 2\lambda(1-p_1) + \lambda^2(1-p_1)^2 \qquad w.p.s_1 \tag{18}$$

$$= 1 - 2\lambda + 2\lambda p_1 + \lambda^2(1 - 2p_1 + p_1^2) \tag{19}$$

$$= 1 - 2\lambda + 2\lambda p_1 + \lambda^2 - 2\lambda^2 p_1 + \lambda^2 p_1^2. \tag{20}$$

Using Eq. (20), we can write $E\left[p_1^2(n+1) \mid P(n) = P\right]$ as:

$$E\left[p_1^2(n+1) \mid P(n) = P\right] = \lambda^2 p_1^2 s_2 + (1 - 2\lambda + \lambda^2)s_1 + 2\lambda(1-\lambda)p_1 s_1 + \lambda^2 p_1^2 s_1 \tag{21}$$

$$= \lambda^2 p_1^2 + 2\lambda(1-\lambda)p_1 s_1 + (1-\lambda)^2 s_1. \tag{22}$$

From Eq. (22), we observe that as $n \to \infty$, both $E\left[p_1^2(n)\right]$ and $E\left[p_1^2(n+1)\right]$ converge to $E\left[p_1^2(\infty)\right]$. Thus, by gathering terms involving $E\left[p_1^2(n)\right]$, Eq. (22) can be written as:

$$E\left[p_1^2(\infty)\right] (1-\lambda^2) = 2\lambda(1-\lambda)E\left[p_1(\infty)\right]s_1 + (1-\lambda)^2 s_1, \tag{23}$$

which can also be expressed as:

$$E\left[p_1^2(\infty)\right] (1+\lambda) = 2\lambda E\left[p_1(\infty)\right]s_1 + (1-\lambda)s_1 \tag{24}$$

$$= 2\lambda s_1^2 + (1-\lambda)s_1, \tag{25}$$

where the last equalities hold since $E\left[p_1(\infty)\right] = s_1$. Thus, we have:

$$E\left[p_1^2(\infty)\right] = \frac{2\lambda s_1^2 + (1-\lambda)s_1}{1+\lambda}. \tag{26}$$

We finally compute the variance of $p_1(\infty)$ as below:

$$Var[p_1(\infty)] = E[p_1^2(\infty)] - E[p_1(\infty)]^2 \tag{27}$$

$$= \frac{(1-\lambda)s_1 s_2}{1+\lambda}, \tag{28}$$

and since $s_2 = 1 - s_1$, the theorem is proved. ■

When $\lambda \to 1$, the variance tends to zero, implying mean square convergence. The *maximum* value of the variance is attained when $\lambda = 0$, and the *minimum* value of the variance is achieved when $\lambda = 1$.

Our result seems to be contradictory to our initial goal. When we motivated our problem, we were working with the notion that the environment was non-stationary. However, the results we have derived are asymptotic, and thus, are valid only as $n \to \infty$. While this could prove to be a handicap, realistically, and for all practical purposes, the convergence takes place after a relatively small value of n. As we will see later, in practice, choosing a value of λ in the interval $[0.9, 0.99]$ yields quite good results. Thus, if λ is even as "small" as 0.9, after 50 iterations, the variation from the asymptotic value will be of the order of 10^{-50}, because λ also determines the rate of convergence, and this occurs in a geometric manner [19]. In other words, even if the environment switches its Bernoulli parameter after 50 steps, the SLWE will be able to track this change. Observe too that we do not need to consider the use of a "sliding window".

5.2 The WEFTR algorithm

In [15], Oommen and Misra used the above-mentioned weak-estimation learning scheme to propose a new fault-tolerant routing algorithm, named the *Weak-Estimation-Based Fault Tolerant Routing* (*WEFTR*) *Algorithm*, which is capable of efficiently estimating the probability of the delivery of packets through the paths available at any moment. Like the E2FT algorithm [10], the WEFTR algorithm involves, among other steps, a *route estimation* phase and a *route selection* phase. The *route estimation* phase is used to estimate the packet delivery probability of all the routes at the disposal at any time instant, whereas the *route selection* phase is used to select those routes that are confirmed to have satisfied a certain optimization constraint, and to drop the unnecessary multipath routes between a pair of source-destination nodes.

In the *route estimation* phase, N packets are sent along a path p. The source node estimates the fraction of packets delivered, $\hat{\gamma}(p)$ from the number of packets, N′, received along that path[3].

In our strategy, the estimate of the packet delivery probability is refined with the increase in the number of iterations. At every iteration, a set of packets is transmitted through each of the multipath routes between a pair of source-destination nodes. We can have two possible scenarios for any path: The nodes in a path either forward the packets correctly, or they do not. Consequently, we can use a binomial estimation scheme (based on the above SLWE) as follows:

$$\hat{\gamma}_0(p) = \begin{cases} \lambda \times \hat{\gamma}_0(p) & \text{if the path does not forward the packet correctly} \\ 1 - \lambda \times \hat{\gamma}_1(p) & \text{if the path forwards the packet correctly} \end{cases} \tag{29}$$

where λ is the learning parameter, such that $0<\lambda<1$, and $\hat{\gamma}_1(p) = 1 - \hat{\gamma}_0(p)$.

In our *route selection* algorithm, for a path to be confirmed, the following condition should be satisfied: $\hat{\gamma}_{WE}(p) \geq \gamma^*$, where γ^* is the minimum packet delivery probability required for a path to be confirmed, and $\hat{\gamma}_{WE}(p)$ is the packet delivery probability estimate using the SLWE scheme presented in Eq. (29). Once a path is confirmed, it is considered to be useful for routing future requests, and consequently, no further estimation is carried out on that path.

[3] Traditionally, this is estimated as: $\hat{\gamma}(p) = \dfrac{N'}{N}$.

The dropping algorithm selects a path, p_{min}, from all the available paths, π, with the minimum packet delivery estimation value, where the latter is examined to see if the following dropping condition is satisfied [10]:

$$\begin{aligned} &\hat{\gamma}_{WE^{1/m}}(\pi') \ \geq \ \gamma^{*} \\ &\text{where } \hat{\gamma}_{WE^{1/m}}(\pi') = 1- \prod_{p\in\pi'}(1-\hat{\gamma}_{WE^{1/m}}(p)), \text{ and } \pi' = \pi - \{p_{min}\} \end{aligned} \tag{30}$$

With the above as a background, we present below a high level sketch of the WEFTR algorithm [15].

Algorithm WEFTR

Input

- A graph (network) with a set of nodes, and a set of links connecting the nodes.
- The nodes are mobile, and links connecting them can be reset with the change in the position of the nodes.
- Some of the nodes in the network are faulty with a certain packet delivery rate dependent on the distance of the node from the center of the area of mobility of the mobile nodes, which, for the purpose of this study, is the "simulation area".

Output

- All the incoming packets are delivered from the source node to the destination node, with the intention of maximizing the packet delivery rate, and minimizing the network overhead.

Algorithm

```
BEGIN
Step 0 (initialization) - Initialize a vector WEFTR_MP that stores all the paths in use, and
WEFTR_Nodes that stores all the nodes in the graph, along with the information about their
estimated packet delivery probabilities.
At each time unit, do the following:
Case 1: If the unit of time is a simulation pause then
          Step 1. Save the estimated packet delivery probability of each node in the vector
                  WEFTR_Nodes.
          Step 2. Update the edges and probabilities in the graph to reflect the current
                  position of the nodes, and calculate the new paths from the source to the
                  destination.
          Step 3. Use the values stored in WEFTR_Nodes in order to calculate the estimated
                  (using the SLWE) packet delivery probability of each path.
Case 2: At each unit of time
          Step 1. Try to confirm or drop paths. Paths dropped are removed from the
                  WEFTR_MP vector.
          Step 2. Use all the paths in the WEFTR_MP vector to send the packets, and
                  calculate the number of packets that are received for each path and the
                  total number of non-duplicated packets that are received.
END
```

5.3 Experimental setup

In order to determine how the performance of the proposed algorithm compares with other competing algorithms[4], we simulated an *ad hoc* network with mobile nodes and dynamically changing topologies, and then ran our proposed algorithm along with the other benchmark algorithms (described in the next Section) in the simulated environment. The results, which appeared in [15], are presented here again.

5.3.1 Simulation environment

The simulated environment that we considered consisted of a flat square of length 500 meters. There were 50 nodes in the network, each having a different data delivery probability which decreases as they move away from the center of the square, and increases as they move closer to it. In other words, if we fix a node in the centre of the square, the reliability of data delivery to its peer nodes (and vice versa) decreases as those peer nodes move away from it. This can happen due to the diminishing signal strength between any pair of communicating wireless devices when they move away from each other. Furthermore, to assume that things are done in a systematic manner (i.e., as per the benchmark accepted "standards") we assumed that each node moves randomly, following the *random waypoint model*[5]. If after a random move as per Eq. (31) and (32) below, a node reaches the edge of the square, then the move is canceled and a new random move for this is done until it lands in a valid position. In our simulated *ad hoc* network, we assumed that the maximum speed with which the mobile nodes can travel is 20m/s. Observe that the nodes move at each time unit, but the links between them are only recalculated at a simulation pause[6]. The maximum speed of a node specified above (i.e., 20 m/s) is needed to calculate how much a node can move in a second. This is because the position of a node at the i^{th} second is calculated as:

$$\text{Xpos}(i) = \text{Xpos}(i-1) + \text{randnum} \tag{31}$$

$$\text{Ypos}(i) = \text{Ypos}(i-1) + \text{randnum} \tag{32}$$

In the above, Xpos(i-1) and Ypos(i-1) denote the abscissa and ordinate of a node in the previous second (or time instant), and Xpos(i) and Ypos(i) denote the abscissa and ordinate, respectively, of the corresponding node at the current second (or time instant). If the

[4] There are currently quite a few algorithms (and their variants) reported in the literature that claim to solve the present problem. It is clearly impossible to compare any single algorithm with all of them. But we had opted to compare the WEFTR algorithm with individual schemes that represent the various "families" of strategies reported earlier. The rationale for choosing these was that we believed that it represents a reasonably fair comparison against the entire spectrum of philosophies motivating the algorithms. We are currently considering undertaking a more comprehensive comparison (including a testing on "real-life" network topologies).

[5] The details of this model can be found at http://www.netlab.tkk.fi/~esa/java/rwp/rwp-model.shtml

[6] Here, we assumed in [15] that the links between the nodes in the network do not get "torn down" with every movement of the nodes in the network. In other words, we assumed that the links in the network remain connected until a certain time (i.e., the Pause Time). The alternative could have been to re-compute the links in the network with a unit movement of nodes. The former, according to our view, although debatable, is more realistic. Additionally, re-computing the links with every movement of the nodes in the network would lead to a prohibitively large computational overhead.

maximum speed is 20m/s, the randnum shown above is a random number generated between -20 and +20.

The maximum distance that two nodes can have for which they are connected (i.e., that they can deliver packets to each other) is directly dependent on the simulation parameter referred to as the network's *"Sparsity"*. The *Sparsity* of the network is an attribute that signifies how the nodes connect with one another, and is a coefficient whose value ranges between 0 and 1 as follows: A value of 1 signifies that no edges (100% sparse coefficient) connect with one another, whereas a value of 0 signifies that the maximum possible number of nodes connect with one another (i.e., a 0% sparse coefficient). The reader should observe that in the simulation there is no fixed number of links in the networks. The links are recalculated at each simulation pause. This is because two nodes are considered to have a link, if they are within a certain distance of each other. Thus, the *Sparsity* directly influences this distance.

Another parameter that was used in the simulations is the so-called *Pause Time*. It signifies how the algorithm is able to accommodate node mobility. This parameter defines the time interval after which the links are recomputed. Each of the simulations was run for 500 seconds. During the simulation period, random Constant Bit Rate (CBR) traffic was generated between a pair of nodes, where this random traffic had a rate of 10 KB/s. Also, during the simulations the SLWE's learning parameter was kept constant, although, as mentioned earlier, the learning parameter does not influence the mean of the final estimate.

We also determine how far a node is from the center of the square, by measuring its Euclidean distance from the center.

5.3.2 Benchmark algorithms

In order to assess how our algorithm performs when compared to the existing algorithms, we had selected three algorithms [15], all of which were executed together with our proposed algorithm in the simulated environment. The three benchmark algorithms that we chose were:

1. DSR Algorithm
2. Multipath Routing Algorithm
3. E2FT Algorithm[7]

Of these three algorithms, the E2FT represented the the state-of-the-art in the area of fault-tolerant routing in MANETs, and so, we reckoned that the performance comparison between our algorithm and the E2FT was crucial. However, since the DSR and the Multipath routing algorithms are currently widely used in deployed MANETs, they were also considered. Also, although the DSR is a simple routing algorithm, it is weak when it concerns routing information in the presence of malfunctioning nodes. On the other hand, multipath routing is, perhaps, a very strong routing algorithm when there are misbehaving nodes. But, as mentioned earlier, the most significant limitation of multipath routing is that it possesses a large network overhead, as it "loads" all the relevant routes between a pair of source-destination nodes with redundant packets so as to ensure that the destination node receives at least one correct copy of the packet sent from the source.

5.3.3 Performance metrics

Two metrics were used in [15] for evaluating the performance of the algorithms invoked in the experiments:

[7] In our study [15], to be fair to the competition, we had considered the optimized version of E2FT that provides an optimization methodology – namely the one that takes the mobility of the nodes into account.

1. *Percentage of packets delivered*: This represents the rate of the successful delivery of packets to the destination, and is calculated as follows: At each second, the packet delivery probability of all the paths in use is calculated. Then, for each packet sent at that time unit, a random number between 0 and 1 is generated. If the number is lower than the packet delivery probability, the packet is considered as having been delivered. Thereafter, after all the iterations, the percentage of delivered packets is calculated as follows:

$$\text{percentage delivered packets} = \frac{\text{total number of delivered packets}}{\text{total number of sent packets}}\,.$$

2. *Overhead*: This represents the overall number of packets sent. This *Overhead* index is calculated as the product of the total length of all the paths in use, and the number of packets sent per second (time unit).

5.3.4 Experimental results

Several experiments were conducted [15] to assess the performance of WEFTR (the proposed algorithm) with respect to the benchmark algorithms. The results of the following three sets of experiments are presented below (also available in [15]):

- Variation in *Pause Time*
- Variation in *Sparsity*
- Variation in the faultiness of nodes

Variation in Pause Time: As noted earlier, the *Pause Time* is a parameter specific to the simulation, which indicates how much an algorithm is capable of accommodating the mobility of the nodes. The results of the simulation for this scenario are given in Figure 1. From this figure, we notice that with respect to the *Overhead*, while the *blind* multipath routing is the worst, the DSR is the best, and the metric for the E2FT lies somewhere in between the DSR and the multipath curves. This is, of course, understandable. Our proposed algorithm further improves on the performance of the E2FT scheme by decreasing the *Overhead* by 25-50%. For example, when the *Pause Time* is 250 seconds, the *Overhead* for the multipath routing is 19,790, that for E2FT is 8,740, while that for the WEFTR is 7,225. On the other hand, from Figure 2, we observe that the WEFTR achieves an almost similar order of performance when compared to the E2FT. However, by examining Figures 1 and 2 together, one can infer that our proposed algorithm (WEFTR) is capable of *significantly* reducing the *Overhead* of the best fault tolerant routing algorithm (E2FT) currently available, while achieving a performance packet delivery guarantee of at least 80%. Thus, if one considers *both* these issues simultaneously, it is clear that our algorithm always performs much better than both the DSR and the blind multipath routing schemes.

Variation in Sparsity: In the second set of experiments, we intended to study how the algorithms compared with respect to each other with the variation in the *Sparsity* of the nodes in the network. As mentioned earlier, the value of *Sparsity* ranges between 0 and 1, where 0 represents the smallest percentage of *Sparsity*, and 1 represents the largest percentage of *Sparsity*. Since the nodes are mobile, the question of how often they connect with each other depends on how close they can get to one another, and, clearly, this is directly related to the *Sparsity*. The different *Sparsity* values used in our experiments indicate the relative number of edges between the nodes in the network.

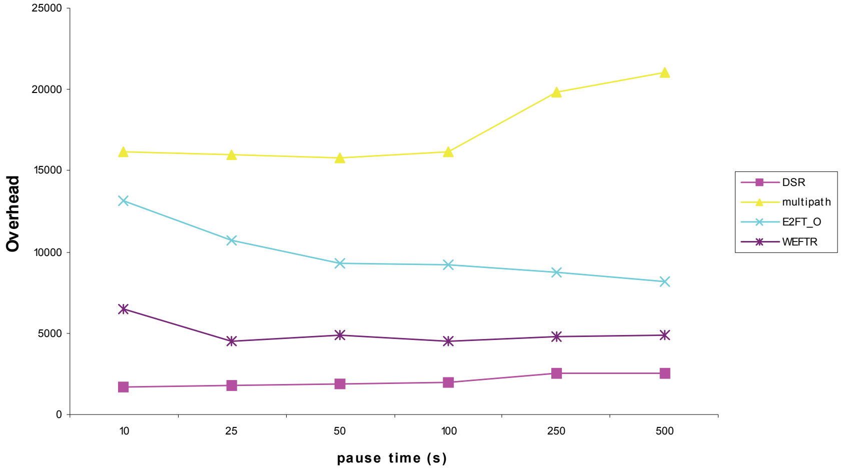


Fig. 1. Plot of the *Overhead* versus the *Pause Time* for the various algorithms tested.

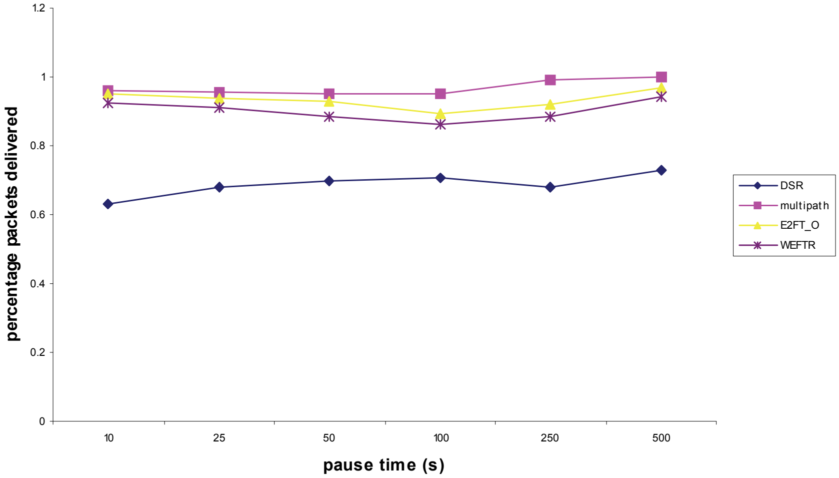


Fig. 2. Plot of percentage delivered packets versus *Pause Time* for the various algorithms tested.

Figures 3 and 4 depict the performance comparison of all the examined algorithms with respect to the overall *Overhead*, and the percentage of packets successfully routed by the algorithms. From Figure 3, we can clearly observe that even at different values of *Sparsity*, the E2FT is capable of significantly reducing the overall *Overhead*. For example, when the value of the *Sparsity* is 0.25, the *Overhead* for the multipath routing is 32,320, while that of the E2FT scheme is 11,410. As opposed to this, the *Overhead* for our proposed algorithm is only 5,570. It should also be observed that the performance of E2FT is much better at lower *Sparsity* values than at the higher ones. On the other hand, if one considers Figure 4, one can observe that, in general, the percentage of packets delivered by both E2FT and WEFTR are almost identical. Thus, for this set of experiments, we observed that the WEFTR significantly reduces the *Overhead* when compared to both the E2FT and blind multipath routing algorithms. This

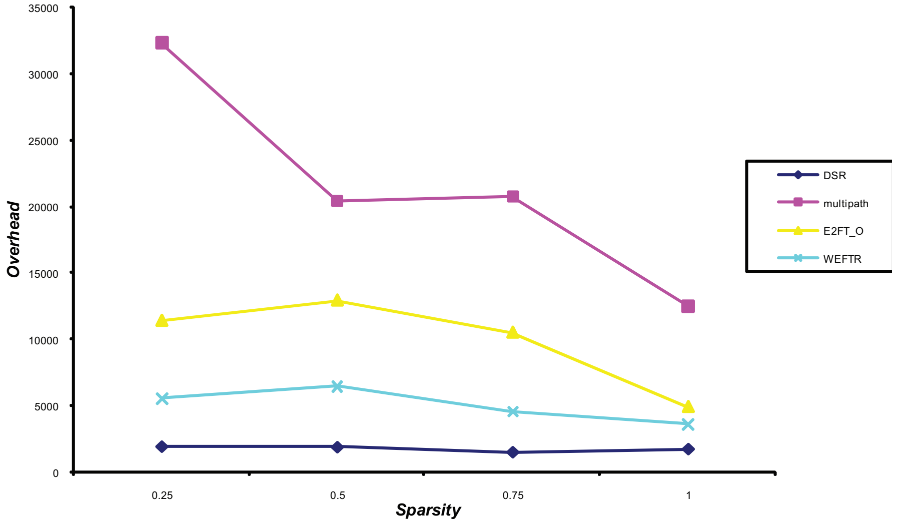


Fig. 3. Plot of the *Overhead* versus the *Sparsity* for the various algorithms tested.

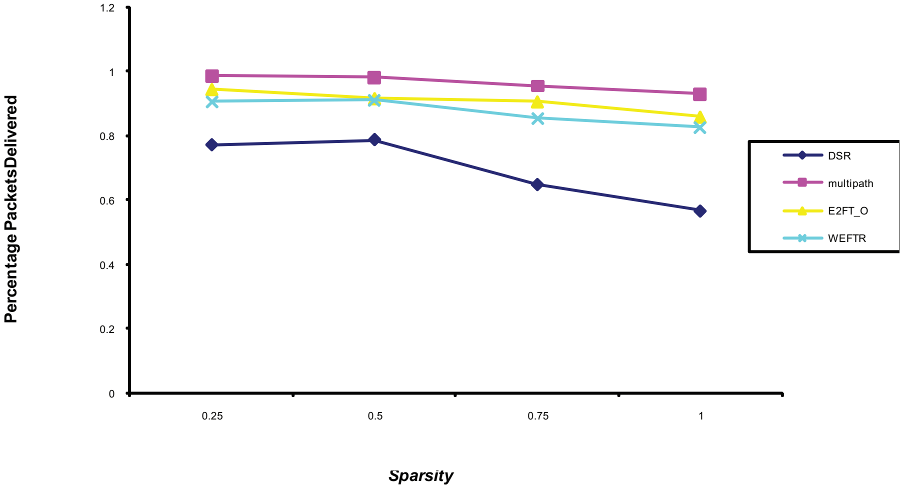


Fig. 4. Plot of the Percentage of Delivered Packets versus the *Sparsity* for the various algorithms tested.

was done while simultaneously achieving a performance comparable to that of the E2FT or the multipath schemes (and certainly yielding a performance noticeably superior to that of the DSR algorithm) with respect to the number packets successfully routed.

Variation in Faultiness: *Faultiness* is an internal simulation parameter that indicates how many nodes will be faulty[8] in a given environment. It influences the faultiness behavior of the nodes, given their distance from the center of the region of operation of the nodes. Figures 5 and 6

[8] In our simulations [15], we assumed that the faulty nodes do not deliver any packets at all.

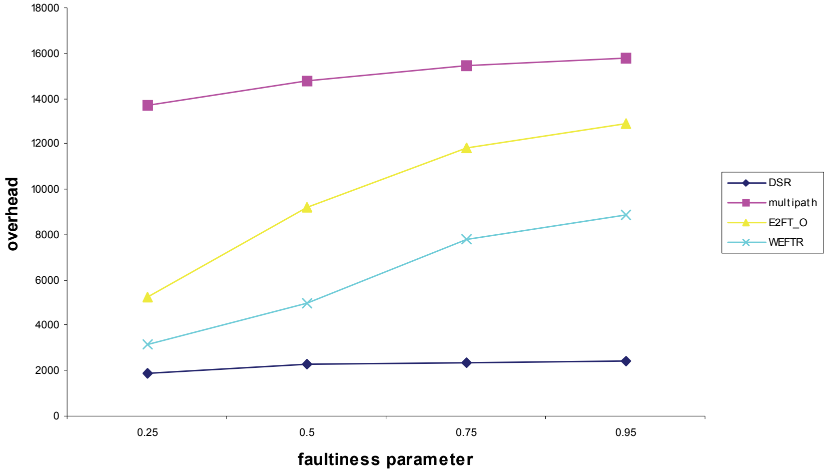


Fig. 5. Plot of the *Overhead* versus the *Faultiness* parameter for the various algorithms tested.

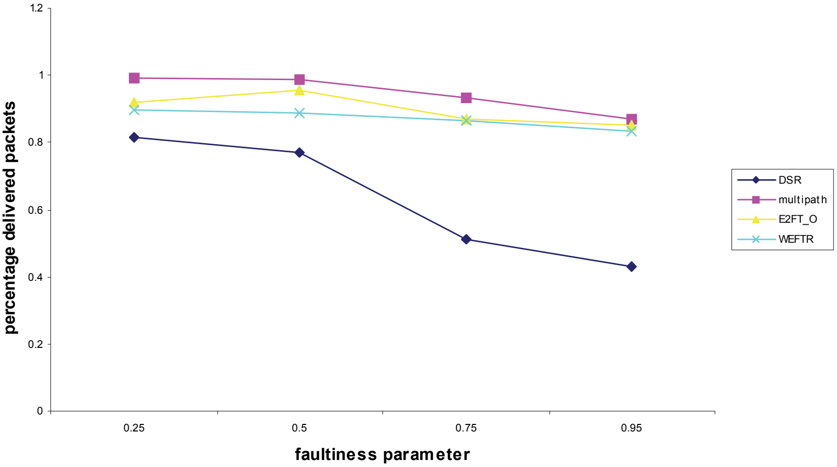


Fig. 6. Plot of percentage of delivered packets versus the *Faultiness* parameter for the various algorithms tested.

depict the variation in the *Overhead*, and the percentage of delivered packets, with the variation in the *Faultiness*. In our experiments, we had used the *Faultiness* parameter to vary from a very low value to a very high value (i.e., on a scale of 0 to 1). We observed that, even in this set of experiments, our proposed algorithm delivers much better performance, when compared to the other algorithms. For example, when the *Faultiness* parameter has a value of 0.25, the *Overhead* for the blind multipath routing is 13,690, for the E2FT is 5,240, while for the WEFTR, it is 3,150. Thus, in this case, our algorithm showed an improvement of about 62 % over multipath routing, and an improvement of about 40 % over the E2FT algorithm. All of these algorithms, however, in general, showed comparable performance with respect to the percentage of successfully delivered packets.

6. Conclusions

We have considered the problem of routing in MANETs, and reported the results of studying the interesting, yet challenging, problem of fault tolerant routing in MANETs, which also appeared in [15]. The problem is that of *efficiently* routing packets in MANETs in adversarial environments particularly, in the presence of misbehaving nodes. Apart from surveying the families of algorithms useful for non-fault tolerant schemes, we have considered state-of-the-art fault tolerant methods, and also devised an algorithm, which is able to successfully route packets by "tolerating" faults in the network. There are two principal metrics that characterize the quality of any fault tolerant routing algorithm designed for MANETs, namely: (1) The *Overhead*, and (2) The percentage of successfully delivered packets. The traditional algorithms, the DSR and the multipath routing, have the potential to attain two extremes of each of these metrics. While the multipath routing is a very strong algorithm for maximizing the number of successfully delivered packets, it introduces an extremely large *Overhead* into the network. On the other hand, the DSR has a low *Overhead*, but, simultaneously, is a very poor fault-tolerant routing algorithm, because it will drop packets if there are problems in the route identified by the algorithm. The E2FT algorithm, proposed by Xue and Nahrstedt [10], is capable of minimizing the *Overhead* when compared to the multipath routing algorithm, while achieving a similar order of performance (slightly inferior, to be more specific) with respect to the number of packets successfully delivered.

Since the nodes are mobile, it turns out that the random variables encountered are non-stationary, implying that estimation methods for stationary variables are inadequate. Consequently, in this chapter, we have also presented a fault-tolerant routing scheme [15] that invokes a stochastic learning-based weak estimation procedure to enhance a *route estimation* phase, which, in turn, is then incorporated in a *route selection* phase. Our algorithm significantly reduces the *Overhead* over the E2FT algorithm, while achieving a comparable performance when it concerns the number of successfully delivered packets. By rigorous simulations, we had shown in [15] that this new algorithm was successful in achieving the above goal.

In the future, we intend to test our proposed scheme on more realistic networks and topologies, and to also consider how *alternate* sequence-based estimates can be utilized advantageously to solve the same problem.

7. Acknowledgements

Most parts of this book chapter appeared in the *Telecommunication Systems* (*Springer*), Vol. 44, Nos. 1-2, June 2010, pp. 159-169. Some parts also appeared in *Pattern Recognition*, Vol. 39, 2006, pp. 328-341 and the Proceedings of *IEEE WiMob'08, pp.* 603-607.

8. References

[1] P. Bickel and K. Doksum, *Mathematical Statistics: Basic Ideas and Selected Topics*, Vol. 1, Prentice Hall, 2nd Edition, 2000.

[2] G. D. Caro, F. Ducatelle and L. M. Gambardella, "AntHocNet: An Ant-Based Hybrid Routing Algorithm for Mobile Ad Hoc Networks", Technical Report No. IDSIA-25-04-2004, Dalle Molle Institute for Artificial Intelligence, Switzerland, August 2004. (Also appeared in the *Proceedings of Parallel Problem Solving from Nature VIII*, LNCS 3242, Springer-Verlag, 2004, pp. 461-470).

[3] R. Duda, P. Hart and D. Stork, *Pattern Classification*, John Wiley and Sons, New York, 2nd Edition, 2000.

[4] R. Herbich, *Learning Kernel Classifiers: Theory and Algorithms*, MIT Press, Cambridge, MA, USA, 2001.

[5] D. B. Johnson and D. A. Maltz, “Dynamic Source Routing in Ad Hoc Wireless Networks”, *Mobile Computing*, 1996, pp. 153-181.

[6] M. K. Marina and S. R. Das, “On-Demand Multipath Distance Vector Routing in Ad Hoc Networks”, *Proceedings of the 9th International Conference on Network Protocols*, Riverside, California, 2001, pp. 14-23.

[7] S. Mueller R. P. Tsang and D. Ghosal, “Multipath Routing in Mobile Ad Hoc Networks: Issues and Challenges”, In *Lecture Notes in Computer Science*, Vol. 2964, Maria Carla Calzarossa and Erol Gelenbe (Eds.), 2004.

[8] S. Nelakuditi and Z. –L. Zhang, “On Selection of Paths for Multipath Routing”, *Proceedings of the 9th International Workshop on Quality of Service*, LNCS, Vol. 2092, Springer-Verlag, London, 2001.

[9] C. E. Perkins and E. M. Royer, “Ad-Hoc On-Demand Distance Vector Routing”, *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, Louisiana, 1999, pp. 207-218.

[10] Y. Xue and K. Nahrstedt, “Fault Tolerant Routing in Mobile Ad Hoc Networks”, *Proceedings of the IEEE Wireless Communications and Networking Conference* (*WCNC*), New Orleans, Louisiania, March 2003, pp. 1174-1179.

[11] Y. Xue and K. Nahrstedt, “Providing Fault-Tolerant Ad Hoc Routing Service in Adversarial Environments”, *Wireless Personal Communications*, Vol. 29, 2004, pp. 367-388.

[12] B. J. Oommen and L. Rueda, “Stochastic Learning-Based Weak Estimation of Multinomial Random Variables and Its Applications to Pattern Recognition in Non-stationary Environments”, *Pattern Recognition*, Vol. 39, 2006, pp. 328-341.

[13] B. J. Oommen and L. Rueda, “A New Family of Weak Estimators for Training in Non-Stationary Distributions”, *Proceedings of the 2004 International Symposium on Structural, Syntactic, and Statistical Pattern Recognition*, Lisbon, Portugal, August 2004, pp. 644-652.

[14] K. Wu and J. Harms, “On-Demand Multipath Routing for Mobile Ad Hoc Networks”, *Proceedings of EMPCC*, Vienna, February 2001, pp. 1-7.

[15] B. J. Oommen and S. Misra, “Fault-Tolerant Routing In Adversarial Mobile Ad Hoc Networks: An Efficient Route Estimation Scheme For Non-Stationary Environments”, *Telecommunication Systems Journal*, pp. 159-169, 2010.

[16] K. Wu and J. Harms, “On-Demand Multipath Routing for Mobile Ad Hoc Networks”, *Proceedings of EMPCC*, Vienna, February 2001.

[17] Z. Ye, S. V. Krishnamurthy and S. K. Tripathi, “A Framework for Reliable Routing in Mobile Ad Hoc Networks”, *Proceedings of IEEE INFOCOM*, San Francisco, 2003.

[18] V. Srinivasan, C. –F. Chiasserini, P. S. Nuggehalli and R. R. Rao, “Optimal Rate Allocation for Energy-Efficient Multipath Routing in Wireless Ad Hoc Networks”, *IEEE Transactions on Wireless Communications*, Vol. 3, No. 3, 2004.

[19] K. Narendra and M. A. L. Thathachar,. *Learning Automata. An Introduction*. Prentice Hall, 1989.

[20] S. Misra, S. K. Dhurandher, M. S. Obaidat, K. Verma and P. Gupta, "A Low Overhead Fault-Tolerant Routing Algorithm for Mobile Ad-Hoc Networks Based on Ant Swarm Intelligence" *Simulation Modelling Practice and Theory* (*Elsevier*), Vol. 18, No. 5, 2010, pp. 637-649.

18

LLD: Loop-free Link Metrics for Proactive Link-State Routing in Wireless Ad Hoc Networks

Takuya Yoshihiro
Wakayama University
Japan

1. Introduction

As wireless communication has become to be popular all over the world, the next promising network technology is the multi-hop wireless networks so called wireless ad hoc networks. Since wireless ad hoc networks require far lower cost to construct than wired networks, it is expected as technology to expand network coverage in both physical areas and applications. As infrastructure, wireless mesh networks (WMNs) (2) which consist of stationary nodes are considered to expand coverage of the broadband Internet. And further, with users' terminals such as PDAs, note PCs and mobile phones (mobile nodes), ad hoc networks will be more useful and flexible tools to enable further useful applications. Although many technical problems remain to be solved, the challenge to put ad hoc networks into practice is hopefully continuing.

One important issue on wireless ad hoc networks is how to supply stable and reliable communications between nodes over vulnerable wireless links. The current ad hoc networks adopt the same strategy as the traditional wired networks; nodes deploy a common routing protocol to recompute new paths in case of topology changes such as link failure. From this approach, currently four routing protocols AODV(3), DSR(4), OLSR(5) and TBRPF(6) have been standardized in IETF. Each of them adopts several mechanisms to handle ad hoc network specific properties such as mobility or wireless instability. They, however, basically compute the hop-count based shortest paths for destination nodes regardless of link state or quality.

Consequently, much work has been tried to improve communication quality by way of taking link stability and quality into account. In mobile scenario, since node mobility is a main factor which brings link breakage, the main concern to improve communication stability is to comprehend link duration. Many analytical results contributed to clarify the characteristics and the statistics of link duration from various points of view (8)(9)(11)(15)(16). Based on those knowledge, also several routing metrics are proposed to improve communication stability by using long lifetime links as communication paths (9)(10)(13)(18)(17)(19)(20)(21).

On the other hand, mainly for the network which consists of stationary nodes such as wireless mesh networks (WMN) (2), several link quality metrics are proposed (22)(23)(24)(26)(29). They are also deeply related to communication stability since link quality effects on communication speed or probability of disruption. In this area of literature, link qualities are defined as several link quality related measurements such as average transmission count, delay, or bandwidth to utilize links with the best availability. Also, load balancing is possibly included in the requirement of routing metrics since congestion may cause degradation of link quality in WMNs. Several proposals actually include this point of view in their design of routing metrics.

It is, however, worth noting that none of above dynamic routing metric proposal takes routing loops into account. Routing loops is an important factor of route stability since they may make communications degraded and unstable in link state routing scheme such as OLSR. Routing loops occur due to propagation delay of metric information which causes temporary inconsistency of the routing tables among network nodes. Routing loops are well studied in the literature of wired networks and has been also recognized as harmful phenomenon which brings severe congestion and loss of wireless connectivity (33)(34)(35)(36). In wireless networks, due to interference the harm of congestion coming from routing loops goes far severer than wired networks (37). It is also important for wireless multihop networks to eliminate routing loops in order to stabilize communications.

In this article, we present a dynamic link stability metric called LLD (Loop-free Link Duration) which guarantees loop-freeness even against transition of metrics. Our dynamic metric LLD and its mechanisms are designed to improve stability of communications in wireless ad hoc networks with low-cost extension for existing proactive link-state routing schemes such as OLSR. Specifically, we introduce our dynamic metric as a simple function of time, and give theoretical analysis on the two instability factors: routing loops and path oscillation. Although LLD concentrates on only one aspect of mobility metrics, i.e., link duration in stable state, this will be a practical example of low-cost loop-free mechanisms in dynamic metric environments. To the best of our knowledge, LLD is the first loop-aware routing metric for wireless ad hoc networks.

The rest of this article is organized as follows: We first describe the literature of the related studies in Section 2. Then, we explain our dynamic metric LLD and its mechanisms in Section 3. In Section 4, we define the problem of Loop-freeness and give a theoretical result on the condition of loop-freeness. In Section 5, we give the simulation results on path oscillation problem and show that path oscillation can be suppressed in our scheme. We also give traffic simulation results in Section 6. Finally concluding remarks are given in Section 7.

2. Related work

In this Section we describe the literature of several research topics related to this article. We first describe related work on routing metrics for wireless mobile ad hoc networks, and then we go to routing metrics for wireless networks which consist of stationary nodes, called wireless mesh networks (WMNs). After that, we show the literature of the study for eliminating routing loops in dynamic routing protocols.

2.1 On mobile networks

In mobile ad hoc networks, mobility of nodes is the biggest factor of losing communication stability since mobility brings link disruption. Although new paths are recomputed by routing protocols deployed, temporary disruption of communications is not negligible when we consider stability of networks.

The natural idea to make communications stable as long as possible is to use stable links which have long residual lifetime. From this idea, much work is done to estimate the residual link lifetime to select stable links for each destination. One major approach for this is to grasp mobility statistics using several mobility models. (Description of major mobility models is shown in (7)) In fact, from empirical experiment in (8), it is reported that the breakage of long lifetime links are mostly brought from mobility (not from interference or congestion) and the statistics on node degree and link lifetime observed are quite similar to the simulation result of typical mobility models such as Random WayPoint Model (RWP). In this context, Gerharz et al. (9) studied the distribution of link duration through simulations under several typical mobility models. They also estimated the average residual lifetime for the links of given age, and their simulation result showed that the residual lifetime estimation based next hop selection improves communication stability. Cheng et al. (10) proposed the routing metric based on the balance between route lifetime estimation and path length, and then compared the performance with shortest path routing. Further, Zhao et al. (11) focused on more practical factors on mobility models, i.e., they consider more accurate patterns of node movements and practical dynamics of transmission range, to estimate link statistics (on such as link lifetime) more accurately.

Wireless signal strength is also available to improve communication stability. In the early stage of the literature, simple signal strength based routing schemes such as SSA (Signal Stability Adaptive Routing) (12) are proposed. they classifies the links into 'strong' and 'weak' groups based on the measured signal strength, and a source node requests paths of only strong links first, and later allows weak links in on-demand routing schemes. Later, more sophisticated utilization of signal strength appears, e.g., Tickoo et al.(13) proposed a new routing metric RFC (Route Fragility Coefficient) based on relative speed of two terminals of a link estimated from the transition of wireless signal power observed. As another work in this category, Triviño-Cabrera et al. (14) proposed a path metric computed from signal strength based link metrics where, for instance, they try to choose the path which has the maximum value of minimum signal strength among all paths.

On the other side, since the amount of mobility is directly related to the link stability, the study of mobility metrics to measure the amount of performance effect coming from mobility. They measure several metrics in various mobility models and mobility levels to find proper mobility metrics. Yawut et al. (15) studied link duration (LD), link state change (LC: the number of link state change, e.g., link creation and deletion), and link stability (LS) defined as LD/LC under several scenarios. Qin et al. (16) studied node degree, average link duration and the number of link breakage that a node observed under several mobility models, and concluded that the number of link breakage observed can be used as a mobility metric.

Other than the studies shown above, several practical work proposed various routing metrics (used in the best path computation) which improves communication quality in mobile scenario. Zhao et al. proposed a routing metric called PARMA, which considers lower layer information such as physical layer link speed and estimated channel congestion. Karbaschi et al. (17) also proposed a cross-layer metric which considers link-quality and congestion. Cao et al. (18) proposed an integrated metric computed from hop counts, link load and power consumption. Guo et al. (20) proposed the method to predict three metrics of queuing delay, energy cost and link stability, and compute the optimal route heuristically by integrating those three metrics. Badis et al. presented QOLSR(21), a QoS extension of OLSR to operate multiple metric values, and provided so-called shortest-widest paths for best-effort traffic, which are the shortest paths computed under a kind of restriction on available bandwidth and delay.

2.2 On wireless mesh networks

Wireless Mesh Networks (WMNs) are regarded as a sort of ad hoc network in which nodes are stationary (2). WMNs are expected to be used as a infrastructure for wider areas where wired networks are hard to be built due to geometrical conditions or building cost.

In WMNs, the main factor of communication instability comes from the variation on wireless link quality and congestion since no mobility is assumed. Thus, much work tries to measure link quality as routing metrics through simple and low-cost measurement method.

The most widely used routing metric in WMN is ETX (Expected Transmission Count) (22) proposed by Couto et al., which is defined as the expected number of transmission required to deliver a packet. The ETX of a link is computed from the ratio of success transmissions which is measured by periodical probe messages sent on the link. ETX metric is the first routing metric which shows that routing metrics actually improve traffic throughput in MANETs against instability of wireless communications. Later, Draves et al. presented ETT (Expected Transmission Time) (23) which extends ETX by taking link speed into consideration as $\mathrm{ETT}= \mathrm{ETX}\frac{S}{B}$, where S is the packet size and B is the link bandwidth. WCETT was also proposed in the same paper (23), which takes bottleneck channel affection into account to compute path metrics under multi-channel environments. WCETT is computed as WCETT $= (1-\beta)\sum_{i=1}^{n}\mathrm{ETT}_i + \beta \max_{1\leq j\leq k} X_j$, where n is the number of hops on a routing path, k is the number of available channels for multi-radio operation, and $X_i = \sum_{\mathrm{Hop}\ i\ \mathrm{on\ channel}\ j}\mathrm{ETT}_i$. Note that $\beta\mathrm{max}_{1\leq j\leq k}X_j$ represents the level of bottleneck channel affection.

Note that WCETT is not a link metric but a path metric. ("Link metrics" here means the *additive metric* where a path metric is computed as the summation of all the link metrics included in the path.) Yang et al. (24) pointed out that path metrics such as WCETT may create loops even under static metric situation, and that the necessary and sufficient condition for path metrics to be statically loop-free is to satisfy the property called *isotonicity* introduced in the work of Sobrinho (25). Yang et al. also proposed a new path metric called MIC (Metric of Interference and Channel-switching) (24), which metric values can be decomposed to the isotonic metrics in a virtual network. This means that MIC is statically loop-free and is computed efficiently using the general shortest-path computation algorithms such as dijkstra's algorithm.

Note also that MIC (24) considers both intra-flow and inter-flow interference. In MIC, an interference-aware resource usage (IRU) of link l between nodes i and j using channel c is proposed as $\mathrm{IRU}_i = \mathrm{ETT}_i + N_{ij}(c)$, where $N_{ij}(c)$ is the number of nodes interfered with by node i and node j while using channel c. Also, to consider intra-flow interference, the channel switching cost (CSC) metric of node i is considered as $\mathrm{CSC}_i = w_1$ if incoming and outgoing hop of i in a routing path use different channels, and $\mathrm{CSC}_i = w_2$ otherwise, where $w_2 > w_1$. Then, MIC metric of a path p is represented by $MIC_p = \frac{1}{N\times\mathrm{min(ETT)}}\sum_{\mathrm{link}l\in p} IRU_l + \sum_{\mathrm{node}i\in p} CSC_i$, where N is the total number of nodes in the network.

There are several further proposals for this topic. For instance, Jin et al. (26) proposed a routing metric which considers the effect of hidden/exposed terminal interference. Also, Waharte et al. (27) proposed a routing metric which measures time to transmit a packet to its neighbor using a model based on the behavior of Wifi (IEEE 802.11) protocol.

From the other approach, several delay based metrics are proposed. Guo et al. presented a routing scheme called OLSR_NN (28) which based on prediction of delay measurement using the technique of neural networks. Murthy et al. presented LDAR (29), which is computed based on precise measurements of experienced delay in a node and their statistics as follows: $d_i = d_i^{\mathrm{process}} + d_i^{\mathrm{queue}} + d_i^{\mathrm{transmit}}$ where d_i^{process} is the processing delay in node i, d_i^{queue} is the queuing delay, d_i^{transmit} is the transmission delay of the 802.11 MAC protocol, and they all are computed based on experienced measurements. Other delay based routing metrics are summarized in (30).

Among those current proposals on dynamic metrics for ad hoc networks and WMNs, there is no proposal which cares temporary routing loops, which brings us a severe congestion and interference. To improve communication stability in ad hoc networks, preventing routing loops should be regarded as one of the important issues.

2.3 On loop-free routing

Now let us take a look at loop-free mechanisms in dynamic routing schemes. In fact, routing loops are well investigated for wired networks and deeply related with dynamic metrics.

Currently, deploying dynamic metrics in the Internet is unusual due to instability coming from changing paths. (It brings variation of bandwidth, delay, and so on which cause instability of communication throughput.) Dynamic metric schemes have been investigated, however, from the early stage of the Internet (31)(32). Their main object is to improve network performance, i.e., to improve network throughput and to avoid congestion, so that the effect of temporary routing loop formation is not considered.

The first loop-free routing scheme was presented as DUAL (33), which controls the sequence of routing tables to update when the topology (or metrics) changes in distance-vector routing schemes. Now DUAL is implemented in EIGRP routing protocol developed by Cisco systems. Later, Francois et al. (34) presented a loop-free link-state routing scheme from the similar strategy. They are always loop-free, however, since they require control messages for each topology change, the overhead is not sufficiently low for wireless ad hoc networks.

As another side, there are the studies on the safe (i.e., loop-free) range of metric modification (35)(36). They analyzed the case of changing at most one metric value simultaneously, and give an algorithm to compute the safe range of the value to be modified. They actually clarified an important property of routing loops, but it is not practical since they require a kind of central control not to allow changes of more than two metric values simultaneously.

To the best of our knowledge, LLD, which is proposed in this article, is the first loop-aware routing metric for wireless ad hoc networks. LLD is regarded as the method which follows the approach of "safe range" shown in (35)(36), and extends them into distributed network environment. Since this approach is low cost, LLD is suitable for wireless ad hoc networks.

3. Mechanisms and protocols of LLD

3.1 The dynamic metric of link stability

In this section, we describe the mechanisms of our metrics to extend proactive routing schemes such as OLSR (5). Normally in proactive routing, every node has a topology database of the network to compute the shortest path for each destination, where hop-counts are used as routing metrics.

Instead of hop-counts, we introduce dynamic link metrics which represent stability of links. In LLD routing scheme, it is considered that link stability is represented by the duration of time in which the link is in stable state, i.e., the longer a link stays stable, the smaller its metric becomes. Every link metric $\delta^t(l)$ of link l are managed by its directly connecting node with the following formula as long as the link is judged to be "stable."

$$\delta^t(l) = ab^t + c.$$

Here, t is time passed by since the link was born, and $b(0 < b < 1)$ is a ratio of metric decreased per unit time. We assume the same value of b for all links in a network. Then, $a + c$ is a initial metric and c is the value into which the metric finally converge. As for a and c, each link has its own values.

When a link is judged as "unstable," its metric is reset into the initial value, i.e., the value of time 0. This judgment (whether a link is unstable or not) should be done cautiously not to make a careless metric reset of stable links. For the judgment, we can use several metrics

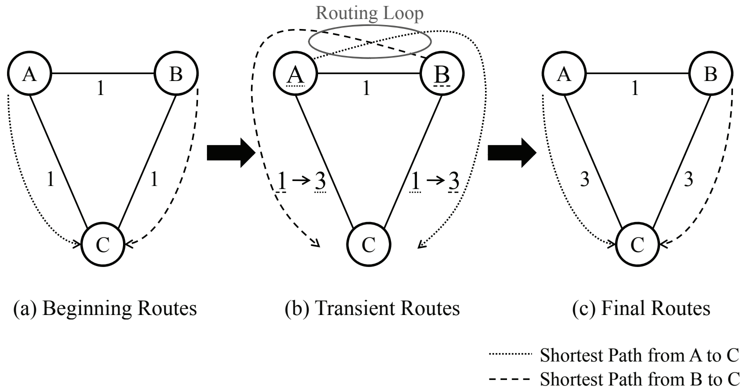


Fig. 1. An example of routing loops

proposed in the literature, e.g., transmission success ratio, physical signal power, and so on. However, since we now concentrate on theoretical aspect of loop-freeness, we do not discuss about this issue.

To say intuitively, a long-time stable link stays with a low metric whereas an unstable link (which metric is frequently reset) stays with high metric. As a result, traffic tends to use stable links from the nature of shortest-path computation so that communications become also stable.

Note that our metrics have to be synchronized periodically, i.e., the metrics of all links in every node's database are updated into the revised values periodically to prevent several different link metrics for a link used in route computation simultaneously. Namely, by periodical synchronization we prevent old metric values used in route computation, which usually occurs due to propagation delay of messages. We do this synchronization by flooding single synchronization message periodically.

3.2 Two issues on route stability in LLD

Routing loops and *path oscillation* are the two essential issues to be considered to realize stable communication in proactive routing. Especially, routing loops cause a severe instability problem so that it is strongly desired to be avoided. The loops occur when the topology (including metrics) of a network changes. During this period of time, two different routing tables computed from old and new topologies work together in the network so that a packet may loop among nodes if those routing tables are inconsistent. See Fig. 1 for instance. There are three nodes A, B and C in the network. The metrics of links (A,B), (A,C) and (B,C) are all 1 at the beginning so that the shortest paths from A and B to C go directly to C (Fig. 1(a)). Assume that the metrics of (A,C) and (B,C) change to 3 simultaneously. It is natural that finally the shortest paths from A and C is the same as the beginning state (shown in Fig.1(c)). In the transient state, however, routing loops possibly form due to propagation delay, where A regards the metrics of (A,C) and (B,C) as 3 and 1, respectively, while B does those as 1 and 3, respectively. This state is shown in Fig. 1(b), where the dotted and broken underlines indicate the metrics that A and B know. Such routing loops frequently occur in ad hoc networks and cause severe congestion and disruption of communications due to heavy packet loss.

On the other hand, the path oscillation problem is the problem that the path from a source to a destination changes frequently among several candidate paths. This also causes instability of communications. Both of those two problems (I.e., routing loops and path oscillation) are discussed in this article in the following sections.

Note that, when we treat loop-freeness, we assume the situation that the two events, i.e., (i) reset of metrics, and (ii) addition/deletion of links, do not occur. Note that this assumption is reasonable because those two events do not cause loops with high probability in typical situations. Considering that a link cause loops only when it is used in some shortest paths, event (i) causes loops only when the metric of the reset link has been low enough (otherwise, the link will not be used in shortest paths), and if the metric is low, the probability of resetting metric is considered also low. As for (ii), newly added links have so high metric values that they will not be used in shortest paths. Also, deleted links are supposed to be unstable with high probability so that their metrics is considered to be high and so will not be used in shortest paths.

As discussed above, as long as stable links keep stable with high probability, those two events generate loops with low probability. This indicates that loop-freeness in other (normal) situations is essential for stability of communications.

From the discussions above, the characteristic of the network to which our scheme is suitably applied is that: (1) stable links keeps stable with high probability and (2) there is sufficient number of stable links to construct stable-paths all over the network. Actually this situation is too realistic, but we have several networks which have similar characteristics, e.g., disaster networks which is expected to work as a simple infrastructure, or mixture network of wired and wireless links.

3.3 Synchronization mechanisms

As an extension to a proactive routing scheme, we have to support periodical update and synchronization of every link metric. The requirement of LLD for synchronization mechanism is that, at any point of time, every node uses the same metric value for a link to compute its routing table. (It is only when the time increases by synchronization message that two consequent metric values may be used simultaneously in a network.) Note that loop formation and path oscillation are easily affected by numerical errors of metric values so that the synchronization should be done accurately, i.e., metrics of different time (old and new) should not be used together in a path computation process. Also, message overhead should be low enough due to reduce communication overhead of wireless networks.

Further, for the synchronization mechanisms to be practical, it should be robust against several irregular cases. Specifically, it should endure (i) (not frequent) loss of messages, (ii) network division (including link/node deletion), and (iii) network integration (including link/node addition).

We need some consideration to meet this requirement. Of course, to flood every new metric value periodically over the network is not allowed since message overhead is too heavy for wireless ad hoc networks. Instead, we may take a method which adds fields of a, c and the generated time of links into link advertisement messages, and each node decreases all link metrics periodically according to each node's attached clock. This mechanism, however, have to assume accurate clock for all nodes. Moreover, joining even single wrong-time node may confuse routing. Alternatively, we can also flood a single "sync" message including sequence number into the network at each synchronization time. If we advertise the generated time of links in sequence number representation in link advertisement messages, every node is able to compute all links' current metric. There is, however, a problem when joining two different networks with different sequence numbers.

Based on the above consideration, our synchronization mechanism is to send periodical sync message (with sequence numbers) which reaches only neighbor node. Also, we prepare fields of a, c and the generated time of links in link advertisement messages, where the link generated time should be expressed by the sequence number based on the sending node's sync messages. Under the behavior above, each node learns the timing to send sync messages so that the timing to send sync messages is synchronized in a network little by little. Thus finally all nodes send periodical sync messages almost simultaneously although some random factor should be considered to avoid interference among messages. Then, if each node changes the metrics of all links in its database every time the node send a synchronization message, the synchronization is done without problems.

This mechanism is able to keep correct metrics even after loss of control packets. There is no problem in both case of division and join of networks, although join process requires synchronization time to converge. Also, additional message overhead is very low if we use existing messages such as hello messages as sync messages in LLD. Note that this mechanism is only an example to do synchronization. But it shows that we can perform required synchronization of LLD in low cost.

4. Theoretical analysis for loop-freeness

4.1 Formulation of loop-freeness

Here we give a condition of synchronization time interval to be loop-free. We start with formulation of loop-freeness.

Let $G = \{V, E\}$ be a network, where V is a set of nodes and E is a set of links. For a pair of nodes $n_1, n_2 \in V$, we call they are *adjacent* if $(n_1, n_2) \in E$. A sequence of nodes $p = (n_1, n_2, \ldots, n_m)$ where $(n_k, n_{k+1}) \in E, k = 1, 2, \ldots, m-1$ are called *path*. The *metric* of link l at time t is denoted by $\delta^t(l) = a_l b^t + c_l$, where a, b and c is a real value and $0 < b < 1$. Note that the value of b must be common in a network but a and c is not. The metric of path p at time t is denoted in similar fashion by $\delta^t(p) = a_p b^t + c_p$, where $a_p = \sum_{l \in p} a_l$ and $c_p = \sum_{l \in p} c_l$. (Note that, theoretically, we can assume all links are generated simultaneously at time 0 without loss of generality. We have only to adjust the value of a_l to do so.) For a pair of nodes $s, d \in V$, the *shortest path* from s to d at time t is the path p that has the shortest value of $\delta^t(p)$.

Now we give the condition of loop-freeness. Routing loops are created only when the composition of the shortest paths computed from two succeeding states of metrics (i.e., before and after metrics change) creates cycles. Formally, let $D_1 = (V, E_1)$ and $D_2 = (V, E_2)$ be the DAGs generated from all the edges of the shortest paths computed from two succeeding states of metrics. (Note that since we consider equal-metric paths, shortest paths do not always form a tree but a DAG(Directed Acyclic Graph).) Then, it is clear that the sufficient condition to guarantee creation of no routing loops is as follows:

Proposition 1 *A sufficient condition to guarantee loop-freeness is that $D = D_1 \cup D_2 = (V, E_1 \cup E_2)$ has no cycle.*

4.2 Behavior of shortest path transition

A lemma is presented before the main result on loop-freeness. The following lemma shows an interesting property of shortest-path behavior under the metric function $\delta^t(l) = ab^t + c$.

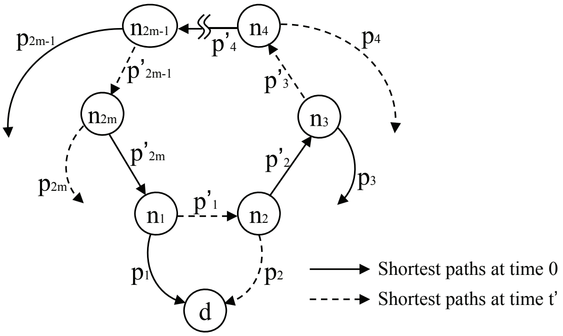


Fig. 2. The General Case of Routing Loops

Lemma 1 *Let p and p' be two paths departing from s to d where p is the shortest path at time* 0*. Then, $t' > 0$ exists such that p' is the shortest path from s to d at time t', if and only if $c_{p'} < c_p$. Further in this case, the length of two paths are reversed only once, i.e.,*

$$\delta^t(p) < \delta^t(p') \quad (0 \leq t < t')$$

$$\delta^t(p) = \delta^t(p') \quad (t = t')$$

$$\delta^t(p) > \delta^t(p') \quad (t > t')$$

proof: Assume that $c_{p'} < c_p$. Then, at time $t = \infty$, p' becomes the shortest path since $\lim_{t\to\infty}\delta^t(p')(= c_{p'}) < \lim_{t\to\infty}\delta^t(p)(= c_p)$. Conversely, assume that $c_{p'} \geq c_p$. If $a_p \leq a_{p'}$, then $\delta^t(p)(= a_p b^t + c_p) \leq \delta^t(p')(= a_{p'} b^t + c_{p'})$ stands for arbitrary t. If $a_p > a_{p'}$, then $c_{p'} - c_p > a_p - a_{p'} > (a_p - a_{p'})b^t = a_p b^t + a_{p'} b^t$ stands from $\delta^0(p) < \delta^t(p')$. This formula leads $\delta^t(p) < \delta^t(p')$. Thus p' cannot be the shortest path for arbitrary t. As above, $c_{p'} < c_p$ is a necessary and sufficient condition for switching the shortest path.

Next, we show that the shortest paths switch only once. Consider t' such that $\delta^{t'}(p')(= a_{p'} b^{t'} + c_{p'}) = \delta^{t'}(p)(= a_p b^{t'} + c_p)$. This leads $c_p - c_{p'} = (a_{p'} - a_p)b^{t'}$. Since $0 < b < 1$ and $0 < c_p - c_{p'} < a_{p'} - a_p$, there is only one value t' which satisfies this formula. □

4.3 A condition for loop-freeness

Now we show the condition of loop-freeness. Fig. 2 shows the general situation of a routing loop created with paths for destination d. Without loss of generality, we assume this network has been synchronized at time 0 and the loop is generated in the next synchronization at time t'. The loop consists of both links in the shortest-path DAGs at time 0 and t'. So we can select a node n_1 from the loop, from which a link of time t' starts and at which a link of time 0 ends. Then, starting from n_1, we can divide the loop into a sequence of paths $p'_1, p'_2, \ldots, p'_{2m-1}, p'_{2m}$ where p'_{2k-1} and $p'_{2k}(0 < k \leq m)$ consist of the links of time t' and 0, respectively. Let the starting nodes of p'_{2k} and p'_{2k-1} be n_{2k} and n_{2k-1}, respectively. Let the shortest path from n_{2k-1} to d at time 0 be p_{2k-1} and the shortest path from n_{2k} to d at time t' be p_{2k}. In this situation, the next statement stands:

Theorem 1 *Assume that a network synchronizes at time* 0 *and subsequently synchronizes at time* t'*. Then, a sufficient condition of* t' *to be loop-free is the following:*

$$t' < \frac{\log \frac{\sum_{k=1}^{m}(c_{2k-1}-c'_{2k-1}-c_{2k})}{\sum_{k=1}^{m}(\delta^0(p'_{2k-1})+\delta^0(p'_{2k})+c_{2k-1}-c'_{2k-1}-c_{2k})}}{\log b}$$

proof: The following formula stands at n_{2k-1}.

$$\delta^t(p_{2k-1}) > \delta^t(p'_{2k-1}) + \delta^t p_{2k} \quad (1)$$

For a path p, its metric is denoted by $\delta^t(p) = ab^t + c = (\delta^0(p) - c)b^t + c)$, Hence,

$$\delta^0(p'_{2k-1}) < \delta^0(p_{2k-1}) - \delta^0(p_{2k}) + \frac{(1-b^t)}{b^t}(c_{2k-1} - c'_{2k-1} - c_{2k}) \quad (2)$$

Similarly, the following formula stands at n_{2k}.

$$\delta^0(p'_{2k}) < \delta^0(p_{2k}) - \delta^0(p_{2k+1}) \quad (3)$$

Summing formulas (2) and (3) for $n_1, n_2, \ldots, n_{2m}$, we obtain

$$\sum_{k=1}^{m}(\delta^0(p'_{2k-1}) + \delta^0(p'_{2k})) < \frac{1-b^t}{b^t}\sum_{k=1}^{m}(c_{2k-1} - c'_{2k-1} - c_{2k}) \quad (4)$$

Since formula (4) is a necessary condition of creating loops, the following formula is a sufficient condition to be loop-free.

$$\sum_{k=1}^{m}(\delta^0(p'_{2k-1}) + \delta^0(p'_{2k})) \geq \frac{1-b^t}{b^t}\sum_{k=1}^{m}(c_{2k-1} - c'_{2k-1} - c_{2k}) \quad (5)$$

Transforming the formula (5) in respect of b^t, we obtain

$$b^{t'} > \frac{\sum_{k=1}^{m}(c_{2k-1} - c'_{2k-1} - c_{2k})}{\sum_{k=1}^{m}(\delta^0(p'_{2k-1}) + \delta^0(p'_{2k}) + c_{2k-1} - c'_{2k-1} - c_{2k})} \quad (6)$$

For $0 < b < 1$ and $0 <$ the right side of (6) < 1, the following conclusion is obtained.

$$t' < \frac{\log \frac{\sum_{k=1}^{m} c_{2k-1}-c'_{2k-1}-c_{2k}}{\sum_{k=1}^{m} \delta^0(p'_{2k-1})+\delta^0(p'_{2k})+c_{2k-1}-c'_{2k-1}-c_{2k}}}{\log b}$$

□

Now we discuss the meaning of this condition. For instance, suppose the situation where $a = 1000$ for all links and metrics converge to $c + 0.5$ when 1 week (10080 minutes) past. In this case, $b = 0.9992462$ if the unit of time is "minutes." Also, suppose the diameter of the network, which we define in this paper as the maximum hop-count of shortest paths, is at most 20. And c for each link takes a value between 10 and 50. In this situation, we consider the maximum value of the following K:

$$K = \frac{\sum_{k=1}^{m}(c_{2k-1} - c'_{2k-1} - c_{2k})}{\sum_{k=1}^{m}(\delta^0(p'_{2k-1}) + \delta^0(p'_{2k}) + c_{2k-1} - c'_{2k-1} - c_{2k})}$$

Here, $c_{2k-1} - c'_{2k-1} - c_{2k} > 0$ from *Lemma 1*, and $\delta^0(p'_{2k-1}) + \delta^0(p'_{2k}) > 0$. Thus K takes the maximum value when $\delta^0(p'_{2k-1}) + \delta^0(p'_{2k-1})$ takes the minimum and $c_{2k-1} - c'_{2k-1} - c_{2k}$ takes the maximum value. In this situation, the minimum values of $\delta^0(p'_{2k-1})$ and $\delta^0(p'_{2k-1})$ are both 10, and the maximum value of $c_{2k-1} - c'_{2k-1} - c_{2k}$ is $50 * 20 - 10 - 10 = 980$, hence,

$$K \leq \frac{(980m)}{(20m) + (980m)} = 0.98$$

Therefore, since $0 < b < 1$ and $0 < K < 1$,

$$t' < \frac{\log 0.98}{\log 0.9992462} = 26.79104797 \leq \frac{\log K}{\log b}$$

As shown above, the proposed scheme is loop-free if the synchronization interval is less than 26.7 minutes. For reference, if every link takes the same value of $c = 10$, the upper bound of t to be loop-free is 139.71982 minutes. Also, if we consider faster convergence such as 1 day instead of 1 week, the upper bound is merely $\frac{1}{7}$ of the above. Further, in this situation, if we always increment t by 1 in every synchronization, the condition $b > 0.9$ guarantees loop-freeness.

5. Preventing path oscillation

5.1 Rounding errors and path oscillation

In the theoretical analysis, we can assume that metrics are real values. Under this assumption, *Lemma 1* guarantees that path oscillation does not occur. However, in practice, values should be represented in computers by a finite length of bits. In fact, in our scheme, the rounding errors coming from this can cause severe path oscillation. In this section we describe the problem and solution for it.

Path oscillation between two paths occurs when the metrics of those two paths are very close. If we consider the case of using integer values as metrics, the range of possible rounding error per link is from -0.5 to $+0.5$. If a path has k links, its rounding error is from $-0.5k$ to $+0.5k$. Therefore, if metrics of two paths are closer than the each other's error range, frequent path oscillation will occur with high possibility.

As a solution, we use floating-point numbers (38) to represent metrics. Floating-point numbers have a useful property that when a value is smaller, the rounding error also becomes smaller. Since, in our metrics, the difference of metrics between two paths goes smaller as time passes, this property is so convenient. As we present later, floating-point numbers truly suppress the oscillation.

Here, note that there is a small problem. The variable c in the metric formula would have an integer value in many cases so that the range of rounding errors stops to go smaller. As a result, oscillation may occur when the path metrics become to be very small values. In order to suppress this kind of oscillation, we enforce to converge metrics into c when ab^t becomes sufficiently small, e.g., such as 0.5.

5.2 Simulation results on path oscillation

To measure how many times paths oscillate, we prepare the simulation scenario in which oscillation will likely to occur the most frequently. We suppose two paths which have the same number of links, have the same source and the destination node, have almost the same metrics, and have the same metric to converge. Fig. 3 is a snapshot of an example situation.

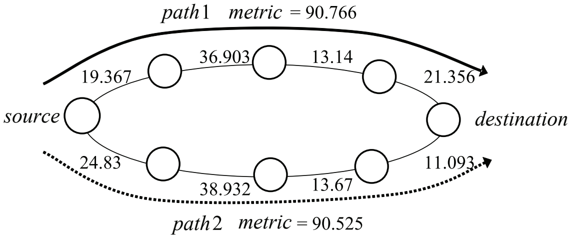


Fig. 3. Simulation Scenario

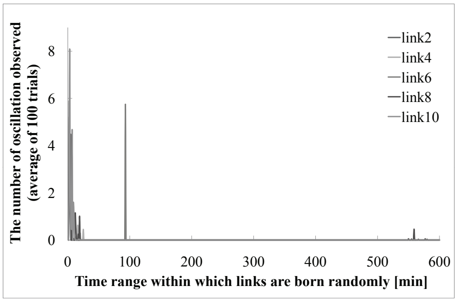


Fig. 4. The Result in Case of Using Single-precision Floating Points

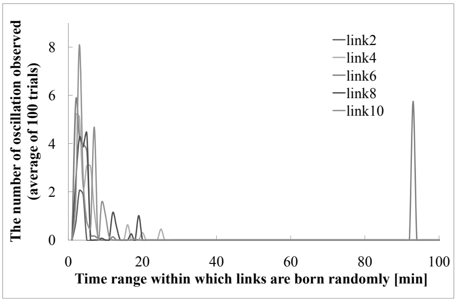


Fig. 5. Zooming Fig. 4 of first 100 minutes

Specifically, we set $a = 1000$ and $c = 20$ for all links, and set $b \simeq 0.9992462$. Namely, the initial metric of every link is 1020 and it takes a week (10080 minutes) to converge into 20.5. As mentioned previously, a metric is enforced to be 20 if the metric become less than 20.5. We assume SyncInterval is 1 (minute) so that path computation is invoked every 1 minute. This is far severe condition than usual. Note that if the result is safe in this condition, every other integer values of SyncInterval are guaranteed to be safe. We test the cases of 2 to 10 links included in each of the two paths, and for each case we generate the links at almost the same time, i.e., we randomly generate links within the time range of 1 to 30 minutes. Every case is tested 100 times and we measure the average number of oscillation occurred.

In Fig. 4, we show the result of the case that metrics are represented by single-precision floating point (32 bits) defined in (38). Although we observe some accidental oscillations (e.g., around 90 minutes of the time range), the number is totally small and practically permissible. There is a trend that the number of oscillations arises when the time range is small, but has no relation with the number of links. As another result, in case of applying double-precision floating point (64bits), we do not observe any oscillation at all. For reference, when the metric to converge is different between two paths even by 1, no oscillation is observed with single-precision floating point (32bits), either. We conclude that floating-point representation can suppress the oscillation.

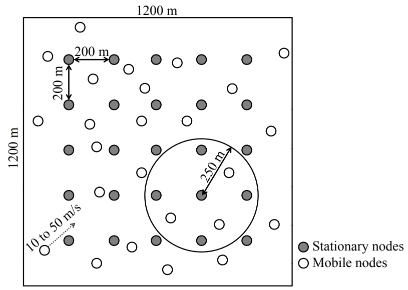


Fig. 6. Traffic Simulation Scenario

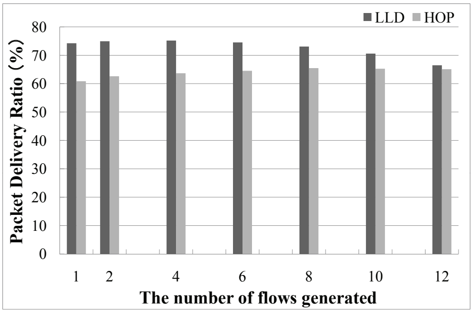


Fig. 7. Packet Delivery Ratio

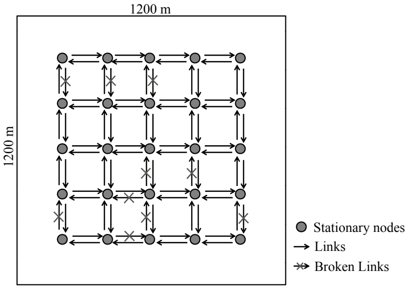


Fig. 8. Link State in 8 Flows Scenario

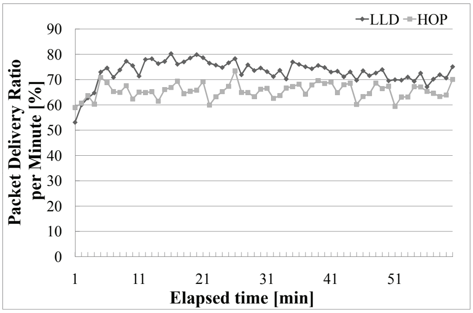


Fig. 9. Delivery Ratio per Minute

6. Traffic simulation

6.1 Simulation scenario

We compared the performance of LLD with conventional hop-count routing (HOP) through traffic simulation using NS-2 ver.2.29(39). We use UM-OLSR ver.0.8.8(40) for the base OLSR module and modify it to implement LLD. Note that the synchronization mechanism is not actually implemented, i.e., time synchronization is done using global variables in C language. Namely we assume that the synchronization mechanism works ideally.

Simulation scenario is illustrated in Fig. 6. To measure the communication performance in the network with both stable links and unstable links, we prepare 25 stationary nodes and 25 mobile nodes i.e., the links between two stationary nodes are regarded as stable, and others are relatively unstable. The field size is 1200m x 1200m and the stationary nodes are placed to form 5 x 5 grid where every interval of adjacent nodes is 200m and the communication range is set as 250m. The moving pattern of mobile nodes is generated by BonnMotion(41) to follow Random Way Point (RWP) Mobility Model. The moving speed is randomly determined between 10.0 m/s and 50.0 m/s. Those nodes communicate with each other via Wifi (IEEE 802.11) of 2Mbps bandwidth. Total simulation time is set as 1 hour.

LLD parameter is set as $a = 1000$, $b = 0.9$ and $c = 1$ for all nodes. Synchronization time interval is set as 1 minute so that every node updates all the link metrics every 1 minute and recompute

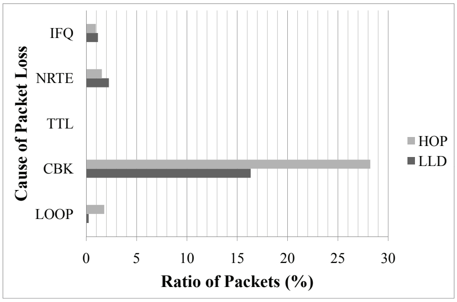


Fig. 10. Packet Loss Specification (4 Flows)

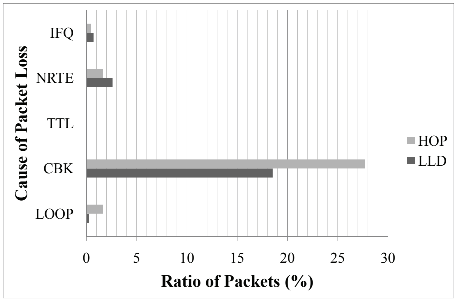


Fig. 11. Packet Loss Specification (8 Flows)

its routing table. Note that we set $b = 0.9$, which is the minimum value of b to guarantee loop-freeness in the network where synchronization time interval is 1 min, maximum hop count among all possible paths (the diameter of the network) is 20, and c takes a common value over the network. For the loop-free condition of b, see Section 4.

In our traffic scenario, several 10kbps CBR (Constant Bit Rate) flows with packet size of 512 bytes are generated between two randomly selected mobile nodes. We compare the communication performance between LLD and HOP by taking the average of 4 trials under variation of the number of flows generated.

6.2 Results of traffic simulation

Fig. 7 shows the packet delivery ratio for each number of flows 1, 2, 4, 6, 8, 10, and 12. When the number of flow is low enough (i.e., 1, 2, and 4), LLD keeps more than 10% higher delivery ratio than HOP. This is because LLD tends to use stable links (which connects two stationary nodes) to support stable communications, resulting in low probability to meet unavailable links. However, as the number of flows goes higher the difference of the performance goes smaller. This result comes from the property of LLD that traffic tends to be concentrated on specific stable links, which brings link breakage to increase loss of packets. For this link situation, see Fig. 8 which shows the state of 8 flows LLD scenario at 1800 second. As is seen in this figure, always several links are broken in such congested state.

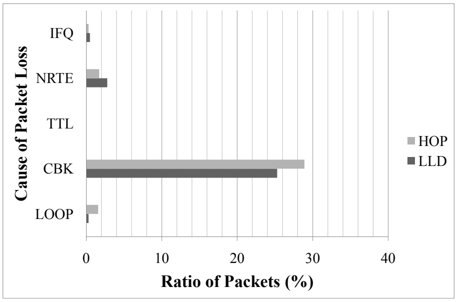


Fig. 12. Packet Loss Specification (12 Flows)

Fig. 9 shows the transition of packet delivery ratio with time course in the 8 flows scenario. Although in the first few minutes the difference are hardly seen since there are not enough difference in metrics between stable and unstable links, after that differences are constantly seen between LLD and HOP. It is found that LLD delivery ratio gradually decreased little by little, which we infer is the effect comes from the breakage of stable links.

Fig. 10-12 shows the specification of drop packets for possible drop reasons in the 4, 8 and 12 flows scenarios. Each loss ratio is represented as the value out of all transmitted packets in the 1-hour scenario. There are five reasons where IFQ is the loss coming from sending queue overflow, NRTE is the loss of no route found in the routing table, TTL is the loss from expiration of TTL (time to live) counter, CBK is the loss from radio interference, and LOOP is the dropped packets when they return to their source nodes.

From the results, we find that the ratio of looping loss (LOOP) is significantly decreased in LLD in comparison with HOP. Note that there are still a little looping packets in LLD although we use the value $b = 0.9$ which guarantees loop-freeness; the reason of the looping is considered link breakage due to radio interference. Note that even in the 4 flow (low load) scenario considerable packets are lost by radio interference (CBK).

The main difference between LLD and HOP is found in CBK. This difference includes not only normal radio interference but also that generated by looping packets. Since it is natural that normal interference between LLD and HOP will not differ considerably, the difference surely comes from looping packets. Consequently, we conclude that LLD improves both stability and throughput of communications by decreasing looping packets.

Incidentally, we found that the difference of CBK between LLD and HOP goes closer as the number of flows (traffic load) increases. This implies that LLD is not good at traffic capacity since LLD tries to concentrate traffic on only stable links. The load balancing performance would be one of the drawbacks of LLD.

7. Concluding remarks

We presented a new dynamic link stability metrics which achieves loop-freeness throughout dynamic metric transition. We gave a theoretical analysis on the condition of loop-freeness, and through simulations we presented that the instability coming from path oscillation can be suppressed by applying floating-point representation of metrics. Further, we presented a traffic simulation result in which LLD improved communication stability unless link load is too high. I wish our new strategy of loop-free routing suggests a new viewpoint of mobility metric in proactive routing schemes.

8. Acknowledgment

The author would like to express sincere appreciation to Ayaka Kuroki and Takahiro Iida for the help for simulation work of this article. This research is partly supported by Strategic Information and Communications R&D Promotion Programme (SCOPE), the Ministry of Internal Affairs and Communications, Japan.

9. References

[1] Yoshihiro, T. (2009). Loop-free Link Stability Metrics for Proactive Routing in Wireless Ad Hoc Networks, *Proceedings of IEEE ICC2009*, pp.1–5.

[2] Akyildiz, I.F. & Wang, X. (2009), *Wireless Mesh Networks*, John Wiley & Sons Ltd Publication.

[3] Perkins, C., Belding-Royer, E. & Das, S. (2003), Ad Hoc On-demand Distance Vector (AODV) Routing, *IETF Request For Comments (RFC) 3561*, IETF.

[4] Johnson, D., Hu, Y. & Malts, D. (2007). The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4, *IETF Request For Comments (RFC) 4728*, IETF.

[5] Clausen, T. & Jacquet, P. (2003). Optimized Link State Routing Protocol (OLSR), *IETF Request For Comments (RFC 3626)*, IETF.

[6] Ogier, R., Templin, F. & Lewis, M., (2004). Topology Dissemination Based on Reverse-Path Forwarding (TBRPF), *IETF Request For Comments (RFC 3684)*, IETF.

[7] Camp, T., Boleng, J. & Davies, V. (2002). A Survey of Mobility Models for Ad Hoc Network Research, *Wireless Communications and Mobile Computing*, Vol.2(Issue.5), pp.483–502.

[8] Lenders, V., Wagner, J., Heimlicher S., May M., Plattner B. (2008). An Empirical Study of the Impact of Mobility on Link Failures in an 802.11 Ad Hoc Network, *IEEE Wireless Communications*, Vol. 15(No. 6), pp. 16-21.

[9] Gerhartz, M., Waal, C., Frank, M. & Martini, P. (2002). Link Stability in Mobile Wireless Ad Hoc Networks, *Proceedings of IEEE Local Computer Networks LCN*.

[10] Cheng, Z. & Heinzelman, W.B. (2004). Exploring Long Lifetime Routing (LLR) in ad hoc Networks, *Proceedings of 7th ACM International Symposium on Modeling, Analysis and imulation of Wireless and Mobile Systems*, pp.203–210.

[11] Zhao, M. & Wang, W. (2007). The impacts of radio channels and node mobility on link statistics in mobile ad hoc networks, *Proceedings of IEEE Global Telecommunications Conference (Globecom2007)*, No.1, pp.1205-1209.

[12] Dube, R., Rais, C.D., Wang, K.Y. & Tipathi, S.K. (1997). Signal Stability based Adaptive Routing (SSA) for Ad-Hoc Mobile Networks, *IEEE Personal Communications*, Vol.4(No.1), pp.36-45.

[13] Tickoo, O., Raghunath, S. & Kalyanaraman, S. (2003). Route Fragility: A Novel Metric for Route Selection in Mobile Ad Hoc Networks, *Proceedings of IEEE ICON03*, pp.537–542.

[14] Triviño-Cabrera, A., Nieves-Pérez, I., Casilari, E. & González-Cañete, F.J. (2006). Ad Hoc Routing Based on the Stability of Routes, *Proceedings of the 4th ACM International Workshop on Mobility Management and Wireless Access*, pp.100–103.

[15] Yawut, C., Paillassa, B. & Dhaou, R. (2007). On Metrics for Mobility Oriented Self Adaptive Protocols, *Proceedings of in Wireless and Mobile Communications 2007*

(ICWMC2007).

[16] Qin, L. & Kunz, T. (2006). Mobility Metrics to Enable Adaptive Routing in MANET, *Proceedings of IEEE Wireless and Mobile Computing, Networking and Communications 2006 (WiMobapos2006)*, pp.1–8.

[17] Karbaschi, G. & Fladenmuller, A. (2005). A Link-Quality and Congestion-aware Cross Layer Metric for Multi-hop Wireless Routing, *Proceedings of IEEE International Conference on Mobile Adhoc and Sensor Systems*.

[18] Cao, L. & Dahlberg, T. (2006). Path Cost Metrics for Multi-hop Network Routing, *Proceedings of IEEE International Conference on Performance, Computing and Communications 2006 (IPCCC2006)*.

[19] Zhao, S., Wu, Z., Acharya, A., & Raychaudhuri, D. (2005). PARMA: A PHY/MAC Aware Routing Metric for Ad-Hoc Wireless Networks with Multi-Rate Radios, *Proceedings of IEEE WoWMoM2005*, pp.286–292.

[20] Guo, Z. & Malakooti, B. (2007). Predictive Multiple Metrics in Proactive Mobile Ad Hoc Network Routing, *Proceedings of LCN2007*, 755–762.

[21] Badis, H. & Agha, k. A. (2005). QOLSR, QoS Routing for Ad Hoc Wireless Networks using OLSR, *European Transactions on Telecommunications*, Vol.16 (No.5), pp.427–442.

[22] De Couto, D., Aguayo, D., Bicket, J. & Morris, R. (2003). A High-Throughput Path Metric for Multi-Hop Wireless Routing, *Proceeding of ACM Annual International Conference on Mobile Computing and Networks (MOBICOM2003)*, pp.134–146.

[23] Draves, R., Padhye, J. & Zill, B. (2004). Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks, *Proceedings of ACM Annual International Conference on Mobile Computing and Networks (MOBICOM2004)*, pp.114–128.

[24] Yang, Y., Wang. J. & Kravets, R. (2005). Interference-aware Load Balancing for Multihop Wireless Networks, *In Technical Report UIUCDCS-R-2005-2526, Department of Computer Science, University of Illinois*.

[25] Sobrinho, J. L. (2003). Network Routing with Path Vector Protocols: Theory and Applications, *Proceedings of the ACM 2003 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM2003)*, pp.49–60.

[26] Jin, S., & Mase K., A Hidden-Exposed Terminal Interference Aware Routing Metric for Multi-Radio and Multi-Rate Wireless Mesh Networks, *IEICE Transactions on Communications*, Vol.92-B(No.4), pp.709–716.

[27] Waharte, S., Ishibashi, B., Boutaba, R. & Meddour, D.E. (2009). Design and Performance Evaluation of IAR: Interference Aware Routing Metric for Wireless Mesh Networks, *Mobile Networking Applications*, Vol.14(No.5), pp.649–660.

[28] Guo, Z. & Malakooti, B. (2007). Predictive delay metric for OLSR using neural networks, *Proceedings of the 3rd international conference on Wireless Internet*.

[29] Murthy, S., Hegde, P. & Sen, A. (2009). Design of a Delay-based Routing Protocol for Multi-rate Multi-hop Mobile Ad Hoc Networks, *Proceeding of IEEE ICC2009*.

[30] Zaki, S.M., Ngadi, M.A. & Razak S.A. (2009), A Review of Delay Aware Routing Protocols in MANET, *Computer Science Letters*, Vol.1(No.1).

[31] Khanna, A. & Zinky, J. (1989). The Revised ARPANET Routing Metric, *ACM SIGCOMM Computer Communication Review*, Vol. 19 (Issue 4).

[32] Chen, J., Druschel, P. & Subramanian, D. (1999). A New Approach to Routing With Dynamic Metrics, *Proceedings of IEEE INFOCOM '99*, pp.661–670.

[33] Garcia-Luna-Aceves, J. J. (1993), Loop-free Routing using Diffusing Computations, *IEEE/ACM Transactions on Networking*, Vol.1 (No.1), 130–141.

[34] Francois, P. & Bonaventure, O. (2007), Avoiding Transient Loops During the Convergence of Link-state Routing Protocols, *IEEE/ACM Transactions on Networking*, Vol. 15 (No. 6), 1280–1932.

[35] Ito, H., Iwama, K., Okabe, Y. & Yoshihiro, T. (2003). Avoiding Routing Loops on the Internet, *Theory of Computing Systems*, Vol.36, 597–609.

[36] Francois, P., Shand, M. & Bonaventure, O. (2007). Disruption-free Topology Reconfiguration in OSPF Networks, *Proceedings of IEEE INFOCOM2007*.

[37] Speakman L., Owada Y., Mase K., Looping in OLSRv2 in Mobile Ad-Hoc Networks, Loop Suppression and Loop Correction, *IEICE Transactions on Communications*, Vol.E92-B(No.4), pp.1210–1221.

[38] IEEE. (1985). IEEE Standard for Binary Floating Point Arithmetic, *ANSI/IEEE* Std 754-1985.

[39] The Network Simulator NS-2 (2010). http://www.isi.edu/nsnam/ns/.

[40] MASIMUM (2010). http://masimum.dif.um.es/.

[41] BonnMotion (2010).
http://web.informatik.uni-bonn.de/IV/Mitarbeiter/dewaal/BonnMotion/.

19

Stability Oriented Routing in Mobile Ad-Hoc Networks Based on Simple Automatons

Miklós Molnár[1] and Raymond Marie[2]
[1]*University of Montpellier 2, IUT / LIRMM*
[2]*University of Rennes 1 / IRISA*
France

1. Introduction

With recent performance advancements in wireless communications technologies, advanced mobile wireless networking realizations are expected to be seen shortly. Meshed wireless network infrastructures and ad hoc wireless networks are growing to facilitate services for mobility-free networks users everywhere and anytime. Mobile ad hoc networks (MANETs) consist of a collection of independent mobile nodes that can communicate to each other via radio and that are free to move and even to switch off arbitrarily. These networks can be installed in isolation, or can be connected to other networks.

The ad hoc networks are often multihop networks. Some neighboring mobile nodes that are in communication range of each other can directly communicate, while others needs to use one or more intermediate router nodes to communicate with far nodes. So, a reliable routing functionality is fundamental in these networks.

Nodes and links can appear and disappear spontaneously due to the behavior of users (who may turn off/on a node), the depletion of battery power, etc. As nodes are free to move arbitrarily and may disappear, such networks can be characterized by a dynamic, often rapidly-changing, random and multihop topology. This ad hoc topology may also change when the nodes adjust their transmission and reception parameters (Corson & Macker, 1999). The continuous presence of these phenomenon implies a very dynamic and randomly evolving topology in both time and space.

Since wireless ad-hoc networks with mobile nodes have not stable topology, the classical network functions such as the routing are difficult to realize. The router nodes and the links between them are not stable and can appear and disappear randomly. So, classic routing algorithms cannot be used successfully. Many special reactive, proactive and hybrid routing algorithms have been proposed to solve the data transmission in multihop ad hoc networks. The principal propositions are cost, delay or energy oriented. However, new approaches should be used which deals with these dynamic changes. In the case of reactive routing, the proposed route to satisfy a new request can be volatile and so, the communication concerned by it may be frequently interrupted and new routes may be computed. As an example, we can cited AODV, which is a well known reactive on-demand routing protocol proposed in (Perkins & Royer, 1999). The dynamic source routing (DSR) also proposes the dynamic allocation of routes (Johnson & Maltz, 1996). Trivially, the establishment of the new routes involves additional latency and intensive communication for control purposes. When a proactive routing algorithm is applied, the topology changes must be broadcasted in the whole network. The topology information is first monitored then periodically distributed and stored in routers. A typical example is the protocol DSDV (Perkins & Bhagwat, 1994) and an optimized control flooding based solution can be found in OLSR (Jacquet et al., 2001). The update of the topology information in proactive case is not immediate and cannot be executed permanently. The broadcasted control messages pass through the randomly congestioned network and achieve the different nodes with different random delays. In the case of both the reactive and proactive routing, the route information may be outworn and may not correspond to the routing objectives. So, the routing is based on uncertain and not necessary adequate information. In both cases, this information can be obsolete at the moment of its utilization. A wrong routing implies packet losses or additional delay. A route breaking issue from an expired information initiates expensive mechanisms to find a new route. Moreover, topology and/or route maintenance involves important control message overhead.

An alternative model for routing can be a solution that try to minimize the route changes and to estimate and to control the validity of the selected routes. A pertinent model should allow to handle uncertainties on routing information caused by unpredictable changes and information propagation delay. We will see that these kinds of uncertainties can be taken into account in some probabilistic routing models. To avoid frequent route requests and volatile routes due to uncertain information, the objective of the routing should correspond to the route stability. For instance, the goal of the route selection may be the selection of routes assuring a good and controlled longevity. Since the description and the handling of network elements which change randomly needs random variables as models, the route computation can also be based on random variables and becomes probabilistic routing. This chapter focuses on modeling the resilience of these information for ad hoc networks where topology information is uncertain. Our basic model is based on a dynamic graph where the existence of the nodes and of the communication capability between them are modeled by simple two state automaton where the transitions are initiated by random events. We present several methods to model the evaluation of the state of the network elements and of the relation between them and we exhibit closed form expressions for the existence probabilities of these network elements assuming different distributions for the random delay concerning the propagation of the routing information. Moreover we analyse, how additional information on the location and on the mobility of the nodes can improve the prediction of the network state.

2. Stability based routing in ad hoc networks

Generally classic proactive and reactive routing protocols apply a simple additive cost metric (often the hopcount) to compute shortest paths towards destinations. Often, shortest paths are not reliable when the network topology changes dynamically. To illustrate, let us imagine a shortest path with minimum number of hops. Such a path corresponds to links (hops) that rely far nodes in the space. This kind of links may be very unstable due to the mobility of the extremities. Finding more stable routes is an important goal in dynamic multi-hop ad hoc networks. Identifying stable paths permits to decrease control traffic and the number of route interruptions. A new routing paradigm can be obtained by considering the route stability or resiliency as routing metric. *Stability based routing* aims at choosing routes which are more stable in time. So, these latter can be more resilient to dynamic changes in the network topology.

If the events (such that the exact trajectory of the nodes, the power battery level, the associated user behavior, the network failures, etc.) are predictable and known in a MANET, then the best route can be computed to satisfy a communication request. Generally, these factors are not predictable and so it is not possible to create a good deterministic routing model. Practical observation based and more sophisticated, statistical and probability based routing models exist to deal with longlife routing. In this section, we discuss the representative protocols in this domain.

2.1 Observations on link stability

Several simple propositions aiming with the improvement of the routing decision can be found in the literature. Some works established that choosing routes based on additional information as the position of the nodes, their battery level and the mobility pattern permits to design routing algorithms promising paths with better resilience to topological changes.

Most of stability oriented works on routing consider the link stability in the MANETs fundamental. Effectively, the stability of a path depends on the stability of its composing links. Several propositions for stability based routing use a classification of the network links. In (Dube et al., 1997) authors use the strength of the received signal of neighboring nodes to determine whether the associated link is either *weakly connected* or *strongly connected*. Routing is then performed through paths maximizing the received signal strength. The signal strength is used as central metric to establish a route with long lifetime in (Agarwal et al., 2000). In their routing method, the authors suggest a routing protocol called Route-Lifetime Assessment Based Routing protocol (RABR) wherein the route selection is done using an intelligent residual-route-lifetime prediction on the basis of affinity appraisal of the candidate routes. Affinity is an estimate of the time after which a neighbor will move out of the threshold signal boundary of a mobile host, and hence is a measure of link availability. A similar, cross-layer information based solution is proposed in (Trivino-Cabrera et al., 2006) to find long-live paths in ad-hoc networks. Since a path is broken if one of the used links is broken, the authors propose the minimal received signal strength along the candidate path as indirect measurement of the path livetime. Targn and al. approach also the link stability by considering the radio propagation effect on signal strength in (Targn et al., 2007). A stochastic radio propagation model is proposed to compute received signal strength between adjacent nodes and to predict path loss. The authors consider that the link stability is equal to the probability of the receiving signal strength exceeding a predefined threshold. So, the estimation of link stability is derived from the prediction of signal strength. Considering the stability of a route, they state that the least stable link within a route would be the bottleneck for the route. The life spans of multi-hop routes decrease by increasing the route length. To model the effect of route length on estimating route stability, the authors consider the route stability as the product of link stabilities (this assumption will also be discussed in Section 3). With awareness of link and route stabilities, then an Ad-hoc On-demand Stability Vector (AOSV) routing protocol is proposed to reactively discover and maintain stable routes adapted to radio channels.

Similarly, the authors in (Chen & Nahrstedt, 1999) also propose the classification of network links from the point of view of their stability. Their objective is to find a path with sufficient resources to satisfy a given delay or bandwidth requirement in a dynamic multihop mobile environment with the help of a distributed QoS routing algorithm. In their classification, links between stationary or slowly moving nodes are considered as *stationary links*. In contrast to this, links which exist only for a short period of time are handled as *transient links*. Newly formed links are also considered to be transient as they are more likely to break down. Routing should use then stationary links whenever this is possible. To discover stable routes, a multi-path scheme and a ticket-based probing procedure is proposed. Multiple paths are searched in parallel to find the most qualified one. The ticket-based probing scheme achieves a balance between the less efficient single-path routing algorithms and the very expensive flooding algorithms. So, a near-optimal performance can be achieved with modest overhead by using a limited number of tickets and making intelligent hop-by-hop path selection. The algorithms can tolerate the imprecision of the available state information even if the degree of imprecision is high.

A simple and bandwidth-efficient distributed routing protocol based on the concept of associativity of nodes is proposed in (Toh, 1997). The Associativity Based Routing (ABR) apply a new metric called associativity which defines the stability of the link between two nodes: its corresponds to a counter which indicates the presence of a node in the neighbors. ABR considers that the longer two nodes have been neighbors, the longer they would stay connected. The optimal route towards a destination is the one maximizing its cumulative associativity metric. As the author state, the association property allows the routing protocol to exploit the spatial and temporal relationships of network nodes to construct long-lived routes, resulting in fewer route re-constructions and hence higher attainable throughput.

The study in (Lim et al., 2002) addresses to analyze the link stability and the lifetime of the routes computed by shortest path algorithms and also in SSA and ABR. The authors state that in a highly dense mobile ad-hoc network, shortest path routing finds unstable routes. Let us notice that they compare the analyzed algorithm with an ideal one which has a perfect link stability knowledge. In real world, this knowledge is impossible to have, but the ideal case illustrates the limits of the stability of the routes.

So, an interesting question can be formulated: if shortest paths are not good candidates for long-life routing, can longer routes be considered for this reason? Trivially, long paths using several short hops increase the congestion in the MANET. The distributions of the link lifetime and the corresponding route lifetime are also analyzed in (Cheng & Heinzelman, 2004). The analysis shows that increasing the route length by choosing short initial link distances may not be effective in extending the route lifetime. Moreover, the authors derive a closed form for link lifetime distribution and route lifetime distribution and try to characterize the relationship between node mobility and link stability. A polynomial time algorithm to determine the longest lifetime routes at different route lengths from all the possible routes between a source and a destination is developed. Using this algorithm, statistical results on the achievable maximum route lifetime improvement in random networks are gathered.

In (Sridhar & Chan, 2005) Sridhar and Chan investigated on the analysis of route stability MANET. In their empirical study, they demonstrate how residual link lifetime is affected by parameters such as speed and mobility pattern. The result shows that residual link lifetime is naturally a function of current link age and mobility of extremities but does not vary monotonically with age. The authors state that the conjecture saying that older links are more stable (which is used in existing stability-based routing algorithms like Associativity Based Routing), does not hold across a large spectrum of mobility speeds and models. In some cases, the reverse can be true. To find a good trade-off between cost, efficiency and route stability, this paper proposes an interesting, empirical based routing algorithm. The algorithm called SHARC is based on stability and hop-count information using DSR (cf. (Johnson et al., 2000)) as the basic routing protocol. The stability of a path is calculated using a simple histogram based estimator.

The availability of nodes is considered in the paper (Punde et al., 2003) which deals with the improvement of the reactive routing procedure. To find good paths, the authors propose a new parameter: the stability of the nodes which depends on the speed and on the observed packet processing ratio of the given node. Unstable nodes should not be used for routing.

2.2 Probability based route computations

Intensive research activities bring into focus the need for finding new and efficient models for ad hoc dynamic networks. The random behavior of ad hoc networks has been analyzed with the help of random graphs where the existence of an edge is characterized by a probability (for example in (Jacquet & Laouiti, 1999)). Let us notice that traditional random graph cannot model the dynamic behavior of the ad hoc networks.

The first analysis of the uncertainties were related to wired networks. In (Guérin & Orda, 1999) the network state is modeled with the help of random variables. In (Kuipers et al., 2005) the authors analyze the stability of paths in Internet like networks. They establish that monitoring any change along the Internet is not possible and they distinguish two kinds of changes: frequent changes and changes which occur infrequently. The infrequent changes can be broadcasted efficiently in the network; the convergence is fast compared to the change frequency. In the case of slight changes, it is not necessary to update the network with useless information. Since a constant update procedure of the network state information is very expensive (even not possible), the available network state information for routing is inherently inaccurate and the established paths can be broken at any time.

Two mobility models (respectively for a single node and for joint mobility of two nodes) are used to determine link and path availability in (Mcdonald & Znati, 1999b). Availability $A_{i,j}(t)$ of a link (i,j) is defined as the (conditional) probability that the link is active between two nodes i and j at time $(t_0 + t)$ given that it was active at time t_0. Similarly, the directed path availability $\Pi^k_{m,n}(t)$ is defined as the probability of the existence of a given path k between the source m and the destination n at time $(t_0 + t)$, supposing the existence of this same path at time t_0. If the independence of link failures is assumed, the path availability is given by the link availabilities that composed it: $\Pi^k_{m,n}(t) = \prod_{(i,j)\in k} A_{i,j}(t)$. In the opinion of the authors, this metric can be used for routing. The link and path availabilities are developed using the known random walk mobility model (cf. also in Section 6).

The authors in (Turgut et al., 2001) are interested by longevity computation of routes. Their goal is finding the right time to re-configure a route before its disruption and for this they study the predictability of the lifetime of the routes in different deterministic and Brownian mobility cases. Their analysis covers only the relation between two neighboring nodes. Trivially, the expected lifetime highly depends on the mobility pattern of the nodes. If the mobility is deterministic, the lifetime of links can be computed exactly. If there is no information about the mobility of nodes the prediction of the lifetime becomes difficult. Probabilistic methods may be used to determine the expected value of the lifetime of the different links. The expected value of a path lifetime corresponds to the minimum value of expected values of link lifetimes on the path.

In (Gerharz et al., 2003) statistical methods are proposed to estimate the stability of paths in a mobile wireless ad hoc environment. Several simulations illustrate the stability (and so the instability) of paths chosen according to a variety of strategies, including those used by the protocols AODV and DSR, under a variety of different mobility patterns. The authors state that selecting a path based on its estimated probability to persist for a minimum amount of time proved to reduce the number of route discoveries significantly. The number of route discoveries can also be reduced significantly by using a simple categorization of links into stable and unstable links, where the stability criterion is the age of the links. At the same time, the authors notices that this simple classification can lead to a severe increase in route failures.

They also confirmed the instability of shortest paths and state that for connections over longer distances, it is advantageous not to use a shortest path.

A fundamental question is: how to characterize the path stability? Different objectives and potential stability metrics are enumerated in (Gerharz et al., 2003) to find stable routes (the authors search additive link metrics for simple routing algorithms but the following propositions do not always permit the additive computation of path metrics). The proposed objectives are:

- *Minimize the number of unstable link.* If links can be classified as stable and unstable, the number of unstable links along the path can be used as a simple stability metric; the objective being to minimize this number.
- *Maximize the expected residual lifetime.* The residual lifetime of a path is equal to the lifetime of its more critical link. The expected residual lifetime of a link may be calculated from collected statistical data.
- *Maximize the persistence probability.* Similarly to the expected residual lifetime, the computation of the persistence probability of a link is proposed based on statistical data. Supposing the independence of the different links, the persistence probability of a path is a simple multiplicative metric. This metric directly aims at minimizing the number of interruptions during a certain time span.
- *Maximize a residual lifetime quantile.* The minimum residual lifetime that a path will reach with a given probability can be calculated from the path persistence probabilities. Using this objective, the path which maximizes this residual lifetime (depending on the desired probability) can be computed.
- *Avoid Instable Links.* Here the weakest link rule is applied: the stability of a path is the stability of the most instable link along the path. Obviously, the path avoiding the instable links is wanted.

A formal model to predict the lifetime of a route based on the random walk model is proposed in (Tseng et al., 2003). Links states are described with the help of a hexagonal cellular based system and node movements are modeled with the help of a state transition automaton, then route lifetime is derived. Lifetime of a route is computed as a function of the existence probability of each link component. The model permits to estimate the residual lifetime of routes which can be used for routing decision. Routing can be then made through the ones with maximum residual lifetime.

In (Chung, 2004) Chung proposes a probabilistic analysis of routes in MANETs when nodes move corresponding to a Random WayPoint model (cf. (Camp et al., 2002a)). An estimated probability density function is computed and analyzed. The analysis illustrates that the exponential distribution is a well candidate function to predict route behavior for stability and the time distribution of the routes is similar to a discrete Gamma distribution depending on the route length in hops (which is evident supposing independent links).

(Beraldi et al., 2006) suggests a "probabilistic" routing protocol in a MANET. Unlike the usual routing protocols, the packet forwarding in MANET's node is not driven by a previously computed path. Rather, the concerned nodes of the network exploit a set of routing meta-information which are called hints to discover the path to the destination on-the-fly. A node gathers hints from the nodes located within a small range of neighbors limited by a maximal number of hops (called the protocol look-ahead). A hint h_{id} from the node i with respect to the node d is defined as $h_{id} = \frac{\Delta_{id}}{\tau_{id}}$, where Δ_{id} is the time elapsed since d has most recently moved out of the transmission range of i, and τ_{id} is the duration of the last wireless link established between i and d. The hint value indicates the chance of i being in the neighborhood of d. Packets are forwarded as follows. On receiving a packet with destination d by a node i, this latter forwards the packet to the neighbor with the best (lowest) hint which not yet received the packet. If the selected node cannot be reached, another one can promptly be used. Since there is no guarantee that the next-hop node lies on a path towards the destination, the protocol is considered as a "probabilistic" one (cf. (Beraldi et al., 2006)). The main statistical properties of hints have been investigated through an analytical model in the paper. Since hint values are periodically updated, the protocol assure robustness against topological changes, while requiring a communication overhead. The hint-based probabilistic protocol was recently improved by omitting control messages. In the new proposition, the nodes reuse the feedback information carried in unicast packets for routing purpose without introducing any extra overhead. The efficiency of the proposed scheme is demonstrated through both mathematical analysis and an extensive simulations study in (Nejad et al., 2010).

A route discovery mechanism in mobile ad hoc networks called FResher Encounter SearcH (FRESH) using encounter ages is proposed in (Dubois-Ferriere et al., 2003). The proposition focus on the case of "blind routing protocols", *i.e.* , where nodes have no notion of coordinates (there is no GPS or localization). In their solution, nodes maintain a record of their recent encounter times with all other nodes. Two nodes encounter each other when they are directly connected. The authors consider that the history of last encounters between nodes contains valuable, but noisy information about the current network topology. To create a route, the source node searches for any intermediate node that encountered the destination more recently than did the source node itself (if it exists). The intermediate node then searches for a node that encountered the destination yet more recently, and the procedure continues until the destination is reached. So, FRESH performs a succession of small searches instead of a large route request, resulting in a cheaper route discovery. The performance of the solution depends on the node mobility. With standard mobility conditions the route discovery cost can be decreased by an order of magnitude.

The family of Parametric Probabilistic Routing (PPR) protocols which are called light-weight adaptive routing protocols is proposed in (Barrett et al., 2005). The protocols are considered probabilistic, because a re-transmission probability function is associated with each node, indicating with what probability a node will forward a received message. As it is explained, the re-transmission probability function can depend on various factors such as the hop-distance to the destination, the hop-distance of the source to the destination, the number of hops that the packet has already traveled, the number of times a node has already forwarded the same packet, the number of neighbors a node has, etc. PPR protocols thus perform a multipath-routing using directed, controlled flooding with multiple packet copies.

A general ad hoc network model and some derived statistical results on link and path availability properties using a particular random walk mobility model is presented in (Yu et al., 2003) (cf. also Section 6). In the proposed mobility model, each node moves with a randomly chosen velocity vector corresponding to a direction and a speed. The direction is a uniformly distributed random variable ϕ while the distribution of the speed v can be arbitrary. The availability study details link and path available time. At first, the computation model of the relative velocity and its probability density (PDF) between two nodes are expressed in order to give the distance and the move between two nodes a and b when the nodes are in continuously movement. The probability for a relative velocity vector V_r is trivially the summation over all the possible combination of V_a and V_b that forms a triangle with V_r. Using a mobility model, in some cases the distribution can be formulated or in other cases it can be approached. The computation method of the cumulative density function (CDF) of relative velocity is expressed in general case and then the link available time distribution computation of a one-hop link is given, which serves as the basis for multi-hop cases. This paper gives a good mathematical formulation of the bases for probabilistic routing design.

Zhang and Dong found out in (Zhang & Dong, 2007) that traditional path stability computation methods which suppose link independence are idealized and so their results do not accord with the reality. As it was presented earlier, if the assumption of link independence is adopted, then the route stability can be computed by multiplying the stability of the links along this path. However, the traditional methods have not considered the dependencies among links, and Zhang and Dong state that the so computed routes may be far from the real case. In fact, in ad hoc networks, the existence of wireless links is more or less correlated. To take into account the correlation, the usual methods are too complicated to be practical. To solve this dilemma, the authors propose a novel path stability computation model, in which a correlation factor is introduced to describe the dependency degree between arbitrary adjacent links. Based on the correlation factors, a simple and universal expression for computing the stability probability of a path is derived. The construction of the dependency structure between adjacent links is proposed which permits the computation of the joint stability of adjacent links.

To complete the state of the art on stability based routing methods, some special propositions based on localization and mobility models will be presented in Sections 5 and 6.

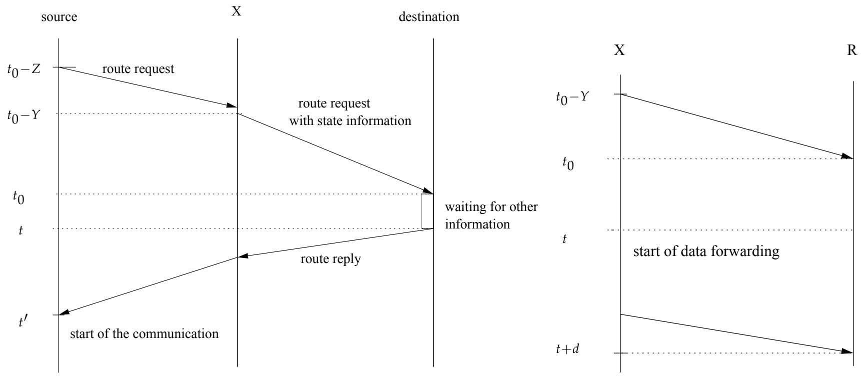
3. Probabilistic routing framework

In highly dynamic networks, when a route request arrives in a router, the information used to select a path is more or less inaccurate. We now enumerate rapidly the major causes of uncertainties.

3.1 Uncertainties of routing information

In the case of *reactive* routing protocols, the route is build following an explicit route request and without any *a priori* knowledge on the network state. The request corresponds to a new communication or to the restoration of a broken route. Generally, the source broadcasts a request and the destination answers using a route selected on the base of the received state information on the components of the route. Then the route is configured in the routers and used until its break. The state information can be volatile. The temporal progression of the process is shown in Figure 1 (a). The figure illustrates the use of the state information on an intermediate element X of the selected route. The source sent the request at time $(t_0 - Z)$ which is treated at time $(t_0 - Y)$ by an intermediate element. The state information on X arrives to the destination at time t_0. The destination selects the route on the base of received overall information at time t. The route reply containing the decision arrives at the source at t' and the communication can begin. If the component X does not exist when the route reply reaches the configuration of X, then the route is considered as broken. Even if the configuration succeeds, the state of the element X can change rapidly after the configuration of the selected route.

When using a *proactive* routing protocol, state information on the network component are broadcasted regularly (often periodically) and each router maintains a database on the network state. If a route request is presented, the router decides which route should be used toward the given destination. The decision is based on the local network state database. This database can contain obsolete information. The typical use of the state information on an



(a) Events in a reactive routing (b) Events in a proactive routing

Fig. 1. Typical scenarii in routing protocols

element X by a router R is illustrated in Figure 1 (b). The state information of X is sent to R at time $(t_0 - Y)$ and received at t_0. This information will be used at time t for routing until the end of the routing period if all right.

In both cases state information for routing can be non correct because of rapid and unpredictable changes in the network and because of (often) random propagation delay of information.

3.2 Our network model

The aim is to model the availability of the network elements in dynamic ad hoc networks. We distinguish two kind of elements:

- *nodes* (often mobile devices) which can manage their presence in the network autonomously (for example, they can arbitrarily disappear for power conservation reason using PSM option of the IEEE 802.11 standard).
- *potential communication capabilities* between node pairs. In our model, two nodes are capable to communicate if they are in the communication range one of the other, independently of their operational state. This concept is not equal to the traditional link concept. Naturally, a communication link exists between the nodes if the potential communication capability is true and if the two nodes are on.

Our model (which was basically proposed in (Marie et al., 2007)) contains also nodes representing the network nodes and edges corresponding to the potential communication capabilities (which are binary relations). At a given time, each element can be in the state UP (U) or in the state DOWN (D). The whole network model corresponds to a complete graph with n nodes and $n(n-1)/2$ edges representing the communication capability relations. The state of an element (which can be a node or an edge) can be modeled by a two state automaton. In the current analysis, we suppose that these automatons correspond to continuous time Markov chains.

3.3 The existence probability of a path

A path contains an ordered set of consecutive links $\{L_1, L_2, ..., L_m\}$ between the ordered, adjacent nodes $\{n_0, n_1, ..., n_m\}$ which are also in the path.

Let us consider the link L_i between two consecutive nodes n_{i-1} and n_i on the path. Let $R(n_{i-1},n_i)$ be the binary random variable which indicates the potential communication capability between them. $R(n_{i-1},n_i) = 1$ when they are one with the range of the other (corresponding to the state UP) and 0 otherwise. In the following, we use in the same way the name n of a node to indicate the fact that the node is present (it is in the state UP). So, n corresponds likewise to a discrete random variable having two possible values. Also to simplify, in the place of the probability $\mathbb{P}(x = 1)$ we will use the expression $\mathbb{P}(x)$ and this probability will indicate the probability that the element x is available. Moreover, we are interested by the availability of the elements on the time interval $[t, t + \Delta]$ without interruption.

Inside an arbitrary time interval, the link L_i exists between n_{i-1} and n_i, if and only if both nodes are in state UP and are also one with the range of the other during the period. Naturally the link does not exist if one of the extremities shuts down or if $R(n_{i-1},n_i)$ becomes equal to zero.

$$\mathbb{P}\left(L_i\right) = \mathbb{P}\left(n_i\ ,\ n_{i-1}\ ,\ R(n_{i-1},n_i)\right) \tag{1}$$

Generally, the existence of a node is independent from the existence of an other and from the location of the nodes. The potential communication capability $R(n_{i-1},n_i)$ between the nodes depends only on the distance between them, on the radio communication circumstances and on their ranges. This potential communication capability is completely independent from the activity state of the nodes. Due to the independence, the probability of the existence of a link in the MANET can be determined by

$$\mathbb{P}(L_i) = \mathbb{P}(n_i) \cdot \mathbb{P}(n_{i-1}) \cdot \mathbb{P}\left(R(n_{i-1},n_i)\right) \tag{2}$$

The computation can be generalized easily to express the probability of the existence of a whole path. Let us consider that a potential path $(s,d) = \{s, n_1, ..., d\}$ relies the source s to the destination d (so, $n_0 = s$ and $n_m = d$ in the previous definition of the path). This path exists if and only if all the nodes on the path are in UP states and if each successive pair of nodes on the path can communicate (the potential communication capability is true between them).

$$\mathbb{P}(s,d) = \mathbb{P}\left(n_0\ ,\ ...\ ,\ n_m\ ,\ R(n_0,n_1)\ ,\ ...\ ,\ R(n_{m-1},n_m)\right) \tag{3}$$

We suppose that the existences of the nodes are always and completely independent from the other facts. So, the nodes do not appear and disappear in a correlated manner. As it is indicated in (Zhang & Dong, 2007), generally, the neighbor links are not independent in MANETs because of the uncertain presence of the common extremity nodes and their mobility. The presence of a node n_i is separately represented in our model and its probability corresponds to $\mathbb{P}(n_i)$.

Concerning the communication capabilities, the discussion on path stability given in (Zhang & Dong, 2007) is worth considering. In our model, an end-to-end communication capability of a path is expressed by $R(n_0,n_1)\ ,\ ...\ ,\ R(n_{m-1},n_m)$. Since the communication capabilities which do not share any common endpoint are independent, the probability of the end-to-end communication capability is equal to:

$$\begin{aligned}
\mathbb{P}(R_{s,d}) &= \mathbb{P}\left(\bigcap_{i=1}^{m} R(n_{i-1},n_i)\right) \\
&= \mathbb{P}(\bigcap_{i=1}^{m-1} R(n_{i-1},n_i)) \cdot \mathbb{P}(R(n_{m-1},n_m)| \bigcap_{k=1}^{m-1} R(n_{k-1},n_k)) \\
&\vdots \\
&= \mathbb{P}\left(R(n_0,n_1)\right) \prod_{j=2}^{m} \mathbb{P}(R(n_{j-1},n_j)| \bigcap_{k=1}^{j-1} R(n_{k-1},n_k))
\end{aligned} \tag{4}$$

But, since the communication capabilities which do not share any common endpoint are independent,

$$\mathbb{P}(R_{s,d}) = \mathbb{P}\left(R(n_0,n_1)\right) \prod_{j=2}^{m} \mathbb{P}(R(n_{j-1},n_j)|R(n_{j-2},n_{j-1})) \tag{5}$$

or

$$\mathbb{P}(R_{s,d}) = \mathbb{P}\left(R(n_0,n_1)\right) \prod_{j=2}^{m} \frac{\mathbb{P}\left(R(n_{j-1},n_j),R(n_{j-2},n_{j-1})\right)}{\mathbb{P}\left(R(n_{j-2},n_{j-1})\right)} \tag{6}$$

If we suppose that the different potential communication capabilities are independent, that is to say, we suppose that the nodes approach one to the others in an uncorrelated way as it has been made in (Yu et al., 2003), then the probability of the common event can be computed as the product of the probabilities of elementary events:

$$\mathbb{P}(s,d) = \prod_{i=0}^{m} \mathbb{P}(n_i) \cdot \prod_{i=1}^{m} \mathbb{P}(R(n_{i-1},n_i)) \tag{7}$$

A nice observation permits to simplify the path probability computation. Namely, let us notice that the source $n_0 = s$ and the destination $n_m = d$ belong trivially to each path from s to d. So, the existence probability of both nodes does not influence the path selection for any routing algorithm. Moreover, in real cases, the path is wanted to ensure the connection between two existing nodes s and d (the routing has not importance when the source or the destination is failed). The simplified conditional probability of the path existence is the following:

$$\mathbb{P}(s,d|sd) = \prod_{i=1}^{m-1} \mathbb{P}(n_i) \cdot \prod_{i=1}^{m} \mathbb{P}(R(n_{i-1},n_i)) \tag{8}$$

3.4 Routing with maximal probability of existence

Since the path will be used between the source and the destination from the route request at t until the end of the routing period, it should be stable in the time interval $[t, t+\Delta]$. The optimal path corresponds to the path with the higher existence probability in this interval. Remember that the probabilities in our study indicate the probability that the elements are available on the time interval $[t, t+\Delta]$. Let $T_{s,d}$ be the set of paths between the nodes s and d. The most stable path T_{opt} can be found as :

$$T_{opt} = \arg \max_{T \in T_{s,d}} \mathbb{P}(T) \tag{9}$$

Since the logarithm function is a monotone increasing function:

$$T_{opt} = \arg\max_{T \in T_{s,d}} \ln \mathbb{P}(T) \tag{10}$$

or:

$$T_{opt} = \arg\min_{T \in T_{s,d}} (-\ln \mathbb{P}(T)) \tag{11}$$

Taking into account the development of the existence probability of the paths:

$$T_{opt} = \arg\min_{T \in T_{s,d}} \left(\sum_{i=1}^{m-1} -\ln \mathbb{P}(n_i) + \sum_{i=1}^{m} -\ln \mathbb{P}(R(n_i, n_{i-1})) \right) \tag{12}$$

The optimal path corresponds to a shortest path in a valuated graph which contains special values associated to the nodes and to the edges. Let $G' = (V, E)$ be a complete graph with the same nodes as the complete topology graph of the network. Let us associate to each node n the value $-\ln \mathbb{P}(n)$ and the value $-ln\mathbb{P}(R(n,m))$ to each edge (n,m). Such a valuated graph is illustrated in Figure 2. To find the shortest path between s and d, the graph G' can be easily transformed to a graph G'' valuated only on the edges. To obtain G'', it is sufficient to replace each node (except the source and the destination) with a small complete sub graph having as many nodes as the degree of the node is in G'. In each of these complete sub graphs the edges are valuated uniformly with the value of the corresponding original node in G'. This transformation is illustrated on the second part of Figure 2 supposing that the more stable path is asked between the nodes b and d. Since G'' is valuated with positive values, to find the shortest path between the source and the destination, one of the well known algorithms (as the algorithm of Dijkstra or the one of Bellman-Ford, ...) can be applied.

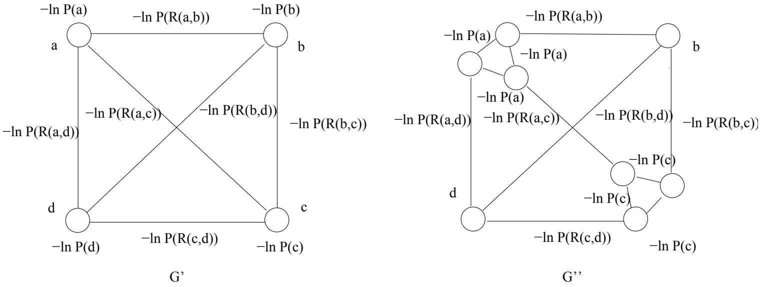


Fig. 2. The valuation of the topology graph

4. On-off automaton based model for the network elements

Our goal is to determine the route which guarantees the maximal existence probability for a time interval beginning with the route selection event. For this, the computation of the existence probability of the different elements based on the beforehand received information is needed.

4.1 Existence probability of an element at a given time

In this section we look for the probability for a given element to be UP at time t given the latest information received. In the following section we will look for the probability for the given element to be UP on all the interval $[t, t + \Delta]$.

We know that at time t_0 we received the information on the latest status of the element but this information was issued and sent at a time $(t_0 - Y)$which is unknown. In our case, we consider Y as a given non negative random variable. It is assumed that the dynamic of the changes of the element states is modeled by a two state continuous time Markov chain. U denotes the UP state ; respectively, D denotes the DOWN state.
When ordering the two states as (U, D), the infinitesimal generator is given by

$$A = \begin{bmatrix} -\lambda & \lambda \\ \mu & -\mu \end{bmatrix} \tag{13}$$

Let us first consider the transition probability function $h_{ij}(x)$ that gives the conditional probability that the element is in state j at time x given that it was in state i at time zero.
Let $H(x)$ denotes the 2×2 matrix such that

$$H(x) = \left(h_{ij}(x)\right) = \begin{pmatrix} h_{UU}(x) & h_{UD}(x) \\ h_{DU}(x) & h_{DD}(x) \end{pmatrix} \tag{14}$$

It is well known (cf. (Trivedi, 1982)), that this function $H(x)$ satisfies

$$H(x) = e^{Ax} \tag{15}$$

where by definition

$$e^{Ax} = \sum_{n=0}^{\infty} \frac{(Ax)^n}{n!} \tag{16}$$

For this two state CTMC, it is not too difficult to find the formal expression of this exponential matrix

$$e^{Ax} = \begin{bmatrix} \eta + (1-\eta)e^{-(\lambda+\mu)x} & (1-\eta) - (1-\eta)e^{-(\lambda+\mu)x} \\ \eta - \eta e^{-(\lambda+\mu)x} & (1-\eta) + \eta e^{-(\lambda+\mu)x} \end{bmatrix} \tag{17}$$

where $\eta = \frac{\mu}{(\lambda+\mu)}$.
In our case $e^{A(t-t_0)}$ will gives the conditional probabilities that the element is in state U or in state D at time t given that it was in state U or in state D at time t_0. But the knowledge we have for sure on the state of the element was issued at time $(t_0 - Y)$ where Y is a random variable. So the problem consists also in finding a probability matrix $M = \left(m_{ij}\right)$ where m_{ij} gives the conditional probability that the element is in state j at time t_0 given that it was in state i at time $(t_0 - Y)$.
It seams reasonable to model the random time Y as the sum of a constant time T_0 and of a random variable Z, where this random variable models the delay due to traffic congestion and the constant part corresponds to the physical propagation delay on the route.
In this study, we consider three possibilities for the distribution of the random variable Z : *i)* Z is exponentially distributed with mean $1/\gamma$, *ii)* Z follows an Erlang-n distribution (representing a random delay with weak dispersion) and *iii)* Z follows an hyperexponetial distribution (representing a random delay with high dispersion).

4.1.1 Case 1 : Z is exponentially distributed

Let us here look for the expression of M when the random variable Z is exponentially distributed with mean $1/\gamma$. First let us consider the probability matrix $B = \left(b_{ij}\right)$ such that b_{ij} is the conditional probability that the element is in state j at time t_0 given that it was in state i

at time $(t_0 - Z)$. Following Ross (see (Ross, 1987)), these conditional probabilities satisfies the following linear system

$$b_{ij}(Z) = \frac{\tilde{a}_i}{\tilde{a}_i + \gamma} \sum_{k \neq i} \frac{a_{ik}}{\tilde{a}_i} b_{kj}(Z) + \frac{\gamma}{\tilde{a}_i + \gamma} \xi_{ij} \tag{18}$$

with $i, j = U, D$; ξ_{ij} corresponding to the Kronecker symbol and $\tilde{a}_i$ denoting the departure rate of state i.
The formal resolution of this linear system gives the following expressions

$$B = \begin{bmatrix} 1 - \frac{\lambda}{\sigma} & \frac{\lambda}{\sigma} \\ \frac{\mu}{\sigma} & 1 - \frac{\mu}{\sigma} \end{bmatrix} \tag{19}$$

where $\sigma = (\mu + \lambda + \gamma)$.
In this case where the random time Y is the sum of a constant time T_0 plus the random variable exponentially distributed with mean $1/\gamma$, then the conditional probabilities are given by the matrix M such that

$$M = e^{AT_0}\, B \tag{20}$$

If M_1 denotes the matrix of the transition probability functions that give the conditional probabilities that the element is in state j at time t given that it was in state i at time $(t_0 - Y)$ (where Y is the sum of the constant time T_0 plus the exponentially distributed random variable with mean $1/\gamma$), then we have :

$$M_1 = Me^{A(t-t_0)} = e^{AT_0}\, Be^{A(t-t_0)} \tag{21}$$

It is possible to prove that matrices e^{AT_0} and B do commute and therefore that M_2 can still be written as

$$M_1 = Be^{A(T_0+t-t_0)} \tag{22}$$

For example, $(M_1)_{UU}$ gives the conditional probability that the element is in state U at time t given that it was declared in state U by the last issued message (at time $(t_0 - Y)$). From the knowledge of matrices B and $e^{A(T_0+t-t_0)}$, we get

$$(M_1)_{UU} = (1 - \frac{\lambda}{\sigma})(\eta + (1-\eta)e^{-(\lambda+\mu)(T_0+t-t_0)}) + \frac{\lambda}{\sigma}(\eta - \eta e^{-(\lambda+\mu)(T_0+t-t_0)}) \tag{23}$$

After some simplifications and in order to help in understanding the influence of the different parameters, we can write the matrix M_1 as :

$$M_1 = \begin{bmatrix} \frac{\mu}{(\lambda+\mu)} + (\frac{\lambda}{(\lambda+\mu)} \frac{\gamma}{(\lambda+\mu+\gamma)})e^{-(\lambda+\mu)(T_0+t-t_0)} & \frac{\lambda}{(\lambda+\mu)} - (\frac{\lambda}{(\lambda+\mu)} \frac{\gamma}{(\lambda+\mu+\gamma)})e^{-(\lambda+\mu)(T_0+t-t_0)} \\ \frac{\mu}{(\lambda+\mu)} - (\frac{\mu}{(\lambda+\mu)} \frac{\gamma}{(\lambda+\mu+\gamma)})e^{-(\lambda+\mu)(T_0+t-t_0)} & \frac{\lambda}{(\lambda+\mu)} + (\frac{\mu}{(\lambda+\mu)} \frac{\gamma}{(\lambda+\mu+\gamma)})e^{-(\lambda+\mu)(T_0+t-t_0)} \end{bmatrix} \tag{24}$$

In such a form, we can see the influence of parameters λ, μ, γ, T_0 and $(t - t_0)$ on these expressions.
If we let $\rho = \lambda/\mu$ and $\beta = \gamma/\mu$, the matrix M_1 can also be written as a function of μ, ρ, β, T_0 and $(t - t_0)$:

$$M_1 = \frac{1}{(1+\rho)} \begin{bmatrix} 1 + \rho\left(\frac{\beta}{1+\rho+\beta}\right) e^{-\mu(1+\rho)(T_0+t-t_0)} & \rho - \rho\left(\frac{\beta}{1+\rho+\beta}\right) e^{-\mu(1+\rho)(T_0+t-t_0)} \\ 1 - \left(\frac{\beta}{1+\rho+\beta}\right) e^{-\mu(1+\rho)(T_0+t-t_0)} & \rho - \left(\frac{\beta}{1+\rho+\beta}\right) e^{-\mu(1+\rho)(T_0+t-t_0)} \end{bmatrix} \tag{25}$$

4.1.2 Case 2 : Z follows an erlang-n distribution

Let us know look for the expression of M when the random variable Z follows an Erlang-n distribution with mean $1/\gamma$. First let us consider the probability matrix $C = \left(c_{ij}\right)$ such that c_{ij} is the conditional probability that the element is in state j at time t_0 given that it was in state i at time $(t_0 - Z)$.

Remembering that an Erlang-n distribution with mean $1/\gamma$ is equivalent to the sum of n independent and identically distributed random variables following the exponential distribution with rate $n\gamma$, the matrix C is such that (cf. (Ross, 1987))

$$C = C_1^n \tag{26}$$

where the probability matrix C_1 is obtained from B by replacing γ by $n\gamma$ (in notation σ):

$$C_1 = \begin{bmatrix} 1 - \frac{\lambda}{\sigma'} & \frac{\lambda}{\sigma'} \\ \frac{\mu}{\sigma'} & 1 - \frac{\mu}{\sigma'} \end{bmatrix} \tag{27}$$

with $\sigma' = (\mu + \lambda + n\gamma)$.

C_1 being a 2×2 probability matrix, it is possible to show after classical technical manipulation that matrix C (also a stochastic matrix) can be written as :

$$C = \frac{1}{1+\rho} \begin{bmatrix} 1 + \rho\left(\frac{n\gamma}{\sigma'}\right)^n & \rho - \rho\left(\frac{n\gamma}{\sigma'}\right)^n \\ 1 - \left(\frac{n\gamma}{\sigma'}\right)^n & \rho + \left(\frac{n\gamma}{\sigma'}\right)^n \end{bmatrix} \tag{28}$$

Then, the conditional probability matrix M will be such that

$$M = e^{AT_0}\, C \tag{29}$$

and the elements of the conditional probability matrix M_1 can be obtained similarly to the previous case. For example we get after some simplifications

$$(M_1)_{UU} = \frac{1}{(1+\rho)} \left[1 + \rho\left(\frac{n\gamma}{\sigma'}\right)^n e^{-(\lambda+\mu)(T_0+t-t_0)}\right] \tag{30}$$

4.1.3 Case 3 : Z follows an hyperexponetial distribution

We now consider the case where, for each outcome, with probability α (resp. $(1-\alpha)$), Z takes the value of an exponentially distributed random variable with rate γ_1 (resp. γ_2). With this hyperexponential distribution, it is always possible to find a triplet $(\alpha, \gamma_1, \gamma_2)$ satisfying a given mean and a given coefficient of variation greater than one, in order to model a random delay with high dispersion. Let us introduce the random variable J such that $J = 1$ (resp. $J = 2$) if the outcome corresponds to the first (resp. the second) exponentially distributed random variable.

Let us consider the probability matrix $D = \left(d_{ij}\right)$ such that d_{ij} is the conditional probability that the element is in state j at time t_0 given that it was in state i at time $(t_0 - Z)$. We have, for all i, j :

$$d_{ij}(Z) = \alpha d_{ij}(Z|J=1) + (1-\alpha) d_{ij}(Z|J=2) \tag{31}$$

This gives us

$$D = \alpha D_1 + (1-\alpha) D_2 \tag{32}$$

where the probability matrix D_1 (resp. D_2) is obtained from B by replacing γ by γ_1 (resp. γ_2) (using the notation $\sigma_i = (\mu + \lambda + \gamma_i)$ for D_i, $i = 1,2$).

If we let σ^* such that $\frac{1}{\sigma^*} = \frac{\alpha}{\sigma_1} + \frac{(1-\alpha)}{\sigma_2}$, it is possible to show that

$$D = \begin{bmatrix} 1 - \frac{\lambda}{\sigma^*} & \frac{\lambda}{\sigma^*} \\ \frac{\mu}{\sigma^*} & 1 - \frac{\mu}{\sigma^*} \end{bmatrix} \tag{33}$$

From that point, the elements of the conditional probability matrix M_1 can be obtained similarly to the previous cases.

4.2 State probability of an element on a given time interval

In this section, we now consider that at time t we want to know the probability for the given element to be UP on all the interval $[t, t + \Delta]$. Let us denote this probability by $\mathbb{P}(x(t, \Delta))$. From classical results on the exponential distribution, we know that the probability that the element stays up on all the interval $[t, t + \Delta]$ given that it is up at time t is equal to $e^{-\lambda\Delta}$.

Then the conditional probability that the considered element stays up on all the interval $[t, t + \Delta]$ given that it was declared in state U by the last issued message is equal to $(M_1)_{UU}.e^{-\lambda\Delta}$. Respectively, the conditional probability that the considered element stays up on all the interval $[t, t + \Delta]$ given that it was declared in state D by the last issued message is equal to $(M_1)_{DU}.e^{-\lambda\Delta}$.

Let us express these conditional probabilities in the special case when Z is exponentially distributed. Using the indicator function I_x such that $I_x = 1$ when the element x is in state U at time $(t_0 - Y)$ (else 0), the probability that this element x stays up on all the interval $[t, t + \Delta]$ is formally equal to:

a) if the element was declared in state U by the last issued message :

$$P(x|I_x = 1) = \frac{e^{-\mu\rho\Delta}}{(1+\rho)} \left[1 + \rho \left(\frac{\beta}{1+\rho+\beta} \right) e^{-\mu(1+\rho)(T_0+t-t_0)} \right] \tag{34}$$

b) if the element was declared in state D by the last issued message :

$$P(x|I_x = 0) = \frac{e^{-\mu\rho\Delta}}{(1+\rho)} \left[1 - \left(\frac{\beta}{1+\rho+\beta} \right) e^{-\mu(1+\rho)(T_0+t-t_0)} \right] \tag{35}$$

The two figures illustrate the existence probability of an element depending on the value of the normalized product $\mu(T_0 + t - t_0)$. In both cases the parameters of the corresponding automaton are $\lambda = 0.20$ and $\mu = 0.1$. Figure 3 presents the obtained probabilities from the three cases of probability distributions considered for the random variable Z with with $\gamma = 1.0$ and $\Delta = 2.0$. When the product $\mu(T_0 + t - t_0)$ is equal to zero, the elapsed time since the element was declared up corresponds to the random delay Z. This is why the probabilities on the figure do not equal one when the product is null. Moreover, since the expectation of the random variable Z is relatively important ($E[Z] = 1$ for the three cases), the influence of the probability distribution of Z is not negligible. Figure 4 presents also the obtained probabilities from the three cases of probability distributions considered for the random variable Z but with $\gamma = 10.0$ and $\Delta = 1.0$. In that situation, the expectation of Z is relatively small ($E[Z] = 0.1$) and the influence of the probability distribution is not important. The curve corresponding to the Erlang-5 distribution is not distinguishable from the curve corresponding to the exponential distribution.

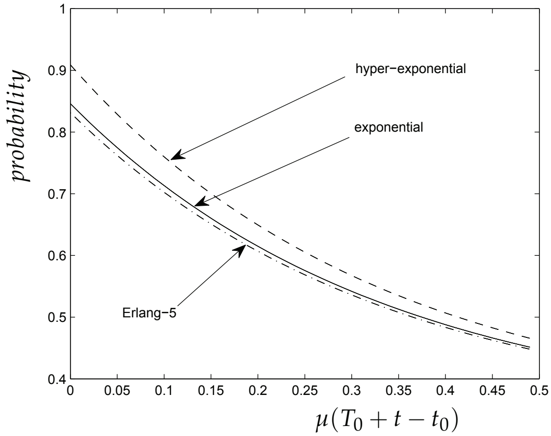


Fig. 3. Existence probability with $\gamma = 1.0$ and $\Delta = 2.0$

5. Localization based computations

Several routing protocols have been proposed using location information of the nodes (obtained for example using the GPS system). In (Ko & Vaidya, 2000), the authors propose two schemes of Location-Aided Routing (LAR) protocol for route discovery. The LAR protocols use this location information (which may be out of date) to reduce the search space for the desired routes. For the nodes concerned in route requests, an "expected zone" is computed based on the knowledge that the node was at a given location at a given time. So, route discovery may be limited to certain zones. Limiting the search space results in fewer route discovery messages.

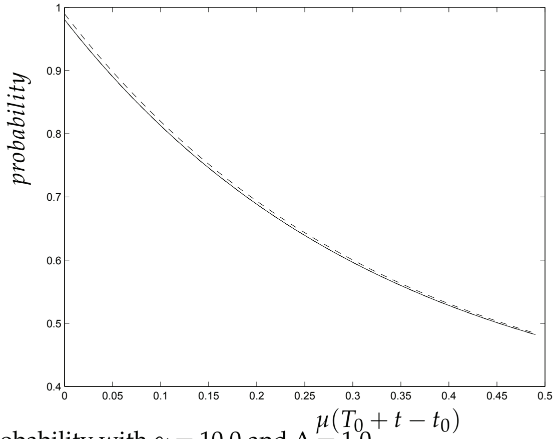


Fig. 4. Existence probability with $\gamma = 10.0$ and $\Delta = 1.0$

The location based protocol called DREAM (as Distance Routing Effect Algorithm for Mobility) was proposed in (Basagni et al., 1998). The DREAM protocol corresponds to a proactive routing procedure. When the sender node s want to send a message to the destination node d, it uses the location information for d to compute the direction of d, The source transmits the message to all its one hop neighbors in the direction of d. The subsequent nodes repeat the same procedure until the destination node is reached.

An other location based protocol, the Greedy Perimeter Stateless Routing (GPSR) has been described in (Karp, 2000). GPSR makes greedy forwarding decisions using only information about the neighbors of nodes in the network topology. By keeping state only about the local topology, GPSR scales better in per-router state than shortest path based ad hoc routing protocols as the number of network destinations increases. Under topology changes, GPSR can use local topology information to find correct new routes quickly. In certain cases of location inaccuracy GPSR disconnects some routes in the graph and hence the packets will not be routed. In (Tomar & Tomar, 2008) an algorithm is proposed, which removes some of the drawbacks of the GPSR algorithm.

A quorum based location service has been proposed in (Stojmenovic et al., 2008), in which source nodes try to find the destinations using their location information. In a previous version of this service, nodes report their new positions to their neighbors whenever a link is broken or created. So, nodes regularly forward their new position to all nodes located in its area of routing to help the stable route creation.

New perspectives are given with the discovery of 3D routing problems in MANETs (cf. a recent routing proposition in (Liu et al., 2008)) but a good survey on 3D routing problems is actually wanted. An earlier survey on position based routing protocols can be found in (Mauve et al., 2001).

To use efficiently our simple dynamic network model based on two-state automatons and presented in Section 3, the transitions of the automatons should be computed as realistic as possible. Since the network elements in a MANET alternate their state between UP and DOWN states, the behavior of the automatons modeling the nodes and the communication capabilities may correspond to the dynamic of the network. Nevertheless, the entire specification of the automaton is difficult. The problem resides in the fact that the transition dates are not known and are random dates. The more the details are known on the (random) behavior of the elements, the more the automatons may be designed precisely.

5.1 Availability of nodes

In our previously proposed model, the automatons associated with the network nodes model the random behavior of both the machines and the human operators. Without any knowledge on the reliability of these actors, a good choice can be the exponential distribution modeling the availability of an existing element. The rate of this distribution can correspond to estimated availability of the machine and its operator. If the machines are unreliable and/or the operators can quit the network, this kind of transition functions are faithful. If the actors are considered stable, then their existence probability is near to 1 and the lifetime of the corresponding node is infinite compared to the (generally) short routing period. To simplify, in this section we suppose than the nodes present at the beginning of a routing period are available in the entire period (the automatons corresponding to the present nodes are in UP state).

5.2 Communication capability between nodes

In the following, we investigate on the determination of simple transition functions associated with communication capability automatons. At first, we describe how the simple information on the node localization can be used to approach the parameters of the two state automatons. The next section illustrates that by learning the parameters of node movements, the communication capabilities and so the link stability can be computed more precisely.

The simple location based computation model for the two state automaton associated with the communication capability was firstly proposed in (Belghith et al., 2008). We assume that the residence time of the element e in state UP is governed by an exponential distribution with rate λ_e. Similarly, the residence time in state DOWN is governed by an exponential distribution with rate μ_e. The choice of the parameters λ_e and μ_e should reflect the dynamic behavior of the network. Indeed, if these parameters increase for a communication capability type element, this indicates unstable communication capacity in time and frequent oscillations between the UP and DOWN states. High values reflect strong mobility of nodes which involves frequent changes in communication capability states. If one of the parameters is very small, this indicates rather stable elements in UP or DOWN state over time. For instance, if λ_e is very small, nodes tend to stay together. Using the two state automatons with exponential transitions, the main question is to choose the values of λ_e and μ_e for the automaton of e. This element e is in state UP if its extremities are within transmission range and independently from the state of these extremities. Nevertheless, it is clear that the communication capability depends on the actual distance separating these nodes. Let R denote the identical transmission range of nodes and let d_e denote the distance separating the nodes corresponding to e. Hence these two nodes are capable of communication only if d_e is less than or equal to R. However, the shorter is d_e the more stable is the corresponding communication capability. So, the automaton of a communication capability is some how related to the distance separating the extremities.

It is known that the transient state of an exponential function based two state automaton falls off rapidly to reach its steady state. Consequently, one can discuss the transition functions by considering and relying on the steady state probabilities. Consider two nodes, which are capable to communicate at the beginning of the routing period. That is d_e for these nodes is no larger than R. The transient probability of being UP $P_{UU}(e,t)$ starts equal to one and then will falls off rapidly to its steady state probability β. This steady state probability should tend to one when the distance d_e is small; yet it should tend to one half as d_e approaches R. Now let us consider an element e in state DOWN at the beginning of the routing period. The transient probability of being in state DOWN $P_{DD}(e,t)$ starts equal to one and then will converge rapidly to its steady state probability $1-\beta$. This latter probability should be near to one when d_e gets farther and should equal to one half as d_e approaches R. So, λ is an increasing function (respectively μ as a decreasing function) of d_e/R. For each automaton, the following parameters were proposed in (Belghith et al., 2008) :

$$\lambda_e = C\left(\frac{d_e}{R}\right)^k \qquad \mu_e = C\left(\frac{R}{d_e}\right)^k \tag{36}$$

with k and C two given positive constants. The value of k permits to define the speed at which λ and μ increases and decreases respectively.

This first computation for transition functions is very simple because it relies only on distances separating nodes, but it can be far from the reality as it is illustrated in the following.

6. Random walk mobility based computations

The more information are known on the behavior of nodes in an ad hoc network, the better can be the probabilistic routing to ensure route stability. Mobility models permit to describe the movement pattern of mobile nodes, *i.e.* their probable location, direction and speed evolution over time. Using mobility models, one can estimate the neighboring relation of nodes without exact knowledge on their future movements. Due to their impact on the network control and management function, the mobility models of nodes are analyzed intensively in the literature (cf. large surveys in (Camp et al., 2002b) and (Lin et al., 2004)).

Frequently used mobility models in MANET are random models and the most known model is the Random Way-point model proposed in (Broch et al., 1998). In this model, nodes move independently to a randomly chosen destination with a randomly selected velocity. This model is the commonly used one when no additional information is available on the mobility, even if stochastic properties of this model are particular (Bettstetter et al., 2002).

The Random Walk model has similarities with the Random Way-point model, but in this model, the nodes change their speeds and directions *at each time interval*. At the beginning of a period, each node chooses randomly and uniformly its new direction and its new speed following a uniform distribution or a Gaussian distribution. It is also referred to as the Brownian Motion (cf. (Kac, 1947)). These random models are Markovian.

Mobility of a node may be constrained by physical laws. The inertia limits the acceleration, the velocity and the changes of direction. Hence, the current velocity of a mobile node may depend on its previous state. So, the mobility is not always Markovian. Several mobility models considering temporal dependency are proposed. Such a model is the Gauss-Markov Mobility Model (Liang & Haas, 1999) in which the velocity of node is correlated with the previous values and modeled as a Gauss-Markov stochastic process. In the Smooth Random Mobility Model (Bettstetter, 2001) proposed by Bettstetter, the nodes change their speeds and directions incrementally and smoothly. The speeds do not follow a purely uniform distribution but nodes chose some preferred speeds with high probability, which corresponds better to the frequently observed behavior of the nodes. The speed change is assumed to be a Poisson process.

In (Su et al., 2001) the authors propose some improvement of location based routing protocols. Their starting-point corresponds to the fact that in typical mobile networks, nodes exhibit some degrees of regularity in the mobility pattern. By exploiting the knowledge on this non-random mobility pattern, the future state of the network topology can be predicted more precisely. The prediction is based on the knowledge of location and on speed parameters of nodes to estimate the lifetime of links. However prediction based on the learning of the inertia of mobiles or the prediction of the speed changes are not developed.

In the previously described random models, nodes move independently one from others. However, in real applications team collaboration may exist between the users. Therefore, the mobility of a mobile node could be influenced by other neighboring nodes. Since the velocities of different nodes are ”correlated” in space, spatial dependency models can improve the mobility model as it is the case in the Reference Point Group Mobility (RPGM) Model and other group mobility based route computations (Hong et al., 1999; Jayakumar & Ganapathi, 2008; Ochirsuren et al., 2008).

In (Cho & Hayes, 2005) a simple but mathematically tractable model of node motion is presented: the constant velocity model, which is used to derive a precise relation between mobility and connection stability. It is demonstrated that *link duration* has a strong invariant relationship with the stability of multi-hop connections for a wide range of mobility models, and thus is an excellent mobility metric.

Random Direction Mobility Model based computation of stable routes is analyzed in (Carofiglio et al., 2009). The authors study the availability and the duration probability of a route that is subject to disruption caused by node mobility. They derive both exact and approximate expressions of these probabilities (using the Random Direction model) and propose an approach to improve the efficiency of reactive routing protocols.

The impact of mobility on the link and route lifetimes in ad hoc networks is analyzed in (Lenders et al., 2006) using real data gathered from a real network of 20 test users. Link and route lifetime distributions are then analyzed. The authors state that besides link breakage due to node mobility, links might also break due to diverse sources of interference or to packet collisions. They develop a statistical framework to distinguish between the mobility and interference or collision errors and determine the lifetime distributions for both error types separately. The paper validates two commonly used stochastic mobility models namely the random way-point and the random reference group mobility model. The results show that the distributions of the two stochastic mobility models match very closely the empirical link lifetime distribution.

Additional knowledge on human operators and on their relations can improve the performance of the routing. For example, in (Musolesi & Mascolo, 2006) the authors model mathematically the possible association between humans. In their paper they propose a new mobility model founded on social network theory. The model allows collections of hosts to be grouped together in a way that is based on social relationships among the individuals.

To limit the perimeter of our study in this chapter, we propose a simple Random Walk (RW) model based computation of the transition functions of our automatons corresponding to the communication capabilities. The computation can easily be extended to other mobility models using the adequate node distribution functions. Our hypothesis are the following.

- At the beginning of a routing period (which is supposed here to start at $t = 0$), the localization of the nodes is known (each node has knowledge on the position of its neighbors).
- We have also information on the node mobility, but only the maximal velocity v_{max} of the nodes is known. We suppose that the nodes move randomly without stopping. So, the RW model can be used to describe the movement (cf. a detailed description here after).
- To simplify, we suppose that there is no obstacle in the space and the communication range of nodes is R.
- As the routing protocol is periodic, we are interesting to determine the relative node position distributions between the nodes for $0 < t < T$. Then this relative node position distribution permits to compute the probability that an existing communication capability at $t = 0$ may exist in the interval $[t, t + \Delta]$ without interruption.

In our model, the instantaneous communication capability of nodes depends only on the distance between them. So, we focus on the evaluation of this distance in order to estimate the probability of the communication capability between the nodes in the routing period. The evaluation can be characterized with a spatial distribution of the nodes. Let A and B be two nodes and d_0 the initial distance between them at $t = 0$. If $d_0 > R$, the nodes can not communicate at this moment. Only node pairs with $d_0 \leq R$, are interesting for an eventual path computation. We determine the distance of the nodes (more precisely the relative position of the nodes) at the moment t as follows.

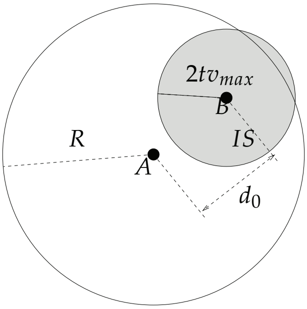


Fig. 5. The relative position af nodes at t.

As the nodes move following the RW model, their distance develops randomly. The moments of changes concerning the relative velocity corresponds to the moments when one of the nodes changes its velocity. Trivially, the relative velocity is the vectorial sum of the node velocities. To simplify the computation of the distance, we can suppose that one of the nodes (for example A) does not move and the other node (node B) moves using a random mobility model with maximal velocity $2v_{max}$. Figure 5 illustrates the position of B relative to A at t. Since the movement of B is limited by $2v_{max}$, B is in the circle of radius $2t \cdot v_{max}$ at the moment t (disk in grey). The nodes can communicate if B is in the communication range of A (with radius R). The probability of this communication capability can be expressed using the geometrical node distribution function of the mobility model at t.

Concerning the dispersion of B in side the disk, we use the random walk model proposed by in (McDonald & Znati, 1999a). This model is a continuous time stochastic process describing the movement of a node in the following way. First consider a Poisson process with rate λ producing events to change the velocity at times $t_i, i = 0, 1, 2, \ldots$. Let $T_i = t_i - t_{i-1}$ be the i^{th} interval. During an interval, the node is supposed to have a constant speed corresponding to a constant vector $\overrightarrow{v_i}$ (with a constant direction Θ_i and a constant module $|\overrightarrow{v_i}|$). At each occurrence of the Poisson process the vector of speed changes from $\overrightarrow{v_{i-1}}$ to $\overrightarrow{v_i}$. The random variables are independent. The module of the speed follows a given distribution function with a mean μ and a finite variance σ^2. The direction follows a uniform distribution over$[0, 2\Pi]$. The speed vectors are independent of the interval duration.

After a time t, if $N(t)$ denotes the number of occurrences of the Poisson process, the position of the node which is initially at $(0,0)$ will be given by the vector

$$\overrightarrow{z(t)} = \sum_{i=1}^{N(t)} \overrightarrow{v_i} T_i \tag{37}$$

The coordinate on the plane are $(x(t), y(t))$. Trivially:

$$x(t) = \sum_{i=1}^{N(t)} |\overrightarrow{v_i}| T_i \cos\theta_i \tag{38}$$

$$y(t) = \sum_{i=1}^{N(t)} |\overrightarrow{v_i}| T_i \sin\theta_i \tag{39}$$

It is known from the theory that the asymptotic distribution of this random position is a two-dimensional Gaussian distribution. $x(t)$ and $y(t)$ are independent, identically and

normally distributed with
$N(0, \frac{t}{\lambda}(\mu^2 + \sigma^2))$.
The probability that B and A can communicate may be computed as the geometrical probability that B is in the intersection IS of the two circles representing the communication range of A and the probable positions of B respectively:

$$P_t(dist(A, B) < R) = \int_{IS} p(x, y) \tag{40}$$

Figure 6 shows the existence probability of an element corresponding to the communication capability between two nodes computed with the following values: $\lambda = 20.0$, $\mu = 0.35$, $\sigma = 0.1$, $R = 10.0$, $d_0 = 6.0$, $v_{max} = 0.5$. The limited speed of nodes permits to adjust the existence probability. If the maximal speed of nodes v_{max} is known, one can compute the first moment t_d when the communication capacity may disappear:

$$t_d = \frac{R - d_0}{2v_{max}} \tag{41}$$

With the data used in Figure 6, t_d equals 4 and we can see that the existence probability stays to 1 from $t = 0$ to $t = 4$. The obtained instantaneous existence probability function can be used to compute the availability of the communication capability between the nodes during the interval $[t, t + \Delta]$.

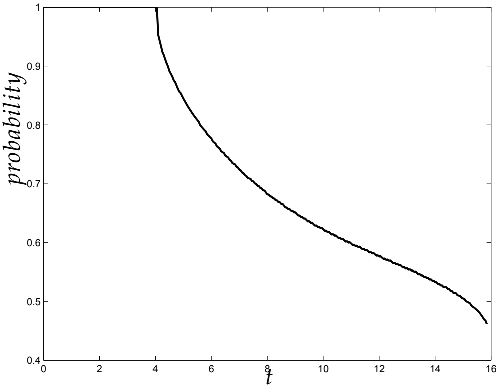


Fig. 6. The existence probability of the communication capability between two nodes when using the random mobility model

Let us notice that the limited speed of nodes permits also to adjust the transition function of the automatons in our dynamic network model. The exponential function based transitions of the automatons corresponding to communication capabilities are not realistic having the information on the position and on the velocity of nodes. Using an exponential transition function as it was proposed in the previous section, the probability of the interruption of an existing communication capability between two nodes becomes positive just after the start-point (cf. the curve with dotted line in Figure 7). In more realistic cases, when the maximal speed of nodes v_{max} is known, the first moment t_d when the communication capacity may disappear can be computed as it is indicated above. Before this date, an existing communication capability cannot disappear. Consequently, the transition function P_{UU} of our automaton modeling this communication capability is more realistic by chosen for example:

$$P_{UU}(t) = \begin{cases} 1 & \text{if } t \leq t_d, \\ \delta + (1-\delta)\exp -(\lambda + \mu)(t - t_d) & \text{if } t > t_d \end{cases} \tag{42}$$

Trivially, if $t_d \leq T$, then the communication capability exists in the whole period. The values λ and μ can be calculated supposing the probabilities $P_{UU}(t_d) = 1$ and $P_{UU}(T) = P_T(dist(A,B) < R)$. Based on this idea, the transition functions $P_{UD}(t)$, $P_{DU}(t)$ and $P_{DD}(t)$ of the automaton corresponding to the communication capability between A and B can also be obtained in the same way.

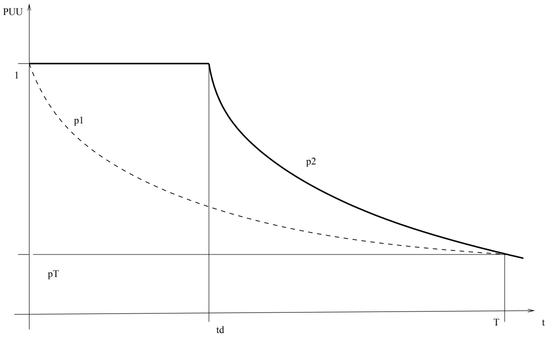


Fig. 7. A more realistic transition function P_{UU} in the time.

7. Conclusion

Because of dynamic topology changes, routes may frequently break in ad hoc networks. Topology and route maintenance involve important control message overhead. Selection of stable routes can diminish the control traffic and the packet losses. Moreover, the topology information in the routers can be inaccurate at the moment of its utilization. In the current chapter, we presented the most important heuristic and probability based ideas to obtain routes as stable as possible. We presented a simple mathematical model in order to predict the uncertainties due to unexpected changes and to random propagation delay of network state information. The proposed network model corresponds to a trade-off between faithfulness (which is expensive) and tractability. Our model deals with mobility, break-down of elements and also disappearances and re-appearances of stations in a simple way: by using an automaton based dynamic graph. It permits to express the joint effect of the state changes of the network elements and of the random propagation delay of control messages. We obtained closed form expressions for the existence probabilities of the network elements. The expressions allow to effect more precisely the network functionality (*e.g.* the routing) which are based on the network state information. The routing algorithm finding the more stable path for a given time interval corresponds to finding a shortest path on an equivalent graph. The estimation of automaton parameters based on the collected topology information is a crucial element for the routing protocol. The chapter illustrates how to use additional information as the location and the mobility of nodes to compute the parameters of the two-state automatons. Future work is needed to adjust the simple automaton-based model for cases where the mobility pattern is different from the here presented random mobility.

8. References

Agarwal, S., Ahuja, A., Singh, J. P. & Shorey, R. (2000). Route-Lifetime Assessment Based Routing (RABR) Protocol for Mobile Ad-Hoc Networks, *ICC'00: Proceedings of the IEEE International Conference on Communications 2000*, pp. 1697–1701.

Barrett, C. L., Eidenbenz, S. J., Kroc, L., Marathe, M. V. & Smith, J. P. (2005). Probabilistic multi-path vs. deterministic single-path protocols for dynamic ad-hoc network scenarios, *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, ACM, New York, NY, USA, pp. 1166–1173.

Basagni, S., Chlamtac, I., Syrotiuk, V. R. & Woodward, B. A. (1998). A distance routing effect algorithm for mobility (dream), *MobiCom '98: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, ACM, New York, NY, USA, pp. 76–84.

Belghith, A., Idoudi, H. & Molnár, M. (2008). Proactive Probabilistic Routing in Mobile Ad-Hoc Network, *ICWN*, pp. 621–627.

Beraldi, R., Querzoni, L. & Baldoni, R. (2006). A hint-based probabilistic protocol for unicast communications in manets, *Ad Hoc Networks* 4(5): 547 – 566.

Bettstetter, C. (2001). Smooth is better than sharp: a random mobility model for simulation of wireless networks, *MSWIM '01: Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, ACM, New York, NY, USA, pp. 19–27.

Bettstetter, C., Hartenstein, H. & Pérez-Costa, X. (2002). Stochastic properties of the random waypoint mobility model: epoch length, direction distribution, and cell change rate, *MSWiM '02: Proceedings of the 5th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems*, ACM, New York, NY, USA, pp. 7–14.

Broch, J., Maltz, D. A., Johnson, D. B., Hu, Y.-C. & Jetcheva, J. (1998). A performance comparison of multi-hop wireless ad hoc network routing protocols, *MobiCom '98: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, ACM, New York, NY, USA, pp. 85–97.

Camp, T., Boleng, J. & Davies, V. (2002a). A Survey of Mobility Models for Ad Hoc Network Research, *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications* 2(5): 483–502.
URL: *http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.16.8792*

Camp, T., Boleng, J. & Davies, V. (2002b). A Survey of Mobility Models for Ad Hoc Network Research, *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications* 2: 483–502.

Carofiglio, G., Chiasserini, C.-F., Garetto, M. & Leonardi, E. (2009). Route Stability in MANETs under the Random Direction Mobility Model, *IEEE Transactions on Mobile Computing* 8: 1167–1179.

Chen, S. & Nahrstedt, K. (1999). A Distributed Quality-of-Service Routing in Ad-Hoc Networks, *IEEE Journal on Selected Areas in Communications* 17(8): 1488–1505.

Cheng, Z. & Heinzelman, W. B. (2004). Exploring long lifetime routing (LLR) in ad hoc networks, *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, ACM, New York, NY, USA, pp. 203–210.

Cho, S. & Hayes, J. (2005). Impact of mobility on connection in ad hoc networks, *IEEE Wireless Communications and Networking Conference*, Vol. 3, IEEE, pp. 1650 – 1656.

Chung, W.-H. (2004). Probabilistic Analysis of Routes on Mobile Ad Hoc Networks, *IEEE*

Communications Letters 8(8): 506508.

Corson, S. & Macker, J. (1999). Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, RFC 2501.

Dube, R., Rais, C. D., Wang, K.-Y. & Tripathi, S. K. (1997). Signal stability-based adaptive routing (SSA) for ad hoc mobile networks, *Personal Communications, IEEE [see also IEEE Wireless Communications]* 4(1): 36–45.

Dubois-Ferriere, H., Grossglauser, M. & Vetterli, M. (2003). Age matters: efficient route discovery in mobile ad hoc networks using encounter ages, *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, ACM, New York, NY, USA, pp. 257–266.

Gerharz, M., Waal, C. d., Martini, P. & James, P. (2003). Strategies for Finding Stable Paths in Mobile Wireless Ad-Hoc Networks, *LCN '03: Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks*, IEEE Computer Society, Washington, DC, USA, p. 130.

Guérin, R. A. & Orda, A. (1999). QoS routing in networks with inaccurate information: theory and algorithms, *IEEE/ACM Trans. Netw.* 7(3): 350–364.

Hong, X., Gerla, M., Pei, G. & Chiang, C.-C. (1999). A group mobility model for ad hoc wireless networks, *MSWiM '99: Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, ACM, New York, NY, USA, pp. 53–60.

Jacquet, P. & Laouiti, A. (1999). Analysis of Mobile Ad-hoc Network Routing Protocols in Random Graph Models, *Research Report RR-3835*, INRIA.

Jacquet, P., Mhlethaler, P., Clausen, T., Laouiti, A., Qayyum, A. & Viennot, L. (2001). Optimized Link State Routing Protocol, *IEEE INMIC'01, 28-30 December 2001, Lahore, Pakistan*, IEEE, IEEE, pp. 62–68.

Jayakumar, G. & Ganapathi, G. (2008). Reference point group mobility and random waypoint models in performance evaluation of manet routing protocols, *J. Comp. Sys., Netw., and Comm.* 2008: 1–10.

Johnson, D. B. & Maltz, D. A. (1996). Dynamic Source Routing in Ad Hoc Wireless Networks, *in* T. Imielinski & H. Korth (eds), *Mobile Computing*, Vol. 353, Kluwer Academic Publishers, chapter 5, pp. 153–181.

Johnson, D. B., Maltz, D. A. & Broch, J. (2000). *DSR: the dynamic source routing protocol for multihop wireless ad hoc networks*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, pp. 139–172.

Kac, M. (1947). Random Walk and the Theory of Brownian Motion, *The American Mathematical Monthly* 54(7): 369–391.

Karp, B. (2000). GPSR: Greedy perimeter stateless routing for wireless networks, pp. 243–254.

Ko, Y.-B. & Vaidya, N. H. (2000). Location-aided routing (LAR) in mobile ad hoc networks, *Wirel. Netw.* 6(4): 307–321.

Kuipers, F., Wang, H. & Van Mieghem, P. (2005). The stability of paths in a dynamic network, *CoNEXT '05: Proceedings of the 2005 ACM conference on Emerging network experiment and technology*, ACM, New York, NY, USA, pp. 105–114.

Lenders, V., Wagner, J. & May, M. (2006). Analyzing the impact of mobility in ad hoc networks, *REALMAN '06: Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, ACM, New York, NY, USA, pp. 39–46.

Liang, B. & Haas, Z. J. (1999). Predictive Distance-Based Mobility Management for PCS Networks, *IEEE Information Communications Conference (INFOCOM)*, pp. 1377–1384.

Lim, G., Shin, K., Lee, S., Yoon, H. & Ma, J. S. (2002). Link Stability and Route Lifetime in Ad-hoc Wireless Networks, *ICPPW '02: Proceedings of the 2002 International Conference on Parallel Processing Workshops*, IEEE Computer Society, Washington, DC, USA, p. 116.

Lin, G., Noubir, G. & Rajaraman, R. (2004). Mobility models for ad hoc network simulation, *INFOCOM*.

Liu, S., Fevens, T. & Abdallah, A. E. (2008). Hybrid Position-Based Routing Algorithms for 3D Mobile Ad Hoc Networks, *MSN '08: Proceedings of the 2008 The 4th International Conference on Mobile Ad-hoc and Sensor Networks*, IEEE Computer Society, Washington, DC, USA, pp. 177–186.

Marie, R. A., Molnár, M. & Idoudi, H. (2007). A simple automata based model for stable routing in dynamic ad hoc networks, *PM2HW2N '07: Proceedings of the 2nd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*, ACM, New York, NY, USA, pp. 72–79.

Mauve, M., Widmer, J. & Hartenstein, H. (2001). A survey on Position-Based Routing in Mobile Ad-Hoc Networks, *IEEE Network* 15: 30–39.

McDonald, A. B. & Znati, T. (1999a). A mobility based framework for adaptive clustering in wireless ad-hoc networks, *IEEE Journal on Selected Areas in Communications* 17: 1466–1487.

Mcdonald, A. B. & Znati, T. (1999b). A Path Availability Model for Wireless Ad Hoc Networks, *Proc. IEEE WCNC*, Vol. 1, pp. 35–40.

Musolesi, M. & Mascolo, C. (2006). A community based mobility model for ad hoc network research, *REALMAN '06: Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, ACM, New York, NY, USA, pp. 31–38.

Nejad, K. K., Ahmed, S., Jiang, X. & Horiguchi, S. (2010). Probabilistic proactive routing with active route trace-back for MANETs, *Ad Hoc Netw.* 8(6): 640–653.

Ochirsuren, E., Indrusiak, L. S. & Glesner, M. (2008). An Actor-Oriented Group Mobility Model for Wireless Ad Hoc Sensor Networks, *ICDCSW '08: Proceedings of the 2008 The 28th International Conference on Distributed Computing Systems Workshops*, IEEE Computer Society, Washington, DC, USA, pp. 174–179.

Perkins, C. E. & Bhagwat, P. (1994). Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, *ACM Conference on Communications Architectures, Protocols and Applications, SIGCOMM '94, London, UK*, ACM, ACM, pp. 234–244.

Perkins, C. E. & Royer, E. M. (1999). Ad-hoc On-Demand Distance Vector Routing, *2nd IEEE Workshop on Mobile Computing Systems and Applications, WMCSA '99, February 25-26, 1999, New Orleans, Lousiana, USA*, IEEE, IEEE, pp. 90–100.

Punde, J., Pissinou, N. & Makki, K. (2003). On Quality of Service Routing in Ad Hoc Networks, *LCN '03: Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks*, IEEE Computer Society, Washington, DC, USA, pp. 276– 278.

Ross, S. M. (1987). Approximating Transition Probabilities and Mean Occupation Times in Continous-time Markov Chains, *Probability in the Engineering and Informational Sciences* 1: 251–264.

Sridhar, K. & Chan, M. C. (2005). Stability and hop-count based approch for route computation in MANET, *14th International Conference on Computer CCommunications and Networks(ICCCN 2005)*, pp. 25–31.

Stojmenovic, I., Liu, D. & Jia, X. (2008). A scalable quorum-based location service in ad hoc and sensor networks, *Int. J. Commun. Netw. Distrib. Syst.* 1(1): 71–94.

Su, W., Lee, S.-J. & Gerla, M. (2001). Mobility prediction and routing in ad hoc wireless networks, *Int. J. Netw. Manag.* 11(1): 3–30.

Targn, J.-H., CHuang, B.-W. & Wu, F.-J. (2007). A Novel Stability-Based Routing Protocol for Mobile Ad-Hoc Networks, *IEICE Transactions on Communications* E90-B(4): 876–884.

Toh, C.-K. (1997). Associativity-Based Routing for Ad Hoc Mobile Networks, *Wirel. Pers. Commun.* 4(2): 103–139.

Tomar, G. S. & Tomar, R. S. (2008). Position Based Routing for Mobile Ad Hoc Networks, *EMS '08: Proceedings of the 2008 Second UKSIM European Symposium on Computer Modeling and Simulation*, IEEE Computer Society, Washington, DC, USA, pp. 555–560.

Trivedi, K. S. (1982). *Probability and Statistics with Reliability, Queuing and Computer Science Applications*, Prentice Hall PTR, Upper Saddle River, NJ, USA.

Trivino-Cabrera, A., Nieves-Pérez, I., Casilari, E. & González-Canete, F. J. (2006). Ad hoc routing based on the stability of routes, *MobiWac '06: Proceedings of the 4th ACM international workshop on Mobility management and wireless access*, ACM, New York, NY, USA, pp. 100–103.

Tseng, Y.-C., Li, Y.-F. & Chang, Y.-C. (2003). On Route Lifetime in Multihop Mobile Ad Hoc Networks, *IEEE Transactions on Mobile Computing* 2(4): 366–376.

Turgut, D., Das, S. K. & Chatterjee, M. (2001). Longevity of routes in mobile ad hoc networks, *In Proceeding of IEEE 53rd Vehicular Technology Conference*, Spring, pp. 2833–2837.

Yu, D., Li, H. & Gruber, I. (2003). Path Availability in Ad Hoc Networks, *IEEE 10th International Conference on Telecommunications (ICT 2003)*, Vol. 1, IEEE, IEEE, p. 383387.

Zhang, H. & Dong, Y. (2007). A Novel Path Stability Computation Model for Wireless Ad Hoc Networks, *IEEE Signal Processing Letters* 14: 928–931.

20

Capacity, Bandwidth, and Available Bandwidth in Wireless Ad Hoc Networks: Definitions and Estimations

Marco A. Alzate[1], Néstor M. Peña[2] and Miguel A. Labrador[3]
[1]*Universidad Distrital*
[2]*Universidad de los Andes*
[3]*University of South Florida*
[1,2]*Bogota, Colombia*
[3]*Tampa, Florida, USA*

1. Introduction

Prasad et al. (2003) offered the most widely accepted definition of the capacity of a path, which has been usefully expressed as:

$$C = \min_{i=1...H} C_i \tag{1}$$

where C_i is the link capacity of the i^{th} hop in an H-hop path. Equation 1 has been the foundation of several clever active probing capacity estimation techniques, such as Pathrate, described by Dovrolis et al. (2004), and CapProbe, described by Kapoor et al. (2004), for example.
Similarly, the link available bandwidth for the i^{th} hop in an interval of time $(t - \tau, t]$, has been defined by Prasad et al. (2003) as:

$$A_i = (1 - \bar{u}_i)C_i, \;\; \bar{u}_i = \frac{1}{\tau}\int_{t-\tau}^{t} u_i(s)ds \tag{2}$$

where $u_i(s)$ is the instantaneous utilization of the i^{th} link at time s. The available bandwidth of the path, for the same interval, has been defined as:

$$A = \min_{i=1...H} A_i \tag{3}$$

Definition 3 has also led the way to many active probing available bandwidth estimation techniques and tools, such as Spruce, described by Strauss et al. (2003), IGI/PTR, described by Hu & Steenkiste (2003), Pathload, described by Jain (2002), Pathchirp, described by Ribeiro et al. (2003), TOPP, described by Melander et al. (2000), Delphi, described by Ribeiro et al. (2000), and Traceband, described by Guerrero & Labrador (2009), among others.
Unfortunately, the definitions above do not make any explicit reference to the dependencies that C and A can have on different parameters and network conditions, such as packet length and medium access overhead. This omission suggests that we are dealing with constant parameters of the path, as it is assumed in many cases. Although this simplifying

assumption is highly convenient and approximately correct in wired networks, it does not apply to wireless ad hoc networks (MANETs) due to the shared and unreliable nature of the transmission medium. Nevertheless, many current estimation techniques for MANETs are still based on definitions 1-3, measuring the fraction of time a node senses the channel idle, multiplying this fraction by the physical transmission capacity of the node, and sharing this measurements among the nodes of a path to estimate the available bandwidth (ABW) as the minimum measure among the individual nodes (see, for example, Chen & Heinzelman (2005); Guha et al. (2005); Xu et al. (2003); Ahn et al. (2002); Chen et al. (2004); Lee et al. (2000); Nahrstedt et al. (2005)). This per node estimation is not correct because it does not consider the occupation times of those links that cannot be used simultaneously, nor the additional overhead incurred when trying to use that idle capacity.

In this paper we conduct a theoretical analysis of the capacity (C), the bandwidth (BW), and the available bandwidth (ABW) of a link and a path in a MANET, in order to extend the definitions 1, 2, and 3 to this type of networks. We also develop a procedure to estimate the mean value of these quantities under the particular case of an IEEE 802.11b multi-hop ad hoc network.

Both C and BW are defined as the maximum achievable transmission rate in absence of competing flows, which is the basic notion of capacity used so far. Both of them take into account the shared nature of the transmission medium, but the concept of capacity does not consider the multi-access overhead, while the concept of bandwidth does. The concept of ABW also considers the effect of competing flows to determine the maximum achievable transmission rate.

The fundamental criterion for the extension of these concepts to MANETs is to avoid the elusive idea of a link as a unit of communication resource and to consider the "spatial channel" instead. Here a link is simply a pair of nodes within transmission range of each other, which shares the communication resources of a spatial channel with competing links. Indeed, a spatial channel is just a set of links for which no more than one can be used simultaneously, as defined below. These extensions do not pretend to constitute a detailed theoretical model of the physical phenomena occurring within a MANET, but simply a way to adapt and extend existing definitions. We would like to warn the reader that, during the process, we slightly redefine several well-established concepts in order to adapt them to the conditions we are facing.

After establishing this theoretical framework, we estimate the end-to-end C, BW, and ABW of a path between a pair of nodes in an IEEE 802.11b ad hoc network as a function of the packet length using dispersion traces between probing packet pairs of different lengths. The pairs of packets that suffer the minimum delay are used to estimate C and BW, while the variability of the dispersion trace is fed into a neuro-fuzzy system in order to estimate the practical maximum throughput obtained over the range of input data rates, closely related to the theoretically defined ABW.

In Section 2 we define the spatial channel as a set of links for which only one can be used simultaneously and, based on this simple concept, we develop the new definitions for C, BW, and ABW. In Section 3 we develop a method to estimate C and BW based on the dispersion measures between pairs of probing packets of two different lengths. In Section 4 we use the variability of the dispersion trace in order to estimate the ABW. Section 5concludes the paper.

2. Capacity, bandwidth, and available bandwidth definitions

Two pioneering works on capacity definitions for wireless networks are those of Bianchi (2000) and Gupta & Kumar (2000). Bianchi computed the saturation throughput of a single IEEE 802.11 cell, defined as the maximum load that the cell can carry in stable conditions. Gupta and Kumar Gupta & Kumar (2000) established some basic limits for the throughput of wireless networks, where the throughput is defined as the time average of the number of bits per second that can be transmitted by every node to its destination. These seminal works have been the basis of additional theoretical models Gamal et al. (2004); Grossglauser & Tse (2002); Neely & Modiano (2005); Kwak et al. (2005); Kumar et al. (2005); Chen et al. (2006) based on similar definitions. More recently, some detailed interference models have shown, analytically, the maximum achievable throughput on a specific link given the offered load on a set of neighbor links Kashyap et al. (2007); Gao et al. (2006); Takai et al. (2001); Sollacher et al. (2006); Koksal et al. (2006). However, these definitions neither extend to the end-to-end throughput nor lead to practical estimation methods.

Several methods have been proposed for the end-to-end capacity and available bandwidth estimation in wireless ad hoc networks based on definitions 1, 2, and 3 Chen & Heinzelman (2005); Xu et al. (2003); Ahn et al. (2002); Sarr et al. (2005); de Renesse et al. (2004); Renesse et al. (2005); Shah et al. (2003). Nonetheless, they are fundamentally inaccurate because, by measuring locally the utilization of the medium, they ignore the self interference of a flow at consecutive links and the simultaneous idle times of neighbor links. The authors of Chen et al. (2009) define the capacity of an end-to-end path as the length of a packet divided by the inter-arrival gap between two successfully back-to-back transmitted packets that do not suffer any retransmission, queuing, or scheduling delay. This definition led to AdHocProbe, but the estimation is only valid for the probing packet length utilized and does not say anything about the available bandwidth. Other authors Chaudet & Lassous (2002); Sarr et al. (2006); Yang & Kravets (2005) consider the interference by estimating the intersection between idle periods of neighbor nodes, so their estimations have better accuracy; but, still, taking the minimum among the individual measurements in the path considering only immediate neighbors, leads to significant inaccuracies. Finally, some estimations of the available bandwidth in a MANET end-to-end path are based on the self congestion principle, under the definition of available bandwidth as the maximum input rate that ensures equality between the input and output rates Johnsson et al. (2005; 2004). This method raises serious intrusiveness concerns in such a resource-scarce environment.

In this section we propose extended definitions for C, BW, and ABW more appropriate for MANETs, where the unit of communication resources is not the link but the spatial channel, so the definitions can take into account the channel sharing characteristic of this type of networks.

2.1 The spatial channel

The concepts of capacity, bandwidth, and available bandwidth are intimately related to the idea of a link between a pair of nodes and a route made of a sequence of links in tandem. However, the main difficulties and challenges with MANETs come, precisely, from the volatility of the concept of a link. While in a wired network every pair of neighbor nodes are connected through a point-to-point link, in a wireless MANET the energy is simply radiated, hoping the intended receiver will get enough of that energy for a clear reception, despite possible interfering signals and noise Ephremides (2002). In this context, a link is simply a pair of nodes within transmission range of each other. In defining bandwidth-related

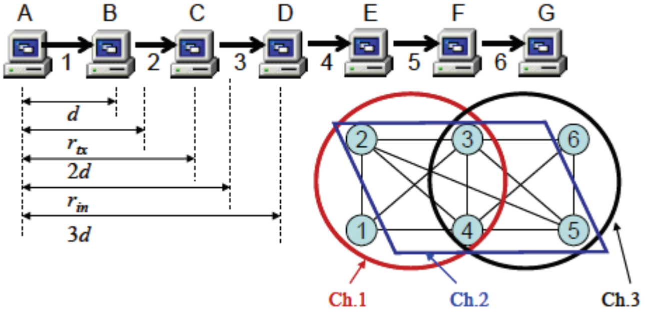


Fig. 1. A six-hop path and the corresponding contention graph showing three spatial channels.

metrics, one of the most important characteristics of MANETs is that two links cannot be used simultaneously if the intended receiver of one of the transmitters is within the interference range of the other transmitter. Accordingly, let us consider a wireless ad hoc network as a contention graph ($\mathcal{L}$,$\mathcal{E}$), where the set of vertices, $\mathcal{L}$, corresponds to the active links of the network, and the set of edges, $\mathcal{E}$, connect pairs of active links that cannot be used simultaneously Chen et al. (2004).

Definition 1. Spatial Channel. A spatial channel is a maximal clique (a complete subgraph not contained in another complete subgraph) in the contention graph ($\mathcal{L}$,$\mathcal{E}$) of a network, i.e., a spatial channel is a set of links for which no more than one can be used simultaneously.

Figure 1 shows a six-hop path in which nodes A through G, connected by links 1 through 6, are uniformly placed on a straight line at a distance d between them. Assuming that the transmission range (r_{tx}) and the interference range (r_{in}) of each node satisfies $d < r_{tx} < 2d < r_{in} < 3d$, there would be three spatial channels in this network, as shown on the contention graph in the bottom-right corner of the figure.

In what follows, we consider the spatial channel as the unit of communication resource, similar to the link in a point-to-point wired network, so we can extend the concepts of C, BW, and ABW.

2.2 Link capacity and end-to-end capacity

We keep the concept of the capacity of a link as the physical transmission rate of the node sending packets over it. But, in a wireless ad hoc network, several links share the same transmission medium, so we take this effect into account to define the concept of path capacity, omitting the effects of multi-access protocols. First, we consider a single pair of nodes, for which we simply define the link capacity as follows.

Definition 2. Link Capacity. For a pair of nodes within transmission range of each other, we define the capacity of the link between them as the physical transmission bit rate of the source node.

Now consider a path that traverses h spatial channels, with n_i links in the i^{th} spatial channel. If every resource is available for the source/destination pair of the path, an L-bit long packet will occupy the i^{th} spatial channel n_i times, during a total effective time of $t_i = \sum_{j=1}^{n_i}(L/C_{i,j})$, where $C_{i,j}$ is the link capacity of the j^{th} link in the i^{th} spatial channel in the path. In order not to saturate the path, the time between consecutive packets sent at the source node must be no less than $t_{min} = max_{i=1...h}t_i$. The maximum achievable transmission rate is $C^{path} = L/t_{min}$.

Definition 3. End-to-End Capacity. The end-to-end capacity of a multi-hop path that traverses h channels, where channel i is composed of n_i links with capacities $\left\{C_{i,j}, i = 1 \ldots h,\right.$ $\left. j = 1 \ldots n_i\right\}$, is defined as:

$$C^{path} = \min_{i=1\ldots h} \frac{1}{\sum_{j=1}^{n_j} \frac{1}{C_{i,j}}} \tag{4}$$

Note that Equation 4 becomes Equation 1 if each channel were a single link, as it is the case of paths composed of point-to-point wired links. However, differently to Equation 1, we cannot interpret Equation 4 as the transmission rate that a source would achieve in absence of competition because, so far, we have ignored completely the overhead introduced by the medium access mechanisms, which lead to the following concept.

2.3 Link bandwidth and end-to-end bandwidth

In absence of competing stations, the time to get and release the medium in a one-hop transmission is a random variable T, distributed as $f_T(t)$. The time required to transmit an L-bit long packet at a link transmission rate of C bps will be $T + L/C$, which means that, if the link is completely available for that packet, the link bandwidth is a random variable:

$$BW^{link}(L) = \frac{C \cdot L}{L + C \cdot T} \tag{5}$$

distributed as Alzate (2008):

$$f_{BW^{link}(L)}(b) = \frac{L}{b^2} f_T \left(L \cdot \left(\frac{1}{b} - \frac{1}{C} \right) \right) \tag{6}$$

Although the exact form of the expected value of $BW^{link}(L)$ depends on $f_T(\cdot)$, we can consider that, since the average time it takes an L-bit long packet to be transmitted is $t = E[T] + L/C$, the link bandwidth would approximately be L/t, suggesting the following definition:

Definition 4. Link Bandwidth. The expected value of the bandwidth of a C-bps link transmitting L-bit packets is defined as:

$$E\left[BW^{link}(L)\right] = \frac{L}{\frac{L}{C} + E\left[T\right]} \tag{7}$$

where T is the time required to get and release the transmission medium at that link.

Now consider a path that traverses h spatial channels, with n_i links in the i^{th} channel and link capacities $\left\{C_{i,j}, i = 1 \ldots h, j = 1 \ldots n_i\right\}$. Under perfect scheduling, an L-bit long packet will take an average time T_i^{ch} to traverse the i^{th} channel, given by:

$$T_i^{ch} = \sum_{j=1}^{n_i} \left(\frac{L}{C_{i,j}} + E\left[T_{i,j}\right] \right) \tag{8}$$

In order not to saturate the path, the average time between consecutive packets sent at the source node must be no less than $t_{min} = max_{i=1\ldots h} T_i^{ch}$. Under these assumptions, the maximum achievable bandwidth is $BW^{path} = L/t_{min}$:

Definition 5. End-to-End Bandwidth. The average end-to-end BW of a multi-hop path using L-bit long packets that traverse h spatial channels, where channel i is composed of n_i links with capacities $\left\{C_{i,j}, i = 1\ldots h, j = 1\ldots n_i\right\}$ and where the time it takes a packet to get and release the medium in order to be transmitted at the j^{th} link of the i^{th} channel is a random variable $T_{i,j}$, is defined as:

$$E\left[BW^{path}(L)\right] = \min_{i=1\ldots h} \frac{L}{\sum_{j=1}^{n_i}\left(\frac{L}{C_{i,j}} + E\left[T_{i,j}\right]\right)} \tag{9}$$

2.4 Link available bandwidth and end-to-end available bandwidth

As stated before, the available bandwidth (ABW) is highly dependent on the competing cross-traffic, which could have a complex correlation structure and interfere in many different ways with a given flow. Therefore, we will no longer look for the ABW probability density function, as we did above. Instead, if we assume that the cross-traffic is stationary and mean-ergodic, and that the queueing dynamics within the network nodes have achieved a stochastic steady state, we can find appropriate definitions for the mean value of the ABW on a link and an end-to-end path.

Consider a network composed of n active links, $j = 1\ldots n$, and h spatial channels, $i = 1\ldots h$. The i^{th} spatial channel is composed of n_i links $L_i = \left\{l_{i,j}, j = 1\ldots n_i\right\}$ with $l_{i,j} \in \{1, 2, \ldots n\}$. Let V_j be the set of spatial channels to which link j belongs to, $j = 1, 2, \ldots, n$. Clearly, $i \in V_j \iff j \in L_i$. In the interval $(t - \tau, t]$ the j^{th} link transmits $\tau\lambda_{j,k}$ packets of k bits, $j = 1\ldots n, k \geq 1$ (note that $\tau\lambda_{j,k}$ is not a per-source rate but a per-link rate, i.e., it includes forwarded packets too). Each k-bit packet transmitted over link j occupies each channel in V_j during $k/C_j + T_j$ seconds, where C_j is the j^{th} link capacity and T_j is the time it takes the packet to get and release the transmission medium at link j.

The time a spatial channel $i \in \{1, 2, \ldots, h\}$ is occupied during the interval $(t - \tau, t]$ is:

$$E\left[T_i^{occ}\right] = \sum_{j \in L_i}\sum_{k=1}^{\infty}(\tau \cdot \lambda_{j,k})\left(\frac{k}{C_j} + E\left[T_j\right]\right) \leq \tau \tag{10}$$

If a link x within L_i wants to transmit $\tau \cdot \lambda$ more L-bit long packets during $(t - \tau, t]$, inequality 10 becomes:

$$\lambda\left(\frac{L}{C_x} + E\left[T_x\right]\right) + \sum_{j \in L_i}\sum_{k=1}^{\infty}\lambda_{j,k}\left(\frac{k}{C_j} + E\left[T_j\right]\right) \leq 1 \tag{11}$$

Setting inequality 11 to 1, we can solve it for $\lambda \cdot L$ to obtain the available bandwidth for link x within spatial channel i, for L-bit long packets. Of course, the true available bandwidth for link x would be the minimum of the available bandwidths it has in each of the channels it belongs to, V_x.

Definition 6. Link Available Bandwidth. The mean available bandwidth in link x during the interval $(t - \tau, t]$ is defined as:

$$E\left[ABW^{link_x}(L)\right] = \frac{L}{\frac{L}{C_x} + E\left[T_x\right]}\left(1 - \max_{i \in V_x}\sum_{j \in L_i}\sum_{k=1}^{\infty}\lambda_{j,k}\left(\frac{k}{C_j} + E\left[T_j\right]\right)\right) \tag{12}$$

Using Equation 7, we recognize that Equation 12 is a direct generalization of Equation 2 for the available bandwidth of a link, where the utilization of the link becomes the maximum utilization among the spatial channels the link belongs to.

Now consider a path within this network, composed of a set of m links $X = \{x_1, x_2, \ldots, x_m\}$. If $\tau \cdot \lambda$ additional L-bit long packets were to be sent over the path in the interval $(t - \tau, t]$, then the new flow is to be added in each channel as many times as links in the path are present within the channel. Correspondingly, the first term in the left sum of inequality 11 must include the new flow in each link of the path within L_i. Writing it down for each link x in the path and each spatial channel i the link x belongs to, the set of conditions in Equation 13 must be met.

```
for each $x \in X$ do
    for each $i \in V_x$ do
        $\lambda \sum_{j \in X \cap L_i} \left( \frac{L}{C_i} + E\left[T_j\right] \right) + \sum_{j \in L_i} \sum_{k=1}^{\infty} \lambda_{j,k} \left( \frac{k}{C_j} + E\left[T_j\right] \right) \leq 1$    (13)
    end
end
```

Solving for $\lambda \cdot L$ with equality, we can find the available bandwidth for each link of the path within each spatial channel it belongs to. Taking the minimum bandwidth among the channels, we find the available bandwidth for each link, and taking the minimum among the links, we find the available bandwidth for the path.

Definition 7. End-to-End Available Bandwidth. The mean available bandwidth in a path during the interval $(t - \tau, t]$ is defined as:

$$ABW^{path}(L) = \min_{x \in X} \left\{ \min_{i \in V_x} \left[\frac{L}{\sum_{j \in X \cap L_i} \left(\frac{L}{C_j} + E\left[T_j\right] \right)} \left(1 - \sum_{j \in L_i} \sum_{k=1}^{\infty} \lambda_{j,k} \left(\frac{k}{C_j} + E\left[T_j\right] \right) \right) \right] \right\} \tag{14}$$

Notice again that Equation 14 is a direct generalization of Equation 3. Indeed, in a single spatial channel network, the form it takes is exactly $BW^{path}(L)(1 - u^{channel})$.

2.5 IEEE 802.11b example

Consider the case of the IEEE 802.11b DCF multi-access scheme in RTS/CTS mode, in which the time to acquire and release the transmission medium is $T = T_0 + L_0/C + B_o\sigma$, where T_0 is a constant delay (propagation time, control timers, and PLCP transmissions at the basic rate), L_0 is the length of the overhead control information (RTS, CTS, Header, and Acknowledgment), σ is the length of the contention slot, and B_o is a backoff random integer uniformly chosen in the range $[0, W-1]$, where W is the minimum backoff window. If we approximate T as a continuous random variable uniformly distributed in $[T_0 + L_0/C, T_0 + L_0/C + (W-1)\sigma]$, we get from Equation 6 the following distribution for the link bandwidth, $BW^{link}(L)$:

$$f_{BW^{link}(L)}(b) = \begin{cases} \frac{L}{b^2\sigma(W-1)} & if\ \ b \in I_b \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

$$\text{where}\ \ I_b = \left[\frac{CL}{L + L_0 + C(T_0 + \sigma(W-1))}, \frac{CL}{L + L_0 + CT_0} \right]$$

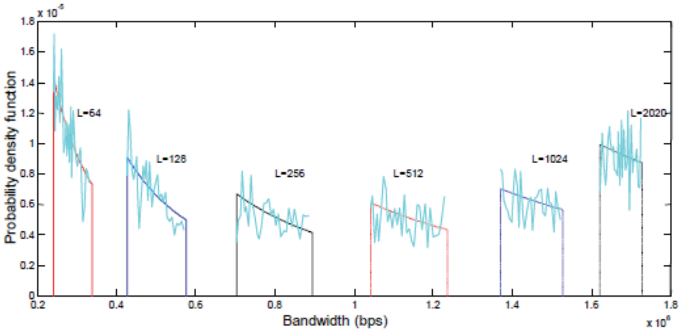


Fig. 2. Bandwidth distribution of a 2 Mbps IEEE 802.11b link.

Figure 2 shows the pdf using a 2 Mbps link as an example with different packet lengths and the corresponding histogram estimations obtained from Qualnet®SNT (2007) simulations. By direct integration, the average link bandwidth becomes:

$$E\left[BW^{link}(L)\right] = \frac{L}{(W-1)\sigma} log\left(1 + \frac{C(W-1)\sigma}{CT_0 + L + L_0}\right) \tag{16}$$

which can be well approximated as Alzate (2008):

$$E\left[BW^{link}(L)\right] \approx \frac{L}{\frac{L}{C} + \left(\frac{L_0}{C} + T_0 + \frac{W-1}{2}\sigma\right)} \tag{17}$$

as in Equation 7.
Consider now a single channel n-hop path for which the total acquisition and release time will be:

$$T^{ch} = \sum_{j=1}^{n}\left(T_0 + \frac{L_0}{C_j}\right) + \sigma\sum_{j=1}^{n} B_{0_j} \tag{18}$$

where B_{0_j} is the backoff selected by the transmitter of link j, uniformly and independently distributed in the range of integers $[0, W-1]$. Defining X as $\sum_j B_{o_j}$, then T^{ch} becomes:

$$T^{ch} = nT_0 + \frac{L_0}{C^{ch}} + \sigma X \tag{19}$$

where C^{ch} is, according to Definition 3, $1/\sum_{j=1\cdots n} C_j$. Assuming B_o is continuous and uniformly distributed in $[0, W-1]$, for $n > 1$ we can approximate X as a Gaussian random variable with mean $n(W-1)/2$ and variance $n(W-1)^2/12$, in which case the distribution of the spatial channel bandwidth becomes;

$$f_{BW^{ch}(L)}(b) = \frac{L}{\sqrt{2\pi} sb^2} exp\left[-\frac{1}{2}\left(\frac{b - L/m}{sb/m}\right)^2\right] \tag{20}$$

where

$$m = \frac{L+L_0}{C^{ch}} + n\left(T_0 + \sigma\frac{W-1}{2}\right) \tag{21}$$
$$s^2 = n\sigma^2\frac{(W-1)^2}{12}$$

are, respectively, the mean and the variance of $L/C^{ch} + T^{ch}$. Figure 3 shows the probability density functions, given by Equations 15 and 20, that correspond to the bandwidth experienced by a 1024-byte long packet transmitted over a completely available channel of n IEEE 802.11b hops at 2 Mbps, for n in $\{1,2,3,4\}$. The plots are compared with the corresponding normalized histograms obtained through Qualnet®SNT (2007) simulations, and with a Gaussian distribution with mean L/m and variance $(sL/m^2)^2$. Correspondingly, we propose that the bandwidth of an n-hop channel in an IEEE 802.11b path is Gaussian distributed with the following mean and variance, where Equation 22 is to be compared with Equation 9:

$$E\left[BW^{ch}(L)\right] = \frac{L}{\frac{L+L_0}{C^{ch}} + n\left(T_0 + \sigma\frac{W-1}{2}\right)} \tag{22}$$

$$V\left[BW^{ch}(L)\right] = \frac{n}{3}\left[\frac{L\sigma\frac{W-1}{2}}{\left(\frac{L+L_0}{C^{ch}} + n\left(T_0 + \sigma\frac{W-1}{2}\right)\right)^2}\right]^2 \tag{23}$$

Figure 4 shows the mean bandwidth given by Equation 22 for a single channel path composed of several 2 Mbps hops. Although the Gaussian approximation seems to be valid for a multi-hop channel but not for a single hop channel, Equations 22 and 23 seem valid for $n \geq 1$ hops, especially if the interest is in first and second order statistics of BW.

The bandwidth of a multi-hop multichannel path is the minimum of the bandwidths of the constituent spatial channels,

$$E\left[BW(L)\right] \leq \min_i E\left[BW_i(L)\right] = \frac{L}{\max_i\left[\frac{L+L_0}{C_i'} + n_i\left(T_0 + \frac{W-1}{2}\sigma\right)\right]} \tag{24}$$

where C_i' is the capacity of the i^{th} spatial channel in the path and n_i is the number of spatial channels.

Finally, as an illustration of the ABW concept, consider the two 2-hop ad hoc paths made of 2 Mbps IEEE 802.11b nodes, as shown in Figure 5. Node 5 routes data traffic between nodes 3 and 4 consisting of L_3-bit long packets at λ_3 packets per second. In order for nodes 1 and 2 to communicate, they must use node 5 as an intermediate router. Figure 6(a) plots the bandwidth of the 1-5-2 path, $E[BW(L_1)]$, as a function of the packet length used by node 1, $L_1/8$ bytes, and Figure 6(b) shows the fraction of available bandwidth, $E[ABW(L_1)]/E[BW(L_1)]$ (which, according to Equation 14 does not depend on L_1), as a function of the cross-traffic data rate, $\lambda_3 L_3$, and the cross traffic packet length, $L_3/8$ bytes.

For example, if node 1 transmits $L_1 = 4096$-bit long packets, Figure 6(a) says that the path could carry up to $\lambda_1 L_1 = 565.2$ kbps if there were no competition. However, if node 3 is generating packets of $L_3 = 8192$ bits at $\lambda_3 L_3 = 400$ kbps, Figure 6(b) says that only 44.6% of the bandwidth would be available for other users, in which case the available bandwidth for the 512-byte packets on the path 1-5-2 would only be 252 kbps.

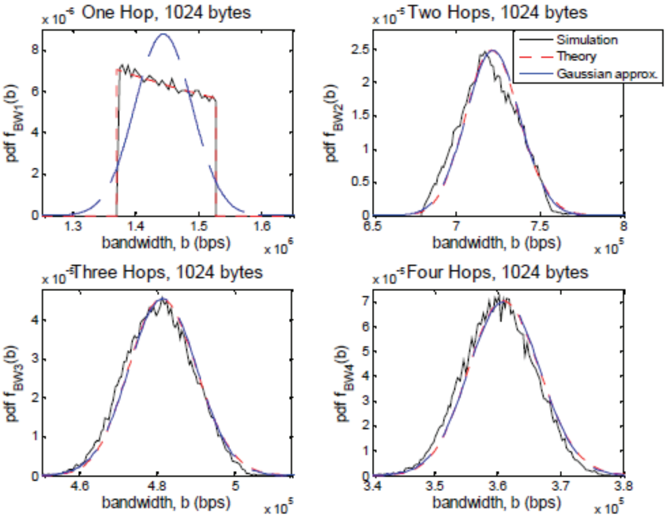


Fig. 3. Comparison of Equations 15 and 20 with QualNet®simulations and the proposed Gaussian approxin

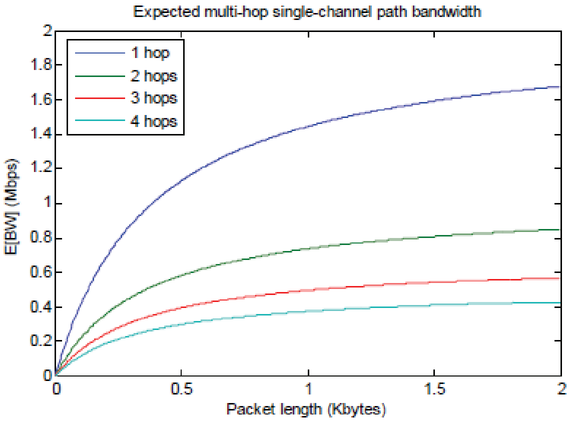


Fig. 4. Expected BW of a multi-hop channel path.

3. End-to-end mean bandwidth estimation as a function of packet length in multi-hop IEEE 802.11b ad hoc networks

It is important to have accurate and timely end-to-end capacity estimations along a multi-hop path for such important applications as source rate adjustment, admission control, traffic engineering, QoS verification, etc. Several methods have been proposed for *BW* and *ABW* estimation in wireless ad hoc networks, especially associated with resource constrained routing Chen & Heinzelman (2005); Guha et al. (2005); Xu et al. (2003) and/or QoS architectures Ahn et al. (2002); Chen et al. (2004); Lee et al. (2000); Nahrstedt et al. (2005). However, these methods depend on the particular routing algorithm and use inaccurate estimators. It would be highly convenient to have an end-to-end estimation tool at the

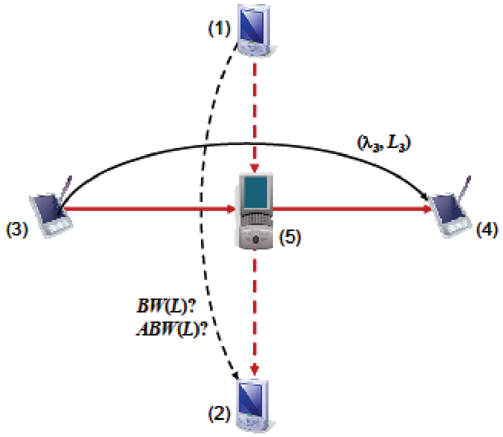
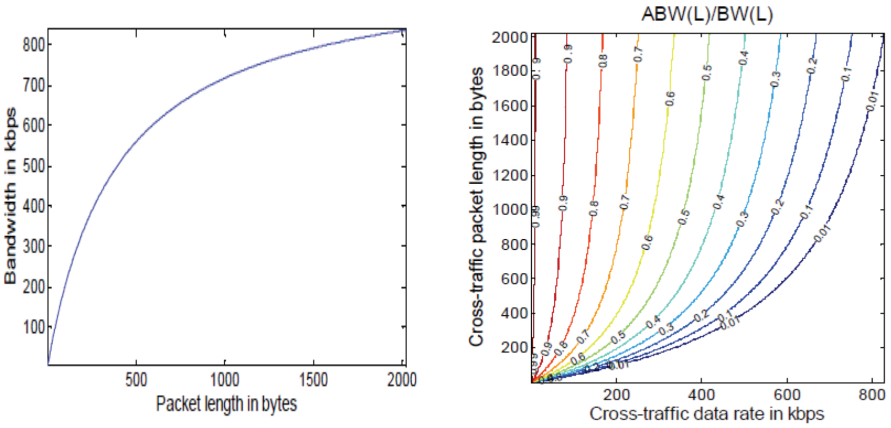


Fig. 5. A simple example to compute BW and ABW.



(a) Bandwidth of a two-hop path.

(b) Fraction of $BW(L)$ still available to the path 1-5-2 when the path 3-5-4 carries a given data rate (horizontal axis) using packets of given length (vertical axis).

Fig. 6. Bandwidth and fraction of available $BW(L)$.

application layer that does not rely on any lower layer assumptions. Ad Hoc Probe Chen et al. (2009), a simple and effective probing method that achieves high accuracy, satisfies these requirements. However, it uses a fixed packet length and returns a sample of the BW associated with that packet length, as if it were constant. In this section we devise a packet pair dispersion method that obtains several samples of BW and use them to estimate the variation range in order to give some confidence intervals for their mean. Our method is fundamentally based on Ad Hoc Probe principles, but we extend it to consider BW as a packet length dependent random variable. We also evaluate the performance of the method in terms of accuracy, convergence speed, and adaptability to changing conditions.

3.1 Measuring procedure

According to Equation 9, the BW experienced by a single packet of length L that finds all path resources completely available, has the form:

$$BW = \frac{L}{\alpha L + \beta} \tag{25}$$

where $\alpha L + \beta$ is the time it takes the packet to traverse the narrowest link in the path. In a single link, for example, $\alpha = 1/C$ is the cost, in seconds, for transmitting a data bit over the link, while $\beta = E[T]$ is the additional cost, in seconds, for transmitting a whole packet, independent of its length. In Equation 9, $\alpha = \sum_{j=1}^{n_i} 1/C_{i,j}$ is the inverse of the capacity of the narrow spatial channel, i, which corresponds to the cost, in seconds, of transmitting one bit over the i^{th} spatial channel, where $C_{i,j}$ is the bit transmission rate of the j^{th} link of the i^{th} spatial channel. Similarly, $\beta = \sum_{j=1}^{n_i} E[T_{i,j}]$ is the sum of the acquisition and release times on each link of the i^{th} spatial channel, the narrow one.

According to Equation 25, if it is possible to estimate $\alpha L + \beta$, the time it takes an L-bit packet to traverse the narrow spatial channel, it would be also possible to estimate the path bandwidth for the given packet length, L. The one way delay (*owd*) would be an appropriate measure of $\alpha L + \beta$ if the path is within a single link channel, but, in any other case, *owd* could be different than $\alpha L + \beta$. Indeed, we can send a pair of back-to-back equal-length packets and measure the interarrival time at the destination as an estimation of $\alpha L + \beta$ but, even in a completely available multi-hop wireless path, there could be scheduling differences that may lead to wrong estimations, as shown in Figure 7 for a two-link channel. The inter-arrival gap at the receiver in the second schedule of Figure 7, corresponding to the minimum *owd* of each individual packet reveals the real value of $\alpha L + \beta$, but the corresponding measure in the first schedule will underestimate $\alpha L + \beta$.

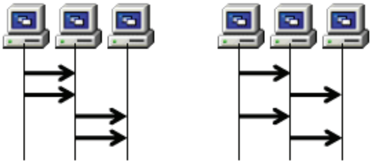


Fig. 7. Two possible schedules for sending two packets on a two-hop path.

In AdHocProbe Chen et al. (2009), the transmitting node sends several back-to-back L-bit long probing packet pairs in order to select the single pair in which each packet suffered the minimum *owd*, and use the gap between them to estimate $\alpha L + \beta$. If the procedure is repeated for a longer (or smaller) packet length, two points in the curve $BW(L)$ of Equation 25 will be obtained, from which the two unknown parameters, α and β, can be estimated. With these parameters, it is possible to interpolate the whole curve for the total range of allowed packet lengths.

Indeed, if the gaps G_0 and G_1 corresponding to the packet lengths L_0 and L_1 can be measured, it would be easy to find α and β, as follows:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} L_0 & 1 \\ L_1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} G_0 \\ G_1 \end{bmatrix} = \begin{bmatrix} \frac{G_1 - G_0}{L_1 - L_0} \\ G_0 - \frac{G_1 - G_0}{L_1 - L_0} L_0 \end{bmatrix} \tag{26}$$

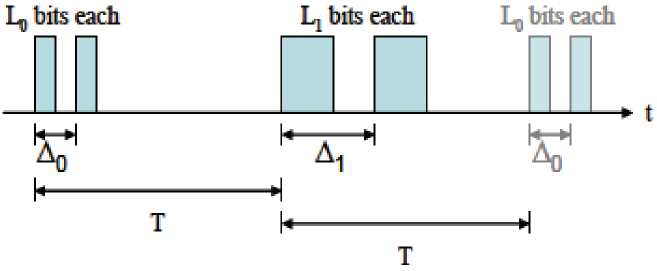


Fig. 8. Probing traffic pattern.

In order to compute Equation 26, the probing traffic will take the form shown in Figure 8. The value of the parameters L_0, L_1, T, Δ_0, and Δ_1 should be selected according to the network environment. In this paper, an IEEE 802.11b network with pedestrian users was used, for which the following parameters were used: $L_0 = 1024$ bits (128 bytes), $L_1 = 11200$ bits (1400 bytes), and $T = 0.25$ seconds. Although Δ_0 and Δ_1 can be adaptively selected to reduce self-interference in an unloaded multi-hop path, in this paper just back-to-back packet pairs were used.

The compromise between adaptability and accuracy is handed by using a window-based analysis. The pairs received during a t-seconds time window will be considered, during which, for each packet length L_0 and L_1, the one way delay of each packet will be measured, and the sum of one way delays of each packet pair, $sowd$, in order to record the minimum $sowd$, $sowd_{min}$. Within the window, those dispersion measurements with $sowd \leq sowd_{min} + (W-1)\sigma$ will be considered as valid realizations of the random variable $\alpha L + \beta$. Clearly, the longer the window length, t, the higher the confidence on the mean BW, and the smaller the window length, the higher the adaptability to changing conditions. Additionally, with several samples, confidence intervals can be found or estimates of the range of BW values, although in this paper only estimates of the mean end-to-end BW will be considered.

An important issue with the mentioned procedure is clock synchronization between transmitter and receiver. Of course, in a simulated environment there is a unique clock system, but in a real implementation, this problem affects dramatically the measurements.

Assume the receiver clock (t_{rx}) and the transmitter clock (t_{tx}) are related as $t_{rx} = (1+a)t_{tx} + b/2$, where there is both a drift term $(1+a)$ and a phase term $(b/2)$, and consider the time diagram of Figure 9, where $td_0^{'}$ and $td_1^{'}$ are the departure times of a pair of packets stamped at the transmitter, ta_0 and ta_1 are the arrival times registered at the receiver, and td_0 and td_1 are the (unknown) departure times according to the receiver's clock.

The correct sum of one way delays would be $sowd_c = (ta_0 - td_0) + (ta_1 - td_1)$, but the measured one would be $sowd_m = (ta_0 - td_0^{'}) + (ta_1 - td_1^{'}) = sowd_c + a(td_0^{'} + td_1^{'}) + b$. This linear tendency can be appreciated by plotting the measured sum of one way delays versus the measuring time, ta_1, as shown in the blue continuous line of Figure 10, corresponding to real measurements on a testbed. Dividing the analysis window in four subwindows, it is possible to compute a least mean square error linear regression on the minimum sowd of those windows, as shown in the diamond marked red dashed line of Figure 10. This linear tendency is subtracted from $sowd_m$ to obtain a new measure, $sowd_m^{'} = sowd_c + c$, where the constant c does not affect the computation of the minimum $sowd$, as shown in the black dotted line.

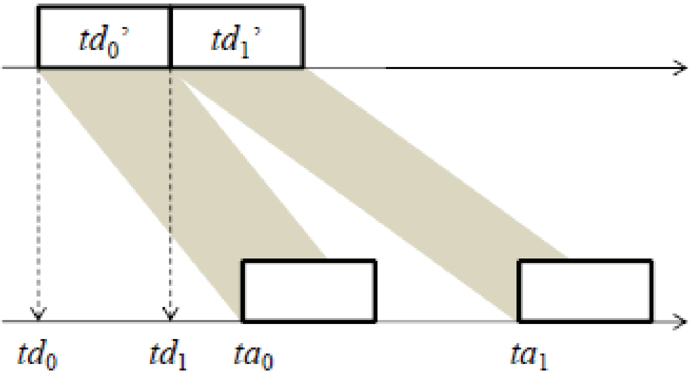


Fig. 9. Time incoherence between transmitter an receiver clocks.

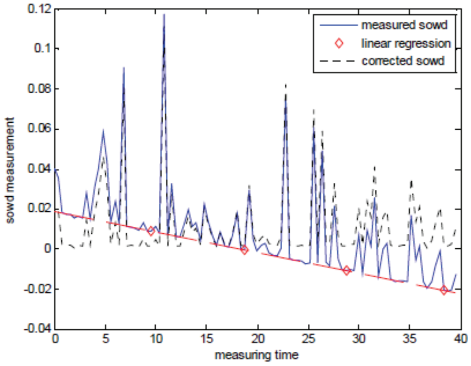


Fig. 10. Correction of clock incoherence through linear regression. Both axis are in seconds.

3.2 Numerical results

QualNet®SNT (2007) was used to evaluate the estimation procedure through simulation experiments using the default physical, MAC, AODV, IP, and UDP parameter values for a 2 Mbps IEEE 802.11b ad hoc network. Figure 11 shows the estimation results using the network shown in Figure 1. The mean *BW* converges quickly to the theoretical value even for windows of only 5 seconds, while the 99% confidence intervals decrease similarly fast, although with longer windows, since they require several valid samples. The 90 seconds results are identical to the corresponding theoretical values of Equation 22, previously shown in Figure 4.

The above encouraging results are obtained without considering additional traffic or mobility. The effects of these characteristics and, consequently, the adaptability of the protocol, are considered in the scenario shown in Figure 12. This scenario consists of a 5 × 5-grid of fixed nodes 300 m away from each other and a 26^{th} node moving around on a spiral trajectory at a speed of 2 m/s. There are two VBR flows of 50 kbps each, one from node 6 to node 10 and another one from node 16 to node 20. The bandwidth of the path between nodes 1 and 26 is to be estimated.

Figure 13 shows the estimated mean bandwidth as a function of time for each packet length when the measurement time window is 30 seconds. Notice how easy is it to detect route

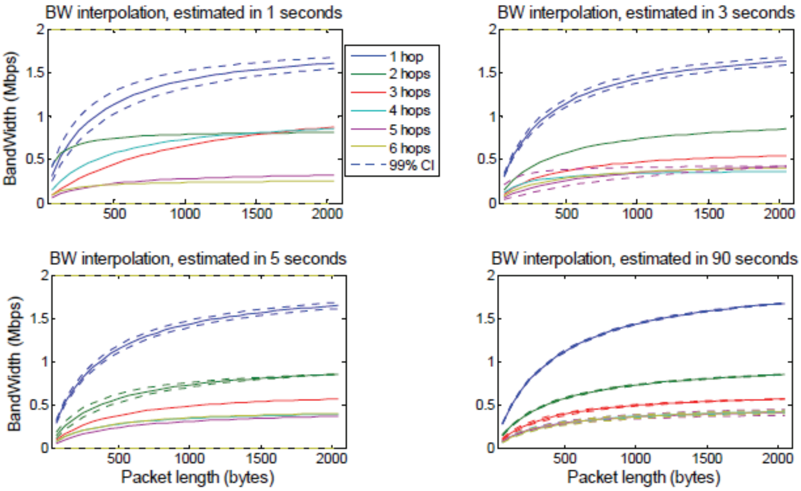


Fig. 11. Convergence speed in absence of cross traffic.

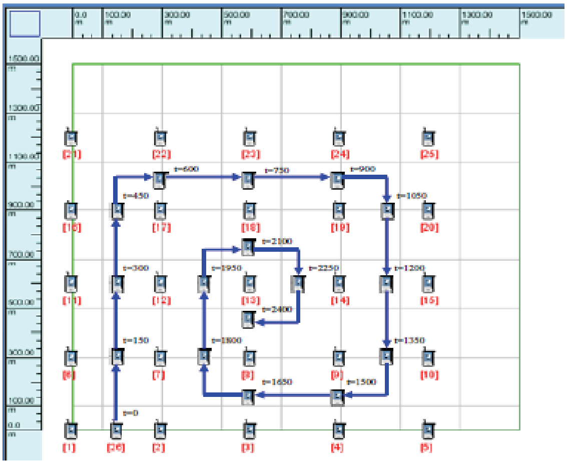


Fig. 12. Mobility scenario for adaptability test.

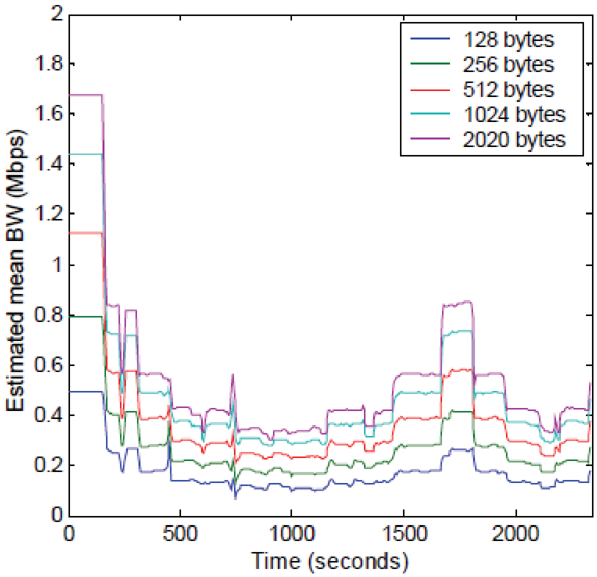


Fig. 13. Mean BW estimation under mobility.

breakdown and reestablishment epochs by inspecting Figure 13. These results show that, as long as the durations of the routes are in the order of several tens of seconds and the network is not highly loaded, the estimation scheme can offer high precision and good adaptability.

4. End-to-end available bandwidth estimation in multi-hop IEEE 802.11b ad hoc networks

In this section, the active probing technique that estimates the bandwidth (*BW*) of an end-to-end path in an IEEE 802.11b ad hoc network, shown in Section 3, is incorporated into a new neuro-fuzzy estimator to find the end-to-end available bandwidth, for which the theoretical definition of Equation 14 is an upper bound. The gaps between those pairs of packets that suffer the minimum sum of one-way delays are used to estimate the maximum achievable transmission rate (*BW*) as a function of the packet length, for any packet length, and then the variability of the dispersions is used to estimate the fraction of that bandwidth that is effectively available for data transmission, also as a function of packet length. However, instead of the perfect-scheduling and no-errors approximation of Equation 14, we consider implicitly all the phenomena that jointly affect the truly available bandwidth and the dispersion measures, using a neuro-fuzzy identification system to model their dependence. For example, even in the absence of competing flows, there can be self interference when consecutive packets of the same flow compete among them on different links of the same spatial channel within the path. Furthermore, cross-traffic can do more than taking away some *BW* of the path by interacting through MAC arbitration, as it can also reduce the signal-to-noise ratio at some parts of the path, or can even share some common queues along the path. Another largely ignored aspect that is indirectly captured by the neuro-fuzzy system is the fact that, once the unused *BW* is to be occupied, the arrival of the new flow can re-accommodate the occupation pattern along the neighborhood of its path.

In order to consider all these interacting aspects, the neuro-fuzzy estimator is trained on data collected from a large set of simulated scenarios, for which Qualnet®SNT (2007) is used. The scenarios were carefully selected to have enough samples of each of the effects mentioned above, and combinations of them, over a wide range of configuration parameters, so as to get a representative set of data for training, testing, and validation purposes. With all these data, the estimator learns how to infer the available bandwidth from the variability of the dispersion traces. The system is designed so as to have good generalization properties and to be computationally efficient. As a result, an accurate, efficient, and timely end-to-end available bandwidth estimator is obtained.

4.1 Practical ABW

The definition of ABW above (Equation 14) is an extended version of the widely accepted concept of unused capacity of the tight link. But in wireless multi-hop ad hoc networks this unused bandwidth can differ from the additional achievable transmission rate because, due to interference, the unused capacity may not be completely available. Indeed, once a new flow is established in the given path to occupy some of that unused capacity, the interfering cross-traffic can re-accommodate itself in response to the new flow, changing the perception of the new flow about its available bandwidth. So, it is tempting to define a practical ABW as the throughput achieved by a saturated source. However, due to self interference, a saturated node could reduce its throughput far below of what a less impatient source might obtain. Another practical ABW could also be defined as the maximum achievable transmission rate that does not disturb current flows, but this is a very elusive definition because, due to the interactions in the shared medium, even a very low rate new data flow could affect current flows.

Accordingly, an additional reasonable practical definition of ABW is the maximum throughput achievable by a CBR flow in the path, where the maximization is performed over the range of input data rates. Although, intuitively, this definition makes better sense, it is the most unfriendly for estimation purposes, because it requires the estimator to explore different transmission rates in order to find the one that maximizes the throughput. However, instead of doing this process on-line, it is possible to collect accurate and representative data to feed a machine learning process that would relate the statistics of the packet pair dispersion measures with the true maximum achievable rate in the path. First, an experiment to measure the probing packet dispersions is conducted and, then, the same experiment is replicated to measure the available bandwidth as the maximum achievable throughput. Then the ratio between the maximum achievable throughput and the bandwidth is computed (the "availability", $x = ABW/BW$), in order to relate it to the variability of the dispersion measures. The underlying hypothesis is that, since the probing packet pair dispersions are affected by the same phenomena that determines the current ABW, the costly search of an optimal input rate can be avoided if it can be inferred from the statistics of the dispersion trace. So, our definition of ABW would be given as follows:

$$ABW(L) = \max_{\lambda>0} \left[\lim_{t\to\infty} \frac{n(t;\lambda)L}{t - t_1} \right] \tag{27}$$

where L is the length, in bits, of the transmitted packets, $n(t;\lambda)$ is the number of packets received up to time t when they are sent at a transmission rate of λ packets per second, and t_1 is the reception time of the first received packet. To evaluate Equation 27 experimentally, a large number (1000) of packets is sent at the given rate λ. If the receiver gets less than 25% of the transmitted packets, the loss probability is considered too high and the available bandwidth is set to zero for that input rate. Otherwise, the throughput for this rate is computed as $\Gamma(L;\lambda) = (n-100)L/(t_n - t_{100})$, where n is the last received packet, which arrived at t_n. The first 100 received packets are considered part of a transient period. Then, through bracketing, the value of λ that maximizes Γ is found, which becomes our practical $ABW(L)$. Since it is possible to keep constant conditions in the experimental scenarios, this procedure gives a very accurate measure of ABW.

It is interesting to notice the relationship between Equations 14 and 27. In a wired network, they are supposed to be the same, where Equation 27 is oriented to a self-congesting estimation procedure while Equation 14 is oriented to a packet pair dispersion measure. Indeed, Equation 28 shows two "equivalent" definitions of ABW in the period $(t-\tau, t]$ for a single link of capacity C bps that serves a total traffic of $\lambda(s)$ bps at instant s, widely accepted as equivalent Prasad et al. (2003).

$$\begin{aligned} ABW(t-\tau,t) &= \frac{1}{\tau}\int_{t-\tau}^{t}(C-\lambda(s))ds \\ &= \arg\max_{R}\left(R \mid R + \frac{1}{\tau}\int_{t-\tau}^{t}\lambda(s)ds < C\right) \end{aligned} \tag{28}$$

However, from previous discussion, it is clear that they can be different in wireless ad hoc networks. In this section, the work is aimed at designing a system capable of learning, from sample data, the intricate relations between ABW, as defined in Equation 27, and dispersion measurements. So, a representative set of data must be collected in order to determine whether the dispersion measurements carry enough information for a significant estimation of ABW or not. If that is the case, that data could be used to train a neuro-fuzzy system.

4.2 Data collection and preprocessing

With the procedure described above, it is possible to collect a large data set that relates the dispersion measurements of the active probing packet pairs with the corresponding ABW on different scenarios. The data set must reflect the most important features of the underlying characteristics of any IEEE 802.11b ad hoc network, which include the interaction between competing flows by buffer sharing, by MAC arbitrated medium sharing, by capture effects or, simply, by increased noise.

All these aspects of the dynamic behavior of an IEEE 802.11b ad hoc network (and their combinations) are captured using the network configuration shown in Figure 14, where different parameters can be changed in order to explore a wide range of cross-traffic interference conditions. In the experiments, we varied the value of the distance between nodes (from 50 to 300 m), the physical transmission rate (1, 2 and 11 Mbps), the use of RTS/CTS mechanism, the number of cross-traffic flows (from 1 to 8), the origin and destination of each cross-traffic flow (uniformly distributed among the nodes), the transmission rate of each cross-traffic flow (from 50 kbps to 200 kbps), the packet length of each cross-traffic flow (64, 100, 750, 1400 and 2000 bytes), and the buffer size at the IP layer (50, 150 and 500 kbytes).

For each condition, the ABW between each of the 21 pairs of nodes of the second row was found, for four different packet lengths (100, 750, 1400 and 2000 bytes), averaged over 10 independent simulations. Then each experiment was replicated to take a dispersion trace of probing traffic for each measured ABW. This way 6000 samples were obtained, where each sample consisted of a traffic dispersion trace, a

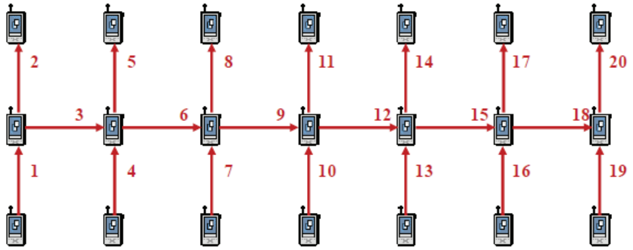


Fig. 14. Test scenario for data collection.

corresponding $BW(L)$ function, and four measured availabilities for four different packet lengths, $\{x(L_i) = ABW(L_i)/BW(L_i), i = 0\ldots 3\}$.

The traffic dispersion trace represents a huge amount of highly redundant data, from which the set of statistics that brings together most of the information about the availability $x(L)$ contained in the whole trace must be selected.

The traces were grouped in analysis windows of 200 packet pairs, overlapped every 4 pairs and, for each analysis window, the following statistics of the dispersion trace for the two probing-packet lengths, L_0 and L_1 were measured:

$$\theta_1(L_i) = \text{mean of the gap between packets of a pair of } L_i\text{-bit packets}$$
$$\theta_2(L_i) = \text{standard deviation of the gap between packets of a pair of } L_i\text{-bit packets}$$
$$\theta_3(L_i) = \text{mean of the } \textit{sowd} \text{ (sum of one way delays) of a pair of } L_i\text{-bit packets}$$
$$\theta_4(L_i) = \text{standard deviation of the } \textit{sowd} \text{ of a single pair of } L_i\text{-bit packets}$$

where, in each analysis window, the gaps and *sowds* are centered and normalized with respect to the gap between the packets that suffered the minimum *sowd*, in order to get comparable magnitudes over different network conditions. The vector of eight input parameters will be denoted as θ, while the vector of four input parameters corresponding to a given packet length L will be denoted a $\theta(L)$.

Figure 15 shows the probability density functions (pdf) of each component of $\theta(L)$ within the collected data for $L_1 = 1400$ bytes, conditioned on a low or high availability, where similar results hold for $L_0 = 100$ bytes. A low availability tends to increase the values of the parameters and disperse them over a wider range, as compared to a high availability. These remarkable differences in the conditional probabilities indicate the existence of important information about the availability $x(L)$ contained in this set of statistics, so the later can be used to classify and regress the former. It is this discrimination property what is to be exploited in the available bandwidth estimator.

4.3 Neuro-fuzzy system design

First, a fuzzy clustering algorithm is used to identify regions in the input space that show strong characteristics or predominant phenomena. Then, the clustered data is used to train simple neural networks, which can easily learn such phenomena. The local training data is selected through alpha-cuts of the corresponding fuzzy sets, and the antecedent membership functions are used to weight the outputs of the locally expert neural networks, according to the following simple rules:

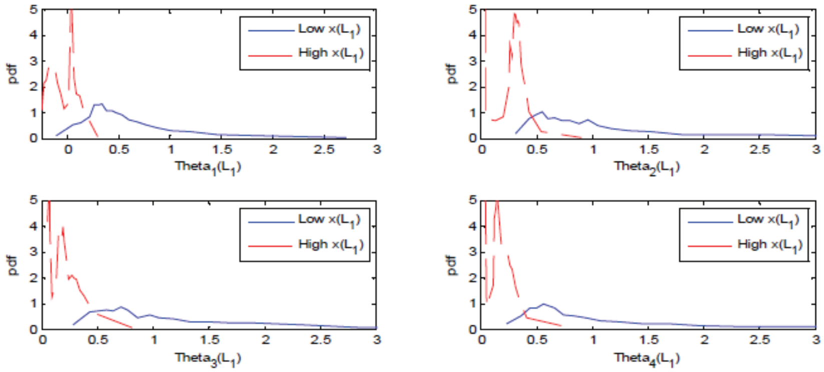


Fig. 15. Probability density functions of the measured statistics conditions on a high or low availability.

```
if $\theta$ is in cluster $j$ then
    $x(L_i) = \text{neuralnet}(i, j)$
end
```

A large number of clusters can give a high accuracy at a cost of a high computational complexity. Since the efficient use of computational resources is an important requirement for ad hoc wireless networks, two neural networks were locally trained on two different subsets of the input data. The local data was selected through a fuzzy *c*-means clustering algorithm on the whole set of input parameters. This choice leads to good regularity and generalization properties and a good compromise between bias and variance errors, while keeps a low computational complexity. The global model takes the following form

$$\hat{x}(L_i) = f_i(\theta \mid r_1)\mu_{r_1}(\theta) + f_i(\theta \mid r_2)\mu_{r_2}(\theta) \tag{29}$$

where $f_i(\theta \mid r_j)$ is the output of the locally expert network for L_i-bit packets in the j^{th} region, and $\mu_{r_j}(\theta)$ is the membership function of the set of input parameters in the j^{th} cluster. The neuro-fuzzy estimator, shown in Figure 16, estimates the availability for four different packet lengths (100, 750, 1400 and 2000 bytes).

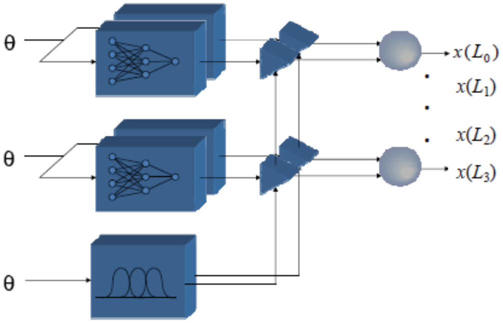


Fig. 16. Structure of the neuro-fuzzy estimator.

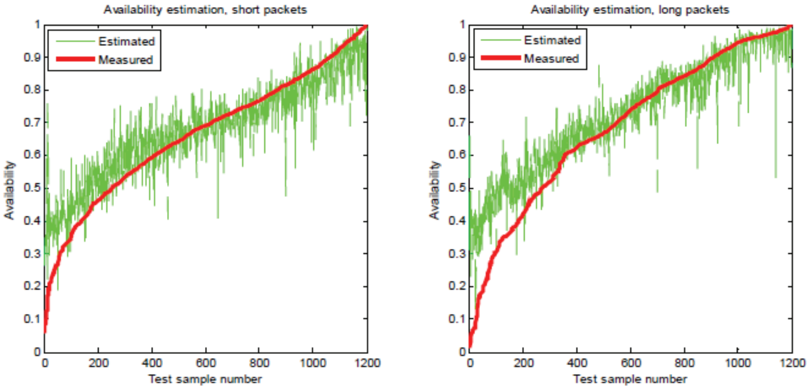


Fig. 17. Availability estimation on the test trace (data not seen by the system during training).

Locally trained submodels increased significantly the learning capacity of the whole system, as can be appreciated in Figure 17, which shows the final estimation results on the collected test data using the following settings: small packet size $L_0 = 100$ bytes, large packet size $L_1 = 1400$ bytes, time between pairs $T = 0.25$ seconds, analysis window size $W = 320$ packets. It can be noticed that, unless the network is heavy loaded, the estimation is highly accurate. Indeed, the accuracy is within 15% for more than 80% of the test samples (which were never seen during training) and, within 10% for more than 90% of the samples with availability greater than 0.5.

Having $BW(L)$ and four samples of the availability $x(L)$, $ABW(L)$ can be interpolated as a function of packet length by adjusting some appropriate functional form. In particular, if the ratio of lost packets is low, a form similar to Equation 25 should be selected, but if there is a high ratio of lost packets, it is assumed that longer packets have more chance to become corrupted. Consequently, the functional form of the ABW is assumed to be

$$ABW(L) = \alpha \cdot BW(L) \cdot \exp(-\lambda L) \tag{30}$$

It is possible to fit the function above to minimize the mean square error with the estimated ABW for the four test packet lengths, as obtained from the neuro-fuzzy estimator, which will allow an interpolated estimate for any packet length.

4.4 Numerical evaluation

Many validation experiments were conducted with several scenarios using different network sizes, mobility conditions, data transmission rates, cross traffic intensities and configuration parameters, all of them with very good results. We present here the experiment shown in Figure 18, where the available bandwidth between nodes 1 and 2 is to be found. The nodes are in a 1100 × 500 m area. All nodes transmit at 2 Mbps and use the RTS/CTS mechanism. Node 1 moves along the dotted trajectory at a constant speed of 2 m/s. Figure 18 shows some intermediate positions of node 1, requiring a path with one, two, three, and four hops. The link between nodes 3 and 4 carries a cross-traffic VBR flow that sends 2000-byte packets at an average rate of 750 Kbps. The results of the probing packet dispersion analysis are shown in Figure 19, where they are compared with the true available bandwidth, obtained by looking

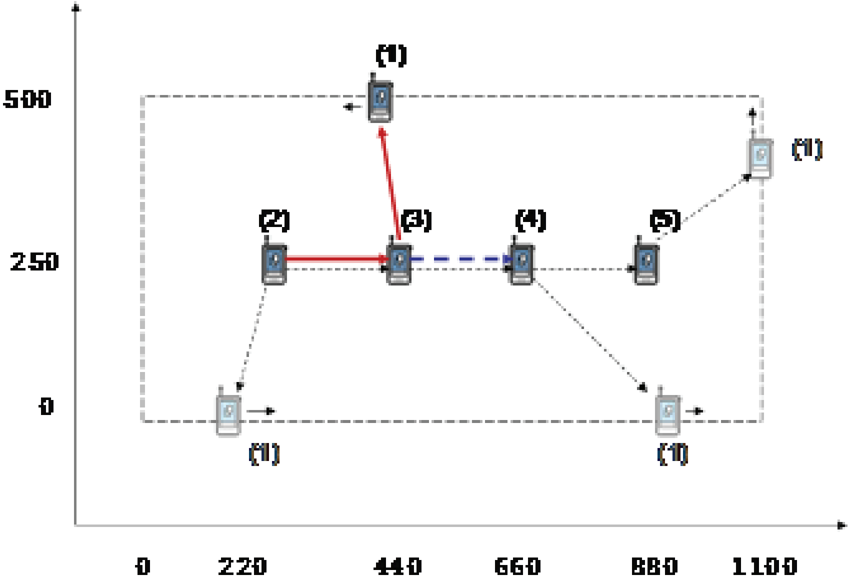


Fig. 18. Mobility scenario for testing the estimation method.

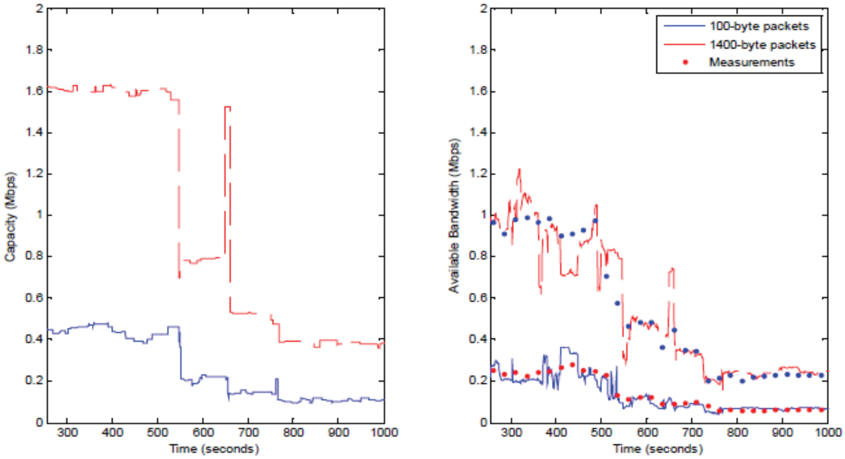


Fig. 19. Bandwidth and available bandwidth in the scenario of Figure 18.

for the maximum achievable throughput at different positions. Notice the high accuracy of the estimation and the detailed resolution of the *ABW* trace. This resolution is achieved by advancing 320-packet analysis windows every 8 packets. Under no losses, this is equivalent to obtaining, every second, the average *ABW* on the past 40 seconds.

5. Conclusions

In this paper we present new definitions of capacity (C), bandwidth (BW), and available bandwidth (ABW) for wireless ad hoc networks based on the concept of a spatial channel as the unit of communication resource, instead of the concept of a link, which is not clearly defined in this type of networks. The new definitions are natural extensions of the widely accepted ones in the sense that they become identical if each spatial channel was composed of a single link with no multi-access overhead, as in a point-to-point wired network. We verify the validity of these new definitions in the case of IEEE 802.11b ad hoc wireless networks, where the definitions become close upper bounds of the measured quantities under the assumption of perfect scheduling and no errors.

Then we present and evaluate an estimation procedure for the capacity and the bandwidth of an IEEE 802.11b multi-hop ad hoc path, according to the newly proposed definitions. The procedure is based on an active probing scheme that considers BW as a packet length dependent random variable. The pairs of packets that suffer the minimum delay are used to estimate the BW for two different probing packet lengths, and then the parameters of the functional form of BW are estimated, among which one parameter is the path capacity, C. Finally, the variability of the dispersion trace is fed to a neuro-fuzzy system in order to estimate the practical maximum throughput obtained over the range of input data rates. The theoretically defined ABW is an upper bound of the estimated quantity, obtained under the assumption of no transmission errors and no collisions. However, the estimation takes implicitly into account all the phenomena that jointly affects the variability of the dispersion trace and the maximum achievable transmission rate, including collisions and transmission errors, which are omitted in the theoretical definition of ABW as the unused BW.

We evaluate the performance of the estimation methods in terms of accuracy, convergence time, and adaptability to changing conditions, finding that they provide accurate and timely estimates in an efficient way, in terms of the use of both radio and computational resources.

6. References

Ahn, G.-S., Campbell, A., Veres, A. & Sun, L.-H. (2002). Swan: service differentiation in stateless wireless ad hoc networks, *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, Vol. 2, pp. 457–466.

Alzate, M. (2008). *End-to-End Available Bandwidth Estimation in IEEE 802.11b Ad Hoc Networks*, PhD thesis, Universidad de los Andes, Department of Electrical and Electronics Engineering.

Bianchi, G. (2000). Performance analysis of the ieee 802.11 distributed coordination function, *Selected Areas in Communications, IEEE Journal on* 18(3): 535–547.

Chaudet, C. & Lassous, I. G. (2002). Bruit: Bandwidth reservation under interferences influence, *Proc. of the European Wireless (EW02*, pp. 466–472.

Chen, C., Pei, C. & An, L. (2006). Available bandwidth estimation in ieee802.11b network based on non-intrusive measurement, *PDCAT '06: Proceedings of the Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies*, IEEE Computer Society, Washington, DC, USA, pp. 229–233.

Chen, K., Nahrstedt, K. & Vaidya, N. (2004). The utility of explicit rate-based flow control in mobile ad hoc networks, *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, Vol. 3, pp. 1921–1926.

Chen, L. & Heinzelman, W. B. (2005). Qos-aware routing based on bandwidth estimation for mobile ad hoc networks, *Selected Areas in Communications, IEEE Journal on* 23(3): 561–572.

Chen, L.-J., Sun, T., Yang, G., Sanadidi, M. Y. & Gerla, M. (2009). Adhoc probe: end-to-end capacity probing in wireless ad hoc networks, *Wirel. Netw.* 15(1): 111–126.

de Renesse, R., Ghassemian, M., Friderikos, V. & Aghvami, A. (2004). Qos enabled routing in mobile ad hoc networks, *3G Mobile Communication Technologies, 2004. 3G 2004. Fifth IEE International Conference on*, pp. 678–682.

Dovrolis, C., Ramanathan, P. & Moore, D. (2004). Packet-dispersion techniques and a capacity-estimation methodology, *IEEE/ACM Trans. Netw.* 12(6): 963–977.

Ephremides, A. (2002). Energy concerns in wireless networks, *Wireless Communications, IEEE* 9(4): 48–59.

Gamal, A., Mammen, J., Prabhakar, B. & Shah, D. (2004). Throughput-delay trade-off in wireless networks, *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, Vol. 1, pp. –475.

Gao, Y., Chiu, D.-M. & Lui, J. C. (2006). Determining the end-to-end throughput capacity in multi-hop networks: methodology and applications, *SIGMETRICS Perform. Eval. Rev.* 34(1): 39–50.

Grossglauser, M. & Tse, D. N. C. (2002). Mobility increases the capacity of ad hoc wireless networks, *IEEE/ACM Trans. Netw.* 10(4): 477–486.

Guerrero, C. G. & Labrador, M. A. (2009). Traceband: A fast, low overhead and accurate tool for available bandwidth estimation and monitoring, *Computer Networks* http://dx.doi.org/10.1016/j.comnet.2009.09.024.

Guha, D., Jo, S. K., Yang, S. B., Choi, K. J. & Lee, J. M. (2005). Implementation considerations of qos based extensions of aodv protocol for different p2p scenarios, *ICESS '05: Proceedings of the Second International Conference on Embedded Software and Systems*, IEEE Computer Society, Washington, DC, USA, pp. 471–476.

Gupta, P. & Kumar, P. (2000). The capacity of wireless networks, *Information Theory, IEEE Transactions on* 46(2): 388–404.

Hu, N. & Steenkiste, P. (2003). Evaluation and characterization of available bandwidth probing techniques, *Selected Areas in Communications, IEEE Journal on* 21(6): 879–894.

Jain, M. (2002). Pathload: A measurement tool for end-to-end available bandwidth, *Passive and Active Measurements (PAM) Workshop*, pp. 14–25.

Johnsson, A., Melander, B. & Björkman, M. (2004). Diettopp: A first implementation and evaluation of a new bandwidth measurement tool, *Swedish National Computer Networking Workshop*.

Johnsson, A., Melander, B. & Björkman, M. (2005). Bandwidth measurement in wireless network, *Technical report*, The Department of Computer Science and Electronics, Mälardalen University, Sweden.

Kapoor, R., Chen, L.-J., Lao, L., Gerla, M. & Sanadidi, M. Y. (2004). Capprobe: a simple and accurate capacity estimation technique, pp. 67–78.

Kashyap, A., Ganguly, S. & Das, S. R. (2007). A measurement-based approach to modeling link capacity in 802.11-based wireless networks, *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, ACM, New York, NY, USA, pp. 242–253.

Koksal, C. E., Jamieson, K., Telatar, E. & Thiran, P. (2006). Impacts of channel variability on link-level throughput in wireless networks, *SIGMETRICS Perform. Eval. Rev.*

34(1): 51–62.

Kumar, V. S. A., Marathe, M. V., Parthasarathy, S. & Srinivasan, A. (2005). Algorithmic aspects of capacity in wireless networks, *SIGMETRICS '05: Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, ACM, New York, NY, USA, pp. 133–144.

Kwak, B.-J., Song, N.-O. & Miller, L. E. (2005). Performance analysis of exponential backoff, *IEEE/ACM Trans. Netw.* 13(2): 343–355.

Lee, S.-B., Ahn, G.-S., Zhang, X. & Capbell, A. T. (2000). Insignia: an ip-based quality of service framework for mobile ad hoc networks, *J. Parallel Distrib. Comput.* 60(4): 374–406.

Melander, B., Bjorkman, M. & Gunningberg, P. (2000). A new end-to-end probing and analysis method for estimating bandwidth bottlenecks, *Global Telecommunications Conference, 2000. GLOBECOM '00. IEEE*, Vol. 1, pp. 415–420 vol.1.

Nahrstedt, K., Shah, S. H. & Chen, K. (2005). *Cross-Layer Architectures for Bandwidth Management in Wireless Networks*, Vol. 16, SpringerLink, chapter Resource Management in Wireless Networking, pp. 41–62.

Neely, M. J. & Modiano, E. (2005). Capacity and delay tradeoffs for ad hoc mobile networks, *IEEE Transactions on Information Theory* 51(10): 3687–3687.

Prasad, R., Dovrolis, C., Murray, M. & Claffy, K. (2003). Bandwidth estimation: metrics, measurement techniques, and tools, *Network, IEEE* 17(6): 27–35.

Renesse, R., Ghassemian, M., Friderikos, V. & Aghvami, A. (2005). Adaptive admission control for ad hoc and sensor networks providing quality of service, *Technical report*, Center for Telecommunications Research, KingŠs College London.

Ribeiro, V., Coates, M., Riedi, R., Sarvotham, S., Hendricks, B. & Baraniuk, R. G. (2000). Multifractal cross-traffic estimation, *ITC Conference on IP Traffic, Modeling and Management*.

Ribeiro, V. J., Riedi, R. H., Baraniuk, R. G., Navratil, J. & Cottrell, L. (2003). Pathchirp: Efficient available bandwidth estimation for network paths, *Passive and Active Measurement Workshop*.

Sarr, C., Chaudet, C., Chelius, G. & Guérin Lassous, I. (2006). Improving accuracy in available bandwidth estimation for 802.11-based ad hoc networks, *Research Report RR-5935*, INRIA.

Sarr, C., Chaudet, C., Chelius, G. & Lassous., I. G. (2005). A node-based available bandwidth evaluation in ieee 802.11 ad hoc networks, *ICPADS '05: Proceedings of the 11th International Conference on Parallel and Distributed Systems - Workshops*, IEEE Computer Society, Washington, DC, USA, pp. 68–72.

Shah, S. H., Chen, K. & Nahrstedt, K. (2003). Available bandwidth estimation in ieee 802.11-based wireless networks, *Proceedings of 1st ISMA/CAIDA Workshop on Bandwidth Estimation (BEst 2003)*.

SNT (2007). Qualnet simulator.
URL: *http://www.scalable-networks.com/*

Sollacher, R., Greiner, M. & Glauche, I. (2006). Impact of interference on the wireless ad-hoc networks capacity and topology, *Wirel. Netw.* 12(1): 53–61.

Strauss, J., Katabi, D. & Kaashoek, F. (2003). A measurement study of available bandwidth estimation tools, *IMC '03: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, ACM, New York, NY, USA, pp. 39–44.

Takai, M., Martin, J. & Bagrodia, R. (2001). Effects of wireless physical layer modeling in mobile ad hoc networks, *MobiHoc '01: Proceedings of the 2nd ACM international* *symposium on Mobile ad hoc networking & computing*, ACM, New York, NY, USA, pp. 87–94.

Xu, K., Tang, K., Bagrodia, R., Bereschinsky, M. & Gerla, M. (2003). Adaptive bandwidth management and qos provisioning in large scale ad hoc networks, *Military Comm. Conf. (MILCOM '03)*.

Yang, Y. & Kravets, R. (2005). Contention-aware admission control for ad hoc networks, *Mobile Computing, IEEE Transactions on* 4(4): 363–377.

21

QoS Routing Solutions for Mobile Ad Hoc Network

Jiwa Abdullah
*University Tun Hussein Onn Malaysia,
Malaysia*

1. Introduction

For the past decade, the field of mobile ad hoc networks (MANETs) [1] has been accepted as a legitimate area of research. It avoids the need for base station infrastructure by being able to self-organized and self-configuring. Hence it provides a spontaneous and yet robust wireless communication systems. Initially, MANETs researchers were focused mainly on designing distributed and dynamic communications protocols for shared channel and for route discovery. It offers best-effort protocols to ensure optimum network operation in an unpredictable wireless environment. Additionally it maintained a network topology view and routes in the face of disruption of links, failure mobile devices and short residual connectivity time. Nonetheless one could not actually experienced any successful practical implementation of MANETs in the real world. Entertainment and some other multimedia services usually made an impact on any technological breakthrough but the potential of MANETs are not truly realized. They must be able to deliver such services, for which best-effort protocols are not adequate. This is because multimedia applications often have stringent delay and reliability sensitive service requirements. Subsequently, the research focus has shifted from best-effort services to the provision of better defined QoS in MANET. QoS routing protocols then play an essential role in a QoS mechanism, since it is their task to find which nodes, if any, can serve an application's requirements. It plays a major part in session admission control (SAC), due to its dependence on the route discovery that support the requested QoS. Alternatively, some QoS routing solutions may not attempt to serve applications' requirements directly, rather to improve QoS under a particular metrics. Most of the solutions proposed in the literature, until now have focused on providing QoS based on throughput and delay. Throughput is the most common metric used. This is due to its character as the lowest common denominator requirement. It is noted that, most voice or video applications require some level of guaranteed throughput in addition to their other constraints. However, many other useful metrics are also used to quantify QoS. In this work we cover most of them and provide examples of their use. The remainder of this article is structured as follows. In Section 2 we discuss related work in terms of QoS routing surveys and summarize their main points. Section 3 describes the problem statement of QoS routing. Section 4 explains in detail the existing heuristics of QoS routing protocol. Section 5 describes a brief review of the challenges posed by the provision of QoS on the MANET environment. Section 6 presents the factors that need to be considered in designing a viable QoS routing protocol, QoS routing protocol performance, the network resources consumable by applications, and some of the trade-offs involved in protocol design. In Section 7 we describes some methods of classifying QoS routing solutions. Section 8 provides some examples of QoS routing protocols that rely on contention-free MAC. Section 9 describes the solutions for operating with a contended MAC. Finally, methods that do not rely on any specific kind of MAC are presented in Section 10. Under each section, we group protocols into different types of approaches, although for some approaches, only one example is provided. In Section 11 we described the QoS routing protocol utilising computational intelligence approach. We discuss our finding and the observed trends in the field of QoS routing in Section 12. Future works are described in Section 13 and finally Section 14 summarizes the chapter.

2. Related work

There are several overviews and surveys of QoS routing issues and solutions. Chen *et al* , provided a fairly comprehensive overview of QoS in networking [2]. Chakrabarti *et al* [3] summarized the important QoS-related issues in MANETs and subsequently produced an updated version [4]. A survey by Zhang *et al* [5], highlighted several significant points: (1) Most of the algorithmic problems, such as multi-constraint routing, have been shown to be NP-complete; (2) QoS and indeed Best Effort routing can only be successfully achieved if the network is stable. This means that the nodes are not moving faster than routing updates can propagate; (3) Techniques of QoS provisioning differ when the network size becomes very large, since QoS state updating mechanism takes longer time to propagate to distant nodes; (4) There is a trade-off between QoS provisioning and minimisation of power utilization. Areas of future work were also identified: (1) Admission control policies and protocols require further attention; (2) QoS robustness; (3) QoS routing protocol security against, for example, denial-of-service attacks. (4) The combination of security and QoS provisioning; (5) Study of QoS preservation under failure conditions; (6) QoS support for multicast applications. Mohapatra et al [56] provides a survey of issues in supporting QoS in MANETs. The paper considered a layered view of QoS provisioning in MANETs. In addition to the basic issues in QoS, it describes the efforts on QoS support at each of the layers, starting from the physical and going up to the application layer.

Al-Karaki et al [6], provided a detailed overview in the field of QoS routing. The following aspects were highlighted: (1) Accommodating multiple classes of traffic, but still allowing the propagation of lower-class traffic with the inclusion of preemptive scheduling; (2) Ensuring QoS guarantees under various failure conditions; (3) usage of localization devices such as GPS; (4) Prioritization of control packets; (5) Using realistic mobility models; (6) Quantifying the impact of cross-layer integration; (7) MANETs and Internet interoperability; (8) Secured QoS routing protocol, preventing malicious and harmful retransmission; (9) Network partitioning in the context of QoS routing; (10) Node heterogeneity in terms of their capacity and capabilities.

In this paper we focus on the essentials of QoS routing, which is the discovery of routes servicing data sessions and admission control. Al-Karaki *et al* [6] also discussed various QoS routing solutions which falls into the following categories: (1) flat, which means that all nodes perform an equal role; (2) hierarchical, where some nodes are group heads; (3) position-based protocol where location information is made available, and (4) power-aware in which battery usage and residual charge are considered. Reddy *et. al* [7] provide a thorough overview of the more widely accepted MAC and routing solutions for providing better QoS. The author provides two varieties of QoS solution, one is based on the QoS routing employed, while the other one is based on the layer at which they operate in the network protocol stack. The QoS routing employed is further classified into the mechanism which is based on (1) interaction between routing protocol. (2) QoS provisioning mechanism; (3) interaction between network and MAC layers; (4) on the routing information update mechanism employed that is on demand, table driven or hybrid.

3. QoS routing protocol: the problem descriptions

3.1 Goal of QoS routing

The main goal of QoS routing is to select, based on information about the state of the network, the path that is most suitable according to traffic requirements [8]. The maximization of network resource utilization is also an important goal of QoS routing. Hence, the QoS routing schemes must present solutions for metrics distribution mechanisms and path selection algorithm. Generally, QoS routing protocols search for routes with sufficient resources in order to satisfy the QoS requirements of a flow. The information regarding the availability of resources is managed by a resource management function which QoS routing protocol in its search for QoS feasible paths. The QoS routing protocol should find paths that consume minimum resources according to the relevant QoS metrics. Finding an optimal path with multiple constraints may be an NP-complete problem if it involves two or more metrics [9]. For a successful QoS routing operation, the topology information can be maintained at the nodes. The topology information needs to be refreshed frequently by sending link state update messages, which consume precious network resources such as bandwidth and battery power. Otherwise, the dynamically varying network topology may cause the topology information to become imprecise. This trade-off affects the performance of the QoS routing protocol. As path breaks occur frequently in MANET compared to wired networks where a link goes down very rarely, the path satisfying the QoS requirements needs to be recomputed every time the current path gets broken. The QoS routing protocol should respond quickly in case of path breaks and recompute the broken path or bypass the broken link without degrading the level of QoS. In the literature, numerous routing protocols have been proposed for finding QoS paths.

3.2 Mechanism for metrics distribution

The state of the network can be represented by a set of metrics, which includes the available bandwidth, delay, jitter, and congestion level. Traffic requirements can be expressed in several ways, depending on the methodology used for traffic characterization. For instance, in the Integrated Services framework, this can be done using the QoS parameters associated with each data flow during resource reservation [10][11]. In the Differentiated Services framework, traffic requirements are associated with each traffic class [12]. The information about the state of the network must be distributed, and kept updated, to all or some routers in the network. The distribution must be done more frequently than in traditional routing, to reflect the dynamic behavior of the network. However, if this frequency is too high, it will induce too much bandwidth consumption, and it is thus undesirable. In these situations, it is advisable to achieve a compromise between the desired actuality of the state information and the overhead that this introduces. Some approaches to this problem include the distribution of quantified values, instead of instantaneous ones. Associated with this value quantification, triggers may be used to control the emission of updates and timers to force a minimum interval between the emission of updates. [13]. A problem that relates to the frequency of the distribution of information pertaining to the state of the network is the inaccuracy that a lower frequency can introduce. Other sources of inaccuracy are the propagation delay of routing messages in large networks, the utilization of estimates, the impact of the metrics measurement mechanism used and information aggregation in hierarchical systems. The study of the impact of routing information inaccuracy on the performance of communication systems and the definition of the mechanisms to overcome its problems has been the subject of several research projects [14] [15] [16] [17].

3.3 Path selection algorithm

The path selection algorithm has a degree of complexity that depends on various factors. Since applications generate traffic with very diverse requirements in terms of QoS, the path selection algorithm must select paths that satisfy a set of restrictions. This is however, a problem with high computational complexity, depending on the rule of metrics composition. The value of a metric along a path, based on its value in each hop, depends on the nature of the metric. There is additive, multiplicative and concave metrics. The rule for additive metrics composition is that the value of this metric over a path is the sum of the values of each hop. Delay and number of hops are examples of additive metrics. With a multiplicative metric, the value of the metric over a path is the product of its values in each hop, as it is the case of losses. The value of a concave metric over a path corresponds to the minimum value observed in all hops of that path. Bandwidth is a common example of a concave metric. In these equations, $m(l_i)$ is the value of a metric on link l, and $m(p)$ is the total metric value of the path composed of links l_1 to l_n. The problem of QoS routing when using two additive or multiplicative metrics, or one additive and one multiplicative metrics is a NP-complete problem [9]. This poses a challenge that must be addressed in order to conceive QoS routing strategies that are efficient and scalable.

4. Existing heuristics for QoS routing

The heuristics of QoS routing can be characterized by several aspects, including the metrics, type of path selection algorithm, instant of application of the path selection algorithm and localisation of the routing decision. In this paper we use as the main characterisation feature, the metrics for path selection, because it is an attribute that determines most of the other aspects. Bandwidth is widely used as a metric for QoS routing, alone or associated with other metrics, such as delay [10][9] and number of hops [13]. It is usually coupled with systems where traffic differentiation is done at the flow level, with the specification of path QoS parameters.

4.1 Metric ordering

Metric ordering requires the identification of highest priority metric and then compute the best paths for it. Where more than one best paths, second metric is invoked, to choose the best path. It is a kind of shortest-widest path and widest-shortest path algorithms. In shortest-widest path algorithms, paths with maximum available bandwidth is located. If there are paths of the same available bandwidth, it would then select the path with shortest number of hops. These algorithms support load balancing, showing top performance with low network loading. However, this approach damages best-effort traffic performance because it contributes to resource consumption. Another shortest-widest path algorithm uses propagation delay, as the second metric. Wang *et al* presented the related path computation algorithms which are based on distance-vector and link-state [9].

4.2 Sequential filtering

In sequential filtering, the network links that do not have enough available bandwidth are excluded from the network graph. The shortest path is then computed. For on-demand path computation, bandwidth value is obtained on request through resource reservation protocol. If paths are pre-computed, bandwidth ranges must be established. On-demand path computation requires parameter specification. For path pre-computation it is necessary to compute and store several pre-computed paths that satisfy the defined range of bandwidth values. Sequential filtering can also be used to find paths subject to more than two constraints. An example is the cheapest-shortest-feasible path algorithm presented by A. Shaikh *et al*[15]. This source routing algorithm aims at finding feasible paths according to a bandwidth constraint, minimizing simultaneously cost and resource consumption.

4.3 Scheduling disciplines

The complexity of route selection algorithms can be overcome by using the relationships among QoS parameters, determined by the nature of scheduling disciplines. In particular, when a Weighted Fair Queuing (WFQ) scheduling mechanism is used, it is possible to find a route, in polynomial time, subject to constraints of delay, jitter and bandwidth [18]. WFQ is a rate proportional scheduling discipline that isolates each guaranteed session from the others. It also has delay bounds that can be mathematically determined.

4.4 Admission control

In some QoS architectures, the admission of new flows in the network is subject to a mechanism of admission control. This mechanism interacts closely with routing. Typically, the routing module can produce information about the network state which contribute to admission control decision [12]. Admission control and QoS routing are connected to resource reservation. The resource reservation protocol can express the flow QoS requirements that are used by the QoS routing protocol to compute suitable paths. The resource reservation protocol can then proceed to flow establishment on the paths produced by the QoS routing algorithm. If this establishment is successful, the flow is accepted; otherwise it is rejected. Rampal in reference [19] presented path computation algorithms that considered QoS requirements and admission control restrictions of multimedia traffic. These algorithms used information associated with the admission control module, the minimum delay and probability of rejection by the admission control module. This information is used for pruning from the network graph the links that do not satisfy admission control restrictions. The remaining graph is then presented to the routing algorithm.

4.5 Control theory approach

Control theory approach offers a successful track record in physical process control. It gives somewhat a performance guarantees in the face of uncertainty, non-linearities and time-variations system. It does not require accurate system models and utilise feedback mechanism. Performance of software services is governed by queuing dynamics which may be expressed by differential equations akin to those of physical systems. Bao Li *et al* [20] presented a model for QoS mechanism employing feedback control theory. The ideal objectives of the model consist of two fold. First, it could accommodate variable QoS requirements, in a timely fashion. Second, it could accommodate concurrency of resource access among multiple applications sharing the same pool of available resources.

4.6 Computational intelligence approach

QoS routing is a key MANET function for the transmission and distribution of multimedia services. It has two objectives; (1) finding routes that satisfy QoS constraints and, (2) making efficient use of limited resources. The complexity involved in the networks may require the considerations of multiple objectives at the same time, for the routing decision process. In this paper we also introduced the use of Fuzzy Logic [21] and Genetic Algorithm based QoS routing for MANET [22].

5. Challenges to QoS routing mechanism

The following is a summary of the major challenges in providing QoS routing mechanism for MANETs.

5.1 Unreliable wireless channel

The wireless channel is prone to bit errors due to interference from other transmissions, thermal noise, shadowing and multi-path fading effects [23]. This makes it impossible to provide hard packet delivery ratio or link longevity guarantees.

5.2 Dynamic topology

The issue of mobility does not exist in fixed wireline networks and in infrastructured wireless networks. The topology of MANET will change dynamically due to mobile host changing their point of connectivity, the rate of node survivability and nodes leaving or joining the network. Saving current knowledge of the network topology and the frequent changes is an important requirement in MANET management system. However the frequent exchanges of topology information may lead to considerable signaling overhead, congesting low bandwidth wireless links, and possibly depleting the limited battery life of the nodes involved. Hence the choice of mechanism used to collect topology information is critical. These complications imposed by mobility in MANETs may severely degrade the network quality. The frequent route breakage is a natural consequence of mobility, which complicates routing. As a result, design of QoS routing protocols in MANETs is challenged by frequent topological changes.

5.3 Node mobility

The nodes in a MANET may move completely independently and randomly as far as the communications protocols are concerned. This means that topology information has a limited lifetime and must be updated frequently to allow data packets to be routed to their destinations. Again, this invalidates any hard packet delivery ratio or link stability guarantees. Furthermore, QoS state which is link-position dependent or node position dependent must be updated with a frequency that increases with node mobility. An important general assumption must also be stated here: for any routing protocol to be able to function properly, the rate of topology change must not be greater than the rate of state information propagation. Otherwise, the routing information will always be stale and routing will be inefficient or could even fail completely. This applies equally to QoS state and QoS route information. A network that satisfies this condition is said to be *combinatorially stable* [3].

5.4 Lack of centralized control

The major advantage of an ad hoc network is that it may be set up spontaneously, without planning and its members can change dynamically. This makes it difficult to provide any form of centralised control. As such, communications protocols which utilise only locally-available state and operate in a completely distributed manner, are preferred [24]. This generally increases an algorithm's overhead and complexity, as QoS state information must be disseminated efficiently.

5.5 Channel contention

In order to discover network topology, nodes in a MANET must communicate on a common channel. However, this introduces the problems of interference and channel contention. For peer-to-peer data communications these can be avoided in various ways. One way is to attempt global clock synchronization and use a TDMA-based system where each node may transmit at a predefined time. This is difficult to achieve due to the lack of a central controller, node mobility and the complexity and overhead involved [25]. Other ways are to use a different frequency band or spreading code (as in CDMA) for each transmitter. This requires a distributed channel selection mechanism as well as the dissemination of channel information. However data communications take place, without a central controller, some set-up, new neighbour discovery and control operations must take place on a common contended channel. Indeed, avoiding the aforementioned complications, much MANET research, as well as the currently most popular wireless ad hoc networking technology (802.11x) is based on fully-contended access to a common channel with Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA). However, CSMA/CA greatly complicates the calculation of potential throughput and packet delay, compared to TDMA-based approaches.

5.6 Heterogeneity

MANETs are typically heterogeneous networks with various types of mobile nodes with diverse nature of communication technologies employed. Its diversity comes in the form of different types of nodes, ranging from sensors, palmtops and laptops within an organisation or a result of multiorganisation consortium. In a military application, different military units ranging from soldiers to tanks can come together, hence forming a MANET system. Nodes differ in their energy capacities and computational abilities. Hence, mobile nodes will have different packet generation rates, routing responsibilities, network activities and energy draining rates. Coping with node heterogeneity is a key factor for the successful operation of MANETs.

5.7 Imprecise state information

In most cases, the nodes in a MANET maintain both the link-specific state information and flow-specific state information. The link-specific state information includes bandwidth, delay, delay jitter, loss rate, error rate, stability, cost, and distance values for each link. The flow specific information includes session ID, source address, destination address, and QoS requirements of the flow (such as maximum bandwidth requirement, minimum bandwidth requirement, maximum delay, and maximum delay jitter). The state information is inherently imprecise due to dynamic changes in network topology and channel characteristics. Hence routing decisions may not be accurate, resulting in some of the real time packets missing their deadlines.

5.8 Hidden terminal problem

The hidden terminal problem is inherent in MANETs. This problem occurs when packets originating from two or more sender nodes, which are not within the direct transmission range of each other, collide at a common receiver node. It necessitates retransmission of packets, which may not be acceptable for flows that have stringent QoS requirements. The RTS/CTS control packet exchange mechanism, proposed in [26] and adopted later in the IEEE 802.11 standard [27], reduces the hidden terminal problem only to a certain extent. BTMA [28] and DBTMA [29] provide two important solutions for this problem.

5.9 Insecure medium

Due to the broadcast nature of the wireless medium, communication through a wireless channel is highly insecure. Hence security is an important issue in MANETs, especially for military and tactical applications. MANETs are susceptible to attacks such as eavesdropping, spoofing, denial of service, message distortion, and impersonation. Without sophisticated security mechanisms, it is very difficult to provide secure communication guarantees.

6. Factors to be considered in designing QoS routing protocol

6.1 QoS requirements specifications

For efficient QoS routing implementation, QoS requirements needs to be specified to the routing protocol. Consequently, they may be used as constraints on route discovery and selection. An application may typically request a particular QOS by specifying its requirements in terms of one or more of the following metrics.

i. Minimum throughput or capacity (bps); which is the desired application data throughput. [30];
ii. Maximum tolerable delay; normally defined as the maximum tolerable source to destination delay for data packets transmission[2];
iii. Maximum tolerable delay jitter, which is the difference between the upper bound on end-to-end delay and the absolute minimum delay [31]. This metric can also be expressed as delay variance [32];
iv. Maximum tolerable packet loss ratio (PLR) (%) which is the acceptable percentage of total packets sent, which are not received by the transport layer agent at the packet's destination node [33];

In most cases, the QoS protocol should only admit this data session into the network if it can provide the requested service. The mechanism by which this decision is made is termed admission control.

6.2 Metrics employed for route selection

This section lists many of the metrics commonly employed by routing protocols for path evaluation and selection in order to improve all-round QoS or to meet the specific requirements of application data sessions.

i. Network Layer Metrics

a. Achievable throughput; which is defined as the achievable data throughput of a path or node. The achievable throughput is often termed as the available bandwidth. [30]
b. End-to-end delay; which refers to the measured end-to-end delay on a route [2];
c. Node buffer space, the number of packets in a node's transmission buffer which determine the amount of delay a packet traveling through a node [34];
d. Delay jitter or variance[31].
e. Packet loss ratio (PLR) (%) which is the percentage of total packets sent, which is not received by the transport or higher layer agent at the packet's final destination node;
f. Route lifetime (s) which is the statistically calculated as expected lifetime of a route. Normally it depends on node mobility as well as node battery. [35];

ii. Medium Access Control(MAC) and Link Layer Metrics:

a. The time taken to transmit a packet between two nodes in a contention-based system is the MAC time delay. It includes the total time deferred and acknowledgement delay [36]. It provides an indication of packet traffic.
b. Frame delivery ratio (%) which is statistically determined, is the probability of a packet successfully being transmitted over a link and correctly decoded at the receiver [37][38].
c. Link stability which can be described as the predicted lifetime of a node pair connection [35]. It indicates the length of time node pairs are connected;
d. Node relative mobility can be measured as the ratio of the number of neighbours that change over a fixed period to the number that remain the same [39].

iii. Physical Layer Metrics:

a. Signal-to-interference ratio (SIR), where the received SIR at a destination node can be used as a routing metric indicating link quality, via cross-layer communication [40].
b. Bit error rate (BER) determines the level of error correction and/or number of retransmissions required over a connection and has major impact on the connection's reliability metric and on energy consumption. [41];
c. Node residual battery charge or cost [42] [43] [37].

QoS metrics such as the above can be classified as either additive, concave or multiplicative metrics, based on their mathematical properties [7].

6.3 Metrics for performance evaluations

The following metrics may be used to evaluate a QoS routing protocol's performance.

1. Transport/Application Layer:

a. Session acceptance/blocking ratio is the percentage of application data sessions (or transport layer connections) that are admitted into or rejected from the network. The value of this metric reflects both the effectiveness of the QoS protocols as well as conditions outside of their control, such as channel quality;
b. Session completion/dropping ratio is the metric represents the percentage of applications that were successfully/ unsuccessfully served after being admitted to the network. For example, if a VoIP session is accepted and the session is completed properly (by the users hanging up) and not aborted due to route failure or any other error, then that counts as a completed session.

2. Network Layer*:*

a. Throughput measured in bit per second(bps) which is the amount of data traffic the entire network carried to its destination in one second;
b. Node throughput (bps), defined as the average throughput achieved by a single node;
c. Route discovery delay for reactive protocols as a measure of effectiveness of the reactive protocols.
d. In order to measure the operating cost and efficiency of Qos routing protocol, normalised routing load (NRL) could be employed. It is the ratio of routing packets transmitted to data packets received at the destination [44].
e. Network lifetime (s) may be defined as the time until network partitioning occurs due to node failure [42], indicating the energy-efficiency and load balancing ability of the protocol.
f. Average node lifetime (s) [42] shows the effective power usage and optimization.

3. MAC Layer

a. Normalised MAC load is very similar to the normalised routing load (NRL), which represents the ratio of bits sent as MAC control frames to the bits of user data frames transmitted [44].
b. MAC energy efficiency, represent the ratio of energy used for sending data bits to the total energy expended for data plus MAC headers and control frames;

6.4 Factors affecting QoS routing protocol performance

When evaluating the performance of QoS protocols, a number of factors have a major impact on the results. Some of these parameters are a particular manifestation of characteristics of the MANET environment.

1. Node mobility which consists of a number of parameters: the nodes' maximum and minimum velocity, velocity pattern and pause time. The node's velocity pattern determines whether the node moves at uniform velocity at all times or whether it is constantly varying, and also how it accelerates. The pause time determines the length of time, nodes remain stationary between each period of motion. Together with maximum and minimum velocity, this parameter determines how often the network topology changes [44], [45].
2. Network size, the larger the network, the more difficult this becomes in terms of update latency and message overhead [24].
3. Number, type and data rate of traffic sources, a smaller number of traffic sources results in fewer routes being required and vice-versa. Traffic sources can be constant bit rate (CBR) or may generate bits or packets at a rate that varies with time according to the Poisson distribution, or any other mathematical model. The maximum data rate affects the number of packets in the network and hence the network load [44].
4. Node transmission power - some nodes may have the ability to vary their transmission power. This is important, since at a higher power, nodes have more direct neighbours and hence connectivity increases, but the interference between nodes does as well. Transmission power control can also result in unidirectional links between nodes [46], [47], [48].
5. Channel characteristics - as detailed earlier, there are many reasons for the wireless channel being unreliable i.e. many reasons why bits, and hence data packets, may not be delivered correctly. These all affect the network's ability to provide QoS.

6.5 Networking resources utilization for effective QoS routing protocol

1. Node computing time
   Today, mobile devices are manufactured with increasingly powerful processors but are still limited in computing power. It may be that they must not only run the applications, but also protocols necessary to support the network.
2. Node battery charge
   It may be the most critical resource. Node failures due to battery drained, can cause network partitioning, leading to a complete network failure. Hence, power-aware and energy efficient MAC and routing protocols have received a great deal of research attention [42][48].
3. Node buffer space
   Node buffer space must be always available to reduce packet transmission congestion. Data packets must be buffered while awaiting transmission and reception. Furthermore, when the buffers are full, any newly arriving packets must be dropped, contributing to the packet loss rate;
4. Channel capacity
   Channel capacity is measured in bps and affects data throughput. Indirectly, it contribute to the delay, and other metrics too. However, since all nodes must share the transmission medium, we must somehow express the fraction of the medium's total capacity that is granted for each node's use. The way to express this depends on the MAC layer technique employed. In a purely contention-based MAC, transmission opportunities may be envisioned, although no node can be guaranteed channel access, merely granted it with a certain probability.

6.6 A balanced trade-offs design

This section discusses some of the common trade-offs involved in QoS routing protocol design.

1. Route discovery and state dissemination, Proactive, On-Demand or Hybrid
   It refers to two problems under one heading. Firstly, should routes be discovered proactively or on-demand? Secondly, how should QoS state required for path selection, be discovered? If both the route and QoS state discovery mechanisms are proactive, then the session establishment time is greatly reduced from an application's point of view. Also, a proactive protocol is largely unaffected by an increase in the fraction of nodes acting as data sources, since routes to all destinations are maintained anyway. However, a large overhead is incurred in keeping routes and state up-to-date, especially in highly mobile scenarios. Additionally, such a mechanism does not scale well with an increasing number of nodes. These are well-known problems of proactive protocols [24]. A major advantage of discovering QoS state proactively surfaces in situations where different applications specify their requirements with different metrics. As long as it is decided which QoS states to keep up to date, a route may be computed from the routing table based on any QoS metric, without the need for a separate discovery process for each metric [49]. A purely reactive routing solution avoids the potential wastage of channel capacity and energy by not discovering routes and QoS state which are not currently needed. However, a discovery delay is incurred when an application requires a route to a destination [39].
2. Between Capacity and Delay
   It has been shown that in MANET, capacity can be traded off with packet delay [50][51]. If delay constraints are relaxed, then the capacity of the network can be

increased by exploiting multiuser diversity [50]. More specifically, if delay is not constrained, a source can split the packets of a session and send them to many different neighbours. These neighbours then forward the packets onto the destination when they move into its transmission range. This scheme has been shown to improve throughput, since far fewer intermediate nodes are transmitting packets and causing interference, but incurs the cost of greatly increased delay [50]. Another strategy is to improve delay by increasing redundancy, at the cost of network capacity utilization efficiency [51]. If multiple copies of a packet are forwarded on multiple paths, it has been shown that the destination receives the packet with less delay on average. On the other hand, more network capacity is consumed in sending duplicate packets [51]. Clearly, increased redundancy also reduces the protocol's energy efficiency.

3. The packet loss rate against the capacity and energy-efficiency
   In a similar way to the trade-off between delay and capacity, PLR can also be traded off against capacity. Increasing the redundancy by sending mueltiple copies of packets over different routes, results in a higher chance of the destination receiving a copy, but reduces the useful capacity of the network. This technique can be more useful in sensor networks where data is often broadcast without a reliable handshaking protocol being employed at the MAC layer. Once again, redundancy also increases the energy expended per packet.
4. Energy consumption vs. responsiveness and accuracy of QoS state information
   Routing can only be accurate if the frequency of neighbour discovery is high enough to reflect frequent topological changes. However, a high-responsiveness to change comes at an increased energy cost [52]. If we consider QoS routing, this tradeoff between accuracy and energy consumption is even more acute, since not only the topology view, but the QoS state information also requires frequent updating, to enable accurate QoS routing decisions to be made.
5. Transmission power control between long and short hops
   Varying the transmission power to adjust the number of hops required to forward a packet to its destination, can yield many advantages and drawbacks. This has often been called the .long hops vs. short hops dilemma [53]. Another question is whether protocol designers should assume the use of transmission power control (TPC) at all. Assuming TPC constrains the type of devices that can be employed, since not all nodes may be equipped with radios with TPC capability. Furthermore, employing TPC can often result in uni-directional links. For example, a node X may be able to transmit to a node Y, but Y cannot reply since it is using a lower transmission power, unless it knows the distance to X and can calculate the transmission power required to reach it.
6. Global goals or individual requirements
   In the eyes of a network designer, the goal is to please, by providing an all-round high QoS. The secondary goal is to increase the network lifetime, by proper management of the battery usage. However, each individual user or data session has its own specific requirements, and to satisfy the user, the network must match their requirements. In more complicated scenarios, an application may specify a variety of QoS constraints. For example, it may specify maximum tolerable values for PLR as well as packet delay. In this case, we desire the routing protocol to find a stable path with a light traffic load. However, from a network lifetime point of view, a path that has the least cost, is preferred. Our goal of low delay matches the aim of load balancing, although the path with the least traffic may not be a stable path. In this case, there’s a clear conflict between various requirements. A protocol designer must decide how to address this trade-off.

6.7 Measurement techniques of all QoS routing parameters

1. Bandwidth

   The challenge in wireless ad hoc networks is that neighboring hosts must share the bandwidth, and there is no centralized control for allocating bandwidth among the nodes. Furthermore, intermediate hosts take part in forwarding packets. Therefore, the total effective capacity achievable is not only limited by the raw channel capacity, but is also limited by the interaction and interference among neighboring hosts. Thus, in order to offer bandwidth-guaranteed routing, bandwidth estimation is needed, yet accurately estimating available bandwidth at each host is a challenging problem. Most QoS-aware routing protocols, such as CEDAR, Ticket-based QoS Routing, ADQR and TDR, assume that the available bandwidth is known. However, some routing protocols try to propose an appropriate way to estimate the available bandwidth, such as OLSR-based QoS routing, AQDR, DSDV/TDMA and Reactive/TDMA. Various methods are proposed in these protocols for estimating the available bandwidth at the nodes.

   a. Exploit the carrier-sense capability of IEEE 802.11 and measure the idle and busy time ratio as used in OLSR-based QoS routing protocol and the QOSRGA [22].
   b. Add bandwidth consumption information to AODV routing packets (or Hello messages) and exchange this information with neighbor hosts (used in AQOR).
   c. Monitor and schedule free time slots using a TDMA scheme (used in DSDV / TDMA and Reactive/TDMA).
   d. Broadcast queries with limited hop count to actively contact all neighbors in the carrier-sensing range (used in CACP [25]-Multihop).
   e. Take advantage of power control and send queries to cover the carrier-sensing range (used in CACP-Power).
   f. Approximate the available bandwidth by using a moving average (used in CACP-CS). A drawback of AQOR's bandwidth estimation method is that it assumes that the interference range is same as the transmission range, which is not true in general. Thus AQOR's bandwidth estimation method will not correctly incorporate the bandwidth being used by neighbours in the interference range of the node.[54]

   The available bandwidth depends on the MAC scheduling, and several of the bandwidth estimation techniques currently proposed are associated with the underlying MAC protocols. Therefore, bandwidth estimation should be done with the assistance of the MAC protocol. A cross-layer design between the MAC and routing layers is the key to solve this problem.

2. Delay

   Only two routing protocols incorporate delay estimation: Ticket-based QoS aware routing and AQOR [54]. Ticket based QoS aware routing does not support a specified delay; it only determines the shortest delay route during route discovery. AQOR uses half the round-trip time of the route discovery process as the estimated path delay. These two schemes do not consider that changes in contention levels will impact the end-to-end delay significantly after the flow is started. Also, the effect of intra-flow contention on delay has not been sufficiently studied. Therefore, the second open issue in QoS-aware routing is: how should end-to-end delay be estimated to support delay-constrained real-time data transmission?

3. Jitter

   Jitter is the variation of delay over a period of time. Among the delay components are fixed delay components and variable delay components. Jitter results from the variable delay components, specifically changes in queuing delays at network switches due to variations in the short term network load. Jitter is the statistical variance of the packet interarrival time. The IETF in RFC 1889 [55] define the jitter to be the mean deviation of the packet spacing change between the sender and the receiver. A node sends packets of identical size at constant intervals which implies that $\boldsymbol{S_j}$ *and* $\boldsymbol{S_i}$ that is sending times of two consecutive packets is constant. The difference of the packet spacing, denoted *D*, is used to calculate the interarrival jitter. According to the RFC, the interarrival jitter should be calculated continuously as each packet *i* is received. The jitter of a packet stream is defined as the mean deviation of the difference in packet spacing at the receiver compared to the sender, for a pair of packets. If $\boldsymbol{S_i}$ is the time packet $\boldsymbol{i}$ was sent from the sender, and $\boldsymbol{R_i}$ is the time it was received by the receiver, the jitter sample $\boldsymbol{J_i}$ is given by, $J_1 = (R_{i+1} - R_i) - (S_{i+1} - S_i)$ or $J_1 = (R_{i+1} - S_{i+1}) - (R_i - S_i)$ and the average jitter is the average value over $\boldsymbol{n}$ packets. The jitter is particularly important quantity for time sensitive data such as real-time audio and video, since a large jitter can have a profound effect on the perceived quality.

4. Packet Loss

   The monitor program uses the sequence number, the time stamp, and the local time information to determine the two qos parameters: Packet Loss Rate, $\boldsymbol{L_{r,}}$, delay jitter, $\boldsymbol{J_t}$. During each measuring period, it counts the total number of packet received, $\boldsymbol{N_{total}}$ and the total number of packet loss, $\boldsymbol{n_{lost}}$. It also record the arrival and send times of the last packet in the measuring period, as $\boldsymbol{t_{last\_arrival}}$ and $\boldsymbol{t_{last\_send}}$. The arrival time taken to be the simulation time when the packet arrive to the receiver, while the send time is derived from the packet using the time stamp function. At the end of every period $\boldsymbol{k}$, the network monitor computes the two qos parameters with the following calculations, $L_r(k) = n_{lost}(k)/n_{total}(k)$.

5. Power

   A lot of work on energy efficient routing in MANET has been done [42]. These efforts tried to maximize the time for network partition and reduces variations in power level of the nodes. The model proposed by Rishiwal *et al* [56] can be used to calculate energy values at different times. Energy consumption of a node after time $\boldsymbol{t}$ is calculated as follows: $E_{c(t)} = N_t * \alpha + N_r * \beta$ where $E_{c(t)}$ is the energy consumed by a node; N_t is the number of packets transmitted by the node after time t; N_r is the number of packets received by the node after time $\boldsymbol{t}$; α and β are constant factors having a value between 0 and 1. Hence by using this formula, the energy usage can be known.

7. Categorisation of existing QoS routing protocol

In [6], QoS routing protocols are classified mainly by their: (i) treatment of network topology, which is either flat, hierarchical or location-aware, and (ii) approach to route discovery, either proactive, reactive, hybrid, or predictive. Whereas, in [7], they are classified in three ways; (i) the interaction between the route discovery coupled with QoS provisioning mechanism; (ii) decoupled interaction between the route discovery and QoS provisioning, (iii) the interaction with the MAC layer; either independent or dependent, (iv) the approach to route discovery. In Section 4, another form of QoS Routing classification was presented based on heuristics: (i) admission control; (ii) scheduling discipline; (iii) ordering of QoS metrics; (iv) sequential filtering; (v) control theory approach and (vi) computational intelligence approach.

In this Section, we elaborate on the classification based on MAC protocol interaction, by considering the following classes of QoS routing solutions.

7.1 Contention-free MAC layer

In this classes of the solutions, those that rely on accurately-quantified resources such as channel capacity, availability and resource reservation, definitely requires a contention-free MAC solution. TDMA is one of them providing near hard QoS guarantee, typically afforded by wired network. Solutions employing a contention-free MAC, QoS guarantees are essentially hard, except for when channel fluctuations or node movements occur. This unpredictable conditions result in MANET would be unsuitable for providing truly hard QoS guarantees.

7.2 Contending MAC layer

These QoS routing algorithm rely only on a contended MAC protocol and therefore only on the available resources at that instant. These resources need to be statistically estimated. Such protocols typically use these estimations to provide soft guarantees. Implicit resource reservation may still be performed, by not admitting data sessions which are likely to degrade the QoS of previously admitted ones. However, all guarantees are based on contended and unpredictable channel access or are given only with a certain probability and are thus inherently soft.

7.3 Independent of MAC layer

There are QoS routing that are independent from the MAC protocol. Such protocols cannot offer any type of QoS guarantees. They typically estimate node or link states and attempt to route using those nodes and links in which more favourable conditions may exist. However, the achievable level of performance is usually not quantified or is only relative. The aim of such protocols is typically to foster a better average QoS for all packets according to one or more metrics. This comes often at the cost of trade-offs with other aspects of performance, increased complexity, extra message overhead or limited applicability.

8. Protocols relying on contention-free MAC

8.1 QoS routing using CDMA over TDMA network

The problem of concerned to QoS routing protocol designers was the process of discovering paths that satisfy a session's throughput requirement. This was due to the fact that assured throughput seemed to be the lowest common denominator among multimedia data sessions' requirements. Since throughput depends largely on a node gaining sufficient transmission opportunities at the MAC layer, where the first part of the solution is the ability to measure the channel capacity at a node. Then, a mechanism is required to estimate the achievable throughput on a path, utilising the available channel capacity of the node. Finally, this information can be used to perform session admission control, admitting only data sessions for which a path with adequate throughput has been found. Chen *et al* presented an early channel-capacity estimation scheme for mobile wireless networks [57].

The authors proposed a kind of clustering scheme to group nodes where each cluster employs a different spreading code under a CDMA scheme. Within clusters, the channel was time-slotted to deterministically allocate channel access opportunities for each node. Hence, the channel capacity could be measured in terms of time slots. Additionally, time slots may be reserved as a way of promising channel capacity to individual data sessions.

The ideas in [57] were taken further by Lin *et al* [30], wherein they devised a detailed algorithm for calculating a path's residual traffic capacity, seemingly filling in the gaps in detail left by [57]. Similar to the aforementioned work, they propose using a CDMA over TDMA network. The channel is time-slotted accordingly, but several communicating pairs can share a time slot by employing different spreading codes. A path's capacity is expressed in terms of free time slots. Route discovery is based again on DSDV [58]. Routing updates are used to refresh the *free slots* information in routing tables. The proposed algorithm first calculates the best combination of free slots on the path for maximum throughput and then attempts to reserve them for a particular data session. In brief, the algorithm deals with nodes in groups of three. Below each node we show the time slots that were free prior to a data session being admitted. In this case, the same six slots were free at each node. At a first trivial glance it appears that the path capacity is six slots. This illustrates that nodes must have some common free slots to communicate, but if all nodes have the same set of free slots, the efficiency of utilisation is not very high. Then, the effective path capacity usable by a new session is only two slots, despite six being initially free at each node. Once the available time slots and path capacity have been determined, reservation signaling takes place to reserve the necessary time slots for satisfying the requesting session's throughput requirement. The two described schemes offer a clear-cut definition of path capacity in terms of time slots and allow a routing protocol to provide throughput guarantees to application data sessions by reserving these slots. However, this comes at the cost of many assumptions. First of all, assuming a CDMA network assumes that each group of nodes is assigned a different spreading code. These must either be statically assigned at network start-up, or dynamically assigned. The former mechanism does not deal with nodes/clusters leaving/joining the network, which is one of the most basic characteristics of ad hoc networks. The latter scheme assumes that there is some entity for assigning spreading codes, which is against the ad hoc design principle of not relying on centralized control. A second assumption is that of time-slotting. For each frame to begin at the same time at each node, the network must be globally synchronised. Synchronisation signaling incurs extra overhead, and as stated in previous work [7], [25], in the face of mobility this becomes practically unfeasible. Furthermore, time slot assignments must be continually updated as nodes move, and sessions are admitted or completed. Since these designs were published, new TDMA based MAC protocol designs have come to fruition, such as the IEEE 802.15.3 standard [59]. However, this protocol is designed for use in wireless personal area networks where every node is in range of a controller which provides the time-slot schedule. Thus, it is not suitable for wider-area MANETs. The conclusion is that there is currently no ideal feasible solution for implementing TDMA in a multihop MANET environment.

8.2 Multiple path routing using ticket

Chen *et al* [2] proposed a QoS routing protocol which reduces route discovery overhead while providing guaranteed throughput and delay. The main novelty of their approach was in the method of searching for QoS paths. First of all, a proactive protocol, such as DSDV [58] is assumed to keep routing tables up-to-date, with minimum delay, bottleneck throughput and minimum hop to each destination. When a QoS-constrained path is required for a data session, probes are issued by the source node, to discover and reserve resources through a path. Each probe is assigned a number of tickets and each ticket represents the permission to search one path. If greater number of tickets are issued, then the delay and throughput requirements are more stringent. Each intermediate node uses its routing table to decide which neighbours to forward the probe to and with how many of the remaining tickets. Neighbours through which a lower delay or higher achievable throughput to the destination is estimated, are assigned more tickets. So, for example, the source sends a probe with three tickets, which splits at the second node. Two tickets are issued to the bottom path since it is deemed to have a higher chance of satisfying the delay requirement. Due to the nature of MANETs, the state information is not assumed to be precise and therefore, each delay and bottleneck channel capacity estimated is assumed to be within a range of the estimate. Eventually all probes reach the destination allowing it to select the most suitable path. It then makes soft reservations by sending a probe back to the source. This probe also sets the incoming and outgoing links for the connection in each node's connections table, setting up a soft connection state. The reservations and states expire when data is not forwarded via that virtual connection for a certain period of time, hence the terms soft reservation/state. Speaking in its favour, this protocol can handle sessions with either a delay or throughput constraint. When such a constrained path is required, flooding is avoided via the ticket mechanism, while at the same time ensuring that more paths are searched when requirements are stringent, increasing the chance of finding a suitable route. Imprecise state information is also tolerated. However, the method has several drawbacks. Firstly, the protocol used to maintain routing tables for guiding the search probes is proactive, requiring periodic updates, thus incurring a large overhead and not scaling well with network size. Secondly, Chen *et al* [2] mentions that a TDMA/CDMA MAC is assumed to take care of channel capacity reservation, which has the drawbacks discussed in the previous section.

8.3 SIR and bandwidth guaranteed routing with additional transmit power

Another TDMA-based QoS routing protocol is presented by Kim *et al* [40] with channel capacity expressed in terms of time slots. Furthermore, this protocol aimed to concurrently satisfy the application's throughput requirement and its BER constraint. For BER constraint, it aims to achieve by assigning adequate transmit power to produce the necessary signal to interference ratio (SIR) between a transmitter and receiver pair, with lower BER. This is in contrast to the previous candidate solutions, which aimed merely to satisfy a single QoS constraint at a particular moment. The protocol is on-demand and in essence, follows a similar reactive route discovery strategy to DSR [61]. An advantage of this protocol is that it gathers multiple routes between a source and destination and allows them to cooperatively satisfy a data stream's throughput requirement. However, only paths that fulfill the SIR requirement on every link qualify as valid routes. However; the maximum achievable SIR is limited by the maximum transmit power. Time is split into frames with a control and data phase, each containing several time slots. In the control phase, each node has a specified slot and uses this to broadcast data phase slot synchronization, slot assignment and power management information. This broadcast is made at a predefined power level. The received power can be measured and knowing the transmit power, the path loss can be calculated. From this, it is possible to calculate the received SIR. This in turn leads to an estimation for the required link gain and thus the required power at the transmitter, $p_{j-1}^{(i)est}$, where j is the current node in the path and i is the time slot index. When a route is required, a RREQ is broadcast by the source and is received by direct neighbours. As in previous TDMA examples, forwarding nodes must be careful not to transmit in a slot in which their upstream node is receiving contains the number of time slots and SIR requirements. Time slots at the current node must be idle and not used for receiving, to be considered for reservation. Slots for which $p_{j-1}^{(i)est}$ is lower, are preferred. As long as one free slot exists, the node is appended to a list in the RREQ packet, along with the required power estimate for the transmitter for that particular transmission slot. The destination eventually receives multiple RREQs, hence the need for only one free slot on each path, since multiple paths can cooperatively serve the throughput requirement. It returns RREPs to the source along the discovered paths, which deliver the estimated power information so that the correct power can be set in the relevant transmission time slots.

8.4 Node state routing

Most designers wrongly adopted wireline paradigm in designing QoS routing protocols [49]. According to this paradigm, nodes are connected by physical entities called links and routing should be performed based on disseminating the state of these links. It was suggested that the correct wireless paradigm assumed the sharing of a geographical space and the frequency spectrum with other node pairs nearby. It must be asserted that links cannot be considered independently of each other. The author instead proposed the Node State Routing(NSR)[49]. In NSR, each node maintains the state information about itself and the surrounding environment, in a routing table. This includes states such as its IP address, packet queue size and battery charge. However, to avoid relying on link state propagation, NSR requires GPS input. This provides extra states, the node's current location, relative speed and direction of movement. It is assumed that nodes can estimate the path loss to neighbouring nodes, using a pre-programmed propagation model and knowledge of the node positions. In this way, connectivity would be inferred. Using the aforementioned states, it would be possible to predict connectivity between nodes, whereas in most other protocols, links must be discovered. In order to perform routing functions nodes must periodically advertise their states to neighbours. Neighbours should further advertise selected states of their neighbours, for example, only those that have changed beyond a threshold. Using the states of its neighbours, a node may then calculate metrics that may be conceived as link metrics, except that measurements at both ends of the link can be taken into account. Moreover, since node states are readily available, they can be used to calculate QoS routes as required. As opposed to most other QoS routing protocols, the node states allow different QoS metrics to be considered for each requesting session, without re-discovering routes. A route can be calculated from the propagation map at each node, and its lifetime can be estimated. This approach shows huge potential for practical multiconstraint QoS routing in the future. Furthermore, since link states are not used, there is no need to update several link states when a single node moves, as in other protocols. Instead, only that one node's state needs to be updated in neighbours' state tables. Despite its many advantages, NSR also has several drawbacks. First and foremost, it relies on accurate location awareness, which limits its usefulness to devices that are capable of being equipped with GPS receivers or such. Secondly, as described in [49], throughput-constrained routing depends on a TDMA-based MAC protocol for capacity reservation and throughput guarantees.

9. Protocols based on MAC contention

9.1 Core Extraction Distributed Ad Hoc Routing (CEDAR)

The CEDAR algorithm was proposed by Sivakumar *et al* [60]. Its name is derived by the fact that it is a topology management algorithm with core extraction mechanism as the main function. The core of a network is defined as the minimum dominating set (MDS). It means that all nodes are either part of this set or have a neighbour that is part of the set. The MDS calculation is a known NP-hard problem [60]. Therefore the algorithm only finds an approximation, of it. MDS is calculated in order to set the core nodes, hence be able to provide a routing backbone. It ensures that all nodes are reachable but not every node need to participate in route discovery. Non-core nodes could save energy by not participating and its overhead would also be reduced. Generally, local broadcasts are unreliable due exposed and hidden node problems [60]. Reliable local unicasts may be used to propagate routing and QoS state information. It utilised the uses of RTS-CTS handshaking to avoid hidden and exposed node problems. Additionally it ensures the broadcast packet is delivered to every neighbouring core nodes. This scheme is termed core broadcast. Using [60] only local state for QoS routing incurs little overhead, but far from optimal routes may be computed. Worst still no QoS route may be found, even if one exists. On the other hand, gathering the whole network state at each node results in a very high overhead. Theoretically it allows the computation of optimal routes, although there's a possibility of using stale information. CEDAR compromises, by keeping up to date, information at each core node about its local topology, as well as the link-state information about relatively stable links with relatively high residual capacity further away. This is done via increase and decrease waves. For every link, the nodes at either end are responsible for monitoring the available capacity on it and for notifying their dominators when it increases or decreases by a threshold value. The method of estimating available link capacity is not specified in [60]. However, nodes only have link capacity information from a limited radius due to the wave propagation mechanism. Thus, the QoS core path is determined in stages with each node routing as far as it can see capacity information, then delegating the rest of the routing to the furthest .seen. node on the core path. This process iterates until the final destination is reached and all links satisfy the achievable throughput requirement. The greatest novelties of this technique were the core broadcast and link capacity dissemination mechanisms. These ensure efficient use of network resources and relatively accurate and up-to-date knowledge of the QoS state, where it is required. Furthermore, this protocol does not rely on a TDMA network, as the protocols discussed in the previous section do. However, the problem of estimating available link capacities was left open.

9.2 Interference- aware QoS routing

In [62] the authours consider throughput-constrained QoS routing based on knowledge of the interference between links. The so-called clique graphs are established, reflecting the links that interfere with each other, hence preventing occurrence of simultaneous transmission. It operates by first recording the channel usage in *bps* of each existing data session on each link. It was noted that the total channel usage of the sessions occupying the links within the same clique should not exceed the channel capacity. A link's residual capacity is then calculated by subtracting the channel usage of all sessions on links in the same clique from the link's nominal capacity. This link capacity information may be utilised to solve the throughput-constrained MANET routing problem. Additionally, Yang *et al* [25] published and discussed the problems of achievable throughput estimation in a contended-access network which depend on the node’s transmission range, ***R.*** Nodes within the Carrier-Sense rang are termed as CS-neighbours, and this set of nodes is the CS-neighbourhood. The CS-range which is equivalent to ***2R*** model simulates the physical layer characteristics of network adapters which are able to sense the presence of a signal at a much greater range than that at which they are able to decode the information it carries. In a contention-based MAC protocol such as the 802.11 distributed coordination function(DCF)[63], a node may only transmit when it senses the channel idle. Therefore, any nodes transmitting within its CS-range may cause the channel to be busy and are thus in direct contention for channel access. This is one of the key realizations in [25] such that all nodes in the CS-range (CS-neighbours) must be considered when estimating a node's achievable throughput. More specifically, in 802.11, the channel is deemed idle if both the transmit and receive states are idle and no node within ***R*** has reserved the channel via the network. The major advantage of this protocol is that no extra control packets are introduced, since bandwidth information is piggybacked on the existing HELLO packets. While the approaches discussed in this section represent significant progress in achievable throughput estimation and admission control, and hence throughput constrained QoS routing, there are still shortcomings. It is well-known that as a network nears saturation, ready-to-send and data packet collisions (in a multihop network) become more frequent, wasting capacity. Additional capacity is wasted due to the 802.11 backoff algorithm, as the level of contention for the channel increases. The protocols discussed in this section do not consider these sources of wastage when calculating the residual capacity at each node.

9.3 Cross-layer multi-constraint QoS routing

Fan *et al* [36] proposed MAC delay metric, which was defined as the time between a packet being received by the MAC protocol from the higher layers, and an ACK being received for it, after it is transmitted. This includes the time deferred when awaiting channel access and is thus a useful metric for avoiding busy links. Link reliability and throughput constraints are also considered in [36], but they use pre-existing definitions and methods of calculation. The focus of the paper is on performing multiconstraint QoS routing with the aforementioned three metrics. The authour reiterates the fact that the multi-constraint QoS routing problem is NP-complete [2] when a combination of additive and multiplicative metrics are considered. Among the above metrics, delay is additive, link reliability is multiplicative and achievable throughput is concave. However, methods have been proposed for reducing this NP complete problem to one that can be solved in polynomial time. In one such method, all QoS metrics, except one, take bounded integer values. Then, the task of finding a path to satisfy all constraints can be performed by a modified Dijkstra's algorithm. The multiplicative metric is reduced to an additive one by taking the logarithm of the reliability percentage of a link. Also, the delay metric is reduced such that each link is represented by the percentage of the allowable total delay it introduces. The resulting problem in the new metric space can be solved in polynomial time. Then, a modified Bellman-Ford or Dijkstra's algorithm with the new reliability metric for link weights can be used to find an approximation to the optimal path. In each iteration, the total MAC delay along a path is checked and also paths which do not satisfy the channel capacity constraint are eliminated. An obvious advantage of this approach is the concurrent consideration of several important QoS metrics in path selection. However, QoS state for all paths must be discovered and kept fresh. This incurs extra overhead. Furthermore, such a protocol requires the participation of other mechanisms which could measure the link reliability, MAC delay and available channel capacity at each node.

9.4 On-demand delay-constrained unicast routing protocol

Zhang *et al* proposed [5] a protocol with delay constrained routes for data sessions. The operation of the protocol are as follows: firstly, a proactive distance vector algorithm is employed to establish and maintain routing tables consists the distance and next hop along the shortest path to each destination node. When a delay constrained path is required, this information is used to send a probe to the destination along the shortest path to test its suitability. If this path satisfies the maximum delay constraint, the destination returns an ACK packet to the source, which reserves resources. For this purpose a resource reserving MAC protocol is assumed. If the minimum hop path does not satisfy the delay constraint, the destination initiates a directed and limited flood search by broadcasting a RREQ packet. Intermediate nodes forward the RREQ if the total of their respective distances from the destination and source is below a set threshold and also the path delay is below the delay constraint value. When a copy of the RREQ reaches the source with a path that meets the delay constraint, the route discovery process is complete. While this protocol aims to minimize the hop-distance between source and destination and discovers paths that satisfy a session's delay constraint, extra overhead is incurred by the proactive distance-vector protocol which maintains the routing tables.

9.5 QoS greedy perimeter stateless routing for ultra-wideband MANETs

A proposal by Abdrabou *et al* [33] highlights new direction for MANETs, that of employing an ultra-wideband (UWB) signal. Using UWB, a node's position can easily be estimated via triangulation techniques. This provides location information, without having to rely on GPS, for enabling a position-based routing protocol. The proposed algorithm extends to another protocol, Greedy Perimeter Stateless Routing (GPSR) for QoS routing, referring as QoS of GPSR for UWB MANETs (QGUM). Each node broadcasts beacons containing its ID and position to all of its neighbour nodes. The destination's position is learnt at the same time as its ID. When a route is required, the source node sends a RREQ to the neighbour node which is closest to the destination. The RREQ specifies, among other information, the requesting data session's total delay bound, its PLR constraint and the accumulated PLR so far. A node receiving the RREQ factors in its own PLR and compares the result with the PLR bound. If it is unacceptable, a <*Route Failure*> is sent back to the source node. In this case, the source node begins route discovery again, starting with a different node in its neighbour list. If the PLR bound is not exceeded, the intermediate node appends its ID to the RREQ, in a manner akin to other source-routing protocols. It also adds its location before performing the same procedure as the source to find the next node to forward the RREQ to. Each intermediate node performs the PLR checks and passes the RREQ to the neighbour closest to the destination, until the destination receives the RREQ. The above procedure describes route discovery. The methods for ensuring QoS on routes are as follows. QGUM can operate[33] with a contended MAC protocol, similar to the 802.11 DCF. After a route to the destination is discovered as detailed above, the session admission control procedure begins. Owing to the available position information, the destination can calculate which nodes on the route are inside each other's CS-ranges and thus can transmit simultaneously. The destination then calculates the channel capacity required at each node for the data session to be admitted. It then sends an admission request (AdReq) back along the route. Each intermediate node checks its locally available capacity and the capacity of its csneighbours by flooding an AdReq. If the intermediate node and all its CS-neighbours have sufficient capacity, they temporarily reserve the necessary capacity for the session and the AdReq is forwarded to the next hop in the route back towards the source node. If any nodes or their CS-neighbours on the route have insufficient capacity, they generate an admission refused message, towards the source, which then invokes a route repair mechanism. However, the advantages of QGUM, must be balanced against the typically shorter range offered by UWB radios, which is only 10m at 110Mbps [64]. Hence, current standardisation efforts involving UWB radio technologies for wireless networks are targeted at personal area networks [65] [54] and not larger-scale ad hoc WLANs as 802.11x is. This limits the applicability of protocols based on a UWB physical layer.

10. Protocols independent of the type of MAC

10.1 QoS optimized link state routing

A QoS routing protocol based on Optimized Link State Routing(OLSR) is presented by Badis et al [65]. OLSR is a pro-active protocol in which information about 1-hop and 2-hop neighbours is maintained in each node's routing table. This information is disseminated via periodically broadcast HELLO messages. OLSR minimises the control overhead involved in flooding routing information by employing only a subset of nodes, termed multi-point relays (MPRs), to rebroadcast it. As a consequence, only MPRs are discovered during route discovery and are used as intermediate nodes on routes. Since only a subset of nodes are MPRs, the best links may not be utilised for routing. In QoS-OLSR (QOLSR) [65], this problem is solved by proposing new heuristics for building nodes' MPR sets in order to enable QoS routing to take place. QOLSR employs both a variation on the MAC delay metric and the achievable throughput metric for QoS routing. In contrast to many of the protocols discussed so far, although the analysis in [65] is based on the 802.11 MAC, QOLSR does not rely on the MAC protocol to provide residual channel capacity. These values are estimated statistically, using the periodic HELLO messages. The total expected MAC delay of a packet is a product of the average estimated delay or expected service time (EST) of one packet and the total number of packets awaiting transmission. The value of EST in turn depends on packets' transmission times and the expected number of retransmissions the MAC layer will have to perform. The FER (Frame Error Ratio) is approximated by taking the ratio of the number of HELLO messages received during a monitoring window to the number expected, which is calculated from the known HELLO sending rate. The FER provides an estimate of the number of retransmissions required for successful delivery of a data packet. The transmission delay of a packet depends on the amount of time a node spends backing off and resolving collisions. A detailed analysis in [65] shows that this is a function of the average backoff window size and the FER. Using these, the derived formulae yield an estimation for the EST of each packet and therefore the total MAC delay of a link between a node and its neighbour. The achievable throughput of a link is also calculated statistically. The MAC delay or EST of a packet is estimated as described above. Using this, and knowledge of the overhead posed by packet headers and MAC control frames, the throughput experienced by packets can be estimated.

10.2 Link stability-based routing

Rubin *et al* [35], considered the link stability as an important QoS metric. Stability is defined as the expected lifetime of a link, which is largely dependent on the node movement pattern. The paper describes the probability distribution functions (PDF) of link lifetimes under various node mobility models. The remaining link lifetime is estimated as the area under the PDF for a given mobility model, taken between the link's measured lifetime so far, and the infinity. For example, in the random destination mobility model, nodes do not change direction after selecting a destination, until they reach it. This mobility model was found to produce a link lifetime PDF similar to a Rayleigh distribution [35]. To find the probability that a link's remaining lifetime is greater than a time $\boldsymbol{t}$, the PDF of the link lifetime is integrated between ($\boldsymbol{t} + \boldsymbol{L_p}$) and infinity, where $\boldsymbol{L_p}$ is the link's past lifetime. A link lifetime model such as the one above is proposed for each of a selection of mobility models. An application may specify a lower limit for acceptable path failure probability, $\boldsymbol{P}_{\text{fail}}$. This value can be calculated based on a data session's delay, delay jitter and packet loss rate requirements. It is proposed [35] that this mechanism is combined with AODV for QoS routing. The value $\boldsymbol{P}_{\text{fail}}$ is inserted into RREQ packets. Intermediate nodes test that the cumulative failure probability of links up to that point (also stored in the RREQ and updated by each node), is not greater than $\boldsymbol{P}_{\text{fail}}$. Therefore, using an appropriate model such as the above and given the data session's duration, it is possible to calculate the probability of a path remaining intact for the duration of the data session, $\boldsymbol{P_{survive}}$. If this is unacceptable i.e. $\boldsymbol{P_{survive}} < \boldsymbol{P}_{\text{fail}}$, the session is not admitted. This simple mechanism could be useful for statistically predicting link lifetimes and therefore avoiding links and paths that have a high probability of failure while a session is active. An obvious difficulty with this approach is that the node mobility pattern must be known and must be modeled accurately for the lifetime estimation to be useful. However, combined with other stability metrics, as shall be discussed later, this could be a useful component of a more sophisticated QoS provisioning mechanism. Another approach that considers link and path stability as an important QoS metric, is presented in [66]. A new variation on the stability metric is introduced in the form of the entropy metric. This is defined for a link as a function of the relative positions and velocities, and the transmission ranges of the link's two end nodes. A path's entropy is defined as the product of the link entropies along it. The lower the entropy, the higher the path stability. This scheme is incorporated into a source-routed scheme somewhat akin to DSR, and during route discovery, the path entropy (among other metrics) is calculated. A destination receives RREQs over multiple paths and waits a specified interval after receiving the first one, before selecting the path with the lowest entropy i.e. highest stability. This route is returned to the source in the RRep, thereby completing the route discovery. This approach has the potential to be more accurate than that in [35], since it considers nodes' relative positions and velocities for calculating the probability of link failure, rather than just a general PDF for a given mobility model. However, this comes at the price of assuming that each node is capable of determining its position via GPS or some similar system [42].

10.3 Hybrid Ad hoc Routing Protocol

The Hybrid Ad hoc Routing Protocol (HARP) is introduced in [39]. It uses the notion of quality of connectivity (QoC) as its routing metric. This is defined as a function of two nodes states: residual buffer space and relative stability. The latter is defined for node $\boldsymbol{x}$ over a chosen period of time, $\boldsymbol{t_1}$-$\boldsymbol{t_0}$, as $stab(x) = \left| \frac{N_{t0} \cap N_{t1}}{N_{t0} \cup N_{t1}} \right|$, where $\boldsymbol{N_{t0}}$ and $\boldsymbol{N_{t1}}$ are the set of neighbours of $\boldsymbol{x}$ at times t_0 and t_1 respectively. Thus, stability is greater, the fewer the number of neighbour nodes that change between t_0 and t_1. The higher a node's residual buffer space and relative stability, the better the QoC to it is. The QoC of each node is used in a logical topology construction algorithm. Each node periodically broadcasts a beacon to all of its neighbours, which contains its address and QoC. Then, each node selects as its preferred neighbour (PN) the neighbour node with the highest QoC. A link between a node and its PN is called a preferred link. A logical tree is constructed by connecting nodes together using only preferred links. A tree's growth terminates where a node's preferred link is with a node that is already part of the tree. This heuristic has been proven to yield a forest of trees [39]. In brief, each tree is then considered a routing zone, within which proactive routing occurs. Inter-zone routing is performed on-demand, and hence the hybrid route discovery of this protocol. In inter-zone routing, other zones may be abstracted as nodes, thus a packet can be routed to another zone, and on arrival, the intra-zone routing mechanism can direct the packet to its final destination. HARP also includes route discovery optimizations which reduce overhead. Firstly, the forest structure can be used to avoid having to flood route request (RREQ) packets used in inter-zone routing. This is done by forwarding RREQs only via gateway nodes; a node is considered to be a gateway, if it is the neighbour of a leaf node, but it is in another zone. Secondly, features of the Relative Distance Microdiscovery (RDM) routing protocol (RDMAR) [67] are incorporated into HARP. RDMAR does not limit the number of neighbours propagating a flooded packet, but limits the scope of the flooding instead. Thus, RREQs do not propagate to areas of the network where they will be useless, thereby wasting resources. The time-to-live (TTL) field in a RREQ is set based on an estimation of the relative distance of the destination in terms of hops. However, the estimation can only be made if there is some previous knowledge of the destination, and a replacement path to it is sought. In this case, the relative stabilities of each node on the path, combined with the time elapsed since the stabilities were recorded, yields an estimation for the total maximum change in the positions of the nodes on the path. This is added to the previous known distance in metres of the destination. The sum is divided by the radio range to obtain an estimated upper bound on the distance of the destination in number of hops. This value is used for the TTL.

10.4 Delay-Sensitive Adaptive Routing Protocol

The Delay-Sensitive Adaptive Routing Protocol (DSARP) [34] employs reactive route discovery, is completely decoupled from the MAC protocol and provides delay guarantees for time-sensitive data sessions. Its basic operation is very similar to classical reactive MANET routing protocols such as DSR. However, when a path is required for delay-sensitive traffic, a different algorithm is employed. The source node sends a route request (RREQ), as usual. This is allowed to propagate to the destination, which sends a route reply (RRep). When forwarding the RRep, each intermediate node on the path attaches the number of packets awaiting transmission in its buffer. Multiple RReps may be received by the source node, which then selects several shortest paths, if there are multiple. Alternatively, the shortest path plus the next shortest path are selected. Using the information about buffer usage at each node, the source calculates the total number of packets on each selected path. Finally, the traffic flow on each path is adjusted such that the new traffic allocated to it is greater if the existing traffic on it is lower and the number of packets on other paths is greater. This algorithm pushes the network towards a state where each path has an equal flow of traffic on it and thus is likely to produce the same packet delay. Essentially, this implements a form of load balancing, ensuring that the energy usage of nodes is also distributed evenly. After adjusting the traffic on each path, a statistical guarantee can be made about the delay on that path. DSARP is simple to implement and provides delay guarantees without relying on the MAC protocol, but has the following disadvantages. The number of buffered packets on each path must be rediscovered each time a new session begins, regardless of whether the route has failed or not. This incurs extra overhead. Also, the delay guarantees may fail in the face of mobility, if other nodes move into contention range and cause greater channel access delays for nodes on a session's path.

10.5 Application-aware QoS routing

A rather unique approach to QoS routing is presented in [32]. It is unique because instead of using lower layer (MAC) information, it is based on the aid of the transport layer. The proposal, referred to as Application Aware QoS Routing (AAQR) in the literature, assumes the use of the real-time transport protocol (RTP) [68]. The delay between two nodes is estimated statistically by examining the difference between time stamps on transmission and receipt of RTP packets between those two nodes. The delay variance is also calculated. Furthermore, each node records the throughput requirement of RTP sessions which are flowing through it. Subtracting the total of these throughput values from the raw channel capacity gives an estimate for the total remaining capacity at that node. When a QoS-route is required, applications may specify throughput and delay constraints. In [32] delay is considered the most important constraint for multimedia applications. Routes are discovered on-demand, although the details of the route-discovery procedure are not discussed. A subset of the discovered routes is selected, such that all paths satisfy the delay constraint of the application. From this subset a further subset of routes is selected, which also satisfy the application's throughput constraint. Finally, from this second subset, the route with the lowest variance in RTP packet transmission delays, is chosen. If there are no routes that meet the throughput requirement, the route with the highest available channel capacity, which satisfies the delay constraint, is selected. A major advantage of AAQR is that no extra overhead is incurred for QoS routing, since the existing transport layer packets are used for QoS metric estimation. Additionally, both delay and throughput constraints may be considered. However, the use of RTP is assumed, and therefore the range of application scenarios for this protocol is obviously limited.

10.6 Ad hoc QoS On-demand Routing (AQOR)

AQOR [54] is a QoS-aware routing protocol with the following features: (1) available bandwidth estimation and end-to-end delay measurement, (2) bandwidth reservation, with the optimal bandwidth path is the path with the largest bottleneck bandwidth among all possible paths and (3) adaptive route recovery. AQOR is an on-demand QoS-aware routing protocol. When a route is needed, the source host initiates a route request, in which the bandwidth and delay requirements are specified. The intermediate hosts check their available bandwidth and perform bandwidth admission hop-by-hop. If the bandwidth at the intermediate host is sufficient to support the request, an entry will be created in the routing table with an expiration time. If the reply packet does not arrive in the allotted time, the entry will be deleted. Using this approach, a reply packet whose delay exceeds the requirement will be deleted immediately in order to reduce overhead. To estimate available bandwidth for assisting in call admission, each node puts its reserved bandwidth in periodic

Hello messages that are sent to their neighbors. AQOR uses the sum of a node's neighbors' traffic as the estimated total traffic affecting the node. This estimated traffic can be larger than the real overall traffic. This overestimation imposes a stringent bandwidth admission control threshold. The available bandwidth is thus a lower bound on the real available bandwidth. End-to-end one way downstream delay is approximated by using half the round trip delay. With the knowledge of available bandwidth and end-to-end delay, the smallest delay path with sufficient bandwidth is chosen as the QoS route. Temporary reservation is used to free the reserved resources efficiently at each node when the existing routes are broken. If a node does not receive data packets in a certain interval, the node immediately invalidates the reservation. This avoids using explicit resource release control packets upon route changes. The adaptive route recovery procedure includes detection of broken links and triggered route recovery at the destination, which occurs when the destination node detects a QoS violation or a time-out of the destination's resource reservation.

10.7 Adaptive QoS Routing algorithm (ADQR)

Hwang *et al* proposed an adaptive QoS routing algorithm (ADQR) to find multiple disjoint paths with long lifetimes [41]. ADQR differs from other QoS routing protocols by using signal strength to predict the route breaks and initiate a fast reroute of data. Three levels of signal strength, $Th1$, $Th2$, and Sr ($Th1 > Th2 > Sr$), are defined. Sr is the minimal signal strength to receive a data packet. Three different classes are also defined for nodes, links and routes. If the received signal strength from a neighbor node is higher than $Th1$, that neighbor node is in the first node class. If the received signal strength from the neighbor is between $Th1$ and $Th2$, that neighbor node is in the second node class. If the signal strength is between $Th2$ and Sr, that neighbor node is in the third node class. Links between the first node class nodes are in the first link class; links between the second node class nodes are in the second link class; and links between the third node class nodes are in the third link class. Also, three route classes are defined, where the bottleneck link determines the path class. Each node keeps a neighbor table, which records the node's neighbors and their corresponding cumulative signal strength, defined as:

$$SSnew-cummulative = \delta \times SSold-cummulative + (1-\delta) \times SSnew-measured$$

where δ is adjusted according to network conditions and is the current received signal strength. Also, two symbols are used to indicate the relative motion of the two nodes: "+" indicates that the two nodes are moving away from each other; while "-" indicates that the distance between the two nodes is shrinking. Each node also keeps a routing table, of the form <*source, destination, next hop, hop count, available bw, reserved bw, active, route class, first class link, second class link, third class link*>. The source node sends a *Route Request* packet, which carries the information <*source, destination, request id, hop cnt, QoS metric, route class, int nodes, first class link, second class link, third class link*>. Intermediate nodes append their own address in the *int nodes* field, update the parameters *QoS metric*, *route class*, and *hop cnt*, and forward the *Route Request* to their neighbors. The destination node checks whether this path is disjoint from other paths already found and whether *route class* is anything but "+3" . If the first condition is true and the second is false, the destination node does the same procedure as an intermediate node, creates a *Route Reply* packet, and inserts the route information into its routing table. When an intermediate node receives a *Route Reply* packet, the node inserts the route into its local routing table, if there is no corresponding route entry; or the node updates its routing table, if the route already exists. When the source node receives multiple routes, the choice of which route to use is based on the *route class* information. The *first route class* routes obtain higher priority than the *second route class* and the *third route class* routes. Similarly, the *second route class* routes obtain higher priority than the *third route class* routes. After selecting the desired route(s), bandwidth is reserved by sending a *QoS Reserve* packet from the source to the destination along the selected route(s). ADQR uses a fast route maintenance scheme, called two-phase monitored rerouting, which is composed of *Pre Rerouting* and Rerouting. The *Pre Rerouting* phase occurs when the route changes from *first route class* to *second route class*, and the Rerouting phase is invoked when the route changes from *second route class* to *third route class*. In *Pre Rerouting*, the source node finds alternate paths in advance, before the current path becomes unavailable, and in Rerouting, the source node switches to one of these alternate paths in advance of the current path becoming unavailable.

11. Computational intelligence approach

11.1 Genetic Algorithm-based QoS routing

In [38], a Genetic Algorithm-based source-routing protocol for MANETs (GAMAN) is proposed, which uses end-to-end delay and transmission success rate for QoS metrics. Genetic Algorithms (GAs) may be employed for heuristically approximating an optimal solution to a problem, in this case finding the optimal route based on the two QoS constraints mentioned. The first stage of the process involves encoding routes so that a GA can be applied; this is termed gene coding. For this purpose, paths are discovered on-demand and then a network topology view is constructed in a logical tree-like structure. Each node stores a tree routed at itself with its neighbour nodes as child nodes and in turn their neighbour nodes as their children. Tree reductions are used to avoid duplicate subtrees. Each tree junction is considered a gene and multiple genes make up a chromosome which represents a path. The route discovery algorithm is assumed to collect locally computed metrics such as average delay over a link and the link reliability for the links on each path. After the gene encoding stage, the fitness, T of each path, is calculated as follows:

$$T = \frac{\sum_{i=1}^{n} D_i}{\prod_{i=1}^{n} R_i}$$

where D_i and R_i are the delay and reliability of link *i* respectively. The fitness values are used to select paths for cross-over breeding and mutation operations. The fittest path (with the smallest T) and the offspring from the genetic operations are carried forward into the next generation. While this method is a useful heuristic for approximating the optimal value over the delay and link reliability metrics at the same time, it requires many paths to be searched in order to collect enough .genetic information for the GA operations to be meaningful. This means that the method is not suited to large networks, as the authors themselves admit [38]. The methods of calculating Di and Ri are not detailed, but we assume they can be calculated statistically by the end nodes of each link. Collecting and maintaining sufficient route and

QoS state information to make a GA useful for QoS routing is costly in terms of both overhead and energy consumption. However, heuristic methods are often the only feasible way of solving NP-complete multi-constraint multihop QoS routing problems. Thus, while their general applicability to MANETs is limited, GAs may play a niche role in finding near-optimal routes, while satisfying multiple QoS constraints in certain environments. For example, MANETs which are less power-constrained and experience lower levels of mobility, and/or MANETs having topologies where a relatively small number of nodes can be combined in a relatively large number of ways to construct valid routes. The GAMAN protocol discussed in this section provides an exploratory example of how GAs may possibly be applied in such networks.

Another QoS routing algorithm was proposed by Peng et al [69]. The authors proposed route discovery technique, RLGAMAN. It tries to increase the probability of success in finding QoS feasible routes and integrates a distributed route discovery scheme with a reinforcement learning (RL) method. It utilizes the local information for the dynamic network environment; and the route expand scheme based on genetic algorithms (GA) method to find more new feasible paths and avoid the problem of local optimize. The performance of the RLGAMAN was investigated by simulation experiment using NS2. The authours claimed that when compared to traditional method, the experiment results showed the network with RLGAMAN had improved its efficient and effectiveness.

11.2 QOSRGA

QOSRGA (QoS Routing Using GA) was designed to select QoS route based on QoS metrics such as bandwidth, delay and node connectivity index (nci) [70]. QoS Routing for MANET posses several challenges that must be addressed. In selecting the most optimal route from source to destination, one has to choose from a set of routes with the corresponding quality of connectivity and resources. Due to the nature of node mobility the protocol demands an exceptional performance. It needs to select a single route with the longest residual node-pair connectivity time simultaneously. The proposed QOSRGA is based on source routing which effectively select the most viable routes in terms of bandwidth availability, end-to-end delay, media access delay and the sum of nci. The NDMRD protocol [22] initially determined a number of potential routes by calculating the number of returning Route Reply(RREP) packet from destination. The returning RREP packets extract the QoS parameters from each node along the routes. Genetic Algorithm (GA), then operates on the accumulated set of routes and the corresponding set of QoS parameters. A genetic algorithm for this particular problem must have these five issues resolved before the application of the generic GA framework: (1) a genetic representation for potential solutions to the problem called *chromosomes*. (2) a methodology to create an initial population of potential solutions. (3) an evolution function that plays the role of the environment, rating solutions in terms of their *fitness*. (4) GA operators that alter the structure of chromosomes. (5) values for various parameters that the genetic algorithm uses such as population size and probabilities of applying genetic operators.

11.3 Fuzzy logic approach

Gomathy and Shanmugavel [21] have shown how to integrate the techniques of fuzzy logic and scheduling principles to produce a fuzzy-based priority scheduler. The paper analyzed the performance of the novel fuzzy-based priority scheduler, for data traffic and evaluated the effect of inclusion of this scheduler with different underlying multicast routing protocols, like NTPMR, CAMP, and ODMRP, run over IEEE 802.11 as the MAC protocol. Queuing dynamics with different degrees of mobility and routing protocols show that the composition of packets in the queue determines the effect of giving priority to control packets or setting priorities among data packets, for the average delay. During low mobility, the average delay is dominated by network congestion due to data traffic. During high mobility, it is dominated by route changes. We have addressed a fuzzy-based priority scheduler for data packets, which improves the QoS parameters in MANETs. The fuzzy scheduler attaches a priority index to each packet in the queue of the node. Unlike the normal sorting procedure for scheduling packet, a crisp priority index is calculated based on the inputs such as queue length, data rate, and expiry time of packets, which are derived from the network. The membership functions and rule bases of the fuzzy scheduler are carefully designed.

Sun *et al* [71] proposed QoS routing algorithm based on fuzzy logic. They proposed Fuzzy controller based QoS Routing Algorithm with a multiclass scheme (FQRA) for mobile ad hoc networks. In FQRA, a routing table is maintained to manage the lifetime of the active routes. Then FQRA applies a fuzzy logic system to dynamically evaluate the route expiry time. The fuzzy logic is chosen because there are uncertainties associated with node mobility and the estimation of link crash; moreover, there exist a mathematical model capable of estimating the node mobility. In addition, FQRA is able to take some controlling factors into consideration. The performance of the FQRA is studied using NS2 and evaluated in terms of quantitative measures such as improved path success ratio, reduced average end-to-end delay and increased packet delivery ratio. Generally it shows a promising approach.

11.4 Biologically inspired algorithm

In this paper, we propose a new version of the self organized Emergent Ad hoc Routing Algorithm with QoS provisioning (EARA-QoS). This QoS routing algorithm uses information from not only the network layer but also the MAC layer to compute routes and selects different paths to a destination depending on the packet characteristics. The underlying routing infrastructure, EARA originally proposed in [72], is a probabilistic multi-path algorithm inspired by the foraging behaviour of biological ants. The biological concept of *stigmergy* in an ant colony is used for the interaction of local nodes to reduce the amount of control traffic. Local wireless medium information from the MAC layer is used as the artificial *pheromone* (a chemical used in ant communications) to reinforce optimal/sub-optimal paths without the knowledge of the global topology. One of the optimisations of EARA-QoS over EARA is the use of metrics from different layers to make routing decisions. This algorithm design concept is termed as the *cross-layer* design approach. Research [73] has shown the importance of cross-layer optimisations in MANETs, as the optimisation at a particular single layer might produce non-intuitive side-effects that will degrade the overall system performance. Moreover, the multiple-criteria routing decisions allow for the better usage of network characteristics in selecting best routes among multiple available routes to avoid forwarding additional data traffic through the congested areas, since the wireless medium over those *hotspots* is already very busy. The parameters for measuring wireless medium around a node depend largely on the MAC layer. In this paper, we focus on the IEEE 802.11 DCF mode [74], since it is the most widely used in both cellular wireless networks and in MANETs. This cross-layer technique of using MAC layer information can be applied easily to other MAC protocols. In addition to the basic routing functionality, EARA-QoS supports an integrated lightweight QoS provision scheme. In this scheme, traffic flows are classified into different service classes. The classification is based on their relative delay bounds. Therefore, the delay sensitive traffic is given a higher priority than other insensitive traffic flows. The core technique of the QoS provision scheme is a *token bucket* queuing scheme, which is used to provide the high priority to the real-time traffic, and also to protect the lower-priority traffic from starvation. Experimental results from simulation of mobile ad hoc networks show that this QoS routing algorithm performs well over a variety of environmental conditions, such as network size, nodal mobility and traffic loads.

11.5 Energy- and reliability-aware routing

The Maximum Residual Packet Capacity (MRPC) protocol is proposed in [37], which considers battery charge as well as link reliability during route selection. Admittedly, MRPC is not intended to be a QoS routing protocol, but we consider it here since it utilizes some QoS-related metrics to improve all-round QoS. Routing based on residual battery charge is considered extensively in the literature [48]. However, in our view, protocols that consider only this state are not useful for QoS routing, since they do not improve the QoS experienced by individual data sessions or packets. On the other hand, MRPC also considers link reliability, as detailed below. In [37] a node-link metric is introduced to capture the energy-lifetime of a link between nodes $\boldsymbol{i}$(transmitter) and $\boldsymbol{j}$, which is defined as:

$$L_{i,j} = \frac{R_i}{E_{i,j}}$$

where R_i is the residual battery charge at node i and $E_{i,j}$ is the energy required to transmit a data packet of a given size over the link (i, j). A suggested formulation for $E_{i,j}$ is as follows

$$E_{i,j} = \frac{T_{i,j}}{\left(1 - p_{i,j}\right)^H}$$

where $T_{i,j}$ is the energy required for one transmission attempt of the aforementioned data packet with a fixed transmission power. Also, $p_{i,j}$ is the packet error probability of the link (***i, j***) and H = 1 if hop by hop retransmissions are performed by the link layer. From the above formulae, it is clear that the lifetime of a link is higher when greater battery charge remains at the transmitter node, and when the reliability of the link is high, resulting in a low energy cost for correctly transmitting a packet. These formulae give an estimation for the expected number of data packets that can be transmitted over a link before the battery of the transmitter fails [37]. Then, if a route failure is said to occur when any single link on it fails, the lifetime of path $\boldsymbol{p}$ in number of packets is simply:

$$Life_p = \min_{(i,j)\in p} \left\{ L_{i,j} \right\}$$

MRPC considers the best route to be the one with the greatest residual lifetime. The authours[23], suggests that the MRPC algorithm may be implemented in AODV [75] for application in MANETs. As routes are discovered, the lifetime of the path is accumulated by calculating the lifetime of each link. The next hop to a destination is always selected to be the neighbour which results in the greatest possible value for $Life_p$. This protocol results not only in load balancing, increasing the life of the network and avoiding congestion, but also yields closer-to-optimal energy consumption per packet, as well as lower packet delay and packet loss probability, due to the preference for more reliable links. It can also be implemented in an on-demand fully distributed routing protocol, such as AODV. However, link reliabilities must somehow be estimated, which may not be a trivial problem. Furthermore, like HARP, MRPC does not cater to particular sessions' requirements, only fosters better all-round QoS, and hence may be unsuitable for many applications. On the other hand, as mentioned above, MRPC is not primarily intended to be a QoS routing protocol, rather an energy-efficient best effort protocol.

12. Progressive trends in this area

As we discussed in Section 6, many of the earlier QoS routing proposals for MANETs were based on contention-free MAC protocols and relied on either TDMA or TDMA/CDMA channel access mechanisms. This was probably due to their well-understood nature from the field of cellular communications. A TDMA approach offers a straightforward method of quantifying channel capacity and access opportunities, as well as allowing such opportunities to be deterministically reserved for particular application data sessions. This enables throughput guarantees to be made, provided that the network dynamics do not invalidate them. Due to mobility, as well as the unpredictable nature of the wireless channel, truly hard guarantees can never be made in a MANET. Even though some newer proposals continue to assume TDMA, it is believed that non-hierarchical TDMA-based methods are highly unfeasible in MANETs[25], since time slotting requires global clock synchronisation, which is difficult to achieve in a mobile environment. A further drawback of this approach is the high signaling overhead incurred by slot scheduling and the potential complexities thereof [57]. Newer MAC protocols such as that specified by 802.15.3 [59] offer feasible TDMA solutions for MANETs by introducing node hierarchies whereby a group of nodes in a piconet is synchronised by a central controller node. However, this protocol is designed only for personal area networks and not for largescale multi-hop MANETs. On the other hand, CDMA based methods introduce the problem of code allocation in a dynamic mobile environment. In light of these conclusions, QoS routing methods that rely on such channel access methods are not the solution for general and especially larger-scale MANETs. This is reflected in the literature, since the majority of later solutions, are based on contended MAC protocols (generally 802.11). In Section 9 we discussed several proposals relying on a contended MAC protocol, such as 802.11. Many less mature solutions in this category did not consider the nature of contention between neighbouring nodes sufficiently accurately and thus reliable QoS provisioning did not become a reality for MANETs. It was through key works such as [25], [76], that the nature of contention and its effect on (primarily throughput-constrained) QoS routing, begun to be well-understood. Other newer proposals take this understanding as a basis for further QoS routing designs. Some proposals greatly further the field of QoS session admission control. Many solutions continue to be based upon 802.11x and its CSMA/CA-based channel access mechanism. Even though 802.11 is an aging standard, the CSMA/CA mechanism has survived into its most recent versions and therefore proposals based on the 802.11 MAC protocol continue to be very relevant. On the other hand, QoS routing proposals based on an ultra-wideband physical layer [33] are emerging. As we discussed though, UWB radios have a limited shorter range compared to 802.11x. Accordingly, current UWB standardisation efforts are all aimed at personal area networks, meaning that UWB-based QoS routing proposals have limited applicability to small-scale MANETs only. Statistical QoS Protocols that make no assumptions about the MAC layer have also received greater attention in the last few years. Such protocols allow a simpler modular network stack design, without the complications of cross-layer issues. However, no guaranteed level of service is provided, as we saw in the proposals discussed in Section 10. Instead, such protocols generally improve the all-round average QoS experienced by packets under some metrics, at the expense of other performance metrics or increased complexity or overhead. Such protocols may not be sufficient for supporting applications with stringent QoS requirements. By contrast, protocols in this category have done much to improve QoS robustness to failures, which was another area identified as future work in previous surveys. The link and node stability-based techniques that were summarised in Section 10 can find longer-lasting routes and thus improve the robustness of QoS solutions against failures caused by mobility. In summary we can say that there is a trend for QoS routing solutions to move away from contention free MAC dependence and towards contended-MAC dependence for throughput-constrained applications. To cater for many other metrics, such as delay and PLR, numerous statistical protocols which are independent of the MAC layer, have been proposed. Another aspect of development considers the metrics themselves. Again, in the earlier proposals, the focus was on providing an assured throughput service only, since throughput was deemed the most important requirement. Some earlier protocols could serve, for example, either a throughput or a delay requirement, but not both simultaneously. In this context, the trend we observe has been to move from single-constraint routing to multi-constraint routing, as demonstrated by the later proposals we have discussed. However, multiconstraint routing remains an NP-complete problem [2], [77] and thus most of the described solutions do not aim to find optimal routes. Instead, they simply apply multiple metrics to route filtering, removing all that do not satisfy a particular constraint. One exception was described in Section 11.2, in which a genetic algorithm is employed as an heuristic to finding the optimal route based on more than one metric.

13. Future works

Following on from this survey, we believe that there is still some way to go in the area of throughput-constrained routing, before perfect QoS Routing protocol is achieved, even in a low-mobility scenario. Works such as [25], [75] consider channel contention, as well as MAC overheads in achievable throughput estimation, but the time wasted due to deferring transmission, random back-off and collisions has not been considered. The wastage due to collisions is especially difficult to calculate in a multi-hop environment. This is important future work, if accurate residual channel capacity estimation is to be realised with contented MAC. The understanding of contention among nodes also needs to be transferred to considerations of other QoS metrics, such as end-to-end packet delay, which is affected by the queues of all nodes within contention range [49]. Delay jitter and energy consumption (due to collisions), are also affected. Quantifying the impact on these metrics and more, in the light of contention awareness and collisions, designing routing protocols that incorporate this knowledge and evaluating them with realistic application layer models, is all future work. A further trend that we have observed, is that many designers place great emphasis on the session admission (QoS route finding) capability of their protocol, which is admittedly very important. In contrast, they often neglect or downplay the importance of session completion i.e. maintaining the routes and the QoS for as long as an application data session requires. An aspect of this, QoS robustness, was highlighted by earlier survey writers. However, more work on the evaluation of QoS sensitive session completion performance with realistic application layers, would be useful. Ultimately, session completion is more important from a user perspective, than session admission. This is because the perceived QoS is better when some sessions are blocked but none are dropped mid-session, rather than all sessions being admitted, but some failing. Furthermore, fast local QoS route-repairing schemes require additional investigation to improve QoS session completion rates and protocols' robustness against mobility. In Section III we reiterated that one of the major challenges to the provision of QoS in MANETS is the unreliable wireless channel. However, we have found that the majority of QoS routing protocol evaluation studies assume a perfect physical channel, ignoring the effects of shadowing and multi-path fading. Therefore, studying the impact of a more realistic physical layer model on QoS routing protocol performance is another interesting area of future work.

As mentioned in the previous section, while simple multi-constraint QoS routing proposals are numerous, there are few that attempt to optimise multi-constraint routing. One example was based on genetic algorithms [38]. However, such methods have limited applicability due to the overhead and energy cost of collecting enough state information. Accurate studies are required to establish, with various networking environments and topologies, whether or not it is feasible to collect and maintain sufficient state information to apply methods such as GAs. For the cases where it is, more research is required on different types of heuristic algorithms for calculating near-optimal paths with multiple QoS constraints. Comparative studies on the performance and impact of the heuristics, are additional future work. Moreover, there is a distinct lack of protocol frameworks for incorporating such methods into practically-realisable systems. One promising, but perhaps not yet mature or feasible approach is that of Node State Routing [49]. Such a solution would provide the mechanism by which to disseminate the information to enable multi-constraint QoS routing.

14. Summary

In this paper we reviewed the challenges to and basic concepts behind QoS routing in MANETs and provided a thorough overview of QoS routing metrics and design considerations. We then classified many of the major contributions to the QoS routing solutions pool published in recent years. The protocols were selected in such a way as to highlight many different approaches to QoS routing in MANETs, while simultaneously covering most of the important advances in the field since the last such survey was published. We summarised the operation, strengths and drawbacks of these protocols in order to enunciate the variety of approaches proposed and to expose the trends in designers' thinking. The protocols' interactions with the MAC layer were also described. Finally, we provided an overview of the areas and trends of progress in the field and identified topics for future research.

15. References

[1] I. Chlamtac, M. Conti, J.J.-N. Liu, "*Mobile ad hoc networking: imperatives and challenges*," Adhoc, vol. 1, pp. 13-64, 2003.

[2] S. Chen, K.Nahrstedt; Distributed Quality-os-Service Routing in Ad Hoc Networks, IEEE JSAC V17, N8, Aug99, pp 1-18.

[3] S. Chakrabarti and A. Mishra, .QoS issues in ad hoc wireless networks,. *IEEE Commun. Mag.*, vol. 39, pp. 142.148, Feb. 2001.

[4] S. Chakrabarti and A. Mishra, .Quality of service challenges for wireless mobile ad hoc networks,. *Wiley J. Wireless Commun.and Mobile Comput.*, vol. 4, pp. 129.153, Mar 2004.

[5] B. Zhang and H. T. Mouftah, .QoS routing for wireless ad hoc networks: problems, algorithms and protocols,. *IEEE Commun. Mag.*, vol. 43, pp. 110.117, Oct. 2005.

[6] J. N. Al-Karaki and A. E. Kamal, .Quality of service routing in mobile ad hoc networks: Current and future trends,. in *Mobile Computing Handbook* (I. Mahgoub and M. IIays, eds.), CRC Publishers, 2004.

[7] T. B. Reddy, I. Karthigeyan, B. Manoj, and C. S. R. Murthy, .Quality of service provisioning in ad hoc wireless networks: a survey of issues and solutions. available online: http://www.sciencedirect.com, Apr 2004.

[8] E. Crawley, R. Nair, B. Rajagopalan, H. Sandick, *A Framework for QoS-based Routing in the Internet*, IETF, Network Working Group, RFC 2386, August 1998.

[9] Z. Wang, J. Crowcroft, "Quality of Service Routing for Supporting Multimedia Applications", *IEEEJSAC*, September 1996.

[10] Z. Zhang, C. Sanchez, W. Salkwicz, E. Crawley, *Quality of Service Extensions to OSPF or Quality of Service Path First Routing (QOSPF)*, IETF, Internet Draft, September 1997.

[11]R. Guérin, S. Kamat, A. Orda, T. Przygienda, D. Williams, *QoS Routing Mechanisms and OSPF Extensions*, IETF, RFC 2676, August 1999.

[12]M. Oliveira, J. Brito, B. Melo, G. Quadros, E. Monteiro, "Quality of Service Routing in the Differentiated Services Framework", *Proceedings of SPIE's International Symposium on Voice, Video, and Data Communications (Internet III: Qu ality of Service and Future Directions)*, Boston, Massachusetts, USA, November 5-8, 2000.

[13]G.Apostolopoulos, R. Guérin, S. Kamat, and S. Tripathi, "Quality of service Based Routing: A Performance Perspective", *Proceedings of ACM SIGCOMM'98*, Vancouver, BC, Canada, August 31-September 4, 1998.

[14] R. Guérin, A. Orda, "QoS-based Routing in Networks with Inaccurate Information: Theory and Algorithms'", *Proceedings of IEEE INFOCOM'97*, Kobe, Japan, April 1997.

[15]A. Shaikh, J. Rexford, and K. Shin, *Dynamics of Quality-of-Service Routing with Inaccurate Link-StateInformation*, University of Michigan Technical Report CSE-TR-350-97, November 1997.

[16]S. Chen, K. Nahrstedt, "Distributed QoS Routing with Imprecise State Information", Proceedings of the International Conference on Computer, Communications and Networks (ICCCN'98), Lafayette, LA, October 1998.

[17] G. Apostolopoulos, R. Guérin, S. Kamat, and S. Tripathi, "Improving QoS Routing Performance Under Inaccurate Link State Information", *Proceedings of the16th International Teletraffic Congress (ITC-16)*, Edinburgh, UK, June 1999.

[18] Q. Ma, P. Steenkiste, "Quality-of-Service Routing for Traffic with Performance Guarantees" *Proceedings of IFIP Fifth International Workshop on Quality of Service*, Columbia University, New York, May 1997.

[19] S. Rampal, *Routing and End-to-end Quality of Service in Multimedia Networks*, PhD Thesis, Department of Electrical and Computer Engineering, North Caroline State University, August 1995.

[20] B. Li and K. Nahrstedt, "A control theoretical model for quality of service adaptations " in *Proceedings of Sixth IEEE International Workshop on Quality of Service,* 1998, pp. 145-153.

[21] C. Gomathy and S. Shanmugavel, Supporting QoS in MANET by a Fuzzy Priority Scheduler and Performance Analysis with Multicast Routing ProtocolsEURASIP Journal onWireless Communications and Networking 2005:3, 426–436

[22] J.Abdullah and D.J.Parish, "QOSRGA Protocol Using Non-Disjoint Multiple Routes in Mobile AD Hoc Networks." The MMU International Symposium on Information and Communication Technologies (M2USIC2007), 19-20 Nov 2007, Petaling Jaya, Malaysia

[23] S. Saunders, *Antennas and Propagation for Wireless Communication Systems Concept and Design*. New York, USA: John Wiley and Sons, 1999.

[24] C. E. Perkins, ed., *Ad Hoc Networking*, ch. 3. Addison Wesley, 2001.

[25] Y. Yang and R. Kravets, .Contention-aware admission control for ad hoc networks,. *IEEE Trans. Mobile Comput.*, vol. 4, pp. 363. 377, Aug 2005.

[26] P. Karn, MACA: A new channel access method for packet radio, in: Proceedings of ARRL/CRRL Amateur Radio 9th Computer Networking Conference, September 1990, pp. 134–140.

[27] IEEE Standards Board, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, The Institute of Electrical and Electronics Engineers Inc., 1997.

[28] F.A. Tobagi, L. Kleinrock, Packet switching in radio channels: Part II––The hidden terminal problem in carrier sense multiple-access and the busy-tone solution, IEEE Transactions on Communications 23 (12) (1975) 1417–1433.

[29] J. Deng, Z.J. Haas, Dual busy tone multiple access (DBTMA): a new medium access control for packet radio networks, in: Proceedings of IEEE ICUPC 1998, vol. 1, October 1998, pp. 973–977.

[30] C. R. Lin and J.-S. Liu, .Qos routing in ad hoc wireless networks,. *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1426. 1438, Aug. 1999.

[31] A. R. Bashandy, E. K. P. Chong, and A. Ghafoor, .Generalized quality-of-service routing with resource allocation,. *IEEE J. Select. Areas Commun.*, vol. 23, pp. 450.463, Feb 2005.

[32] M. Wang and G.-S. Kuo, .An application-aware QoS routing scheme with improved stability for multimedia applications in mobile ad hoc networks,. in *Proc. IEEE Vehicular Technology Conf.*, pp. 1901.1905, Sep. 2005.

[33] A. Abdrabou and W. Zhuang, .A position-based qos routing scheme for UWB mobile ad hoc networks,. *IEEE J. Select. Areas Commun.*, vol. 24, pp. 850.856, Apr. 2006.

[34] M. Sheng, J. Li, and Y. Shi, .Routing protocol with QoS guarantees for ad-hoc network,. *Electronics Letters*, vol. 39, pp. 143.145, Jan. 2003.

[35] I. Rubin and Y.-C. Liu, .Link stability models for QoS ad hoc routing algorithms,. in *Proc. 58th IEEE Vehicular Technology Conf.*, vol. 5, pp. 3084.3088, Oct. 2003.

[36] Z. Fan, .QoS routing using lower layer information in ad hoc networks,. in *Proc. Personal, Indoor and Mobile Radio Communications Conf.*, pp. 135.139, Sep. 2004.

[37] A. Misra and S. Banerjee, .MRPC: Maximising network lifetime for reliable routing in wireless environments,. in *Proc. IEEE Wireless Communications and Networking Conf.*, (Orlando, Florida), March 2002.

[38] L. Barolli, A. Koyama, and N. Shiratori, .A QoS routing method for ad-hoc networks based on genetic algorithm,. in *Proc. 14th Int. Wksp. Database and Expert Systems Applications*, pp. 175. 179, Sep. 2003.

[39] N. Nikaein, C. Bonnet, and N. Nikaein, .Hybrid ad hoc routing protocol - HARP,. in *Proc. Int. Symp. Telecommunications*, 2001.

[40] D. Kim, C.-H. Min, and S. Kim, .On-demand SIR and bandwidth-guaranteed routing with transmit power assignment in ad hoc mobile networks,. *IEEE Trans. Veh. Technol.*, vol. 53, pp. 1215.1223, July 2004.

[41] N. Wisitpongphan, G. Ferrari, S. Panichpapiboon, J. Parikh, and O. Tonguz, .Qos provisioning using BER-based routing in ad hoc wireless networks,. in *Proc. Vehicular Technology Conf.*, vol. 4, pp. 2483.2487, May 2005.

[42] S. Singh, M. Woo, and C. S. Raghavendra, .Power-aware routing in mobile ad hoc networks,. in *Proc. Int. Conf. Mobile Computing and Networking*, pp. 181.190, 1998

[43] C.-K. Toh, .Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks,. *IEEE Trans. Commun.*, vol. 39, no. 6, pp. 138.147, 2001.

[44] C. E. Perkins, E. M. Royer, S. R. Das, and M. K. Marina, Performance comparison of two on-demand routing protocols for ad hoc networks,. *IEEE Personal Commun. Mag.*, vol. 8, pp. 16.28, Feb. 2001.

[45] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, .A performance comparison of multi-hop wireless ad hoc network routing protocols,. in *Proc. Int. Conf. on Mobile Computing and Networking*, Oct. 1998.

[46] J.-H. Chang and L. Tassiulas, .Energy-conserving routing in wireless ad-hoc networks,. in *Proc. IEEE INFOCOM*, vol. 1, pp. 22.31, 2000.

[47] S. Doshi, S. Bhandare, and T. Brown, .An on-demand minimum energy routing protocol for a wireless ad-hoc network,. *Mobile Computing and Communications Review*, vol. 6, no. 2, pp. 50.66, 2002.

[48] C. Yu, B. Lee, and H.-Y. Youn, .Energy-ef_cient routing protocols for mobile ad-hoc networks,. *Wiley J. Wireless Commun. and Mobile Comput.*, pp. 959.973, December 2003.

[49] J. Stine and G. de Veciana, .A paradigm for quality of service in wireless ad hoc networks using synchronous signalling and node states,. *IEEE J. Select. Areas Commun.*, vol. 22, pp. 1301.1321, Sep. 2004.

[50] M. Grossglauser and D. Tse, .Mobility increases the capacity of ad hoc wireless networks,. *IEEE/ACM Trans. Networking*, 2002.

[51] E. Neely, M.J.and Modiano, .Capacity and delay tradeoffs for ad hoc mobile networks,. *IEEE Trans. Inform. Theory*, 2005.

[52] L. Galluccio and S. Morabito, G.and Palazzo, .Analytical evaluation of a tradeoff between energy efficiency and responsiveness of neighbor discovery in self-organizing ad hoc network,. *IEEE J. Select. Areas Commun.*, vol. 22, pp. 1167.1182, Sep. 2004.

[53] D. Haenggi, M.and Puccinelli, .Routing in ad hoc networks: a case for long hops,. *IEEE Commun. Mag.*, vol. 43, pp. 93.101, Oct. 2005.

[54] Qi Xue and Aura Ganz, Ad Hoc Qos On-Demand Routing(AQOR) in Mobile Ad Hoc Networks, Journal of Parallel and Distributed Computing, V63, 2003, Pg 154-165.

[55] RTP: A Transport Protocol for Real Time Applications, IETF RFC1889,
Available: www.ietf.org/rfc/rfc1889.txt

[56] Prasant Mohapatra, Jian Li and Chao Gui, 'QoS In Mobile Ad Hoc Networks ', IEEE Wireless Communications, June 2003.

[57] T.-W. Chen, J. T. Tsai, and M. Gerta, .QoS routing performance in multihop, multimedia, wireless networks,. in *Proc. IEEE 6th Int. Conf. Universal Personal Communications*, vol. 2, pp. 557. 561, Oct 1997.

[58] C. E. Perkins and P. Bragwat, .Highly dynamic destination sequenced distance-vector routing (DSDV) for mobile computers, in *Proc. ACM SIGCOMM '94*, pp. 234.244, 1994.

[59] IEEE Computer Society, *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High-Rate Wireless Personal Area Networks (WPANs)*, 2003. IEEE Std. 802.15.3-2003.

[60] R. Sivakumar, P. Sinha, and V. Bharghavan, .CEDAR: a core extraction distributed ad hoc routing algorithm,. *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1454.1465, Aug. 1999.

[61] R. Gupta, Z. Jia, T. Tung, and J. Walrand, .Interference-aware qos routing (IQRouting) for ad-hoc networks,. in *Proc. Global Telecommunications Conf.*, vol. 5, pp. 2599.2604, Nov. 2005.

[62] D. Johnson, D. Maltz, and J. Broch, *DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks in Ad Hoc Networking*, ch. 5, pp. 139.172. Addison-Wesley, 2001.

[63] IEEE Computer Society, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 1999. ANSI/IEEE Std. 802.11, 1999 Ed.

[64] D. Porcino and W. Hirt, .Ultra-wideband radio technology: Potential and challenges ahead,. *IEEE Commun. Mag.*, vol. 41, pp. 66.74, July 2003.

[65] H.Badis and K.Al Agha, QOLSR, "QoS routing for Ad Hoc Wireless Networks Using OSLR", Wiley European Transactions on Telecommunications, Vol 15, no 4: 427-442, 2005.

[66] H. Shen, B. Shi, L. zou, and H. Gong, .A distributed entropy based long-life qos routing algorithm in ad hoc network,. in *Proc.IEEE Canadian Conf. on Electrical and Computer Engineering*, vol. 3, pp. 1535.1538, May 2003.

[67] G. Aggelou and R. Tafazolli, .RDMAR: A bandwidth-efficient routing protocol for mobile ad hoc networks,. in *Proc. 2nd ACM Int. Wksp. Wireless mobile multimedia*, pp. 26.33, 1999.

[68] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, .RTP: A transport protocol for real-time applications (rfc 3550). IETF RFC, July 2003.

[69] Fu Peng and Zhang Deyun, Hybrid Optimize Strategy based QoS Route Algorithm for Mobile Ad hoc Networks, Journal of Computer Science 2 (2): 160-165, 2006, ISSN 1549-3636 © 2006 Science Publications

[70] J.Abdullah and D.J.Parish, Node Connectivity Index as Mobility Metric for GA-Based QoS Routing in MANET, Proc of ACM/IEEE, Intl Conf on Mobile Technology, Application and Systems, 10-12 Sept 2007, Singapore.

[71] B. Sun, C. Gui, Q. Zhang, H. Chen, Fuzzy Controller Based QoS Routing Algorithm with a Multiclass Scheme for MANET, Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844 Vol. IV (2009), No. 4, pp. 427-438

[72] Z. Liu, M. Z. Kwiatkowska, and C. Constantinou, "A swarm intelligence routing algorithm for manets," in *Proceedings of the IASTED International Conference on Communications Internet and Information Technology*, no. 433, November 2004, pp. 484–489.

[73] E. M. Royer, S. J. Lee, and C. E. Perkins, "The effects of mac protocols on ad hoc networks communication," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 2, 2000, pp. 543–548.

[74] IEEE Computer Society LAN MAN Standards Committee, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Institute of Electrical and Electronics Engineers Std., 1997.

[75] C. E. Perkins and E. M. Royer, .Ad hoc on-demand distance vector routing,. in *Proc. 2nd IEEE Wksp. Mobile Computing Systems and Applications*, (New Orleans, LA), pp. 90.100, Feb. 1999.

[76] L. Chen and W. Heinzelman, .QoS-aware routing based on bandwidth estimation for mobile ad hoc networks,. *IEEE J.Select. Areas Commun.*, vol. 23, pp. 561.572, Mar. 2005.

[77] F. Kuipers and P. Van Mieghem, .Conditions that impact the complexity of QoS routing,. *IEEE/ACM Trans. Networking*, vol. 13, no. 4, pp. 717.730, 2005.

22

A Novel Secure Routing Protocol for MANETs

Zhongwei Zhang
University of Southern Queensland
Australia

1. Introduction

Ad hoc networks is a special kind of wireless network mode. A mobile ad hoc network (known as MANET) is a collection of two or more devices equipped not only with wireless communications and networking capability, but also with mobility. Most applications of MANETs are primarily concentrated at the military, tactical and other security-sensitive operations (Somebody, 2000).

In MANETs, there is no need having fixed infrastructure such as base stations or mobile switching canters. That is to say, all nodes of MANETs are mobile hosts with similar transmission power and computation capabilities. The feature having no fixed infrastructure makes MANETs to exhibit two antagonistic characteristics. For instance, this feature popularize MANETs to be deployed at some place where wired networks are impossible to be laid down on one hand, this feature also renders MANETs in jeopardies that attackers can easily break-in on other hand.

Although many deployments of MANETs are highly sensitive to the message transmitted in the application layer, MANETs often lack security mechanism in place within the network layer or MAC layer. For instance, MANETs are vulnerable to many kinds of attacks with IEEE 802.11 standard in MAC and PHY layers. The mobility of hosts within MANETs adds another dimension of complexity in the network layer such as routing and security. The complexity is reflected by the fact that the security level of mobile devices or nodes always change all the time.

Most research efforts are concentrated on how to secure routing information on the mobile nodes. It is desirable that a good secure routing algorithm should not only prevent each of possible attacks, but also ensure that no node can prevent successful route discovery and maintenance between any other nodes other than by non-participation.

Methodologically looking at many researches which were working towards the security of wireless ad hoc networks, these studies are based on two types of approaches. One approach is to develop the secure protocols for instance, secure routing algorithms. Another approach is to design secure architecture such as Hierarchical Hybrid architecture. In past decades, there are many schemes of secure routing protocols designed for MANETs, unfortunately a limited number of these schemes are practically implemented, their feasibility and performance are yet to be studied. Further to the already implemented schemes, in case that there are two or more routes, none of them guarantee the communication nodes with the most secure route. Another problem is that the schemes are not capable of adapting to the changing in their topology.

In this chapter, we develop a new scheme of secure routing protocol for MANETs. In Section 2, we present an overview of possible attacks on wireless networks. Routing on MANETs is more challenging than conventional wireless networks, a set of routing protocols have been reviewed in Section 3 along with several algorithms of achieving the security. Our implementation is given in Section 5. We demonstrate the feasibility of the proposed scheme and perform a set of simulation experiments using NS2 in Section 5. The chapter is concluded in Section 6 by a discussion, followed by a list of possible questions for the future,

2. Security concerns in wireless networks

Wireless networks generally are more vulnerable to link attacks than wired networks due to the wireless transmission media. A scrutinies reveals that security concerns in wireless networks involve two separate problems: *secure routing discovery* and *secure data transmission* over the wireless networks.

The use of wireless links makes wireless networks susceptible to many attacks. For instance, eavesdroppers can access secret information, violating network confidentiality. Hackers can either directly attack the network to delete messages, inject erroneous messages, or impersonate a node, which violates availability, integrity, authentication, and non-repudiation. Compromised nodes also can launch attacks from within a network.

One approach to address the security on wireless networks is through the authentication of message among the communicating nodes, while another approach to enhance security on wireless networks is through intrusion detection (ID). Intrusion detection is a reactive approach, which has been used with relative maturity in the traditional wired networks.

Associated with routing is that all secure routing protocols do not specify a scheme to protect data or sensitive routing information. Any centralised authority could lead to more vulnerability in wireless networks. Accordingly, a secure routing protocol must be based on the principle of distributed trust. That is for each mobile hosts, there is a relationship of trust to others. Each host has a certain level of trust to other hosts.

2.1 Protocol based approach

Many routing protocols have been developed to defend against link attacks. Dynamic source routing(DSR) is a simple routing algorithm, in which a sending or source node must provide the sequence of all nodes through which a packet will travel. Each node maintains its own route cache, essentially a routing table, of these addresses. Source nodes determine routes dynamically and only as needed; there are no periodic broadcast packets from routes.

2.2 Architecture based approach

Hierarchical Hybrid(HH) architecture is an infrastructure for wireless networking. In a HH wireless network, all mobile nodes are partitioned into groups. Each group has a group agent and some group members. A group agent itself can be a group member of higher level group.

2.3 Hybrid approach

This approach is to combine the advantages of on-demand (AODV) and optimized link-state routing (OLSR) for wireless sensor networks. The algorithm discovers the route to each node only when it is necessary, but route discovery is based on multipoint relays. It works as follows: the algorithm defines three types of nodes: (1) master, (2) gateway, and (3) plain. A group of nodes selects a master to form a *piconet* and then synchronies and maintains the neighbor list. A node can be a master in only one piconet, but it can be a plain member in any number of piconets. Gateway nodes belong to two or more piconets. Only masters and gateways forward routing information; plain nodes receive and process this information, but they do not forward it.

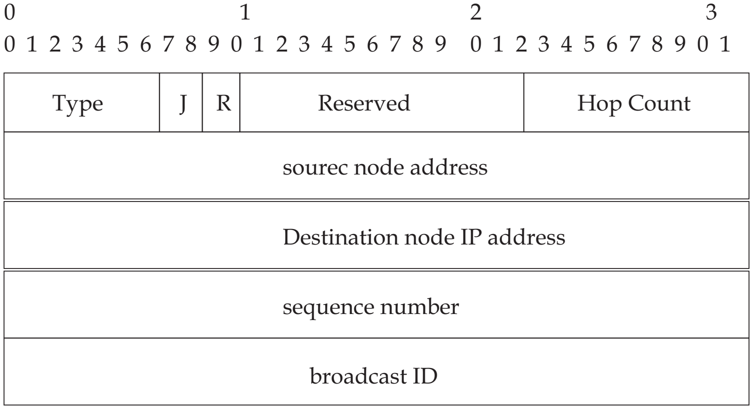
3. Routing protocols and security algorithms for MANETs

Different than conventional wired networks, routing on MANET is characterized by constant changing of route and susceptibility of attacks. Existing routing algorithms include DSR (D. B. Johnson & Hu 2003), AODV (Charles E. Perkins & Das 2003) and SAODV (Zapata 2004).

3.1 Efficient routing protocols for MANETs

In this section, we review one efficient routing protocol for MANETs. Among other routing protocols, Ad hoc On-Demand Distance Vector Routing (AODV) is regarded as the most efficient. With AODV, a source node checks its routing table whether there is a route, if there is no existing route, it then broadcasts an RREQ packet across the MANETs. All nodes that received this RREQ packet will update their information for the source node.

Figure 1 describes the format of a RREQ packet.
Where `Type` is 1, `J` is joint flag and `R` is repair flag.
`HCount`: refers to the number of hops from the Source IP address to the node handling the request.
`BID`: is a sequence number uniquely identifying the source node's IP address.
`DIP`: IP address of destination for which a route is desired.
`DSN`: is the last sequence number received in the past by the source for any route towards the destination.
`SIP`: is the IP address of the node which originated the route request.
`SSN`: the current sequence number to be used for route entries pointing to the sequence of the route request.



```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

<table>
<tr><td>Type</td><td>J</td><td>R</td><td>Reserved</td><td>Hop Count</td></tr>
<tr><td colspan="5">sourec node address</td></tr>
<tr><td colspan="5">Destination node IP address</td></tr>
<tr><td colspan="5">sequence number</td></tr>
<tr><td colspan="5">broadcast ID</td></tr>
</table>

Fig. 1. RREQ packet format

More importantly, AODV has a number of operations, for instance, the unicast communication of nodes including: nodes generating of RREQ and RREP and how the fields in the message are changed.

Figure 2 describes the AODV's *route discovery*. Assume that node *S* intends to explore a route to destination node *D*.

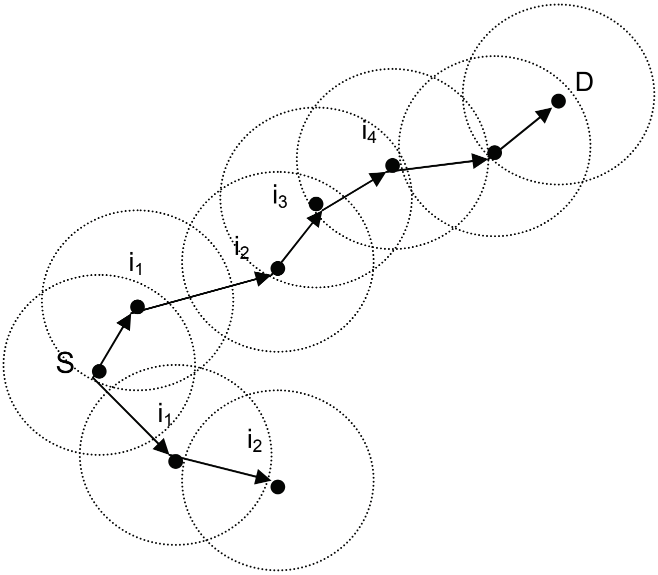


Fig. 2. Route discovery

- Generating route requests: The node S broadcasts a RREQ packet when it determines that it needs a route to a destination and does not have one available in its routing table. After broadcasting a RREQ packet, the node waits for a RREP packet. If the RREP packet is not received within a constant time, the node may rebroadcast the RREQ packet, the rebroadcasting will be repeated up to a fix number of times. Note that each broadcast will increment the broadcast ID in the RREQ packet.
- Forwarding route requests: When a node receives a broadcast RREQ packet, it first checks to see whether it has received a RREQ packet with the same source IP address and a broadcast ID field of equal unsigned integer value within the last RREQ packet. If the checking result is invalid, then it forwards the RREQ packet to its neighbor nodes. The routing table in these nodes will be updated and a reverse path is added.
- Piggyback route reply: When this broadcasted RREQ packet eventually reach an intermediate node on which the checking result is valid or simply the destination node, The intermediate node or the destination node (D) would create a RREP packet, and piggyback it back to the source node (S).

The primary objective of AODV and its routing algorithms is to discover routes for the packets to be delivered from the source node to the destination node, with best efficiency they ever can achieve. Unfortunately, the security in the discovered routes was not seriously considered. AODV is an efficient routing protocol on MANETs which is necessary, but not good enough. If it can not ensure the security, the usability of MANETs would be severely reduced.

3.2 Secure routing protocols for MANETs

Designing efficient routing protocols on MANETS is a primary challenge, but useful for conventional routing protocols. Conventional routing protocols which depend either on distance-vector or link-state usually use periodic broadcast advertisements of all routers to keep routing table up-to-date. In summary, efficient routing on MANETs faces several problems as follows.

- periodically updating the network topology increase bandwidth overhead;
- repeatedly awakening mobile nodes to receive and send information quickly exhausts batteries, which are the main power supply of the mobile nodes.
- the propagation of routing information causes overloading, thereby reducing scalability;
- communication systems often cannot respond to dynamic changes in the network topology quickly enough.

Most secure routing protocols for MANETs use multihop rather than single-hop routing to deliver packets to their destination. The security of mobile nodes is guaranteed by the hop-by-hop authentication, and all intermediate nodes need to cryptographically validate the digital signatures appended with a routing message.

Secure routing protocols usually are based on the efficient routing protocol such as the AODV protocol discussed in Section 3.1. For instance, to add security to AODV, an extension to AODV called SAODV has been designed in recent time (Zapata, 2004). SAODV has extended the AODV by designing a few new extension messages, and a few operations on these new extension message.

Secure routing protocols significantly improve the usefulness of the efficient routing protocol. The idea was to simply incorporate more information in the routing message and routing table, in addition, there are security related operations introduced in the protocols. However, if a secure routing protocol incurs too much overheads, it is possible to render the protocol practically unusable.

3.3 Examples of secure routing protocols for MANETs

A secure on-demand routing protocol for MANETs is developed in (Hu et al, 2002), which is called Ariadne. Ariadne can authenticate routing message using one of three schemes: *shared secrets between each pair of nodes*, *shared secrets between communicating nodes combined with broadcast authentication*, or *digital signatures*.

3.3.1 SEAD: Secure efficient distance vector routing protocol

SEAD (Yih-Chun Hu & Perrig, 2002) is robust against multiple uncoordinated attacks creating incorrect routing state in any other node, even in spite of active attackers or compromised node in the network. The SEAD was designed based on the Destination-Sequenced Distance Vector (DSDV).

During the route discovery process, the source node first selects a random seed number and sets the Maximum Hop-count(MHC) value. By using a hash function, *h*, the source node computes the hash value as `h(seed)`.

3.3.2 Ariadne: A secure on-demand routing protocol

This protocol provides security against one compromised node and arbitrary active attackers, and relies only on efficient symmetric cryptography.

4. Proposed secure routing protocol

We propose a new secure routing protocol for MANETs. It is known as FL-SAODV. The broadcast RREQ packet is an extension of RREQ packet described in Section 3.1, refer to Section 5.1.2.1 for more details. The routing table in each node is same as AODV, more details are given in Section 5.1.2. The FL-SAODV protocol is a secure routing protocol in which the security level is determined by fuzzy logic. FL-SAODV protocol assume that each mobile host uses a secure key with its neighbor nodes. Unlike existing strategies which always assume some security association, our proposed strategy is to rely on the knowledge about the secret key and node's environment such as the wireless link bandwidth and the number of neighbor nodes.

4.1 Node's security association

In spite of the intricate relationship between the security level with these factors, it is obvious that the security level is in the proportional to the number of the neighboring nodes and the length of the key. After having an arduous investigation, we discovered the following knowledge.

- for each mobile node, if its secret key is frequently changed, it is pretty hard for adversary node to decipher the key. In other word, the mobile node concerned is of higher level of security.
  If we represent the frequency of key change by f , then the security level of a mobile node N will has a relationship as $SL \propto f$.
- if a node has many neighbor nodes, the number of possible adversary nodes is higher. The security level the node has can not be very high. The security level of the mobile node $SL \propto \frac{1}{n}$, where n is the number of neighbour nodes.
- if a node has a secret key, its length is l, intuitively, the security level of this node must have a relationship as follows: $SL \propto l$.

4.2 New secure routing protocol operations

FL-SAODV is a new scheme of secure routing protocol for MANETs. Like SAODV that is based on the AODV protocol, FL-SAODV is also an extension to the SAODV. FL-SAODV assumes that each mobile node has a signature key pair from a suitable asymmetric cryptosystem. Each node is capable of securely verifying the association between the address of a given mobile node and the public key of that node. Two mechanisms are used to secure the message: digital signatures to authenticate the non-mutable fields of the message, and hash chains to secure the hop count information, which is the only mutable information in the messages. Every node uses digital signatures to sign the whole message and that any neighbor that receives verifies the signature. FL-SAODV has three operations: (1) determination of the node security level, (2) route discovery, and (3) route maintenance.

4.2.1 Mobile node's security level

The security level of a mobile node in MANETs is determined by the length of the secret key *(l)*, the frequency of the key change *(f)*, and the number of its neighbour nodes *(n)* at a particular time. Its value can be determined by using a fuzzy system described in Algorithm 1, as shown in Fig 3.

Algorithm 1 Security level

```
$n \leftarrow$ number of neighboring nodes
$f \leftarrow$ the frequency of key change
$l \leftarrow$ the length of the key
for all rules in the ruleset do
    get fuzzified value of $n$, $f$ and $l$.
    calculate the individual security level using fuzzy reasoning
    add the individual security level to the total security level
end for
get the defuzzified value of the total security level
```

4.2.2 Route discovery

The route discovery consists of two processes: (1) route request from the source node to the destination node, and (2) route reply from the destination to the source node. The operation of route discovery is described in Algorithm 2.

Algorithm 2 FL-SAODV Route Discovery

```
$S \leftarrow SourceNode$, $D \leftarrow DestinationNode$
$SL_i$ is the security level of node $i$.
$SL_p$ is the security level in the RREQ packet {The Destination node sends RREP back}
Source node broadcasts a RREQ to all of its neighbors
repeat
    for neighbor nodes do
        if there is a route to the destination node then
            authenticate the RREQ using MD5
            calculate its security level using Algorithm 1.
            if $SL_i > SL_p$ then
                update the security level in the RREQ packet
                overwrite the $SL$ in RREQ packet with $S_{ij} = min(S_{ij}, SL_p)$
                update other fields in RREQ
            end if
        else
            broadcast the RREQ to its neighbor nodes
        end if
    end for
until Destination node is reached {The Destination node sends RREP back}
for all RREQ received do
    if Broadcast ID && Security Level in RREQ then
        create a RREP packet
        unicast RREP back to $S$
    else
        drop the RREQ
    end if
    the destination determines which route is the best
    $SL_k = max(S_i)$
end for
```

4.2.3 Route maintenance

A node uses HELLO message to maintain the local connectivity. The route maintenance is described in Algorithm 3.

Algorithm 3 Route maintenance

```
S: the source node
D: the destination node
repeat
    S send a HELLO message to each neighboring nodes
    for all neighbor nodes do
        if the neighbor node does not receive any packets within a certain time then
            the node assume the link is lost
            the node send an RERR packet to all precursors
        end if
    end for
until Route Expired
S starts a new route discovery described in Algorithm 2.
```

5. Implementation and experiments

In this section, we describe an implementation of FL-SAODV, built as an augmentation to the SAODV protocol in the NS2 network simulator (Network Research Group, 1995). The implementation of FL-SAODV involves the changes in routing message format and routing tables.

5.1 Routing message format and routing table

The RREQ packet and RREP packet are the most important packets among others.

5.1.1 Routing request and reply packet

We modify the RREQ packet and the RREP packet formats to carry additional security information. The common fields in RREQ and RREP packet include:

- Destination IP address
- Source IP address
- Broadcast ID
- Expiration time for reverse path route entry
- Source sequence number

We simply adopt other messages such as HELLO message and RERR packet without modification.

5.1.2 Routing table

Every entry in the routing table contains seven fields as follows,

- Destination IP Address
- Destination Sequence Number
- Valid Destination Sequence Number flag
- Security Level
- Hop Count

- Next Hop
- List of Precursors
- Lifetime

Where the field of *Security Level* is an additional than the ones in the routing table of AODV protocol. It is designed to represent the minimum security level of all nodes in the route.

The field of list of precursors contains those neighboring nodes to which a route reply was generated or forwarded. In our implementation, a data structure called *linked list* is used.

The field of lifetime represents the expiration time of the route, the filed of *Hop Count* is the number of hops needed to reach the destination.

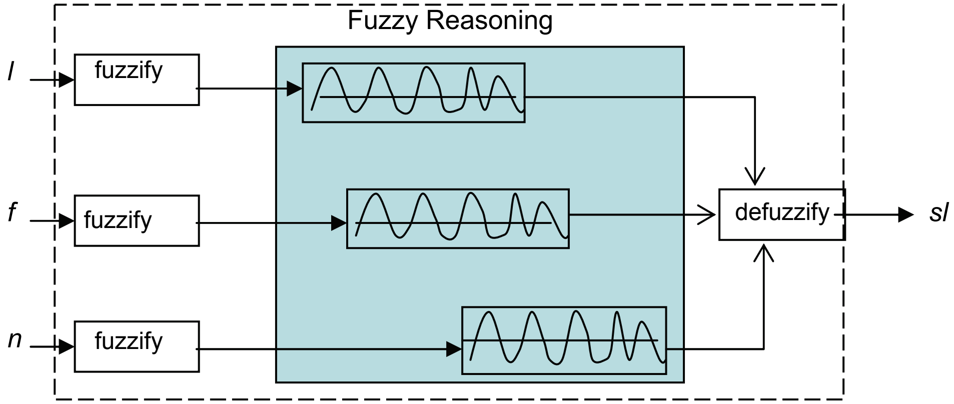


Fig. 3. Fuzzy system

5.1.3 Fuzzy system of determining the security level

The security level of each mobile node is determined by a fuzzy reasoning system. The fuzzy system is implemented using the analysis and knowledge we obtained in Section 4.1. The membership functions of each factor are selected as follows.

Fuzzy membership function for three factors are defined as:

1. key_length: short and long; They are represented in Figure 4.

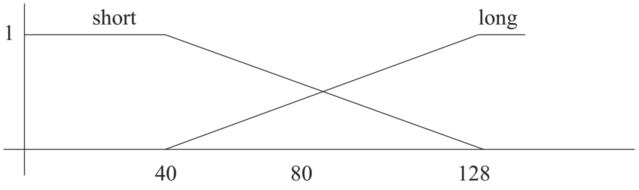


Fig. 4. Membership functions for Key Length

2. frequency: slow and fast; The membership functions looks quite the same as the one above. We would not present them here.
3. number_neighbour: few, normal, and many; These membership functions are shown in Figure 5.

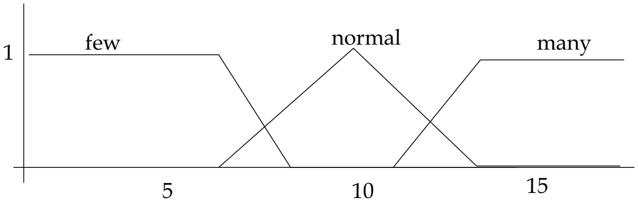


Fig. 5. Membership functions for the density of neighbor nodes.

Fuzzy membership for the security level for each node are: **lowest**, **low**, **normal**, *high* and *highest*.

A fuzzy rule is a representation of knowledge in the form of **IF x is Big and y is Slow Then z is High**. According to the understanding about the mobile nodes in MANETs, we have modeled the relationship between the security level and factors, and presented them in Table 1.

The security level of each mobile node is based on Algorithm 1.

Key_Length (l)	Frequency_key_change (f)	Number_Neighbor_Node (n)	Likehood_security_level (sl)
short	slow	few	least
short	fast	few	low
short	fast	normal	normal
long	slow	many	normal
long	fast	many	high
long	fast	few	highest

Table 1. Fuzzy rules

5.2 Experiment results

The results generated in this section are based on the simulation experiments set up for 4 × 4, 5 × 5 and 8 × 8 and 10 x 10 nodes moving around in 670*m* × 670*m* area. Nodes move according to the random way-point model (Stallings, W., 2005).

When a node sends out the RREQ packet, it is assigned a random number between 0 to 100 as initial security level. The security level at each node `en route` is varying along the time due to the number of neighbor nodes changes. According to the FL-SAODV, the next hop node will be either selected or determined from a few candidate nodes, based on the current security level. If there is only one neighbor node, FL-SADOV will choose that one; The relationship between FL-SAODV and AODV is that AODV is a special case of FL-SAODV, where on the route at each next hop, from the source node to the destination node, there is only candidate node.

In our experiments of 10 x 10 nodes, we shown the security level and the overheads of determining the next hop node. Figure 6 shows the security level at each intermediate node on the route from the source node to the destination node. Figure 7 shows the routing overheads (ie. the calculating time in μsec).

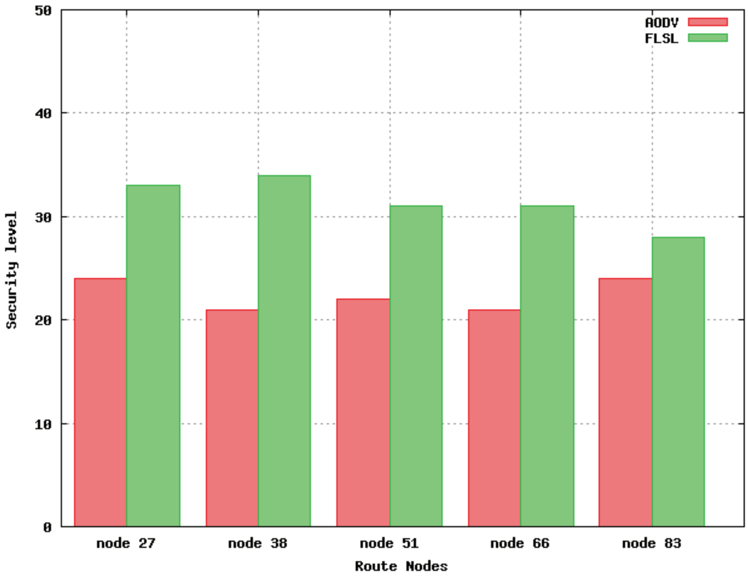


Fig. 6. Security Level on Route Nodes

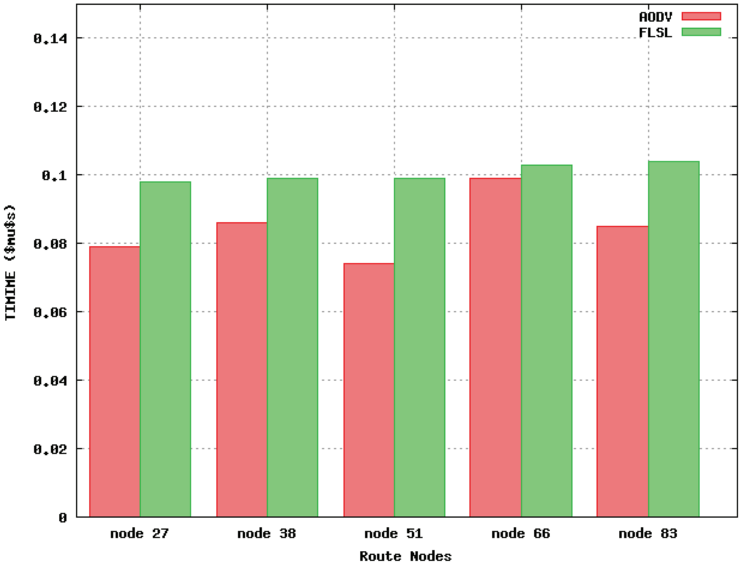


Fig. 7. Times in μs Spent on the Route Nodes

5.3 Analysis

We can see from Figure 6 and Figure 7, using FL-SADOV, the security level on each intermediate node on the route to the destination node has been improved, consequently the security level of the route is of higher value, comparing with the route determined by AODV.

At each intermediate hop node on the route, an addition but minimum overhead is needed for FL-SADOV to calculate the security level before the next hop node is determined. It is worthy pointing out that FL-SADOV has achieved a fair improvement to the route security at a small expense of extra overheads.

In summary, this scheme of secure routing protocol has the following features.

- Protecting routing information from attackers by using hop-by-hop authentication technique: digital signature and hash. This avoids using a CA where other secure routing protocols have to.
- It can adapt itself to the changing environment which is the most salient characteristics of the MANETs.
- FL-SAODV also improves MANETs security from two aspects:
  1. It selects the shortest route which decreases the transmitting time and therefore could shorten the attack time of attackers and improve the MANET's security.
  2. Using security level as metric ensures the updated route to be the most secure one.

6. Conclusion

In this Chapter, we have developed a practical solution to the secure routing on MANETs. First of all, we have reviewed the possibility of attacks to the MANETs, and the security adversaries which compromise a mobile host in ad hoc networks for the purpose of identifying a strategy to beef up hosts security level. Secondly, based on the characteristics of MANETs and the requirements of secure routing, FL-SAODV, a new secure and efficient routing protocol has been developed. A set of algorithms have been **designed for FL-SAODV.** Thirdly, these algorithms have been implemented on the MANETs and many experiments on different scenarios have been carried out on NS2. Lastly, we listed out the security level of the nodes which are on the final route. The route found by using the FL-SAODV protocol have higher security level than the route AODV found. In addition, we shown the timings on its en route nodes and clearly shown that each *en route* node needs more time than AODV to decide their next hop.

There are two open questions for our future research. We believe that the performance of the protocol might be improved by using a better authentication method on one hand. On another hand, how to get the knowledge about the number of neighbor nodes needs more study.

7. References

Charles E. Perkins, E. M. R. & Das, S. R. (2003). Ad hoc on-demand distance vector (AODV) routing, RFC 3561.

D.B. Johnson, D. M. & Hu, Y. (2003). The dynamic source routing protocols for mobile ad hoc networks (DSR).

Hu, Y. C., Perrig, A. & Johnson, D. B. (2002). Ariadne: A secure on-demand routing protocol for ad hoc networks.

Network Research Group, L. B. N. L. (1995). The network simulator NS2, http://www.isi.edu/nsnam/ns.

Stallings, W(2005). Wireless Communications Networks (2nd ed.), Pearson Prentice Hall.

Yih-Chun Hu, D. B. J. & Perrig, A. (2002). Secure efficient distance vector routing in mobile wireless ad hoc networks.

Zapata, M. G. (2004). Secure ad hoc on-demand distance vector (SAODV) routing, RFC 999.

Part 4

Other Topics

23

Security and Dynamic Encryption System in Mobile Ad-Hoc Network

Peter H. Yu and Udo W. Pooch
Texas A&M University, Department of Computer Science and Engineering
College Station, TX
USA

1. Introduction

Wireless network technology enables computing devices to communicate with each other without any physical medium. Compared with wired networks, wireless communication provides better connectivity and mobility, which allows mobile devices to access other local area networks or the Internet at anytime and anywhere. The benefits of flexible routing, global connectivity and a highly adaptive potential make mobile ad-hoc networks (MANET) suitable for a wide range of applications in both military and commercial environments, such as battlefields, disaster relief operations, mobile device/personal networking, mobile information sharing and vehicular networks (Kant et al., 2005); (Liu et al., 2007).

However, maintaining security in wireless ad-hoc networks is quite challenging. First, unlike wired networks that at least have some degree of physical protection, wireless communication over radio waves lacks defined and restricted boundaries. Anyone can connect to the network as long as the transmitted signal strength is strong enough to cover the area (Chan et al., 2005), and therefore, security attacks on data communication, such as passive eavesdropping, packet injection or even violations of confidentiality are widespread. Second, the end-to-end communication in MANET cannot rely on any fixed infrastructure, such as a base station or access points (AP); thus, existing security protocols that are based on a centralized or infrastructure-based network environment will not work in this mobile environment (Hubaux et al., 2001).

Third, in order to achieve better network throughput in such a highly dynamic environment, the default routing protocol does not implement any security protection during end-to-end communication. In addition, the trust relationships between each node are very low as a consequence of the frequently changing topology and membership. Because of this, many attacks can be launched against the routing protocol, giving hackers a major opportunity to insert themselves as one of the cooperative nodes in the network. Therefore, the security protection that is used to ensure the integrity of the mobile ad-hoc network should not only repel external attacks, but also prevent internal attacks launched against the network from any compromised node.

Most security mechanisms rely on data encryption, which is a message combined with a secret key to generate a ciphertext that cannot be revived without the original key. This encryption mechanism can prevent any unauthorized user from gaining access to the secured communication. However, a fixed secret key is vulnerable to deciphering by capturing sufficient packets or by launching a dictionary attack. Therefore, the most efficient way to protect the network from such attacks is to generate the secret key dynamically and replace it periodically (Ramakrishnan et al., 2005). Furthermore, the protocol applied to the mobile ad-hoc wireless network should be sufficiently flexible to adjust to different levels of security protection to fit the needs of applications in different environments and with varied communication speeds. For example, mobile banking and E-commerce require larger encryption keys for stronger protection, while real-time driven applications such as disaster recovery, stream services like VOIP and online video need to preserve data privacy as well as performance to maintain the quality of services (QoS).

In this chapter, we introduced a new, efficient, low-bandwidth cost and security-enhancing data encryption *i-key* protocol for mobile ad-hoc wireless networks via dynamic re-keying during end-to-end communication. Unlike its counterparts, this secret *i-key* is generated using the previous data as the seed and as next packet encryption before delivery; therefore, only the original sender and authorized client are able to decrypt the message using the unique *i-key* in their possession, which ensures the privacy of their communication.

2. Related work

Wired Equivalent Privacy, or WEP, is an encryption protocol designed by the IEEE 802.11 and Home RF group (Lansford & Bahl, 2000) in an attempt to protect link-level data over radio signals for wireless networks, included both Base Station (BS)-oriented and mobile ad-hoc networks, to the security level closer to wired one. The WEP key used to encrypt data sent over wireless networks consists of two parts: the Initialization Vector (IV) and user pre-shared secret key (PSK). The stream cipher, RC4 used in WEP, expands the IV (40 or 104 bits) and PSK into an arbitrary long "key stream" of pseudorandom bits then XOR with the plaintext to obtain the ciphertext. To decrypt it, the receiver side takes the same steps in the reverse order by the same key stream. In addition, a CRC-32 algorithm is applied to check the data integrity for each data packet in WEP encryption.

Many WEP vulnerabilities and security design issues has been discovered and reported by researchers since the IEEE released it as the standard encryption protocol for 802.11 wireless networks (Gast, 2002); (Miller, 2001); (Prasithsangaree & P. Krishnamurthy, 2004); (J. S. Park & Dicoi, 2003). Therefore, wide attention has been paid by many researchers to the design of new protocols to secure the mobile ad hoc network, such as ARIADNE, DSDV, SEAD, ARAN and SPR (Hu et al., 2005); (Perkins & Bhagwat, 1994); (Hu et al., 2003); (Sanzgiri et al., 2005); (Papadimitratos & Haas, 2002) to provide a solutions for the wireless ad-hoc networks.

Hu et al. developed a secure routing protocol called ARIADNE (Alliance of Remote Instructional Authoring and Distributed Networks for Europe) (Hu et al., 2005), which relies on Dynamic Source Routing protocol (DSR) (Johnson et al., 2002) and symmetric cryptography architecture for end-to-end authentication. On the other hand, based on DSDV (Destination-Sequenced Distance Vector Routing) (Perkins & Bhagwat, 1994), Hu and Perrig have proposed the proactive routing protocol SEND (Secure Efficient Ad-hoc Distance vector) (Hu et al., 2003), which runs under a trusted ad-hoc network environment. In order to lower the node’s CPU processing time and achieve better performance, SEND uses one-way public-key signed hash functions instead of asymmetric cryptography.

Authenticated Routing for Ad-hoc Network (ARAN) by Sanzgiri et al. (Sanzgiri et al., 2005) detects and protects the ad hoc network against malicious actions with help from its parties’ or peers' nodes by using pre-determined public key cryptography certificates. However, compared with SEND, ARAN requires a higher computational cost in each node to retain the hop-by-hop authentication.

Using a different approach, SRP (Secure Routing Protocol) (Papadimitratos & Haas, 2002) assures correct connectivity information as well as route discovery by rejecting fabricated, compromised or replayed route replies. SRP assumes a security association between the pair of end-points only, without the need for intermediate nodes to cryptographically validate control traffic (Sanzgiri et al., 2005); (Papadimitratos & Haas, 2002).

Those protocols and traditional security approaches, such as authentication, digital certificates and public-key encryption algorithm, still play important roles in achieving data privacy, integrity, non-repudiation and availability of communication in mobile ad-hoc networks (Zhou & Haas, 1999). However, these mechanisms by themselves are not sufficient, either in terms of computational or communication overhead or lack of ability to prevent attacks launched from inside the network. Therefore, there remains a need for a lightweight and reliable security enhancement protocol for mobile ad-hoc wireless network.

3. Routing and dynamic encryption protocol

3.1 Routing

In an ad-hoc wireless network, routing strategies can be classified as proactive or on-demand (reactive). With proactive protocols, such as Destination-Sequenced Distance Vector Routing (DSDV) (Perkins & Bhagwat, 1994) and Optimized Link State Routing Protocol (OLSR) (Clausen et al., 2003); (Clausen & Jacquet, 2003), the packets route information that is periodically exchanged among hosts, allowing each node to build a global routing table without considering the usage of routing information. In the on-demand approach, such as Ad-hoc Network On-demand Distance Vector (AODV) (Perkins & Royer, 1999) and Dynamic Source Routing (DSR) (Johnson et al., 2001), the nodes build and maintain routes as needed and only toward the nodes involved in the routing, instead of continuously calculating routes in the background.

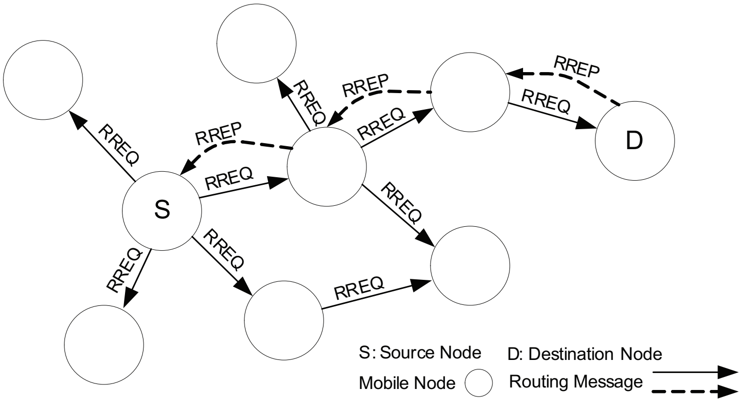


Fig. 1. AODV routing protocol with RREQ and RREP control message

AODV is adapted as the default routing protocol in this dynamic encryption model for the ad-hoc networking because of its high performance and low overhead, which are very important when considering that bandwidth is very limited in wireless communication. In AODV, as shown in Fig. 1. above, the source node first broadcasts a route request (RREQ) message to all adjacent nodes and waits for the corresponding route reply (RREP) message from the destination node to establish routing information. This request and reply query cycle will continue as long as this particular path is not listed in the routing table. Once routes have been built from source to destination, they will continue to be maintained as long as they are needed by the source node. All wireless packets between these two parties will follow the pre-build routing information and will be forwarded node by node until they reach the final destination. When the communication ends, the links will time out and eventually be removed from the table to release space for other routing paths.

3.2 i-key protocol procedures

This *i-key* protocol is primarily based on a dynamic re-keying mechanism that ensures the privacy of communication and prevents unauthorized users from accessing protected data over wireless communication. The key management and cipher stream system in *i-key* architecture is similar to Temporal Key Integrity Protocol (TKIP) used in WPA/WPA2 and RC4 used in Wired Equivalent Privacy (WEP) (Lansford & Bahl, 2000), in which each encryption key contains a pre-shared key (PSK) and a randomly selected key value from the Initialization Vector (IV) pool. In addition to these two keys, an extra dynamic secret *i-key* is applied to the cipher stream that is used to encrypt every data packet before transmission. Fig. 2. illustrates the key stream that is combined with these three different keys and the block diagram of *i-key* encryption and decryption algorithm. The dynamic *i-key* is generated according to the previous data packet and therefore only the sender and authorized recipient are able to decrypt the cipher text by the key stream that is combined with the dynamic *i-key* and static key to reveal the plaintext in the data packet, which becomes the new seed of the *i-key* used in the next data encryption.

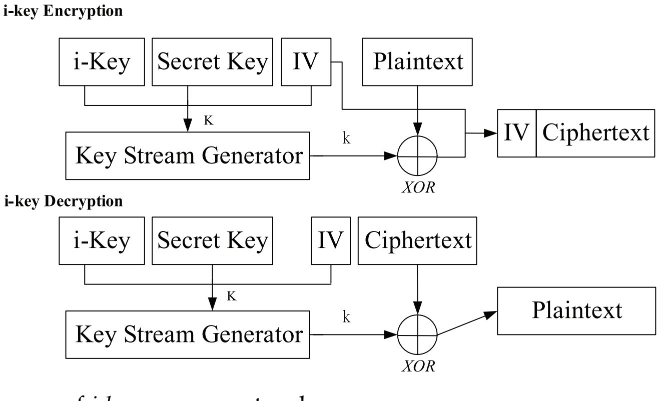


Fig. 2. Block diagram of *i-key* secure protocol

Once routing information and initial handshaking are established for communication between the source mobile node (SMN) and destination mobile node (DMN), the dynamic *i-key* encryption protocol for the wireless ad-hoc network will execute, as seen in Fig. 3.

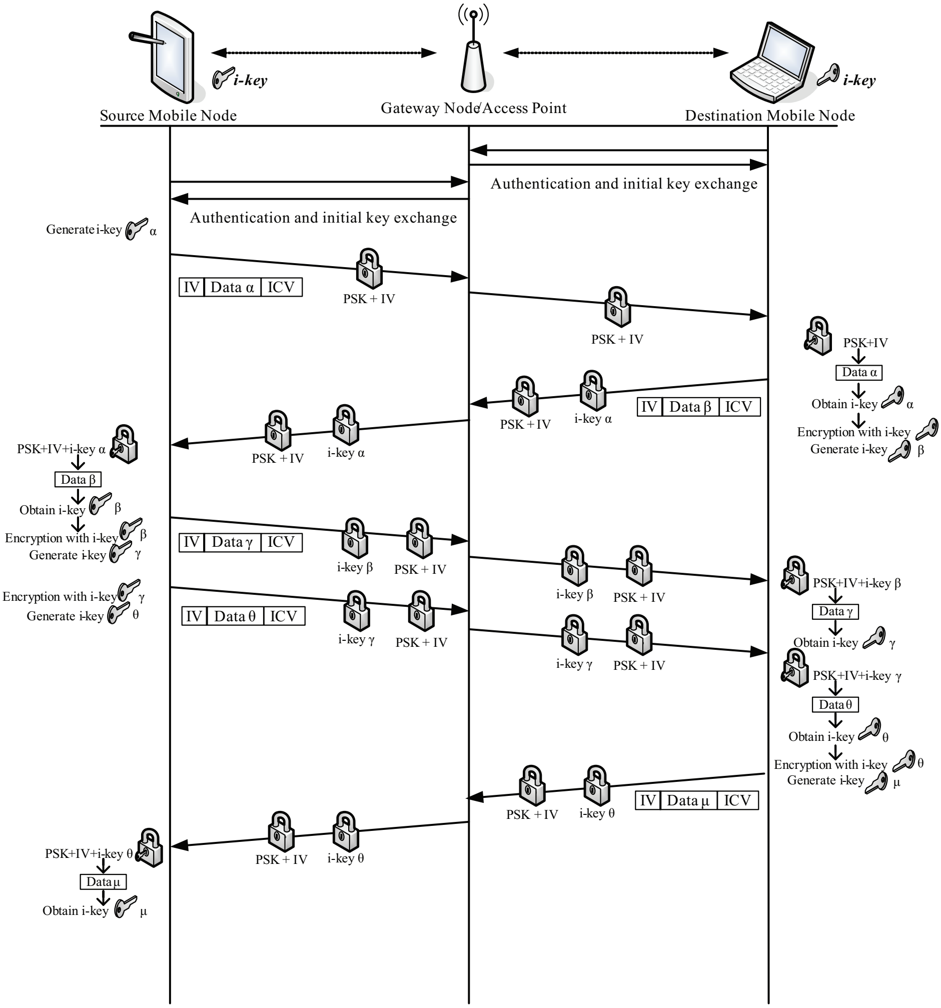


Fig. 3. Dynamic *i-key* encryption and decryption protocol procedures

Step 1. First, the source node S checks the destination node D on its routing information to confirm the proper routing rules been established. Then, source node S generates the secret *i-key*, which is based on the data as the seed contained on the first packet *a*, and keeps this particular secret key to decrypt the next encrypted packet from destination node D. A combination of pre-shared secret key *PSK* and one unique *IV* value is applied for the stream cipher to encrypt the plaintext before routing an adjacent mobile ad-hoc node to relay to the destination node D. Of all the communication between source node and destination node, this is the first and only packet that does not use the dynamic *i-key* for data encryption; however, the security protection remains strong since it needs at least two packets with the identical *IV* value to decode the pre-shard key. Each value in the *IV* pool is generated randomly and uniquely to strengthen the encryption cipher stream and preventing people from cracking it even they are able to capture those wireless packets.

Step 2. The destination node D obtains the data packet *a* as well as the *i-key a* after running a decryption for this encrypted packet from source node S. It will then apply this dynamic *i-key a* to the next data packet's cipher stream to enhance security (because the source node S is the only one that has the same unique secret *i-key a* in this wireless ad-hoc network). Before sending the response/reply packet *β* back to the source node by the same routing strategy, the destination node D will also generate the next *i-key β* based on data in the packet in order to decode the next arrival. From this point forward, every data packet and communication from one side to another in this wireless environment is secured by a dynamic stream cipher that has triple layers of protection: one pre-shared secret key *psk*, one unique *IV* and one dynamic *i-key* possessed only by the original source and destination node.

Step 3. The source node S will use the *i-key a*, generated in Step 1 and which it alone knows, to decode the cipher text along with the pre-shared secret key *psk* and *IV* to acquire the data *β* in the packet that it receives from destination node D. The encryption procedure with *i-key* in Step 2 will repeat again for the next data packet before node S sends it to the destination node D to enhance the security and maintain the data integrity from malicious modification.

Step 4. In cases when node S has more than one data packet to send before it gets a response, the destination node D will apply the corresponding *i-key* to decode the cipher text in accordance with the order of the arrival packets and update *i-key* based on the sequence number in each packet's header to ascertain that the decrypted cipher stream matches the arrival packet and thus passes the integrity checksum in the payload after decryption.

These *i-key* dynamic encryption/decryption procedures will continue running and applying to every packet that is transmitted in the mobile ad-hoc wireless network to ensure the integrity and confidentiality of communication. When any wireless packet fails to be delivered to the destination or is lost during ad-hoc routing (which is common in both IEEE 802.1x based-oriented or an ad hoc network wireless network), an ACK-failed (timeout) or AODV routing error RRER message will be triggered and both sides will be alerted to restore the last successfully received data packet and then re-synchronize the dynamic *i-key* and start the communication over again from Step 2 for the next packet transmission.

Furthermore, before confidential data such as medical records or personal financial information are shared through a wireless ad-hoc network to other mobile devices, the source node can verify the authenticity of the destination node by requesting a response to decrypt a challenge message that the source node encrypted with the latest *i-key* holding with its signature. This sharing continues only when the other side passes the identity challenge; otherwise, the source node will mark the destination as invalid node and reject any further conversations to avoid data leaks or session hijacking. This verify-challenge mechanism in the *i-key* protocol can effectively detect any potential intruders and secure the wireless network by blocking both in-coming and out-going communication to prevent additional attacks.

In addition, this encryption protocol is highly flexible. The dynamic secret *i-key* is regenerated every time for each individual data packet; therefore, the secret key-size can also adjust dynamically to fit different needs in different applications. For example, an online streaming system can temporarily increase the key size during the user identity authentication check to strengthen the complexity of ciphertext from eavesdropping by attackers and then lower the encryption/decryption overhead by reducing the *i-key* size to improve the quality of services (QoS) of real-time live streaming while remaining under solid data protection. Thus, systems with existing security protection, such as SEND and SPR (Hu et al., 2003); (Papadimitratos & Haas, 2002) can still adopt this *i-key* encryption system to enhance data privacy and prevent malicious attacks against the wireless network.

3.3 i-key protocol algorithm

In additional to the RC4 encryption algorithm (Rivest Cipher 4, also know as ARC4 or ARCFOUR) (Rivest, 1992) that also used in WEP and TKIP protocol in IEEE 802.11 wireless networks, dynamic *i-key* protocol also utilizes the stream cipher as the security system model due to its efficiency, reliability and simplicity. Stream cipher takes in one byte to from a stream every time and produces a corresponding but different byte as the output stream, as shown in Fig. 4.

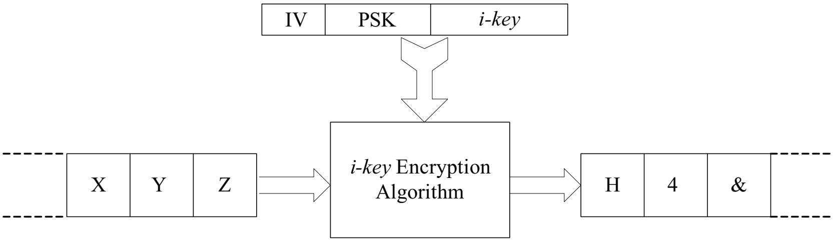


Fig. 4. Dynamic *i-key* encryption stream cipher

Then, this stream cipher combines with the data before transmission over the wireless network by using an exclusive OR (XOR - ⊕) operation. It combines two bytes, one from the cipher and one from the data, and generates a single byte output result as 0 when the values of them are equal, otherwise the result is 1. In general, the strength of an encryption algorithm is primarily measured by how hard it is to decode the ciphertext (Edney & Arbaugh, 2004). Certainly there are stronger encryption procedures than this RC4-like dynamic re-keying algorithm applied in this i-key architecture, however, this simple XOR encryption method is considered very strong among all of the data encryption people use today for both wired and wireless communication (Edney & Arbaugh, 2004).

One of the most important attributes of XOR operation is that if you apply the same value again to the first output result, the original value before the XOR operation is returned:

$$10110010 \oplus 11011001 = 01101011 \tag{1}$$

$$01101011 \oplus 11011001 = 10110010 \tag{2}$$

This characteristic can rewrite as:

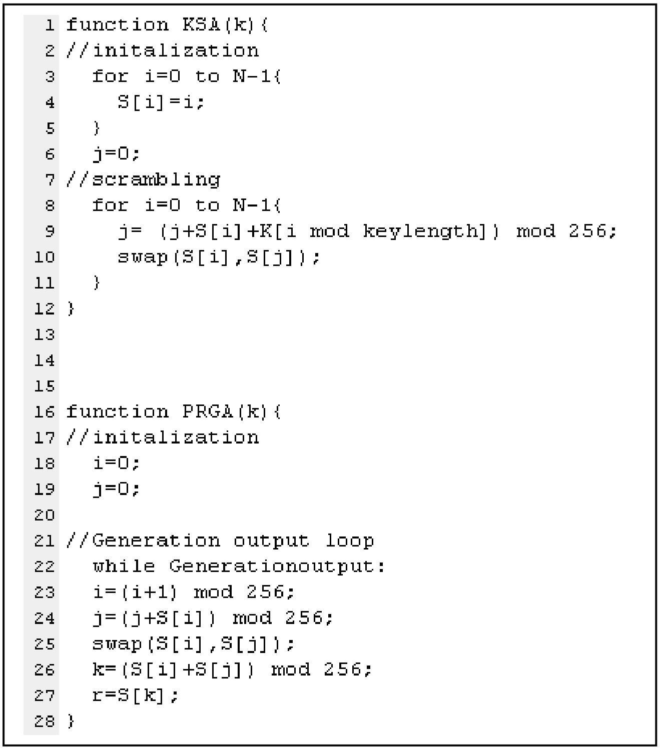
$$\text{if } A \oplus B = C, \text{ then } C \oplus B = A \tag{3}$$

This is also how the decryption procedure works in the dynamic *i-key* system:

$$\text{Encryption: plaintext} \oplus \text{stream cipher} = \text{ciphertext} \tag{4}$$

$$\text{Decryption: ciphertext} \oplus \text{stream cipher} = \text{plaintext} \tag{5}$$

Compared with other encryption systems, such as AES and RSA, XOR operation is relatively resource friendly and lightweight, ideally suited for mobile and hand-held computing devices since they have limited hardware computing ability and power resources. The only remaining challenge is how to generate a strong cipher stream that ensures the quality of encryption to avoid key deciphering and that protects data integrity over wireless radio communication. Encryption algorithms used in this *i-key* protocol consist of a Key Scheduling Algorithm (KSA) that establishes an initial permutation *S-box* of *{0,1,2,.......,N-1}* of the numbers 0 to 255 from a random key array with the typical size of 40 to 256 bits and an Pseudo-Random Generation Algorithm (PRGA) that utilizes this output permutation *S-box* to generate the pseudo-random output sequence. The pseudocode for these two algorithms is shown in Fig. 5.



```
function KSA(k){
//initalization
  for i=0 to N-1{
    S[i]=i;
  }
  j=0;
//scrambling
  for i=0 to N-1{
    j= (j+S[i]+K[i mod keylength]) mod 256;
    swap(S[i],S[j]);
  }
}



function PRGA(k){
//initalization
  i=0;
  j=0;

//Generation output loop
  while Generationoutput:
  i=(i+1) mod 256;
  j=(j+S[i]) mod 256;
  swap(S[i],S[j]);
  k=(S[i]+S[j]) mod 256;
  r=S[k];
}
```

Fig. 5. Pseudocode of KSA and PRGA Algorithm

The KSA algorithm consists of two N loops of round operations that initialized the permutation array with a sequential number starting with 0 in the first loop and then rearranging the order by swapping each individual value with another byte in the same array with the following computational formula:

J(x) = (the value the particular index byte of S-box + the value of the same particular index byte of K-box) with any overflow ignored (6)

The value of *J* is used as an index, as well as the values at that location, and are swapped with the target value in that location in *S-Box*. *Sn* is denoted as the result of the first "n" iterations from the loop of scrambling that represents the process have swapped each of *S[0]...S[n-1]*, with a corresponding value of *S[j]*. The same process will start from the beginning of the initial *S-box* and is continuously repeated until it finishes swapping until the end of the array and produces the final version of *S*, *S256* in our *i-ke*y system as the output permutation *S-box*.

Once the *S-box*, the so-called state array, is initialized, it will be used as input in the next phase of *i-key* encryption algorithm, called the PRGA. This involves more calculation and swapping to generate the final key stream. A Pseudo-Random Number Generator (PRNG) is an algorithm used to generate a random sequence of numbers, the elements of which are approximately independent. The PRGA in the *i-key* protocol is responsible for creating the cipher stream used to encrypt the plaintext based on the *S-box* value, whish is the output from the KSA in the previous step. It first initializes two indices, *i* and *j* to 0, and then loops over five operations that increase the value of *i* in each loop as the counter, increasing *j* pseudo-randomly by adding one value *S[i]* to it, then swapping the two values of the *S-box* pointed by the value of *i* and *j*, and outputs the values of the *S-box* that is pointed to by *S[i]+S[j]*. Note that every block of *S-box/State* array is swapped at least once, possibly with itself, within each completed iteration loop, and hence the permutation *S-box/State* array evolves fairly rapidly during the generation output loop phase (Fluhrer et al., 2001).

The strength of a cryptographic system primary depends on two components: the algorithm and the encryption key. Since a system is only as strong as its weakest link, both components need to be strong enough to protect the unsecure wireless communication via the radio frequency (Edney & Arbaugh, 2004); (Chandra, 2005). In this *i-key* encryption protocol, first of all, the dynamic re-keying algorithm enormously enhances the level of protection by adding the extra secret *i-key* to the *K-box*. This increases not only the complexity of the secret key array but also effectively prevents key cracking and dictionary attacks. Second, it improves the level of data protection by creating a better initialized *S-box/State* array during the KSA algorithm when swapping the blocks based on the *j* index that are mixed with the value of additional secret *i-key*. Finally, it helps generate a better and stronger pseudorandom number stream in the PRGA algorithm phase that is used to encrypt the data packet sent via the wireless network. Therefore, this dynamic *i-key* encryption protocol strengthens the cryptographic system in both ways and provides a solid protection for both individual stand-alone wireless models as well as for mobile ad-hoc wireless networks.

4. Security analysis

Due to the nature of frequent changes in both topology and membership in mobile ad-hoc networks, the initial design of the wireless routing protocol has mainly focused on the effectiveness of packet forwarding and delivery to the target node, and not on security. Consequently, a number of attacks that take advantage of this weakness have been developed for use against data integrity or routing protocol in wireless communication.

Transmitted data packets may be exposed to unauthorized access at anytime and anywhere due to the nature of radio broadcasting; therefore, it is essential to apply security protection that prevents the reading or modification of confidential data by anyone who can receive the wireless signal. Using the secret key for data encryption is currently considered the most common way to protect data privacy in all kinds of computer communication; however, one of the static key or pre-shared key (psk) encryption's biggest vulnerabilities is that an attacker can obtain the original secret key by monitoring the packet transmission or conducting a massive dictionary attack between any two nodes in the network. Theoretically, a 64-bit secret key is decipherable with approximately 1 to 2 million data packets (2 to 4 million for 128-bit secret keys) and in a matter of mere hours, attackers can detect enough data packets in an average busy network environment to decode the pre-shared secret key (Chan et al., 2005).

In addition, mobile nodes are often deployed in a wide area with very limited or no physical protection, rendering them very vulnerable to capture or hijacking. Once a single node has been compromised and the secret key revealed, an attacker can launch far more damaging attacks from inside the network without being detected. Hence, the encryption protocol that applies to the mobile ad-hoc network should not only prevent the encryption key from been revealed, but also be flexible enough to be adopted as a security enhancement by other existing routing protocols in such highly dynamic network environment.

With the advanced dynamic encryption mechanism, *i-key* protocol ensures privacy of communication and protects sensitive data from eavesdropping by dynamically changing the secret *i-key*, which allows only the original sender and authorized receiver to decode the encrypted data packet via the secret *i-key* that they own. Therefore, this protocol overcomes the weakness of pre-shared key encryption and protects the wireless network against other attacks in the methods described below.

4.1 WarDriving

WarDriving is the act of scanning and searching for wireless network signals in a moving vehicle by any devices equipped with a wireless interface, such as PDAs or portable computers. Scanning software likes NetStumbler and Airmon-ng can report detailed information, including Service Set Identifier (SSID), MAC address, communication channel, signal strength and most importantly, the encryption protocol applied for each access point and wireless node. It can also record the location by connected to a GPS (Global Position System) receiver.

In addition, there are several online web sites and databases such as WiGLE/JiGLE, StumbVerter and Google Hotspot Maps where people around the world can report their discovery of each access point's information. In July 2010, WiGLE/JiGLE alone recorded 23,182,272 pieces of access point data from 1,125,930,947 unique observations, which cover most of the major cities on five continents. Therefore, other people who do not have the proper equipment for doing wardriving can simply locate any near by access point by searching these sites. As an example, take the city of College Station, where Texas A&M University is located. More than six thousand access points have been detected and reported to the WiGLE/JiGLE database. Fig. 6. demonstrates the distribution in a Google map.

Those scanning tools, access point information sources and online databases are convenient for wireless network studies and research, but they also provide an advantage by letting hackers pick the most vulnerable entry point from an existing wireless network and expected to spend less time and effort to compromise the target node and its local area network. That is also why running a wardriving scan is usually hackers' first step before they start any other kind of wireless attack.

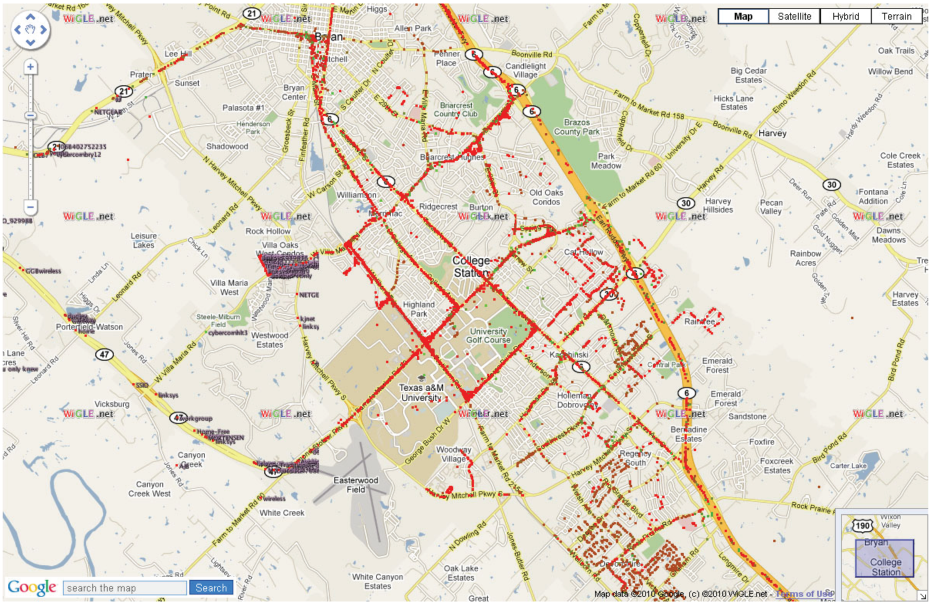
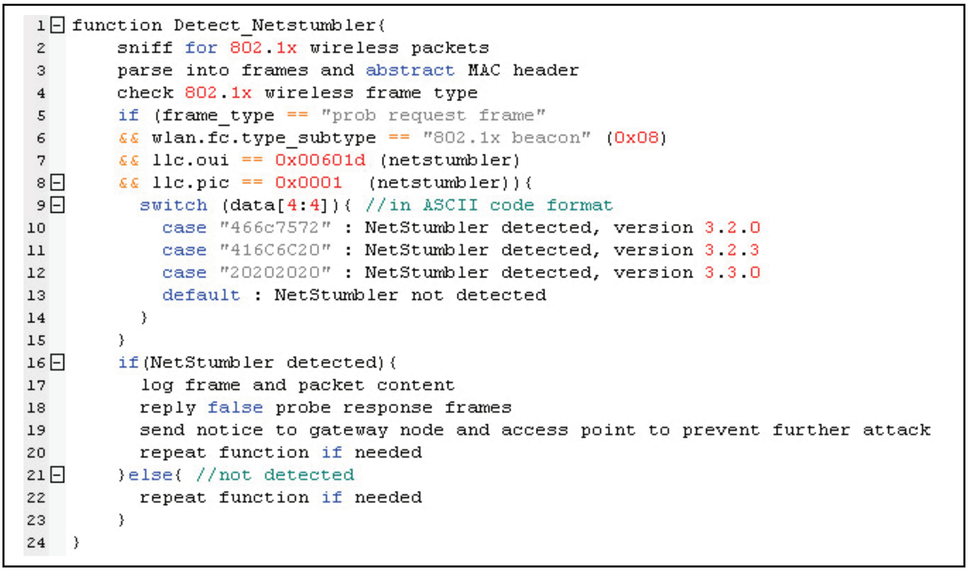


Fig. 6. The distribution of wireless access points in city of College Station, Texas

The dynamic *i-key* encryption protocol can recognize and prohibit wardriving attacks by adding wireless packet pattern analysis to both access point and mobile node. Take NetStumbler for example; this unique pattern can be found in its 802.11 probe request frames (Tsakountakis, 2007). First, LLC encapsulated frames generated by NetStumbler contain the valise 0x00601d for organizationally unique identifier (OID) and protocol identified (PID) of 0x0001. Second, the payload data size is usually 58 bytes with the attached hidden string "Flurble gronk bloopit, bnip Furndletrune!" for version 3.2.0, "All your 802.11b are belong to us" for version 3.2.3 and " intentionally left blank 1" for version 3.3.0. In (Tsakountakis, 2007), authors also illustrate the pseudocode for the above pattern detection in a traditional wireless network and we extended this for dynamic *i-key* protocol used in a mobile ad-hoc wireless network (Fig. 7.). Once the *i-key* system detects the presence of wardriving activities, it generates several false probe requests to prevent any further attacks by misleading attackers with fake MAC address, SSID, channel and encryption protocol. Similar detecting signature parameters and policies shown in Fig. 8 can also add to the intrusion detecting system (IDS) to prevent additional attack on a wireless network.

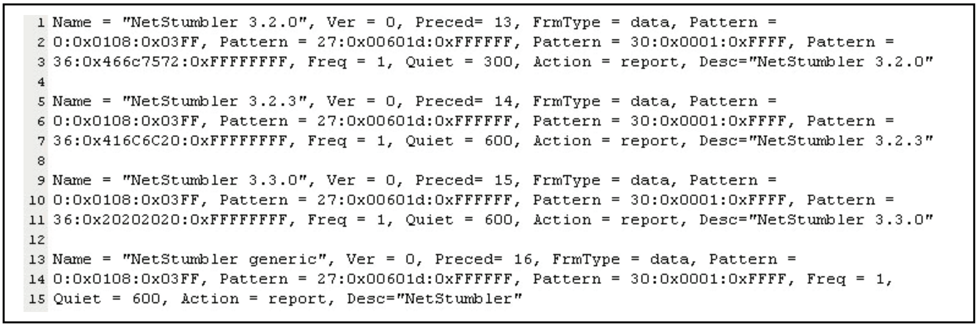
4.2 Man-in-the-Middle (MITM)

In a Man-in-the-Middle (MITM) attack, as shown in Fig. 9., the hacker places himself in the mid-point of the information flow between sender and recipient, which allows him to access all of the communication between them. If no proper security protection and data encryption protocol are applied to the wireless network, the attacker can effortlessly read the data, inject malicious packets, modify the information integrity or even block the communication from one side to another. In addition, a man-in-the-middle attack is hard to detect and prevent in a wireless network environment since everyone can easily capture the wireless packets transmitted from any mobile device to another or from the base stations.



```
function Detect_Netstumbler{
    sniff for 802.1x wireless packets
    parse into frames and abstract MAC header
    check 802.1x wireless frame type
    if (frame_type == "prob request frame"
    && wlan.fc.type_subtype == "802.1x beacon" (0x08)
    && llc.oui == 0x00601d (netstumbler)
    && llc.pic == 0x0001  (netstumbler)){
      switch (data[4:4]){ //in ASCII code format
        case "466c7572" : NetStumbler detected, version 3.2.0
        case "416C6C20" : NetStumbler detected, version 3.2.3
        case "20202020" : NetStumbler detected, version 3.3.0
        default : NetStumbler not detected
      }
    }
    if(NetStumbler detected){
      log frame and packet content
      reply false probe response frames
      send notice to gateway node and access point to prevent further attack
      repeat function if needed
    }else{ //not detected
      repeat function if needed
    }
}
```

Fig. 7. NetStumbler detecting pseudocode



```
Name = "NetStumbler 3.2.0", Ver = 0, Preced= 13, FrmType = data, Pattern =
0:0x0108:0x03FF, Pattern = 27:0x00601d:0xFFFFFF, Pattern = 30:0x0001:0xFFFF, Pattern =
36:0x466c7572:0xFFFFFFFF, Freq = 1, Quiet = 300, Action = report, Desc="NetStumbler 3.2.0"

Name = "NetStumbler 3.2.3", Ver = 0, Preced= 14, FrmType = data, Pattern =
0:0x0108:0x03FF, Pattern = 27:0x00601d:0xFFFFFF, Pattern = 30:0x0001:0xFFFF, Pattern =
36:0x416C6C20:0xFFFFFFFF, Freq = 1, Quiet = 600, Action = report, Desc="NetStumbler 3.2.3"

Name = "NetStumbler 3.3.0", Ver = 0, Preced= 15, FrmType = data, Pattern =
0:0x0108:0x03FF, Pattern = 27:0x00601d:0xFFFFFF, Pattern = 30:0x0001:0xFFFF, Pattern =
36:0x20202020:0xFFFFFFFF, Freq = 1, Quiet = 600, Action = report, Desc="NetStumbler 3.3.0"

Name = "NetStumbler generic", Ver = 0, Preced= 16, FrmType = data, Pattern =
0:0x0108:0x03FF, Pattern = 27:0x00601d:0xFFFFFF, Pattern = 30:0x0001:0xFFFF, Freq = 1,
Quiet = 600, Action = report, Desc="NetStumbler"
```

Fig. 8. NetStumbler signature parameters for CISCO IDS

There are many different ways to interrupt the communication and allow hackers to insert themselves in the middle of the information flow by taking advantage of the protocol's weak security design, for example, by using Address Resolution Protocol (ARP) spoofing (Plummer, 1982); (Wagner, 2001), Domain Name Server (DNS) spoofing (Klein, 2007); (Sax, 2000) or via Border Gateway Protocol (BGP) (Rekhter et al., 2003). Once hackers are able to access the communication channel, the next step is to capture the current session, decode the secret key, decrypt the message and then modify the content and send it back. First, the attacker needs to reveal the secret key before he can successfully alter any data packets and launch an attack on both sender and recipient.

However, due to the natural of this dynamic re-keying protocol, every single packet is secured by a unique and solid cipher stream composed of one hidden pre-shared secret key (psk), one unique IV value and one dynamic *i-key*, which together provide three strong layers of secure enhancement protection for wireless ad-hoc networks. Plaintext messages can only be decoded by authorized recipients and senders who have the legal and updated *i-key*. Therefore, a real-time man-in-the-middle attack would not succeed against this protocol.

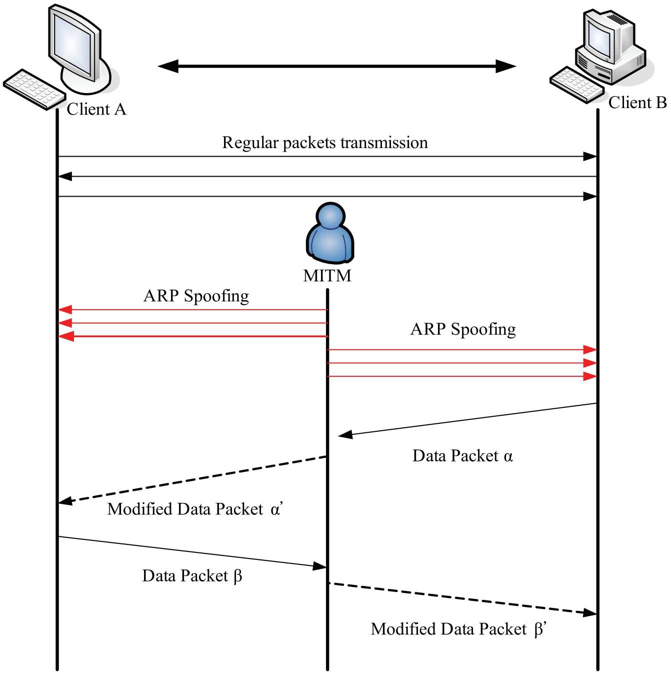


Fig. 9. Wireless man-in-the-middle attack example

4.3 Blackhole attacks

Blackhole attacks (Tamilselvan & Sankaranarayanan, 2008); (Hu & Perrig, 2004); (Chuah & Yang, 2006) (Fig. 10.) are similar to denial of services (DoS) attacks in traditional networks in that a compromised node in MANET participates in a routing protocol and attracts all packets by claiming to have a valid route to all destination nodes, but then drops all received data packets without forwarding them. This attack will not merely prolong the routing delay; in the worst case scenario, it can disrupt the entire network connection.

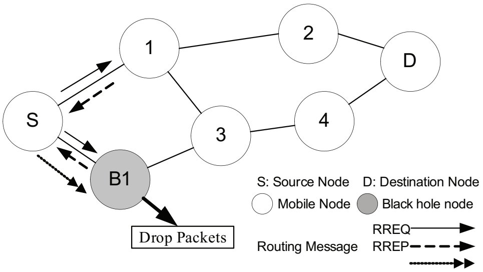


Fig. 10. Black hole attack in MANET

This attack is easily lunched against reactive protocols in a Mobile Ad-Hoc Network such as Dynamic Source Routing (DSR) (Johnson et al., 2001), Temporally Ordered Routing Algorithm (TORA) (V. D. Park & Corson, 1997) and Ad Hoc On-Demand Distance Vector (AODV) (Perkins & Royer, 1999), which assume that all nodes in a given ad-hoc network are trustworthy and that the data packet will forward to the node that first replies to the route reply message (RRM) in routing path discovery. To set in motion a blackhole attack, the attacker needs to decipher not only the pre-shared key (psk) but also the dynamic re-keying secret *i-key*; however, the attacker needs the added advantage of a dynamic re-keying mechanism that provides three different layers of data encryption and unique cipher streams to secure both the data and each mobile host's secret key for every transmitted packet over the mobile ad-hoc wireless network. The *i-key* encryption protocol can easily prevent this form of attack in its very early stages by stopping the node from compromised and controlled by the attacker.

4.4 Wormhole attacks

In wormhole attacks, an adversary establishes a wormhole link by using either in-band or out-of-band communication as illustrated in Fig. 11. This direct link can be set up with a traditional wire, long-range wireless transmission or an optical link. Once this wormhole link is built up, the attacker can receive wireless packets on one end in the network, known as the original point, and then reply to them in a timely fashion at another location, as the destination point.

Using this method, an attacker could relay an authentication exchange to gain unauthorized access without compromising any node or having any knowledge of the routing protocol in use (Chuah & Yang, 2006); (Eriksson et al., 2006). Because a wormhole attack is launched internally against the mobile ad-hoc network, default routing protocols and traditional security protections are unable to effectively detect or prevent this unique attack pattern.

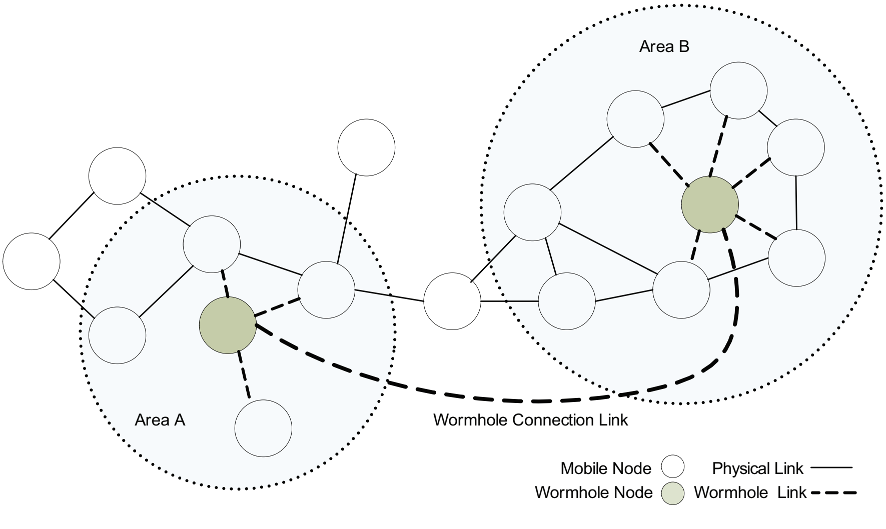


Fig. 11. Wireless wormhole attack

Under the protection of the *i-key* encryption protocol, however, only the original sender and authorized receiver are able to decrypt the cipher text, by using the unique secret key in their possession, ensuring continued confidentiality and integrity for the data communication, as well as the authentication information between source and destination node. Therefore, even if wormhole attacks are launched inside the network, the cryptographic key that is used for both encryption and decryption during each node-to-node communication still remains secret and the authentication information is still valid only to original node as well.

4.5 Session hijacking

In session hijacking, attackers take an authorized and authenticated session away from its owner and use it to establish a valid connection with the peer node, then snoop or modify the secret data. To successfully execute session hijacking, the attacker must accomplish two tasks: He first needs to stop the target node from continuing the session and then disguise himself as one of the legal client nodes (Welch & Lathrop, 2003).

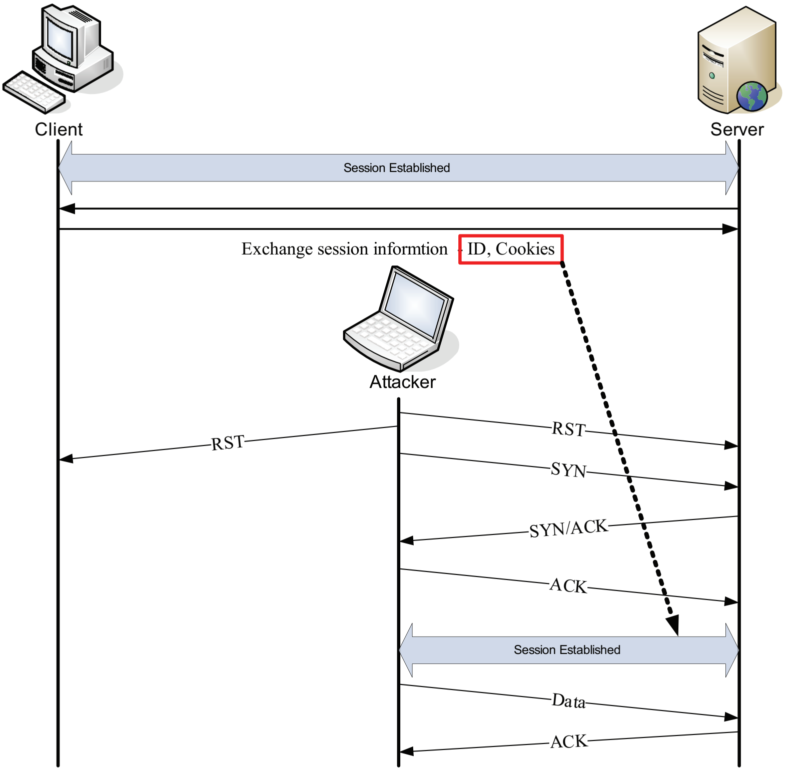
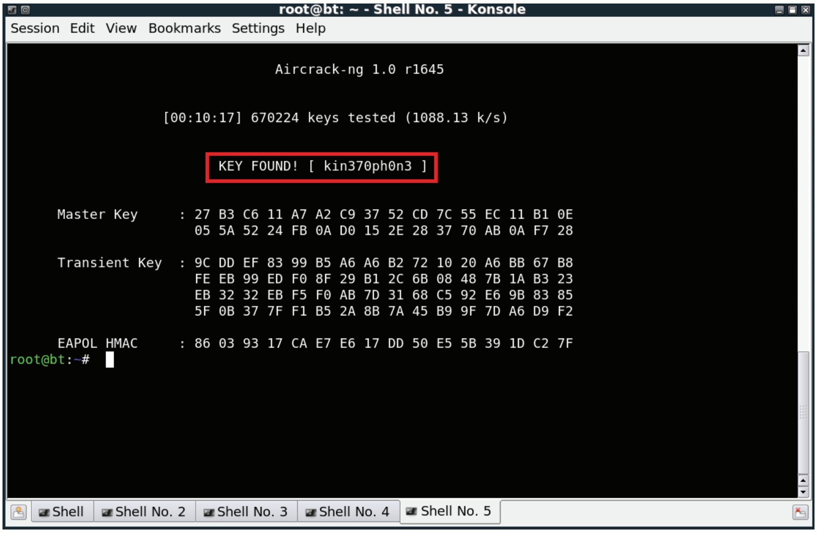


Fig. 12. Session hijacking attack example in IEEE 802.11 wireless network

The attacker can take the advantage of using Denial of Services (DoS) or a flood attack to achieve his first task for the session hijacking to temporarily interrupt the target's session connection; however, in order to masquerade himself as the target, he also needs to obtain the original secret key to maintain communication with the peer node. Because the *i-key* is dynamic re-keys for every packet, the secure key stream remains secret even if the session connection is interrupted. In this protocol design, described in the previous chapter, when communication is stopped or interrupted, the two parties will be notified by an IEEE 802.11 ACK-failed (timeout) or AODV routing error RRER message to restore the last successfully received data packet and the secret *i-key*. Therefore the security protection remains even when consistency session connections are lost.

4.6 Key cracking and dictionary attacks

Any encryption system using only static pre-shared key (psk) or lacking well-defined re-keying mechanisms are vulnerable to key cracking through the capturing of sufficient packets. Also, when choosing passwords for authentication or encryption system, many users select from a small domain and end up with a weak password. Those weak security systems and passwords enable adversaries to launch dictionary attacks that attempt to login into accounts by trying all possible password combinations. Once the correct password is discovered, attackers can crack the ciphertext easily and even carry out other attacks effortlessly (Pinkas & Sander, 2002). Fig. 13. below illustrates the key cracking attack with Aircrack-ng software.


root@bt: ~ - Shell No. 5 - Konsole
Session Edit View Bookmarks Settings Help
Shell
Shell No. 2
Shell No. 3
Shell No. 4
Shell No. 5

```
                              Aircrack-ng 1.0 r1645


                 [00:10:17] 670224 keys tested (1088.13 k/s)


                         KEY FOUND! [ kin370ph0n3 ]


      Master Key     : 27 B3 C6 11 A7 A2 C9 37 52 CD 7C 55 EC 11 B1 0E
                       05 5A 52 24 FB 0A D0 15 2E 28 37 70 AB 0A F7 28

      Transient Key  : 9C DD EF 83 99 B5 A6 A6 B2 72 10 20 A6 BB 67 B8
                       FE EB 99 ED F0 8F 29 B1 2C 6B 08 48 7B 1A B3 23
                       EB 32 32 EB F5 F0 AB 7D 31 68 C5 92 E6 9B 83 85
                       5F 0B 37 7F F1 B5 2A 8B 7A 45 B9 9F 7D A6 D9 F2

      EAPOL HMAC     : 86 03 93 17 CA E7 E6 17 DD 50 E5 5B 39 1D C2 7F
root@bt:~#
```

Fig. 13. Key cracking by Aircrack-ng

Dynamic re-keying in the manner used in *i-key* protocol is advantageous because not only is every stream cipher unique for each packet, but also the *i-key* system provides the wireless ad-hoc network with an innovative and solid security protocol of up to 18,432 bits, the maximum for the data packet size in IEEE 802.1x wireless communication (Borsc & Shinde, 2005), in key size. Therefore, attackers are unlikely to take the time required to capture enough packets before they can start to crack them or launch dictionary attacks against the system, because they know the longer they stay, the more likely their detection by a monitor system or firewall will be.

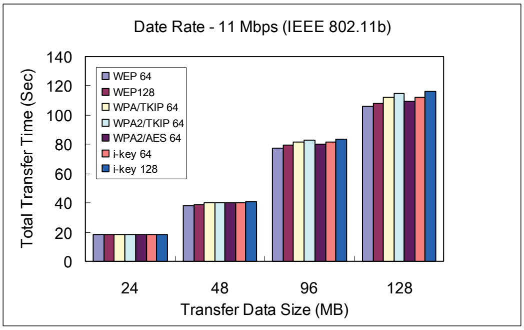
5. Performance evaluation

In these experiments, both 25 and 50 mobile nodes with 2 access points randomly located over an area of 600m x 600m and 1100m x 1100m are simulated with different settings of the size of the secret *i-key* that correspond to other security protocols. Each simulation ran for 200 simulated seconds with a radio transmission range set to 250 meters. Nodes coved by this range can receive the wireless signal and establish communication directly to the nodes within its ad-hoc range, while others rely on packets relayed by adjacent mobile nodes to deliver the message to the destination node. The physical and MAC layer setting is following the standard of IEEE 802.11 protocol with the data rate set from 1 to 20 MB/s.

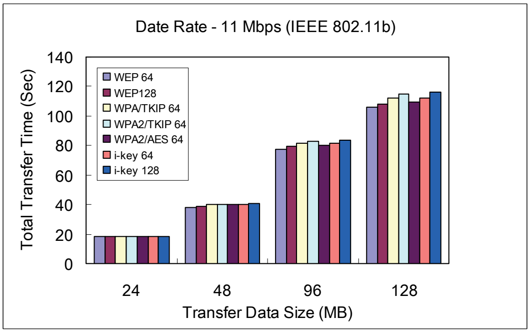
The kernel of this test bed is based on Fig. 3. and Fig. 5. for the *i-key* dynamic encryption protocol with the rewrite extension from CMU Monarch (Monarch Project, 1998) to support this dynamic re-keying architecture model for AODV routing in mobile ad-hoc network.

5.1 Protocol throughput

In the throughput experiment, two mobile nodes are randomly selected in the deployed area and measured the average of total complete time for four different sizes of data transferred between them. This protocol throughput test allowed us easily to compare the performance of *i-key* with WEP, WPA and WPA2 system, which are the most popular and adopted security protocols in today’s wireless networking. As seen in Fig. 14, there is almost no



(a) 25 mobile nodes over 600mx600m area



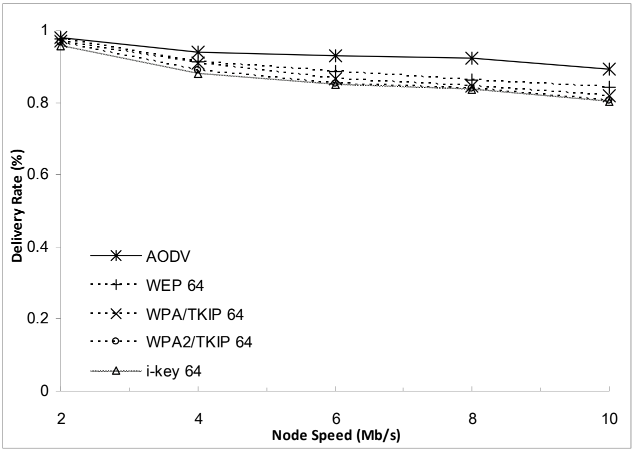
(b) 50 mobile nodes over 1100mx1100m area

Fig. 14. Average total data transfer time for *i-key* encryption protocol

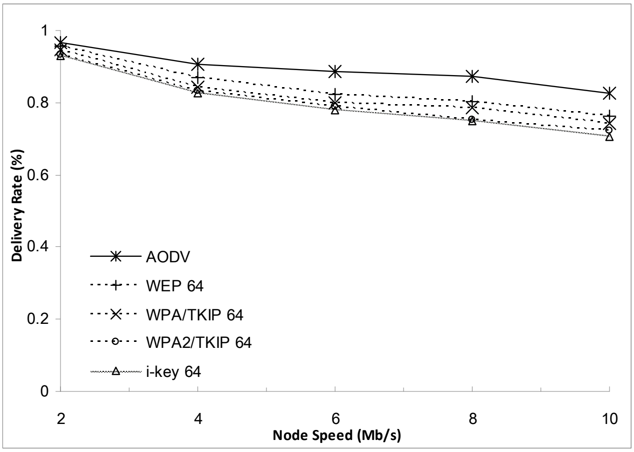
difference between each encryption approach in the lower transfer data size (24 and 48 MBs) and only a very small gap from the quickest WEP protocol with 64 bits to the slowest dynamic *i-key* 128 bits security system while transferred over 96 MBs of data. However, regarding data security, *i-key* encryption protocol not only strengthened the cipher by doubling the secret key size to provide a higher level of protection, but also dynamically re-keying during the end-to-end communication to defend the network from unwanted intrusion and guarantee the privacy of wireless data exchange.

5.2 Protocol delivery rate

The simulation results for protocol average delivery rate are shown in Fig. 15. The percentage of successfully delivered packets is measured from the source to the destination



(a) 25 mobile nodes over 600mx600m area



(b) 50 mobile nodes over 1100mx1100m area

Fig. 15. Average end-to-end delay for AODV and *i-key* protocol

node in five different data rate setting: 2, 4, 6, 8 and 10 MB/s. As expected, delivery rates dropped as the result of a greater number of lost packets and collisions in the wireless environment caused by the increased number of mobile nodes and data transfer speed. The nature of radio communication makes packet loss and collisions during transmission unavoidable. When this happens to the *i-key* dynamic encryption protocol, it only needs to retrieve the secret key from the most recently received data packet and then re-synchronize with both sides to continue the conversation. Consequently, the cost of time and overhead for packet loss and collision in the *i-key* protocol is quite low. This also is why the differences between *i-key* with other secure protocols are minimal.

Both the complexity of the encryption system and the size of the ad-hoc network have a negative effect on performance. Obviously, AODV alone had the best delivery rate in all of the simulations, a result of the trade-off between security and performance. However, the relatively small gap between them also underscores that this *i-key* protocol can perform as efficiently as a non-security protection such as an AODV routing protocol while providing stronger data privacy through the dynamic *i-key* encryption system.

Those results from throughput and end-to-end delay experiments also indicate that the *i-key* security mechanism has very low computational overhead and power consumption during both data encryption and decryption procedure, which is very critical, especially when most mobile nodes in the wireless network depend on limited processing ability and the finite energy provided by batteries (Wang & Chuang, 2004).

6. Conclusion and future research

Data integrity and privacy are the two most important security requirements in wireless communication today. Most mechanisms rely on pre-share key (psk) data encryption to prevent unauthorized users from accessing confidential information. However, maintaining security in the highly dynamic ad-hoc wireless network is full of challenges due to the complexity of data routing and the nature of the wireless transmission medium.

In this chapter, we introduced a novel, efficient and lightweight encryption protocol that fulfils the need for security protection in wireless ad-hoc networks. This protocol ensures the privacy of communication from node to node and prohibits the modification of sensitive data by dynamically changing the secret key for data encryption during packet transmission. Under the protection of this protocol, only the original sender and authorized recipient are able to decode the cipher text using the secret key that is in their possession only. Therefore, the weakness of pre-shared key encryption is overcome and other wireless attacks are prevented. Experiment results with different network configurations and key sizes have been simulated. They indicate that this *i-key* protocol design is efficient, with low commutation overhead, while providing better and stronger data protection compared with other common security protocols in IEEE 802.11 wireless network. Furthermore, the dynamic encryption and decryption architecture in *i-key* protocol is flexible; other secure systems can also adopt it as a secondary security enhancement without compromising system performance.

The future works include the integration of this existing work with the intrusion detection and locating system. This integration provides another layer of defense by effectively pinpointing the location of an attacker and helps the wireless secure system to react correctly and instantly. Also, the implementation of advanced dynamic secure protection for large-scale wireless communication, such as IEEE 802.16 WiMAX network and the 4G (4th generation) of the cellular wireless network is also recommended, with evaluation of protocol performance in both lab software simulations and real-world experiments.

7. References

Borsc, M., & Shinde, H. (2005). Wireless security & privacy. In *2005 IEEE International Conference on Personal Wireless Communications, 2005. ICPWC 2005*, pp. 424-428, 2005

Chan, F., Ang Hee Hoon, & Issac, B. (2005). Analysis of IEEE 802.11b wireless security for university wireless LAN design. In *Networks, 2005*. doi:10.1109/ICON.2005.1635688

Chandra, P. (2005). *Bulletproof wireless security: GSM, UMTS, 802.11 and ad hoc security.* Elsevier, 0750677465

Chuah, M., & Yang, P. (2006). Comparison of Two Intrusion Detection Schemes for Sparsely Connected Ad Hoc Networks. In *Military Communications Conference, 2006. MILCOM '06*, pp. 1–7, 2006

Clausen, T., & Jacquet, P. (2003). RFC3626: Optimized Link State Routing Protocol (OLSR). *RFC Editor United States*.

Clausen, T., Jacquet, P., Adjih, C., Laouiti, A., Minet, P., Muhlethaler, P., Qayyum, A., et al. (2003). Optimized link state routing protocol (OLSR).

Edney, J., & Arbaugh, W. A. (2004). *Real 802.11 security: Wi-Fi protected access and 802.11 i.* Addison Wesley Publishing Company, 0321136209

Eriksson, J., Krishnamurthy, S. V., & Faloutsos, M. (2006). Truelink: A practical countermeasure to the wormhole attack in wireless networks. In *Proceedings of the 2006 14th IEEE International Conference on Network Protocols, 2006. ICNP'06*, pp. 75–84, 2006

Fluhrer, S., Mantin, I., & Shamir, A. (2001). Weaknesses in the Key Scheduling Algorithm of RC4. In *Selected Areas in Cryptography*, pp. 1-24

Gast, M. (2002). Wireless LAN security: A short history. Available online at: http://www.oreillynet.com/pub/a/wireless/2002/04/19/security.html

Hu, Y. C., Johnson, D. B., & Perrig, A. (2003). SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks. *Ad Hoc Networks*, *1*(1), pp. 175–192

Hu, Y. C., & Perrig, A. (2004). A survey of secure wireless ad hoc routing. *IEEE Security and Privacy magazine*, *2*, pp. 28–39.

Hu, Y. C., Perrig, A., & Johnson, D. B. (2005). Ariadne: A secure on-demand routing protocol for ad hoc networks. *Wireless Networks*, *11*(1), pp. 21–38.

Hubaux, J. P., Buttyán, L., & Capkun, S. (2001). The quest for security in mobile ad hoc networks. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*

Johnson, D. B., Maltz, D. A., Broch, J., & others. (2001). DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks. *Ad hoc networking*, *5*, pp. 139–172

Johnson, D. B., Maltz, D. A., Hu, Y. C., & Jetcheva, J. G. (2002). *The dynamic source routing protocol for mobile ad hoc networks*. Internet-Draft

Kant, L., Demers, S., Gopalakrishnan, P., Chadha, R., LaVergne, L., & Newman, S. (2005). Performance modeling and analysis of a mobile ad hoc network management system. In *MILCOM*, Vol. 5

Klein, A. (2007). *BIND 9 DNS cache poisoning*. Available online at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.86.4474&rep=rep1&type=pdf

Lansford, J., & Bahl, P. (2000). The design and implementation of HomeRF: A radio frequency wireless networking standard for the connected home. *Proceedings of the IEEE*, *88*(10).

Liu, X., Fang, Z., & Shi, L. (2007). Securing Vehicular Ad Hoc Networks. In *2nd International Conference on Pervasive Computing and Applications, 2007. ICPCA '07*, pp. 424-429

Miller, S. K. (2001). Facing the challenge of wireless security. *Computer*, *34*(7), pp. 16–18

Monarch Project (1998). Rice Monarch Project and Wireless Mobility Extension to ns-2

Papadimitratos, P., & Haas, Z. J. (2002). Secure routing for mobile ad hoc networks. In *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, Vol. 31

Park, J. S., & Dicoi, D. (2003). WLAN security: current and future. *IEEE Internet Computing*, *7*(5), pp. 60–65.

Park, V. D., & Corson, M. S. (1997). A highly adaptive distributed routing algorithm for mobile wireless networks. In *IEEE Infocom*, Vol. 3, pp. 1405-1413

Perkins, C. E., & Bhagwat, P. (1994). Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of the conference on Communications architectures, protocols and applications*

Perkins, C. E., & Royer, E. M. (1999). Ad-hoc on-demand distance vector routing. In *2nd IEEE Workshop on Mobile Computing Systems and Applications,*New Orleans, LA.

Pinkas, B., & Sander, T. (2002). Securing passwords against dictionary attacks. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 161-170

Plummer, D. C. (1982). RFC-826 An Ethernet Address Resolution Protocol. *Network Working Group*.

Prasithsangaree, P., & Krishnamurthy, P. (2004). On a framework for energy-efficient security protocols in wireless networks. *Computer Communications*, *27*(17), pp.1716–1729.

Ramakrishnan, K., Balasubramanian, A., Mishra, S., & Sridhar, R. (n.d.). Wireless Security Protocol using a Low Cost Pseudo Random Number Generator, 2005.

Rekhter, Y., Li, T., Hares, S., & others. (2003). *RFC-1771 A border gateway protocol 4 (BGP-4)*. RFC 1771, March 1995.

Rivest, R. L. (1992). The RC4 Encryption Algorithm. RSA Data Security. *Inc., March*, *12*.

Sanzgiri, K., LaFlamme, D., Dahill, B., Levine, B. N., Shields, C., & Belding-Royer, E. M. (2005). Authenticated routing for ad hoc networks. *IEEE Journal on Selected Areas in Communications*, *23*(3), pp. 598–610.

Sax, D. (2000). DNS spoofing (malicious cache poisoning). *November*, *12*. Available online at http://www. sans. org/rr/firewall/DNS_spoof. php

Tamilselvan, L., & Sankaranarayanan, V. (2008). Prevention of co-operative black hole attack in MANET. *Journal of networks*, *3*(5), 13.

Tsakountakis, A., Kambourakis, G., & Gritzalis, S. (2007). Towards effective Wireless Intrusion Detection in IEEE 802.11i. *Third International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing*, 2007

Wagner, R. (2001). Address resolution protocol spoofing and man-in-the-middle attacks. *The SANS Institute*.

Wang, Y. H., & Chuang, C. C. (2004). Ad hoc on-demand backup node setup routing protocol. *Journal of Information Science and Engineering*, *20*(5), pp. 821–843.

Welch, D., & Lathrop, S. (2003). Wireless security threat taxonomy. In *Information Assurance Workshop, 2003. IEEE Systems, Man and Cybernetics Society*, pp. 76–83

Zhou, L., & Haas, Z. J. (1999). Securing ad hoc networks. *IEEE network*, *13*(6), pp. 24–30.

24

Security of Access in Hostile Environments Based on the History of Nodes in Ad Hoc Networks

Saud Rugeish Alotaibi
De Montfort University
United Kingdom, England

1. Introduction

An ad hoc wireless network is built on cooperation between two or more nodes with wireless links and networking capability. The major applications of such networks today are tactical military and other security-sensitive operations. For example, military and police units (e.g. soldiers, tanks, police cars) equipped with wireless communication devices can form ad hoc wireless networks when they roam in insecure environments. Such networks can also be used for emergency, law enforcement and rescue missions. Since they have relatively low cost and can be deployed rapidly, they also constitute a viable option for commercial uses such as sensor networks and emergency situations, and there is a trend to adopt them for commercial uses due to their unique properties. The most critical challenge in the design of these networks is their security in hostile environments [81-86].

Their nodes are independent units which rely not on a central infrastructure but on neighbouring nodes to route each packet to the destination node. Ad hoc wireless networks can therefore work properly only if the participating nodes cooperate with each other in routing and forwarding. Nodes lack physical protection and are always under threat of being captured and compromised. They carry user and device histories, as each node can obtain data on all events involving a specific user and a specific device; therefore, each has to be able to document the user and the device at the registration stage.

The security requirements for different services range from highly security-sensitive military tactical operations, such as battlefields, rescue missions and emergency situations, to instantaneous classroom applications and areas where density is too small to justify economically the deployment of a network infrastructure. Attacks on ad hoc wireless networks can come from any direction and can target any node. Thus, ensuring a secure environment is as important as for wired networks, which have several lines of defence such as firewalls and gateways. Security depends on access to the history of each unit, which is used to calculate the cooperative values of each node in the environment. The calculated cooperative values are then used by the relationship estimator to determine the status of the nodes. Every node should be capable of making its own security decisions based on cooperation with other peer nodes.

The rest of the chapter is organized in the following manner. Section 2 discusses the requirements for any security solution, while section 3 explains the secure environment.

Section 4 describes the creation of public/ private keys and digital certificates, section 5 sets out the components of our architecture, section 6 presents and explains the activity diagram and section 7 presents a case study. Section 8 concludes the chapter.

2. Security requirements

The following are the security requirements to be met by a secure environment:

- **Authentication**: Ensures the identity of the node with which the communication is carried out. This avoids impersonation.
- **Availability**: Ensures that the eligible nodes are able to obtain the required services despite denial-of-service attacks.
- **Non-repudiation**: Ensures that a node cannot deny a particular action performed by it at a later stage. This could help in the detection of compromised nodes.
- **Detection of malicious nodes**: Ensures that nodes are capable of detecting the presence of malicious nodes in the environment, thus avoiding the participation of such nodes in the routing process.
- **Stability**: Ensures that a node is able to revert to its normal operating state within a finite time after any attack.

3. Secure environments

In a secure environment, some of the ad hoc nodes are involved in other infrastructure-based wireless networks such as WLANs and cellular systems; therefore, each of the ad hoc nodes will belong to an operation service provider (OSP), as shown in Figure 1. Other non-managed ad hoc network nodes, which are not involved in any other wireless networks, will be managed by the OSP, in order for those undefined nodes or networks to be able to access our secure environment. The following sections show how our SE consists of a number of ad hoc wireless networks interconnecting with each other.

3.1 Node classification

Nodes in the SE are classified thus:

- **User Nodes** are normal ground nodes; typically, soldiers equipped with devices of limited communication and computation ability whose duty it is to collect data and transfer it to a network backbone node.
- **Network Backbone Nodes** are usually units or master nodes located within the same network, for example in towers or tanks. NBBNs can establish direct wireless links to communicate amongst themselves.
- **Operation Service Providers** are usually units in the environment. This type of node will have many management, registration and control functions, such as duty signing and creating new certificates for different nodes in the secure environment.

3.2 Node documentation

All nodes in the secure environment are also placed into three categories according to their documentation status.

- **Documented nodes** are those which are documented by the OSP. Information on these nodes and their history is stored in a database (DB) authenticated by the OSP.

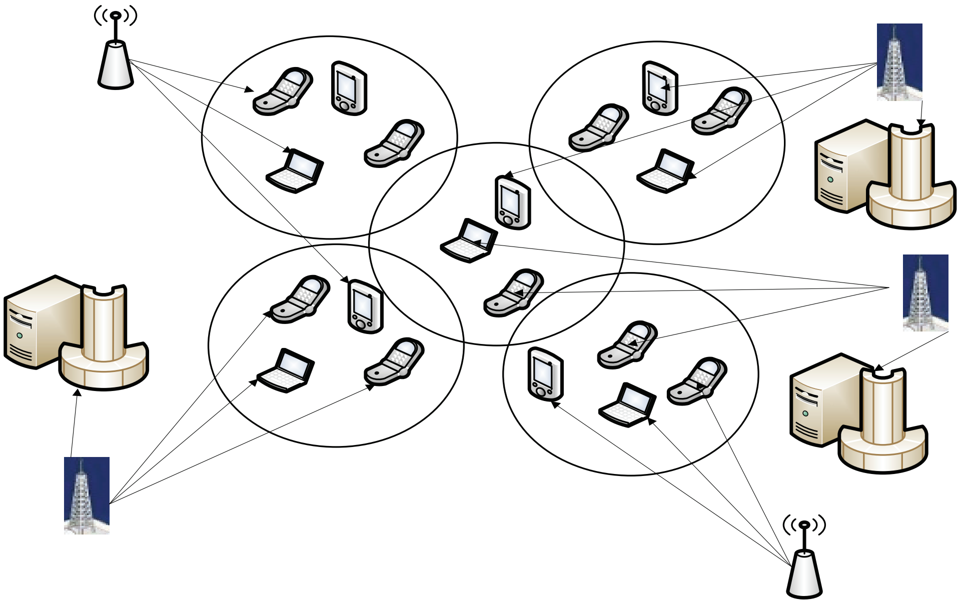


Fig. 1. Secure environment

- **Certificate-documented nodes** are those which possess a certificate issued by the OSP. They will have come into contact with a secure environment earlier and the certificate will verify that they are secure. Information on these nodes is stored in the documented DB of the OSP and they do not have any history in the documented DB.
- **Undocumented nodes** are those in the secure environment which do not fall into either of the above two categories. This category may also contain nodes which could have been certificate-documented by an OSP, but remain undocumented because there has been no need to verify their certificates.

4. Digital operation certificate management framework

This section describes the certificate management system of a secure environment. It shows how public/private keys and digital operation certificates are created. It also illustrates the process of certificate revocation.

4.1 Creation of public/ private keys and digital certificates

The public keys and the corresponding private keys of secure environment nodes are created by the OSP, which also issues the public-key certificates of SE nodes. Since a key is unique, (K_{public}) is unique and thus $H(K_{public})$, the fingerprint of K_{public}, is also unique and is considered the identifier in an SE. The operation certificate is used as permission to access this environment. Each node in the secure environment holds its digital operation certificate in its node database. The main structure of digital operation certificates contains [70] the MAC address of the node, its public key, the name of the OSP issuing this certificate, the certificate issue and expiry dates and the public key of the OSP. Finally, the contents of the certificate are attached to the digital signature of the OSP.

- **Node Identifier (ID)**: Holder of the certificate
- **MAC address of device (Mac)**: The unique serial number of the device
- **Node Public Key (K_{public})**: A unique key that is the fingerprint of the user
- **Certificate Operation OSP Identifier**: Name of the OSP that created and signed the certificate
- **Certificate Issue Date/Time**: The first day on which the certificate is valid
- **Certificate Expiry Date/Time**: The last day on which the certificate is valid
- **OSP Digital Signature**: Digital signature of the OSP.

4.2 Digital operation certificate distribution

Certificate distribution is a very important and low-cost mechanism that allows SE nodes to send the certificates they hold. Each node periodically starts receiving its physical neighbour (in one hop), its digital operation certificate and the corresponding OSP's public key stored in its NDB. Each node receives these certificates, compares them with its NDB and adds whatever new certificates it does not hold, as well as the public keys of its issuer; or it adds the renewal of an expired, extant certificate. The certificate distribution process is repeated at regular time intervals (RTIs). All nodes will have almost all digital operation certificates based on the mobility of the nodes and the RTI.

4.3 Revocation of digital operation certificates

The digital certificate management system provides certificate revocation as one of its basic services. There are two types of certificate revocation in our algorithm. Explicit revocation occurs when any node has a certificate and the OSP revokes it. The OSP sends the corresponding revocation to the other nodes belonging to the SE. If it cannot send the corresponding revocation for any reason, the renewal of the certificate can be denied, resulting in an implicit revocation.

In general, the OSP, when issuing the certificate, determines its issuing and validity times. All certificates are revoked after their expiration time. Therefore, the OSP should be updated about the certificates of SE nodes before the expiration time. In both types of revocation, when the OSP provides the SE nodes with information about any certificate, it should be distributed through the exchange process. In this way the nodes in the secure environment will be provided with this new information. Consequently, the OSP is responsible for the certificate revocation process and for transferring these revocations to all SE nodes. All SE nodes are informed when any of them carries out an explicit revocation and their NDBs are subsequently modified. This revocation will be distributed to the other nodes in the secure environment, both by certificate distribution and the process by which NDBs are merged.

The OSP is responsible for updating those certificates that have been implicitly revoked at regular intervals. Each SE node that has a new certificate will update its NDB, and then transfer the new certificate to its neighbours through the certificate distribution process. If any node does not receive the new certificate through the distribution and merging processes, and needs to validate the key, a new certificate will be requested from the OSP itself.

5. Components of our architecture

The components of our architecture are as follows:

User Nodes, as set out in 3.1 above, are typically soldiers or persons equipped with devices of limited communication and computation ability, whose duty is to deal with nodes, collect data and transfer them to NBBNs.

Network Backbone Nodes are usually units or master nodes located within the same network, for example towers or tanks. NBBNs can establish direct wireless links for communication among themselves. There are three divisions which carry out many functions (management, observation, control and so on) for the network. Their responsibility is to collect data, to observe nodes entering the network and to record the histories and certificates of all other nodes.

Operation Service Providers are usually units in the environment whose five divisions carry out many functions (management, registration, control and so on) for that environment. Their responsibility is to register new nodes, collect and analyse data, update the history of nodes and observe nodes entering the environment. The OSP has six units, which are the Registration Unit (RU), the Operation Certificate Unit (OCU), the Data Packet Collection Unit, the Analyser Unit (AU), the History Model Unit and the Database Unit.

The responsibility of the **Registration Unit** is to register a new node and apply the policy of the unit. Registration is an important stage before issuing a digital operation certificate for a node, as it verifies the identity of the user. This is the function of the RU. The user provides the RU with essential information: the user's name, the MAC address of the device and the fingerprint of the user.

The **Operation Certificate Unit** is the main service provided by the OSP. When the OCU receives a certification request from the RU, the OSP issues a digital operation certificate and signs it with its private key. The structure of the certificate should be defined by being standardised to ITU-T recommendation X.509, for example. All the information needed to complete the certificate will be provided by the RU.

The **Data Packet Collection Unit** collects the data packets in a secure environment and saves them in the main buffer. The data collector enables the packet analyser to use data collection containers to analyse all available data that the system has collected from the different nodes. At the same time it enables the packet analyser to process the transferred information, which can be used to obtain and save data that is gathered from several sources [106].

The **Database Unit** stores information on each node in a secure environment, including information regarding the history model of each node. It also keeps information like H (K_{public}), K_{public}, the fingerprints of each node and the MAC address of each device. Finally, it holds information regarding digital operation certificates and their revocation, to help in restricting future access with the same certificate.

The **History Model Unit** is used to calculate the cooperation values of each node in the environment. Our secure environment access system uses the history of nodes to build several lines of protection, equivalent to firewalls and gateways in wired networks. This unit receives data on the classification of nodes from the analyser base to analyse the packets. There are three kinds of node, as follows.

1. Positive Node (POSN). This is considered a cooperative node which, concerning packets or messages, will:
   - Notify its neighbours of any misbehaviour
   - Send an update to its neighbours when it receives new information
   - Forward any notification it receives from the OSP or NBBN
   - Notify its neighbours about any problem occurring with itself.

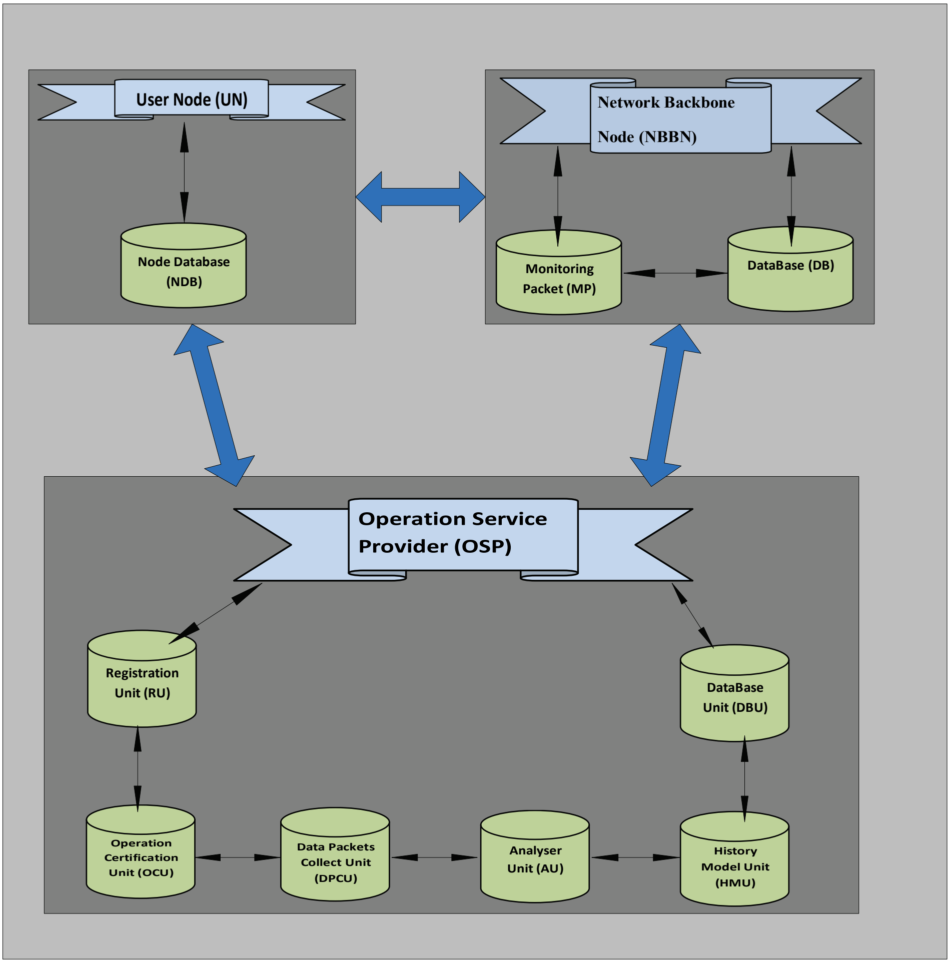


Fig. 2. Components of our architecture

The history of positive node

$$\left(\text{HPOSN}\right) = \frac{\Sigma \text{ all events of (POSN)}}{\Sigma \text{ all events of node}} \tag{1}$$

2. Natural Node (NATN). This is considered an uncooperative node and carries out normal work, such as:
   - Regular forwarding
   - Sending regular updates to its neighbours
   - Sending acknowledgment messages

The history of natural node

$$\left(\text{HNATN}\right) = \frac{\Sigma \text{ all events of (NATA)}}{\Sigma \text{ all events of node}}. \tag{2}$$

3. Negative Node (NEGN). This type misbehaves and does not send natural packets and messages. It is not considered a natural node because it:
    - Does not perform regular forwarding
    - Does not send regular updates to its neighbours
    - Does not send acknowledgment messages
    - Carries out misbehaviour
    - Tries to attack, for example by sending invalid certificates or invalid public keys, or sending many packets to a specific node.

The history of negative node

$$\left(\text{HNEGN}\right) = \frac{\Sigma \text{ all events of (NEGN)}}{\Sigma \text{ all events of node}} \tag{3}$$

The **Analyser Unit** checks each packet or message in the secure environment that the main buffer has collected. It then classifies all packets according to their contents. The analyser deals with all definition of packets and messages. It has the ability to define and classify unknown packets and add to the classification. The analyser will classify the nodes in the secure environment into three categories, POSN, NATN and NEGN, according to Table 1.

Event and behaviour	**Positive Node**	**Natural Node**	**Negative Node**
Regular forwarding		√	
Sends regular updates to its neighbours		√	
Sends acknowledgment messages		√	
Notifies its neighbours of any misbehaviour by others	√		
Send updates to its neighbours when it receives new information	√		
Forwards any notification it has received from OSP or NBBN	√		
Notifies its neighbours of any problem occurring with itself	√		
Carries out misbehaviour			√
Tries to attack (sends invalid certificate or invalid public key, or sends many packets to a specific node)			√

Table. 1. Node classification by analyser unit

6. Activity diagram

The activity diagram (Figure 3) depicts the steps taken by a node while handling requests to access a secure environment. The following are the steps shown in the activity diagram:

- Request from node *J* to node *I*
- Node *I* checks whether node *J* is registered
- If node *J* is not registered and node *I* ignores its request then node *J* transfers to the registration stage.

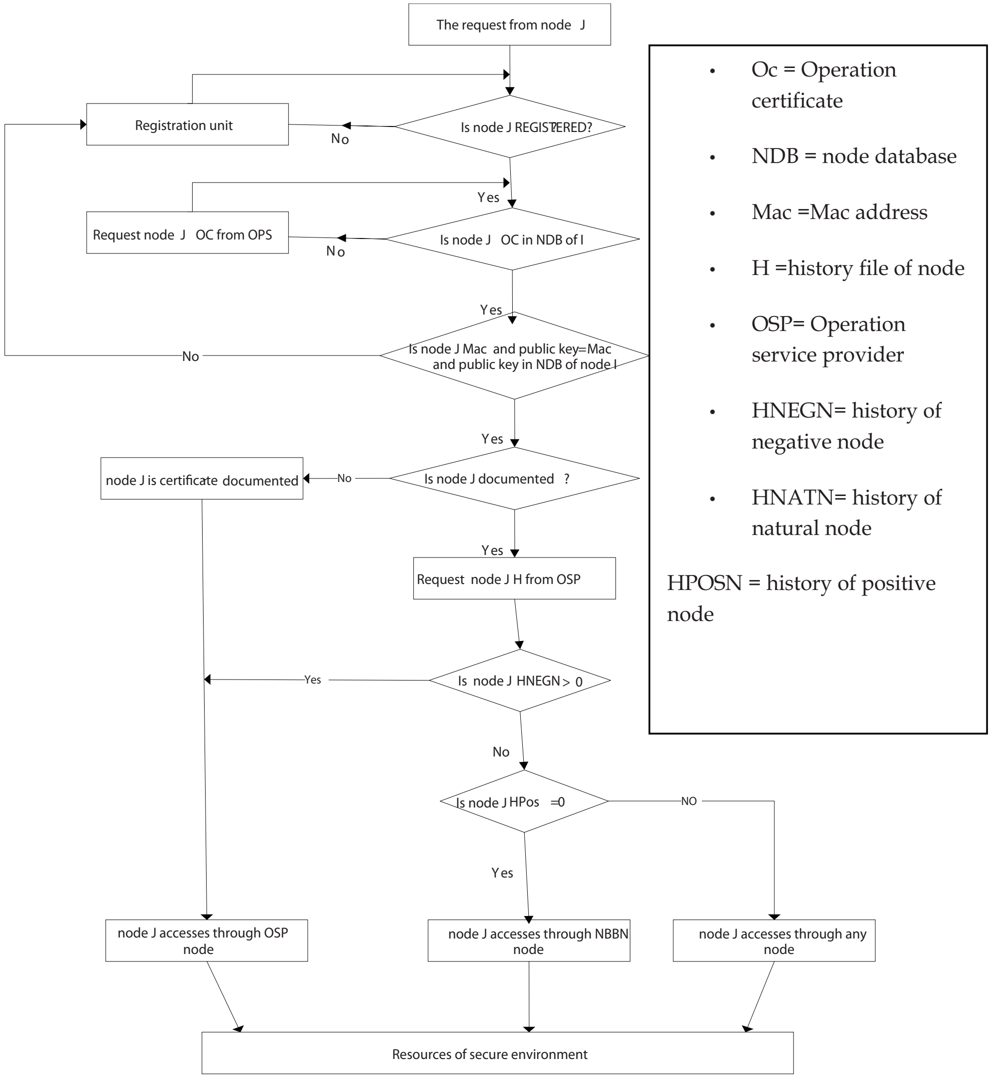

The request from node J
Registration unit
No
Is node J REGISTERED?
Yes
Request node J OC from OPS
No
Is node J OC in NDB of I
Yes
No
Is node J Mac and public key=Mac and public key in NDB of node I
Yes
node J is certificate documented
No
Is node J documented ?
Yes
Request node J H from OSP
Yes
Is node J HNEGN > 0
No
Is node J HPos =0
NO
Yes
node J accesses through OSP node
node J accesses through NBBN node
node J accesses through any node
Resources of secure environment
• Oc = Operation certificate
• NDB = node database
• Mac =Mac address
• H =history file of node
• OSP= Operation service provider
• HNEGN= history of negative node
• HNATN= history of natural node
HPOSN = history of positive node

Fig. 3. Activity diagram

- If node *J* is registered then node *I* checks for the operation certificate (OC) of node *J* in its database.
- If the OC of node *J* is not in the database then node *I* requests it from the OSP.
- If the OC of node *J* is in the database then node *I* checks that the MAC address and public key of node *J* are valid
- If the MAC address and public key of node *J* are not valid then node *I* ignores its request and node *J* transfers to the registration stage
- If the MAC address and public key of node *J* are valid then Node *I* checks whether node *J* is documented

- If node *J* is not documented, then it is certificated-documented
- If node *J* is certificated-documented then it can access the secure environment through the OSP
- If node *J* is documented then node *I* requests the history of node *J* from the OSP.
- If the history of negative node of node *J* > 0 then Node *J* can access the secure environment through the OSP
- If the HNEGN of node *J* < 0 then If the HPOSN of node *J* = 0 then Node *J* can access the secure environment through an NBBN.
- ELSE node *J* can access the secure environment through any node.

7. Formal description

The formal description of the Secure Environment (SE) is as follows.

7.1 Network model

In a secure environment, some of the ad hoc nodes are involved in other infrastructure-based wireless networks such as WLANs and cellular systems; therefore, each of the ad hoc nodes will belong to an operation service provider (OSP), as shown in Figure 1. Other non-managed ad hoc network nodes, which are not involved in any other wireless networks, will be managed by the OSP, in order for those undefined nodes or networks to be able to access our secure environment. Our secure involves a number of MANET interconnecting with each other; in addition all PKI are pre-connected by wireless connection to exchange data, and to update information.

Pki: Public key of network *i*, 1≤*i*≤*n*;

Those OSP are fully trusted by all nodes that belong to this secure environment. Nodes in the SE are classified thus:

- User Nodes(UN) are normal ground nodes;
  n: Number of networks in the SE; networks are numbered from 1 to *n*;
- Network Backbone Nodes(NBBN) are usually units or master nodes located within the same network,
  ni: Number of nodes, including Network Back Bone Node (NBBN), in the network *i*, 1 ≤*i*≤*n*; nodes in a network *i* are numbered from 1 to *ni*;
- Operation Service Providers (OSP) is usually units in the environment.
  k*i*: Number of Operation Service Providers (OSPs) in SE *i*, 1≤ ≤*n*;
  Pki: Public key of network *i*, 1≤*i*≤*n*;

7.2 Behaviour model

In a secure environment, the behaviour of nodes capture by operating service node (OSP) and stores in history file of behaviours that nodes might have (positive node , negative node and natural node). Positive Node (POSN) This is considered a cooperative node which, concerning packets or messages, will: Notify its neighbours of any misbehaviour, Send an update to its neighbours when it receives new information, Forward any notification it receives from the OSP or NBBN and Notify its neighbours about any problem occurring with itself.

POSN: Positive node in the SE, for 1≤ *i* ≤*n*.

Natural Node (NATN), this is considered an uncooperative node and carries out normal work, such as: Regular forwarding, Sending regular updates to its neighbours and sending acknowledgment messages.

NATN: Natural node in the SE, for $1 \le i \le n$.

Negative Node (NEGN), this type misbehaves and does not send natural packets and messages. It is not considered a natural node because it: Does not perform regular forwarding, does not send regular updates to its neighbours, does not send acknowledgment messages, carries out misbehaviour, tries to attack, for example by sending invalid certificates or invalid public keys, or sending many packets to a specific node.

NEGN: Negative node in the SE, for $1 \le i \le n$.

During specific period of time; this capture is always updated depending on the observed node actions, despite the fact that saving all behaviours is impossible; nevertheless, a reasonable number of behaviours must be stored.

7.3 Mobility model

Our secure environment (SC) is proposed for *ad hoc* wireless networks with a minimum number of mobile nodes. The proposed algorithm requires a different minimum number of nodes in the network to guarantee establishment of connection between nodes. In secure environment (SC), each node sends an RREQ packet to only one neighbor or operating service provider (OSP). In an ad hoc network, however, there are many situations where mobile nodes move together or form groups (the heading direction angle of nodes in each group is nearly similar). For example, vehicles on a road or, in a military scenario, a group of soldiers searching a particular plot of land, all working together in a cooperative manner to accomplish a common goal.

The following variables represent the parameters of the SE:

- n: Number of networks in the SE; networks are numbered from 1 to n;
- ni: Number of nodes, including Network Back Bone Node (NBBN), in the network i, $1 \le i \le n$; nodes in a network i are numbered from 1 to ni;
- ki : Number of Operation Service Providers (OSPs) in SE i , $1 \le i \le n$;
- h_i : History of node in SE i , $1 \le i \le n$;
- DOCxj: Digital Operation Certificate of node i in the SE, for $1 \le i \le n$
- DOCM : Documented of node i in the SE, for $1 \le i \le n$
- C-DOCM: Certificate -Documented of node i in the SE, for $1 \le i \le n$
- 2POSN: Positive node i in the SE, for $1 \le i \le n$
- NATN: Natural node i in the SE, for $1 \le i \le n$
- NEGN: Negative node i in the SE, for $1 \le i \le n$
- HPOSN: History of positive node i in the SE, for $1 \le i \le n$
- HNATN: History of natural node i in the SE, for $1 \le i \le n$
- HNEGN : History of negative node i in the SE, for $1 \le i \le n$
- $Pubij$: Public key of the node j in the network i , for 1 $j \le ni$ and $1 \le i \le n$;
- $Prvij$: Private key of the node j in the network i , for 1 $j \le ni$ and $1 \le i \le n$;
- Pki : Public key of network i , $1 \le i \le n$;
- MAC i : MAC address of node i , $1 \le i \le n$

Before defining our access mechanism, healthiness conditions for the above variables must be defined.

- $Pkij \neq Pkuv$ for $i \neq u$ or $j \neq v$
- $Pubi \neq Pubj$ for $i \neq j$
- $Prvi \neq Prvj$ for $i \neq j$
- $MAC\, i \neq MACj$ for $i \neq j$

After showing the healthiness of our variables, our access mechanism can be described by the following steps, where *Ti* denotes the i^{th} component of a tuple *T*:

1. Granting certificate and history authority duties to nodes:
   - $\forall i,j.\ 1 \leq j \leq ni \wedge 1 \leq i \leq n \wedge OSP\ ij{=}ti.$ (1)
2. Issuing digital operation certificates to local nodes of each network:
   - $\forall i,j.\ 1 \leq j \leq ni \wedge 1 \leq i \leq n \wedge DOCdij = <j, i, sdxij, edxij, OSPi, Pkij, MACij, CALij\ \ldots, Sign\ xij>$ (2)

   Where *OSPi* is the OSP of the node *j* in the network *i*; *sdxij* and *edxij* are the start and end date of the digital operation certificate; and the digital signature of the certificates *Sign xij* is calculated by the *OSPi* of the network *i* by performing threshold cryptography. *CALij* is the type of node based on the registration.
3. Recording history certificate to local nodes of each network:
   - $\forall i,j.\ 1 \leq j \leq ni \wedge 1 \leq i \leq n \wedge Hxij = <j, i, shxij, uhxij,$ OSPi, $Pkij, HPOSNij, HNATNij, HNEGNij, \ldots\ \ldots., CLLAij, Signxij>$ (3)

   Where OSP*i* is the OSP of the node *j* in the network *i*; *shxij* and *uhxij* are the start and update of the history file; *HPOSNij*, *HNATNij* and *HNEGNij* are the history file of the node and the OSP of the network *i* calculated from their events. *CLLA ij* is the kind of node that it will be after calculation.

   Each node uses its digital operation certificate and history certificate in order to access services in the SE through another node. A node needs to request from its OSP a new history certificate and operation certificate in order to perform in it.
4. A request for digital certificates from a node *j* of the network *t* to another node can be modelled by a message of the form:

   $$\langle \boldsymbol{j}, \boldsymbol{i}, \boldsymbol{W}, \boldsymbol{Z} \rangle \tag{4}$$

   for some digital operation certificate *W* and some history certificate *Z*.

   Such a request $\langle j, i, W, Z \rangle$ is checked as follows:

   a. The requester is the owner of the digital operation and history certificates, i.e.

   $$(\boldsymbol{W}^1 = \boldsymbol{j}) \wedge (\boldsymbol{Z}^1 = \boldsymbol{j});$$

   b. The OSP is the node where *W* and *Z* were issued, i.e.

   $$(\boldsymbol{W}^2 = \boldsymbol{i}) \wedge (\boldsymbol{Z}^2 = \boldsymbol{i});$$

   c. These certificates have not expired, i.e.

   $$(\boldsymbol{W}^3 \leq CD \leq \boldsymbol{W}^4) \wedge (\boldsymbol{Z}^3 \leq CD \leq \boldsymbol{Z}^4);$$

   Where CD denotes the current date;

   d. Certificates *W* and *Z* are authenticated using the public key Pk_t of the network *t* and a signature verification algorithm for threshold cryptography.

5. The requester node can access services in SE as follows:
   a. It has access through the OSP when the its history certificate is negative
      - *Hxij*=<*j*, *i*, *s*h*xij*, uh*xij*, *OSPi*, *Pkij*, *HPOSNij*, *HNATNij*, *HNEGNij*……. *HNEGNij*, *Signxij*> (9)

$$Node(i)access \rightarrow OSP\, if\; H(i) = negative$$

   b. It has access through NBBN when its history certificate is natural
      - *Hxij*=<*j*, *i*, *s*h*xij*, uh*xij*, *OSPi*, *Pkij*, *HPOSNij*, *HNATNij*, *HNEGNij*……. *HNATNij*, *Signxij*> (10)

$$Node(i)access \rightarrow NBBN\, if\; H(i) = natural$$

   c. It has access through any anther node when the its history certificate is positive
      - *Hxij*=<*j*, *i*, *s*h*xij*, uh*xij*, *OSPi*, *Pkij*, *HPOSNij*, *HNATNij*, *HNEGNij*……. *HPOSNij*, *Signxij*> (11)

$$Node(i)access \rightarrow Node(j)\, if\; H(i) = postive$$

8. Case study

Wireless *ad hoc* networks of networks (WANETs) are considered to be the future of wireless networks owing to their specific characteristics: practicality, simplicity, self-organization, self-configuration, ease of use and low cost when operating in a licence-free frequency band. There are many applications of *ad hoc* networks, ranging from small, static networks that are constrained by power sources, to large-scale, mobile, highly dynamic ones such as:

- In education, for students to interact with teachers during classes via laptops
- Healthcare and telecare systems
- Inter-vehicle communications; for example, sending instant traffic reports and other information between drivers
- Email and electronic file transfer
- Web services that can be used by *ad hoc* network users where a node in the network serves as a gateway to the outside world
- A wide range of military applications, such as a battlefield in unknown territory where an infrastructure network is not available or impossible to maintain
- Collaborative work for business environments
- Emergency search-and-rescue operations in disaster areas, where it is almost impossible to implement an infrastructure network
- Personal area networking and Bluetooth
- Electronic payments from anywhere (i.e. taxi)
- Home wireless networks and smart homes
- Office wireless networks.

In this section, we evaluate our secure environment system, concentrating on access to the SE. A military case study with two scenarios will be introduced. The first highlights our secure environment system, concentrating on our access control prevention technique for predefined armies in an unknown and unstable military environment; this scenario combines authentication, authorisation, confidentiality and integrity to provide privacy protection for elements and tactics. The second scenario illustrates an SE system in an unstable and unknown military environment, showing event detection techniques combined with policies to provide a secure military system against unknown elements.

8.1 Military environment

This military case study considers a battlefield in unknown territory, where infrastructure deployment is hard to achieve or maintain; therefore, SE will be the perfect solution. The military domain is a very challenging environment characterised by ambiguity and the need to be able to deal with significant and disruptive dynamic changes. The goals of military systems are mainly concerned with the ability to provide a secure environment for their components, because opponents (enemies) are always trying their best to break down or destroy our activities. Therefore, our secure environment concentrates on prevention and detection mechanisms.

8.2 Definition of components

In the military environment, a critical system and the specification of the security requirements for its components are essential. Registration, authentication and authorisation are among the most important requirements, but before defining and analysing them, we need to define our military system and its elements. We will be dealing with a military alliance consisting of different armies (e.g. NATO); each army will be defined as a WANET, while the whole alliance is defined as an SE. Each of the armies comprises different elements, from a soldier to the commander-in-chief. Usually in the military, there will be a specific hierarchy, in which each officer will have the authority to give orders or to communicate with different elements based on his/her rank.

- Each army is classified as a WANET
- WANETs merge to create a military alliance which is an SE
- NATO is defined as the OSP for all armies
- Soldiers in our SE will be defined as normal *ad hoc* nodes (negative, positive and nautral)
- Base stations, tanks, trucks and military aircraft are defined as NBBNs
- A set of policies is defined for each WANET (army).

8.3 Securing the military environment

Our military alliance will be a merger of different WANETs creating the SE, depicted in Figure 4.

The first step in providing a secure military system is to set up an operational process for the SE components; this is done by distributing operation certificates, which are initially granted by the SE. These certificates act as identity documents for each element of the military SE. As with the operation certificates, each OSP distribute certificates to enable specific nodes to carry out leading and agile operations.

Two scenarios are now considered for the military environment.

8.4 Scenario one

The first scenario assumes a military alliance of two armies (B and F), each of which has all kind of nodes (positive, negative and neutral). Before defining the SE to provide authorisation and authentication to other nodes, a few points must be clarified in our scenario:

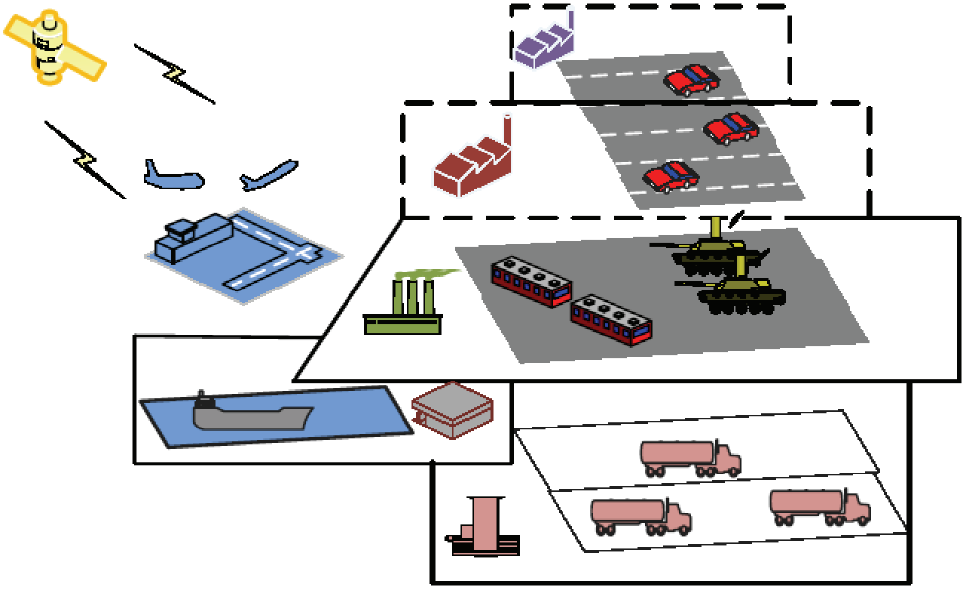


Fig. 4. Secure environment community

- Armies can join and disconnect without affecting the SE system.
- The public keys of the digital operation certificate are known between all elements in the whole SE.
- All nodes (soldiers) have received their operation certificates from their OSPs.
- The OSPs have sent the history of the nodes to the NBBNs.
- Each WANET has a set of policies.
- All nodes in the SE must be registered at the registration stage.

To provide a secure environment in this scenario, the first step is to ensure administration; as previously mentioned, the OSP and the NBBNs will carry out the administration. Their duties are to guarantee that elements from different armies can communicate and engage with different elements in the SE system and to provide all nodes with updates on the history of other nodes.

The second step is to provide or prevent access to the most essential components, which are needed in any community. Such prevention and access are needed for the authentication (by operation certificate) and authorisation (by node history status) of the SE elements. For instance, if a node (soldier) from the British army is trying to connect through the French army, this node will be authenticated (verified) by the NBBN of the French army by his operation certificate. Meanwhile, the granting of history status will be based on the policies (positive, negative and natural) of the node to access the SE through the French army.

The third step is the containment and recovery component. When a problem has occurred during any military operation, specific rules and procedures usually apply; for example, if members of a F platoon have been captured, the enemy will try its best to extract the private key in order to gain access to all secret information and to forge new certificates in order to break the system down. In this situation, the OSP of the SE will try to regenerate new shares of the private key, to make sure that it is kept safe during military operations. Moreover, the history file of this node, updated via links through heterogeneous cards available with NBBNs (e.g. satellites and cellular), will be used to receive orders from the main station OSP of the SE to which the NBBN belongs.

To elaborate on our secure system and to show the components providing authentication and authorisation between the military elements in the SE, the following specification formalism will be introduced:

X*ij*: soldiers *i* from army *j*; *i* ≥1; *j* = NATO countries;

Yij: tanks, trucks and military aircraft *i* from army *j*; *i* ≥1; *j* = NATO countries;

Zi: base station and cellular (OSP) *i* from SE; *i* ≥1.

DOCM: Documented node i in the SE, for 1≤ *i* ≤*n*

POSN: Positive node i in the SE, for 1≤ *i* ≤*n*

NATN: Natural node i in the SE, for 1≤ *i* ≤*n*

NEGN: Negative node i in the SE, for 1≤ *i* ≤*n*

HPOSN: History of positive node i in the SE, for 1≤ *i* ≤*n*

HNATN: History of natural node i in the SE, for 1≤ *i* ≤*n*

HNEGN: History of negative node i in the SE, for 1≤ *i* ≤*n*

Hi: History certificate of node in SE i, 1≤ *i* ≤*n;*

DOCxj: Digital Operation Certificate of node i in the SE, for 1≤ *i* ≤*n*

Pki: public key of network i, 1≤ *i* ≤*n;*

MAC i : MAC Address of node i, 1≤ *i* ≤*n*

8.4.1 Authentication and authorisation between elements of an army in the SE military system

- Authentication (*X1* B, *X2* B) between soldiers in the same army is based on the *X* operation certificate, where *X* is received from base station *Z*. The certificate will be verified using the B public key and the MAC address of *X*.

$$DOCxij =< j,i,sdxij,edxij,OSPi,Pkij,MACij,CALij...,Signxij >$$

$$X(B1 \vee B2) \Rightarrow \begin{cases} drop(req) \quad if \;\; DOC(X(B1 \vee B2)) \neq valid \\ \\ Accept(req) \;\; if \;\; DOC(X(B1 \vee B2)) = valid \end{cases}$$

- Authorisation (*X1* B, *X2* B) between soldiers in the same army is based on the history file of *X* and its status, where *X* is received from base station *Z* or *Y* and will be granted only for positive *X*.

$$Hxij =< j,i,\mathrm{sh}xij,\mathrm{uh}xij,OSPi,Pkij,HPOSNij,HNATNij,HNEGNij...CLLAij,Signxij >$$

$$X(B1 \vee B2) \Rightarrow \begin{cases} drop(req) \quad if \quad H(B1 \vee B2) \neq positive \\ \\ \;Accept(req) \wedge access \rightarrow X \; if \; HB(1 \vee 2) = positive \end{cases}$$

- Authentication (*Y1* B, *Y2* B) between tanks or trucks in the same army is based on the *Y* operation certificate, where *Y* is received from the base station *Z*. The certificate will be verified using the B public key and MAC address of *Y*.

$$DOCxij =< j,i,sdxij,edxij,OSPi,Pkij,MACij,CALij...,Signxij >$$

$$YB(1 \vee 2) \Rightarrow \begin{cases} drop(req) \quad if \quad DOC(YB(1 \vee 2)) \neq valid \\ \\ Accept(req) \ \ if \quad DOC(YB(1 \vee 2)) = valid \end{cases}$$

- Authorisation (*Y1* B, *Y2* B) between tanks or trucks in the same army is based on the history file of *Y* and its status, where *Y* is received from base station *Z* and will be granted only to positive and natural *Y*.

$$\mathrm{H}xij = j,i,\mathrm{sh}xij,\mathrm{uh}xij,OSPi,Pkij,HPOSNij,HNATNij,HNEGNij...CLLAij,Signxij >$$

$$YB(1 \vee 2) \Rightarrow \begin{cases} drop(req) \quad if \quad H(1 \vee 2) = negative \\ \\ Accept(req\ ) \wedge access \rightarrow Y \ if \ H(1 \vee 2) \neq negative \end{cases}$$

- Authentication (*Y1* B, *X2* B) between tanks or trucks and soldiers from the same army is based on *Y* and *X* operation certificates, where *Y* and *X* are received from base station *Z*. The certificates will be verified using the B public key and MAC addresses of *Y* and *X*.

$$DOCxij =< j,i,sdxij,edxij,OSPi,Pkij,MACij,CALij...,Sign\ xij$$

$$XB \vee YB \Rightarrow \begin{cases} drop(req) \quad if \ \ DOC(X \vee Y) \neq valid \\ \\ Accept(req) \ \ if \ \ DOC(X \vee Y) = valid \end{cases}$$

- Authorisation (*Y1* B, *X2* B) between tanks or trucks and soldiers in the same army is based on the history files of *Y* and *X* and their status, where *Y* is received from base station *Z*, and will be granted only to positive and natural *Y*s or *X*s.

$$\mathrm{H}xij =< j,i,\mathrm{sh}xij,\mathrm{uh}xij,OSPi,Pkij,HPOSNij,HNATNij,HNGENij...CLLAij,Signxij$$

$$XB \vee YB \Rightarrow \begin{cases} drop(req) \quad if \ \ H(X \vee Y) \equiv negative \\ \\ Accept(req) \wedge access \rightarrow X \vee Y \ \ if \ \ H(X \vee Y) \neq negative \end{cases}$$

8.4.2 Authentication between elements from different armies in the SE military system

- Authentication (*X1* B, *X1* F). If a soldier from the B army wants to authenticate a F soldier, this will be done using the operation certificate, which will be verified using the F public key and the MAC address of *X*.

$$DOCxij =< j,i,sdxij,edxij,OSPi,Pkij,MACij,CALij...,Sgnxij >$$

$$XF \Rightarrow \begin{cases} drop(req) \quad if \quad DOC(XB) \neq valid \\ \\ Accept(req\ ) \ \ if \ \ DOC(XB) = valid \end{cases}$$

- Authorisation (*X1* B, *X1* F). If a B soldier tries to communicate with a F soldier, then the latter will need to check the history which he receives from the OSP or a F NBBN. If the B *X* is a positive node it will be allowed to communicate directly with the F soldier, while if it is a natural node it will be allowed to do so through a F NBBN.

$$\mathrm{H}xij =< j,i,\mathrm{sh}xij,\mathrm{uh}xij,OSPi,Pkij,HPOSNij,HNATNij,HNEGNij...CLLAij,Signxij >$$

$$XF \Rightarrow \begin{cases} drop(req) \quad if \quad H(XB) = negative \\ Accept(req) \wedge access \rightarrow NBBN \ if \ H(XB) = natural \\ Accept(req) \wedge access \rightarrow XF if \ H(XB) = positive \end{cases}$$

- Authentication (*Y1* B, *Y1* F). If tanks or trucks from the B army want to authenticate a F tank, this will be done using the operation certificate, which will be verified using the F public key and the MAC address of *Y*.

$$DOCxij =< j,i,sdxij,edxij,OSPi,Pkij,MACij,CALij...,Signxij >$$

$$Y(F) \Rightarrow \begin{cases} drop(req) \quad if \quad DOC(YB) \neq valid \\ \\ Accept(req) \ \ if \ \ DOC(YB) = valid \end{cases}$$

- Authorisation (*Y1* B, *Y1* F). If a B tank tries to communicate with a F one, the history of the B *Y* is required and can be received from the OSP. If the B *Y* is a positive or natural node it is allowed to communicate with F tanks, whereas if it is a negative node it can do so only through the OSP of the SE.

$$\mathrm{H}xij =< j,i,\mathrm{sh}xij,\mathrm{uh}xij,OSPi,Pkij,HPOSNij,HNATNij,HNEGNij...CLLAij,Signxij$$

$$Y(F) \Rightarrow \begin{cases} Accept(req) \wedge access \rightarrow OSP \ if \ H(YF) = negative \\ \\ Accept(req) \wedge access \rightarrow NBBN \, if \ H(YB) = postive \vee natural \end{cases}$$

- Authentication (*Y1* B, *X1* F). If a B tank wants to authenticate a F soldier, this will be done using the operation certificate, which will be verified using the F public key and the MAC address of *Y*.

$$DOCxij =< j,i,sdxij,edxij,OSPi,Pkij,MACij,CALij...,Signxij >$$

$$X(F) \Rightarrow \begin{cases} drop(req) \quad if \quad DOC(YB) \neq valid \\ \\ Accept(req\ ) \ \ if \ \ DOC(YB) = valid \end{cases}$$

- Authorisation (*Y1* B, *X1* F). If a British tank tries to communicate with a F soldier, a history of the B *Y* is required and can be received from the OSP. If the B *Y* is a positive or natural node it is allowed to communicate with F tanks, whereas if it is a negative node it can do so only through the OSP of the SE.

$$\mathrm{H}xij =< j,i,\mathrm{sh}xij,\mathrm{uh}xij,OSPi,Pkij,HPOSNij,HNATNij,HNEGNij,...CLLAij,Signxij >$$

$$X(F) \Rightarrow \begin{cases} Accept(req) \wedge access \rightarrow OSP \ if \ H(YB) = negative \\ \\ Accept(req) \wedge access \rightarrow NBBN \ if \ H(YB) = postive \vee natural \end{cases}$$

8.5 Scenario two

As with the first scenario, scenario two assumes a military alliance consisting of two armies (B and F). In addition, new elements will be defined in this scenario (J army). Before stating how the SE provides authorisation and authentication to other nodes, some points must be clarified:

- Armies can join and disconnect without affecting the SE system.
- The public keys of the digital certificates are known by B and F elements in the whole SE system and unknown to the Japanese army.
- All nodes (soldiers and tanks) have received their operation certificates from their own OSPs (each army has its own certificate).
- Any node that is undefined and trying to operate in a different network will receive an operation certificate from its OSP and its history file will be new.
- The OSP sends node histories NBBNs.
- Each WANET has a set of policies.

To illustrate the working of the SE system and to show how the authentication and authorisation components operate between the military elements, the following example is introduced. If during a war the B army needs reinforcements from a non-NATO country such as J, in order for the J army to communicate with B forces and to engage into the battlefield, J soldiers and tanks will need to obtain an operation certificate from the OSP to perform in such situations. As J forces are non-trusted, our OSP will monitor and observe their actions based on their history in order to check whether or not J elements are acting in a normal or malicious manner. This checking is accomplished by tracing their behaviour. Usually, showing all aspects of the tracing of behaviour under the set of policies in our scenario is impossible; therefore, we provide examples showing normal, malicious and positive actions.

The following specific formalism is introduced:

Xij: soldiers *i* from army *j*; *i* ≥1; *j* = NATO countries;
Yij: tanks, trucks and military aircraft *i* from army *j*; *i* ≥1; *j* = NATO countries;
Z: base station and cellular (OSP) i from SE *i* ≥1;
Wik: soldiers *i* from army *k*; *i* ≥1; *k* = non-NATO country;
Mik: tanks, trucks and military aircraft *i* from army *k*; *i* ≥1; *k* = non-NATO country.

In the first instance, the J will deal with the OSP. Four examples, all set in wartime, are given below to show how the OSP observes the behaviour of new nodes and the new army to build a history for every node as a basis for granting authorisation.

In the first example, if an order from an OSP has been given to Japanese troops and the soldiers obey this order, the OSP will observe these acts and decide whether or not they are normal; it will still classify the nodes as new and follow these rules:

- Authentication (*W1* J, *X1* F). If a soldier from the J army wants to authenticate a F soldier in the SE, this will be done using the operation certificate, which will be verified using the OSP's public key and the MAC address of *W*.

$$DOCxij =< j,i,sdxij,edxij,OSPi,Pkij,MACij,CALij...,Signxij >$$

$$X(F) \Rightarrow \begin{cases} drop(req) \quad if \quad DOC(WJ) \neq valid \\ \\ Accept(req) \ \ if \ \ DOC(WJ) = valid \end{cases}$$

- Authorisation (*W1* J, *X1* F). If a Japanese soldier tries to communicate with a F one, the latter will need to check his history, which he obtains from the OSP or a F NBBN; if the J *W* is a new node from a new army it will be allowed to communicate with the F soldier through the OSP.

$$\mathrm{H}xij =< j,i,\mathrm{sh}xij,\mathrm{uh}xij,OSPi,Pkij,HPOSNij,HNATNij,HNEGNij...CLLAij,Signxij >$$

$$XF \Rightarrow \begin{cases} drop(req) \quad if \quad H(WJ) \neq DOCM \\ \\ Accept(req) \wedge access \rightarrow OSP\,if\ H(WJ) = DOCM \end{cases}$$

In the second example, if an order from an OSP has been given to J troops and the soldiers obey it, the OSP will observe these acts and decide whether or not they are normal. If the node continues to obey every order the OSP will classify is as neutral and follows these rules:

- Authentication (*W1* J, *X1* F). If a J soldier wants to authenticate a F one in the SE, this will be done using the operation certificate, which will be verified using the OSP's public key and the MAC address of *W1*.

$$DOCxij =< j,i,sdxij,edxij,OSPi,Pkij,MACij,CALij...,Signxij >$$

$$X(F) \Rightarrow \begin{cases} drop(req) \quad if \quad DOC(WJ) \neq valid \\ \\ Accept(req) \;\; if \;\; DOC(WJ) = valid \end{cases}$$

- Authorisation (*W1* J, *X1* F). If a Japanese soldier tries to communicate with a F one, the latter will need to check his history, which he obtains from the OSP or a F NBBN; if the J *W* is a natural node it is allowed to communicate with the F soldier through a F NBBN.

$$\mathrm{H}xij =< j,i,\mathrm{sh}xij,\mathrm{uh}xij,OSPi,Pkij,HPOSNij,HNATNij,HNEGNij...CLLAij,Signxij >$$

$$XF \Rightarrow \begin{cases} drop(req) \quad if \quad H(WJ) \neq DOCM \\ Accept(req) \wedge access \rightarrow OSP\, if\; H(WJ) = DOCM \\ Accept(req) \wedge access \rightarrow NBBN\, if\; H(WJ) = natural \end{cases}$$

In the third example, if an order and notification from an OSP has been given to J troops and the soldiers obey this order and forward the notification to their neighbours, then the OSP will observe these acts and decide that they are positive. If the node continues to obey all orders and forward all notifications, the OSP will classify it as a positive node and follow these rules:

- Authentication (*W1* J, *X1* F). If a Japanese soldier wants to authenticate a F soldier in the SE, this will be done using the operation certificate, which will be verified using the OSP's public key and the MAC address of *W1*.

$$DOCxij =< j,i,sdxij,edxij,OSPi,Pkij,MACij,CALij...,Signxij >$$

$$X(F) \Rightarrow \begin{cases} drop(req) \quad if \quad DOC(WJ) \neq valid \\ \\ Accept(req) \;\; if \;\; DOC(WJ) = valid \end{cases}$$

- Authorisation (*W1*, *X1* F). If a Japanese soldier tries to communicate with a F one, the latter will need to check his history, which he obtains from the OSP or a F NBBN. If the J *W* is a positive node it is allowed to communicate directly with the F soldier.

$$\mathrm{H}xij =< j,i,\mathrm{sh}xij,\mathrm{uh}xij,OSPi,Pkij,HPOSNij,HNATNij,HNEGNij...CLLAij,Signxij >$$

$$XF \Rightarrow \begin{cases} drop(req) \quad if \quad H(WJ) \neq DOCM \\ Accept(req) \wedge access \rightarrow NBBN\, if\; H(WJ) = natural \\ Accept(req) \wedge access \rightarrow XF\, if\; H(WJ) = positive \end{cases}$$

In the fourth example, a J soldier tries to request a specific tactic from the F army in the SE using an invalid or fake operation certificate. The OSP will observe this act, decide that it is negative and send an update for the history file to all nodes in the SE. If the node continues to behave in this way, the OSP will classify it as a negative node and adopt the following rules:

- Authentication (*W1* J, *X1* F). If a Japanese soldier wants to authenticate a F one in the SE, this will be done using the operation certificate, which will be verified using the OSP's public key and the MAC address of *W1*.

$$DOCxij =< j,i,sdxij,edxij,OSPi,Pkij,MACij,CAL...,Signxij >$$

$$X(F) \Rightarrow \begin{cases} drop(req) \quad if \quad DOC(WJ) \neq valid \\ \\ Accept(req) \quad if \quad DOC(WJ) = valid \end{cases}$$

- Authorisation (*W1* J, *X1* F). If a Japanese soldier tries to communicate with a F one, the latter will need to check his history, which he obtains from the OSP or a F NBBN. If the J *W* is a negative node it will not be allowed to communicate directly with the F soldier.

$$\mathrm{H}xij =< j,i,\mathrm{sh}xij,\mathrm{uh}xij,OSPi,Pkij,HPOSNij,HNATNij,HNEGNij...CLLAij,Signxij >$$

$$XF \Rightarrow \begin{cases} drop(req) \quad if \quad H(WJ) = negative \\ \\ Accept(req\ ) \wedge access \rightarrow NBBN\, if\ H(WJ) \neq negative \end{cases}$$

9. Conclusion

This chapter has proposed ways to control access to a secure *ad hoc* database environment based on the history of its nodes. It also proposes an access algorithm which explains the steps taken by a node while handling requests to access a secure environment.

We have provided a case study, with specific concentration on two military scenarios in unknown and insecure territory. Scenario one assumed two NATO countries (pre-connected) in a battlefield and showed the implementation and evaluation of access to a secure environment providing authentication and authorisation between members of the same network and between other members. Scenario two considered two NATO countries with new elements (non-defined), showing the implementation and evaluation of our mechanism for allowing and preventing access to the secure environment. It detailed the access technique between NATO countries and the undefined elements, presenting a number of different situations that any military system might experience. In each situation our technique was examined to establish whether or not the situation was adequately addressed by our set of policies in order to prevent malicious acts by undefined elements in a secure military environment.

The solution is a combination of the history of the nodes and operation certificates. Each node in a secure environment is uniquely identified by its public key and MAC address. The solution addresses various vulnerability issues affecting wireless links such as active and passive attacks. The dynamic nature of networks and their membership does not affect the solution, since each node makes access decisions on its own and the use of cooperative algorithms is avoided.

10. References

Toh, C.-K. (2002). "Ad Hoc Mobile Wireless Networks: Protocols and Systems", pp: 34-37, Prentice- Hall, New Jersey,

Siva Ram Murthy, C.; and Manjo, B.S. (2004). "Ad Hoc Wireless Networks: Architectures and Protocols", Prentice Hall communications engineering and emerging technologies series Upper Saddle River,

Singh, S.; Raghavendra, C.S. (1998)."Power efficient MAC protocol for multihop radio networks", pp:153 – 157, Personal, Indoor and Mobile Radio Communications, The Ninth IEEE International Symposium .

Perkins, C.E.; Royer, E.M. (1999) "Ad-hoc on-demand distance vector routing Mobile Computing Systems and Applications", pp: 90 – 100, Proceedings. WMCSA'99. Second IEEE Workshop .

Chiang, C.; Gerla, M., Zhang, L. (1998)."Adaptive shared tree multicast in mobile wireless networks", pp:1817 – 1822, Global Telecommunications Conference, GLOBECOM 98. The Bridge to Global Integration. IEEE.

Toh, C.-K. (2001). "Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks", Volume: 39, Issue: 6, pp:138 – 147, Communications Magazine, IEEE.

Holland, G.; Vaidya, N. (1999) "Impact of routing and link layers on TCP performance in mobile ad hoc networks", pp:1323 – 1327, Wireless Communications and Networking Conference, WCNC. 1999 IEEE.

Dahill, B. ; Neil Levine, ;B. Royer E.; Shields, C. (2001). "A Secure Routing Protocol for Ad Hoc Networks", Technical report UM-CS-2001-037, University of Massachusetts, Amherst.

Hu, Y.; Perrig A. .; Jonson, D.B. (2002)."Ariadne: A Secure On-Demand Routing for Ad hoc Networks", pp:12-23, Proceedings of ACM MOBICOM 2002.

25

Trust Establishment in Mobile Ad Hoc Networks: Direct Trust Distribution-Performance and Simulation

Dawoud D.S.[1], Richard L. Gordon[2],
Ashraph Suliman[1] and Kasmir Raja S.V.[3]
[1]National University of Rwanda
[2]University of KwaZulu Natal
[3]SRM University, Chennai,
[1]Rwanda
[2]South Africa
[3]India

1. Introduction

In the previous chapter, we discussed the distinct characteristics of ad hoc networks, which make them very difficult to secure. Such characteristics include: the lack of network infrastructure; no pre-existing relationships; unreliable multi-hop communication channels; resource limitation; and node mobility. We provided a theoretical background to mobile ad hoc networks and the security issues that are related to such networks. We defined the ad hoc networks and their characteristics in terms of trust establishment. As the focus of the two chapters is on the network layer, attacks specific to this layer are identified and explained in Chapter 1. We also presented a survey of the existing key management solutions for mobile ad hoc networks.

The current chapter is a continuation for the previous one. This is why we start this chapter by Section-2 that offers a survey of the existing secure routing protocols for mobile ad hoc networks. This section makes a pertinent observation that most secure routing protocols assume some kind of key management authority exists. Mobile ad hoc networks have little fixed network architecture and it is unlikely that there is a centralised authority member. Section-2 of this chapter together with last section of the previous one identify the problem that the two chapters together are addressing. There exists secure routing mechanisms to address the unique characteristics of mobile ad hoc networks, however, these solutions assume that key management is addressed prior to network establishment. A novel, on-demand solution to the key management problem for mobile ad hoc networks is then described. Section-3 details the functionality and operation of the proposed model: "Direct Indirect Trust Distribution" (DITD). The DITD model focuses on the task of distributing keying information. The DITD model also includes a verification optimization protocol and trust evaluation metric, which maximises the security of distribution.

The implementation and simulation of the DITD model is examined in Section-4. There are various packages used to compare existing and proposed routing protocols. One such package is the ns2 Network Simulator, which is commonly used in the relating literature. A comparative ns2 simulation study between the DITD and the AODV protocols is presented. The DITD model is based on the Ad Hoc On-demand Distance Vector (AODV) routing protocol. Simulations show the performance overhead of including key management functionality and the performance of DITD in the presence of malicious attacking nodes.

Section-5 summarises the contribution of the Chapter to the field of trust establishment in mobile ad hoc networks. Section-5 also provides future direction for research.

2. Secure routing in mobile ad hoc networks

A mobile ad hoc network's routing protocol has unique challenges due to the dynamic nature of ad hoc network. Mobile ad hoc networks do not have the same privileges that fixed, wired networks have. The routing mechanisms are uniquely designed to deal with the lack of infrastructure and unreliable wireless multi-hop communication channels.

This section investigates the procedure of securing of these routing protocols. The routing solutions are briefly visited and an extensive survey is presented for the existing security mechanisms that are used to secure these routing protocols.

Routing in mobile ad hoc networks is divided into two categories: table driven methods and on-demand methods. Table driven methods are also known as proactive routing. They maintain routing tables that contain routes to all the nodes in the network. Theses tables are periodically updated which allows routing information to be available at all times. Examples of table methods include Destination Sequence Distance Vector Routing (DSDV) [Perkins & Bhagwat, 1994] and Optimized Link State Routing (OLSR). Source initiated on-demand routing methods establishes routes in a reactive manner. Routes are established through a route discovery phase. During a route discovery phase node *S* will broadcast a request message *RREQ* into the network. This request message is forwarded until it reaches its target destination node *D*. Node *D* then replies with a reply message *RREP* which is unicast along the reverse route, until it reaches the source and the route is established. Routes are maintained as long as they are required. Examples of on-demand methods include Ad Hoc On-Demand Distance Vector (AODV) [Perkins et al, 2003] and Dynamic Source Routing protocol (DSR) [Johnson et al, 2001]. The reactive on-demand approach is less computationally expensive, in comparison with the proactive table driven approach.

In the previous chapter, it is identified that most security attacks target the network layer, and more specifically the routing protocol. These attacks include: black-hole attacks; wormhole attacks; eavesdropping attacks; byzantine attacks; resource consumption attacks; and routing table poisoning. The routing protocol is found on the network layer and is a significant service for mobile ad hoc network. Adversaries, specifically, target the routing protocol. Thus, a secure routing solution is needed for ad hoc networks to be securely implemented.

This section gives a survey and an analysis of the existing secure routing protocols. Each protocol is presented and investigated based on: functionality; operational assumptions; and security. A summary and discussion is formulated at the close of this section.

2.1 Secure Efficient Ad hoc Distance vector routing protocol (SEAD)

Secure efficient ad hoc distance vector (SEAD) [Hu et al, 2002] is a secure routing protocol which is used in conjunction with the table driven destination-sequenced distance vector (DSDV) routing protocol [Perkins & Bhagwat, 1994]. The DSDV routing protocol uses a distributed version of the Bellman-Ford algorithm to discover the shortest path between two nodes. The SEAD protocol uses symmetric key cryptography and one-way hash functions to protect against security attacks like denial of service and resource attacks.

a. System Overview

The DSDV routing protocol discovers the shortest path based on a route's hop count. Routing packets are assigned sequence numbers to ensure the most recent route is processed. The hop count and sequence number variables are stored in the routing packets. Attackers can create an incorrect routing state in nodes resulting in a denial of service attack (DoS) where the attacker attempts to make other nodes consume excess bandwidth and processing time. SEAD makes the routing process robust against multiple uncoordinated attackers by authenticating the hop count and sequence number of routing packets with a one-way hash function h. Hash chaining is used so that only nodes that are in possession of the previous routing update (identified by a sequence number) can broadcast a new routing update. Authenticated routing updates are computed to prevent against malicious routing updates broadcast by attackers.

b. One-Way Hash Function

SEAD uses a one-way hash function to authenticate routing updates and minimize resource consumption attacks. A formal definition of the hash function H is provided in [Stalling, 2003]. The most commonly used hash functions are MD-5 [Rivest, 1992] and SHA-1 [Publications F IPS, 2008].

A one-way hash function H is used to generate a one-way hash chain h. The one-way hash function H has an input of any bit length * and outputs a variable of fixed bit length p. The one-way hash function H must be computationally impossible to invert.

$$H: (0,1)^{*} \rightarrow (0,1)^{p}$$

A hash chain h_i is created when a node selects a random number $x \in (0,1)^{p}$ and uses it to generate a list of variable which make up a hash chain $h_0, h_1, h_2, h_3, \ldots, h_n$. Here $h_0 = x$ and h_i is calculated using the irreversible one-way hash function H such that:

$$h_i = H(h_{i-1}) \text{ where } 0 \leq i \leq n$$

Assuming there is an existing authenticated element, a node can verify elements later in the chain's sequence. For example if an authenticated element h_i exists, a node can authenticate h_{i-4} by checking that $h_i = H(H(H(H(h_{i-4}))))$. SEAD assumes the existence of an authentication and key distribution mechanism to distribute an authenticated element like h_n allowing for authentication by hash chaining [Hu et al, 2002].

c. Authenticating routing updates

SEAD uses the elements of the hash chain to provide authentication and secure the routing updates in DSDV. SEAD assumes an upper bound on the variable to be authenticated, for example if it were the hop count then SEAD would assume a maximum route distance n in the network (the maximum hop count between two nodes allowed). This also eliminates any routes with a length greater than m to exist, eliminating possible routing loops or the routing infinite problem.

The sequence values that make up the hash chain are calculated from the H function such that $h_1, h_2, \ldots, h_n$ where n is divisible by m. For a routing table entry with sequence number i

let $k = {}^{n}/_{m} - i$. An element from h_{km}, h_{km+1}, …, h_{km+m-1} is used to authenticate the routing entry with sequence number i. If the hop count is j where $0 \leq j < m$, then h_{km+j} is used to authenticate the routing entry found with sequence number i and hop count j [Hu et al, 2002].

Routing updates are sent with the appropriate routing information and a hash chain value is used to authenticate the update. If the authentication value appended is h_{km+j} then only attackers with h_{km+j-1} can modify the authentication value. Nodes receiving a routing update, check the authentication value h_{km+j} by calculating the new hash chain value. Receiving nodes can calculate the new hash chain value by using the earlier hash chain value h_{km+j-1} and the received sequence number i and hop count j. If the new calculated hash value is equal to h_{km+j} then the routing update is verified.

SEAD proposes two methods for routing update authentication. One method uses clock synchronization and a broadcast authentication mechanism like TESLA [Perrig et al, 2001]. The second method requires a shared secret between each communicating node pair. The secret can be used to implement a message authentication code (MAC) between nodes authenticating routing update messages.

d. Analysis

The SEAD protocol protects the ad hoc network from routing attacks that target resource consumption. The SEAD protocol does protect against multiple uncooperative attacks, preventing routing loops but routing loop prevention cannot be guaranteed in the presence of co-operating attackers. The SEAD protocol is vulnerable to intelligent attackers that use the same sequence number and same hop count of the most recent update to corrupt routing information. The SEAD protocol provides protection against denial of service attacks [Perrig et al, 2001], replay attacks and routing table poisoning by authenticating routing updates so malicious nodes cannot corrupt the routing procedure.

2.2 A secure on-demand routing protocol for ad hoc networks (Ariadne)

Ariadne [Hu et al, 2005] is a secure on-demand routing protocol which uses symmetric cryptography. Ariadne is based on the on-demand DSR [Johnson et al, 2001] routing protocol and is developed by the same authors as the SEAD protocol [Hu et al, 2002]. Ariadne provides end-to-end authentication on the routing layer.

a. System Overview

Ariadne assumes a shared secret key between communicating node pairs and uses message authentication code (MAC) to authenticate end-to-end packets between the communication pair. Broadcast authentication is employed, with loose time synchronization, to authenticate route request and other broadcast packets. The TESLA [Perrig et al, 2001] broadcast authentication scheme is used. In TESLA the source generates a one-way key chain and a schedule is made which defines at which time keys of the chain are revealed. This mechanism limits Ardiadne's operation to ad hoc networks which have time synchronization. Ardiane provides end-to-end authentication in an on-demand manner over the DSR routing protocol [Hu et al, 2005].

b. End-to-end Authentication

For communication from a source node S to a destination node D, the source S will broadcast a route request into the network and expect a reply from D. Ariadne assumes a shared secret between *S* and *D*, K_{SD} and K_{DS}, which enables message authentication for each respective direction.

Nodes *S* wanting to start a route discovery for node *D* will first generate an initial hash chain h_0 consisting of: a packet identifier identifying the type of packet (a request packet *RREQ* in this case); the source's address (ID_S); the destinations address (ID_D); a broadcast identity (*bi*) identifying the current route discovery; and a TESLA time interval (*tes*) identifying the expected time of arrival at the destination.

$$h_0 = MAC_{K_{SD}}(RREQ|ID_S|ID_D|bi|tes)$$

Node *S* will broadcast a route request packet which includes: a packet identifier, the hash chain h_0; the source's address (ID_S); the destinations address (ID_D); the broadcast identity (*bi*); the TESLA time interval (*tes*); a node list *N()* and a MAC list *M()*. The packet broadcast is as follows:

$$S \rightarrow broadcast \ : \ RREQ|h_0|ID_S|ID_D|bi|tes|N()|M()$$

A neighbouring node that receives the route request checks the validity of the TESLA time interval, *tes*. The TESLA time interval is valid if the corresponding key that it points to has not been revealed yet and the time interval does not point too far in the future. The neighbouring node *A* will then compute a new hash chain h_1 using the previous hash chain h_0. A message authentication code of the packet to be broadcast is created (MAC_A). MAC_A is calculated using the TESLA key (K_{Ates}). Before forwarding the packet the neighbour node *A* includes: the hash chain h_1; itself in the node list *N*; and the MAC_A calculated in the MAC list *M*. The hash function and broadcast packet are as follows:

$$h_1 = H(A|h_0)$$

$$A \rightarrow broadcast \ : \ RREQ|h_1|ID_S|ID_D|bi|tes|N(A)|M(MAC_A)$$

Intermediate node *P* receiving a forwarded route request first calculates a new message authentication code MAC_P and a new hash chain $h_i = H(P-1|h_{i-1})$ where *P*-1 is the previous node and h_{i-1} is the previous hash chain value. Secondly it includes this information and forwards the route request as follows:

$$P \rightarrow broadcast \ : \ RREQ|h_i|ID_S|ID_D|bi|tes|N(A,\ldots,P)|M(MAC_A,\ldots,MAC_P)$$

The route request is propagated to the destination node *D*. When *D* receives the route request it validates the authenticity of the route request by checking that the TESLA time intervals indicate no keys have been released as of yet and that the hash chain is valid. *D* then generates a message authentication code MAC_D. $\mathrm{MAC_D}$ and an empty key list *K()* are included in the packet and sent back along the reverse path indicated by the node list and DSR protocol. The $\mathrm{MAC_D}$ and reply message are as follows:

$$MAC_D = MAC_{K_{DS}}(RREP|ID_D|ID_S|bi|tes|N(\ldots)|, M(\ldots)$$

$$D \rightarrow P \ : \ RREP|ID_D|ID_S|bi|tes|N(\ldots)|M(\ldots)|MAC_D|K()$$

Intermediate node that receive a reply message will wait for the *tes* time interval to lapse so the corresponding key can be revealed an included in the key list *K()*. The reply message is forwarded until it contains all the TESLA keys of the intermediate nodes and it finally reaches the source node *S*. The source then verifies the validity of all the keys, MAC_D, and the message authentication code contains.

c. Maintenance

Ariadne achieves secure route maintenance by authenticating the DSR error messages. Ariadne authenticates error messages preventing malicious nodes from broadcasting false broken links and causing denial of service type attacks. When an error message is generated TESLA authentication information is included. If authentication is delayed as a result of the TESLA time intervals, the intermediate nodes buffer the error message until the appropriate keys are revealed and the message can be authenticated and action taken [Hu et al, 2005].

d. Analysis

The authors of Ariadne are the same authors of SEAD [Hu et al, 2002] protocol. Ariadne employs an end-to-end approach to authentication while SEAD uses a hop-by-hop approach because of the DSDV routing procedure. The Ariadne proposal is based on the on-demand DSR routing protocol. Ariadne implements TESLA broadcast authentication and message authentication code to provide authentication for routing packets in an ad hoc network environment. The Ariadne proposal assumes that there exists some shared secret between a communication pair, therefore assuming the existence of an authentication and key distribution mechanism. Ariadne relies on TESLA authentication which requires time synchronization in the ad hoc network, synchronization is difficult to achieve without the presence of an outside authorized member or TTP.

Ariadne implements end-to-end authentication to prevent unauthorized nodes from sending error messages and incorrect routing packets in the form of repays attacks. However this proposal does not consider the case where attackers do not cooperate with the routing protocol and drop routing packets which are suppose to be forwarded. An extension is proposed in [Hu et al, 2003] which uses packet leashing to solve this problem.

2.3 Authenticated Routing for Ad hoc Networks

The authenticated routing for ad hoc networks (ARAN) protocol [Sanzgiri et al, 2002] is a securing routing solution which uses cryptographic certificates. ARAN is designed for an on-demand ad hoc routing protocol and achieves authentication, integrity and non-repudiation on the network layer but assumes prior shared secrets at initialization.

a. System Overview

The ARAN secure routing protocol establishes trust in three stages:

1. Issuing of certificates
2. Route Discover process
3. Shortest path Optimization

Initially ARAN assumes the presence of a trusted third party (TTP) which issues valid certificates, and a shared public key for all participating nodes. The route discovery process of ARAN provides end-to-end authentication for communicating nodes. The source node broadcasts a route request which carries the source's certificate. The route request is propagated to the destination node by an end-to-end authentication process. The destination node responds by unicasting a reply message back along the found route using the end–to-end authentication protocol.

b. Issuing of Certificates

This section describes how the certificates are issued and distributed to the participating nodes. The assumption is made that an authenticated trusted third party (TTP) member exists which plays the roles of an initial certificate authority (CA). This TTP CA is known to all the nodes in the network. The ARAN protocol assumes that certificates are generated by the TTP CA and distributed to nodes before they officially join the wireless ad hoc network. No specific key distribution mechanism is described for the ARAN protocol. Node *i* entering the network will receive a certificate *cert*$_i$ from the TTP CA that has the following contents:

$$TTP - CA \rightarrow i \ \ : \ \ cert_i = E_{k_{TTP-CA}}(ID_i|K_i|t|et)$$

The certificate *cert*$_i$ is signed by the private key of the TTP-CA ($k_{CA\text{-}TTP}$) and has the following contents: ID_i representing the identification of node *i* for example a specific IP address; K_i the public key of node *i*; *t* the timestamp for the *cert*$_i$; and *et* the expiry time of the certificate.

c. Route Discovery Process

The route discovery process provides end-to-end authentication which ensures that the packets sent from a source node *S* reach their intended destination node *D*. Each node maintains a routing table which contains the active communication routes between the different source and destination pairs. The route discovery process begins by a source *S* broadcasting a route request. The route request is signed by the source node's private key k_S and contains: the certificate of the source node (*cert*$_S$); the identification of the destination node (ID_D); a nonce (N_S); a timestamp (*t*); and a packet identifier identifying that the packet is a route request packet (*RREQ*). The authenticated route request broadcast by node *S* is:

$$S \rightarrow broadcast \ \ : \ \ E_{k_S}(cert_S|ID_D|N_S|t|RREQ)$$

The nonce value is incremented every time the source sends a route request. The nonce value acts like a sequence number ensuring the most recent route request is dealt with. Each node that receives the route request will process it if it has a higher value of the source's nonce than previously received route requests from the same source node. Each intermediate node *P* receiving the route request will validate the signature with the certificate, update the routing table with the neighbour from whom it received the route request, sign the route request and broadcast it to its neighbours. Node *P* will remove the signature and certificate of the previous node if the previous node was not the source itself. Therefore each forwarded route request is authenticated by the source and the intermediated node and will contain two certificates *cert*$_S$ and *cert*$_P$:

$$P \rightarrow broadcast \ \ : \ \ cert_P|E_{k_P}(E_{k_S}(cert_S|ID_D|N_S|t|RREQ)$$

The route request is propagated to the destination node *D* which will reply with a reply message *RREP*. The reply packet is signed by the destination node's private key k_D and the packet contains: the identity of the source node (ID$_S$); the destination's certificate (*cert*$_D$); a nonce of validity (N_D); a timestamp (*t*); and a packet identifier (*RREP*). The reply packet is unicast along the reverse path toward the source node with a similar authentication procedure to the forwarding of the route request.

$$D \rightarrow reverse\ path \ \ : \ \ E_{k_D}(cert_D|ID_S|N_D|t|RREP)$$

$$P \rightarrow reverse\ path\ \ :\ \ cert_P|E_{k_P}(E_{k_D}(cert_D|ID_S|N_D|t|RREP)$$

The source node will receive the reply packet *RREP* and check the signature and nonce (N_D) to verify that the packet was sent by the destination node and not a malicious attacker. If the nonce or certificate fails an error message is broadcast and the route request process restarted.

d. Shortest Path Confirmation

This is an optional procedure employed by ARAN to ensure that the shortest path is found between source and destination. Path confirmation has a high computational cost. After a route has been found between *S* and *D* the shortest path confirmation process begins. The source will broadcast a packet signed by the public key of *D* (K_D) containing: the certificate of the source; the identity of the destination node; a nonce (N_S); timestamp (*t*); and packet identifier identifying that this is a shortest path confirmation packet (*SPC*).

$$S \rightarrow braodcast\ \ :\ \ E_{K_D}(cert_S|ID_D|N_S|t|SPC)$$

Each intermediate node that receive the *SPC* packet updates its routing table, signs the packet, includes its certificate and signs it with the public key of the destination node.

$$P_1 \rightarrow braodcast\ \ :\ \ E_{K_D}(cert_{P_1}|E_{k_{P_1}}(E_{K_D}\big(cert_S|ID_D|N_S|t|SPC\big)$$

The destination node will verify all the signatures and reply to the first and subsequent *SPC* packets with a recorded shortest path packet *RSP*. The *RSP* is propagated to the source which confirms the shortest path by verifying the nonce N_S sent with the *SPC* packet.

e. Maintenance

The ARAN solution uses error messages and implicit revocation of certificates to maintain routes. Error message packets (*ERR*) are broadcast by any node *P* that discovers a broken route. An *ERR* packet is signed by its originator and includes the certificate of the originator, the source and destination pair describing the broken route, a nonce, a timestamp, and a packet identifier. Each node receiving an ERR packet will check its routing table if it contains the accused route. If it does then the *ERR* packet is rebroadcast unchanged.

$$P \rightarrow braodcast\ \ :\ \ E_{k_P}(cert_P|ID_S|ID_D|N_P|t|ERR)$$

The expiration (*et*) attribute included in each certificate allows for implicit revocation of certificates. Certificates are implicitly checked during the route discovery process. Explicit revocation is achieved by the TTP CA broadcasting a certificate revocation message to nodes which then can forward it. Routes are re-calculated as a result of certificate revocation.

Analysis

The ARAN solution uses asymmetric key cryptography to provide authentication, integrity and non-reputation. Asymmetric cryptography will result in high complexity and computational cost. A trusted certificate authority (TTP CA) is required so that authentication can be made available. In the route discovery process unlike AODV, ARAN disallows intermediate nodes which have a path to the destination to reply with a *RREP* message. This creates addition routing overheads but ensures authentication [Sanzgiri et al, 2002].

2.4 Secure Ad hoc On-demand Distance Vector (SAODV)

Zapata et al [Zapata, 2002] proposes the Secure Ad hoc On-demand Distance Vector (SAODV) protocol as a security extension to the AODV protocol. SAODV secures the AODV protocol by using a hybrid cryptographic approach involving asymmetric cryptography in the form of digital signatures and symmetric cryptography in the form of hash chains.

a. System Overview

SAODV defines the fields in a routing packet into two categories mutable and non-mutable. The non-mutable fields are authenticated using asymmetric cryptographic signatures. The only mutable field in an AODV routing packet is the hop count. A new hash chain is created for each route discovery phase which is used to secure the hop count. SAODV requires that the AODV routing packet is extended to include the security information like digital certificates. The implementation of digital signatures and hash chains provides end-to-end authentication for the AODV routing protocol.

SAODV uses asymmetric cryptography and assumes the presence of a key management scheme to distribute keys in a secure manner [Zapata, 2002]. It also assumes that it is possible to verify the relationship between a public key and an IP address or identity.

b. Packet Extension

SAODV proposes an extension to the standard AODV message format so that security mechanism can be implemented. The SAODV extension contains the following fields as described in Table - 1 and Figure 1 [Zapata, 2002].

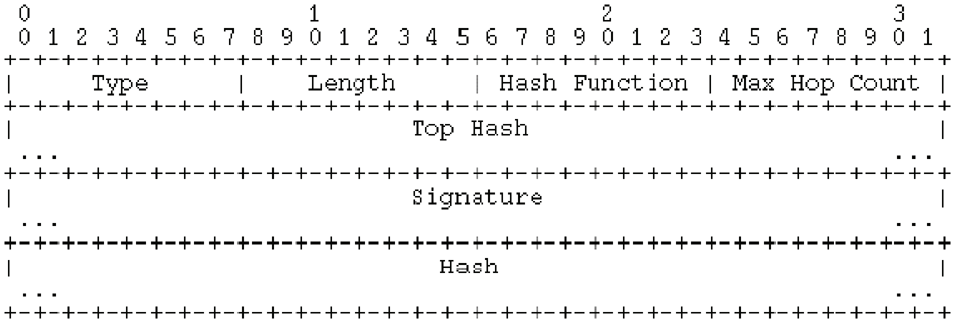
The standard AODV protocol uses packet sizes of 512 bytes but the SAODV extension requires the AODV packet size to be extended to use packets of size 1024 bytes.

c. Route Discovery Process

A source node *S* initiates a route request to a destination node *D* by performing the following steps:

Field	Description
Type	This is a packet identifier where the value 64 identifies a request packet *RREQ* and the value 65 identifies a reply packet *RREP*.
Length	The length of the packet data.
Hash Function	This describes the hash function used for example MD5 [Rivest, 1992] or SHA-1 [Publications F IPS, 208].
Max Hop Count	The maximum hop count (*mhc*) is used for hop count authentication. It defines the maximum number of nodes a packet is allowed to pass through.
Top Hash	The top hash is the result of the hash function *H* applied *mhc* times to a random generated number *x* such that: *top hash* = $H^{mhc}(x)$. Top Hash is vital in the hop count authentication process.
Signature	This field is 32-bit aligned and contains the signature used to authenticate all the non-mutable fields in the AODV packet.
Hash	This field is 32-bit aligned and contains the hash value h_i used to authenticate the mutable hop count variable.

Table 1. RREQ and RREP Signature Extension Fields



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     | Hash Function | Max Hop Count |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Top Hash                           |
 ...                                                         ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Signature                           |
 ...                                                         ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Hash                              |
 ...                                                         ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fig. 1. RREQ Single Signature Extension

1. *S* sets the max hop count (*mhc*) variable equal to the *TTL* (time to live) variable found in the IP header.
2. *S* generates a random number *x* and sets it as the value in the hash field such that $h_0 = x$.
3. The top hash is then generated by applying the hash function *H*, max hop count (*mdc*) times to h_0 such that: *top hash*= $h_{mhc} = H^{mhc}(h_0)$. The hash function *H* is defined in the hash function field in the packet header.
4. *S* digitally signs all the fields in the packet except hop count and hash field and stores the digital signature in the 32-bit signature field.
5. *S* then broadcasts the route request packet to its neighbours.

When an intermediate node receives a route request it verifies the authenticity of the hop count and the integrity of the digital signature. The digital signature is verified using asymmetric cryptography. The hop count is verified by checking $h_{mhc} = H^{mhc-i}(h_i)$, where h_{mhc} is the top hash; h_i is the hash field of the route request; and H^{mhc-i} is the application of the hash function max hop count minus the hop count (*i*) times. The one-way hash chain approach used to authenticate the hop-count is similar to the approach used in the SEAD protocol [Hu et al, 2002]. After the intermediate node verifies the digital signature and hop count, it replaces the packet's hash field with a new hash value computed by applying the hash function to the existing hash value. The intermediate node then rebroadcasts the route request and propagates it until it reaches the destination *D*.

When the route request *RREQ* reaches the destination *D*, *D* checks the validity of the packet and will reply with a reply packet *RREP* if the route request is valid. The *RREP* packet is forward along the reverse route to the source following the same authentication and integrity procedure that the *RREQ* message experienced.

AODV allows for intermediate nodes to also reply to route requests if they have a valid route to the destination node. SAODV proposes two solutions to this security problem. The first is the simplest disallowing intermediate nodes to reply ensuring that the destination node sends the reply message *RREP* and guaranteeing authentication. The second approach uses a double signature extension which allows an intermediate node *P* to reply to a route request from *S* for *D* $RREQ_{SD}$. Intermediate node *P* will reply with a double signature *RREP* message. One signature will sign the intermediate node *P*'s standard reply and the second signature will sign the original *RREP* packet received by *P* for its route to *D*. Both reply message headers are included and sent to the source *S* to establish a secure route to *D*.

d. Maintenance

AODV uses error messages *RERR* to report broken links. SAODV secures these messages using digital signatures. The originator of the error message signs the entire message except the destination sequence number. Each forwarding node also signs the message to prevent unauthorized error messages being broadcast. Nodes using SAODV do not change their destination sequence number after receiving an error message because the error message's sequence number is not authenticated.

Analysis

SAODV authenticates the AODV routing packets preventing certain impersonation attacks. The assumption is made that a node's identity and address can be securely bound to a public key. Such an assumption leaves SAODV vulnerable to Sybil attacks.

SAODV employs asymmetric cryptography which is computationally taxing. The packet size has to be extended to incorporate the security mechanism resulting in a serve communication overhead. For every route discovery a new one-way hash chain is computed resulting in further computational overheads.

The SAODV protocol assumes a key management entity is available in the ad hoc network which can successfully distribute public keys among participants. Such infrastructure is difficult to execute in mobile ad hoc networks and before SAODV is to be implemented either an off-line TTP or distributive key management scheme must be employed.

The SAODV solution uses hash chaining and digital signatures providing security against impersonation routing attacks. It also helps toward preventing denial of service attacks and eavesdropping attacks where malicious users may re-direct a route through a malicious node where eavesdropping may occur.

2.5 Secure Link-State routing (SLSP)

A hybrid scheme is proposed in [Perrig et al, 2001] called Secure Link State Routing Protocol (SLSP). It is a proactive security solution which uses digital signatures and one-way hash chains to provide security to link-state updates and neighbour discovery. SLSP secures link state information in localized manner and can operate alone or be combined with a reactive ad hoc protocol routing protocol like ZRP where SLSP would be the intra-zone routing protocol for the Zone Routing Protocol (ZRP) [Haas & Pearlman, 2001].

a. System Overview

SLSP provides secure neighbour discovery for nodes in a limited hop radius called a zone. Link state updates are authenticated using a hybrid cryptographic method and flooding attacks prevented by a priority ranking mechanism. The main assumption of SLSP is that nodes have existing asymmetric key pairs. SLSP assumes that a key management scheme is present to certify the public keys in the network.

SLSP uses four components: key distribution, secure neighbour discovery, link state update management, and a priority ranking scheme. SLSP's priority ranking scheme prevents denial of service type attacks.

b. Key distribution

SLSP assumes that each node has an existing signed public key before it joins the network; and the certification of keys is performed by an assumed key management method. Public keys are bound to the IP addresses of the network nodes. Nodes then broadcast their public keys into their neighbourhood zone, for example a two hop radius. The received public keys are used to authenticate future packets from the source node.

c. Secure Neighbour Discovery

SLSP uses a Neighbour Location Protocol (NLP) to proactively check that neighbouring nodes do not perform impersonation attacks. Link state information is periodically broadcast by nodes in the form of a NLP hello message. These messages are signed by the source and contain the source's MAC address (a unique hardware address) and IP address (a distinctive network address). A NLP hello message broadcast by source S is described here:

$$S \rightarrow broadcast \; : \; E_{k_S}(IP_S|MAC_S)$$

Notification messages are generated when conflicting link state information is broadcast. An inconsistent mapping of the IP and MAC addresses is considered as conflicting link state information. For example when two nodes with different IP addresses have the same physical MAC address.

d. Link State Update Management

Link state update packets are periodically broadcast to a limited hop radius of nodes. A link state update packet (LSU) contains the IP address of the source, a 32-bit sequence number used for updating and a hop count variable. The LSU hop count variable is authenticated using hash chains as discussed in SEAD [Hu et al, 2002] and SAODV [Zapata, 2002] and the rest of the packet content is authenticated using digital signatures.

When a LSU is received the digital signature is verified using the previously distributed public key. The hash chain is verified and the time to live (TTL) variable is decremented. The hop count authentication protects the LSU packet from travelling too many hops or from link state updates not to be received by nodes.

e. Priority Ranking Scheme

SLSP uses a lightweight flooding prevention scheme which gives priority ranking to nodes. A priority list is maintained for neighbouring nodes which ranks node's priority based on the number of link state updates a node broadcast. Malicious nodes will flood the network with link state update packets to cause resource and denial of service attacks. SLSP gives high priority to nodes that send less link state updates limiting the affects of flooding attacks.

Analysis

The Secure Link State Routing Protocol is a hybrid cryptographic scheme using digital signatures to provide authentication for its NLP hello messages and link state update (LSU) packets. Hash chains are used to authenticate the limited hop broadcast of LSU. Impersonation type attacks are prevented by monitoring the IP and MAC address bindings of neighbours through a neighbour location protocol (NLP). Link state updates are authenticated using digital signatures and hash chains. Flooding type attacks are prevented using priority ranking.

The SLSP protocol provides security to topology discovery but cannot act as a standalone security mechanism as it lacks a data transmission protection agent. Nodes that securely join the network can misbehave during data transmission without being detected or prevented.

SLSP lacks a disciplining agent like a revocation mechanism. For example a malicious node *B* can impersonate another node *A* and flood *A*'s neighbours with LSU packets. SLSP's priority mechanism will limit the effectiveness of the flooding attack. The NLP protocol will detect the impersonation attack but node *A* has no mechanism to correct to the attack and *A* will remain with a low priority.

2.6 On-Demand Secure routing Byzantine Resilient routing protocol (ODSBR)

The on-demand secure routing byzantine resilient routing protocol (ODSBR) is proposed in [Hu et al, 2003b]. Byzantine behaviour is defined by the authors as any action taken by an authenticated node to disrupt the routing procedure. ODSBR is a secure reputation based routing protocol that prevents the effects of byzantine failures on successful routing.

a. System Overview

ODSBR uses weighted paths to select routes in the route discovery process. Paths are assigned weights based on a fault path detection method. A high weight is assigned to an unreliable path. ODSBR is divided into three components: route discovery process, fault detection, and weight path management. ODSBR assumes that a public key infrastructure is present to manage public key authentication.

b. Route Discovery Process

Routes are discovered in an on-demand manner like in DSR. ODSBR extends the standard route request *RREQ* by adding a weight list instead of a node list like in DSR. A weight list includes the list of chained nodes with their associated weights. These weights are defined by link failures detection mechanism. The *RREQ* is signed by the source and broadcast into the network updating its weighted list after each hop until it reaches the destination. The destination then verifies the signature and broadcasts the reply message *RREP*. Each node that receives a *RREP* will then calculate the total weight of the path by summing the weights of the specific path to the current node. The *RREP* forwarded if the total weight is less than the total weight of any previously forwarded *RREP*. Before an intermediate node *P* forwards a suitable *RREP* all the signatures are verified and node *P* appends its signature. The source node receives the *RREP* and calculates the total weight and verifies all the signatures.

c. Fault Detection Method

Each node *i* has a list of probe nodes. Each probe node sends node *i* an acknowledgement message for each data packet *i* sends. If a threshold *t* of acknowledgements is not received a fault accusation is logged against a specific path. Using a binary search technique ODSBR identifies a path as faulty after *log(n)* fault accusations, where *n* is the length of the accused path.

d. Weight Path Management

A low weight is associated with a secure path. The weights associated to paths are influenced by two factors: time and fault detection. When ODSBR identifies a path as faulty based on the fault detection method then the weight for that path is doubled. Path weights are halved after a counter reaches zero, each path has an associated counter.

e. Analysis

The on-demand secure routing protocol (ODSBR) provides ad hoc on-demand routing with byzantine failure prevention. Weights are assigned to paths by a fault detection method and paths are selected based on the weights. The Secure Routing Protocol (SRP) proposed in [Papadimitratos & Hass, 2002] introduces the metric specific path selection method but this proposal is not a standalone secure routing protocol like ODSBR [Awerbuch et al, 2008].

ODSBR assumes the existence of a public key management system. ODSBR further assumes that a shared key exists between source and destination nodes to ensure authenticity and integrity of acknowledgement messages sent by probe nodes. This helps avoid expensive asymmetric per packet computations for acknowledgement messages.

The route discovery process of ODSBR broadcast reply messages instead of unicasting them which results in a computationally expensive operation. This method will result in $2^{\frac{n}{2}+1}-1$ reply packet transmissions where $n+2$ is the number of nodes in a path from node A to B including nodes A and B [Awerbuch et al, 2008]. Furthermore the cost monitoring of data packet transmission is computationally high because the fault detection method requires a threshold of t probe nodes to reply with an acknowledgement for every data packet sent.

ODSBR is identified by authors [Awerbuch et al, 2008] to be vulnerable to wormhole attacks. The wormhole attack may be avoided in the case where the wormhole link node exercises byzantine behaviour.

2.7 Reputation based CONFIDANT

The CONFIDANT protocol representing the ‘Cooperation Of Nodes: Fairness In Dynamic Ad Hoc Networks’ [Buchegger & Boudec, 2002] is a reputation based solution which operates over the DSR routing protocol.

a. System Overview

The CONFIDANT solution does not use any cryptographic techniques to achieve secure routing. The system is solely reputation-based and operates in an on-demand ad hoc network environment as an extension of the DSR routing protocol. Each node in the ad hoc network is required to be involved in the four components of CONFIDANT: monitoring, trust management, reputation system and path management.

b. Monitoring

Each node is a monitor and is responsible for the packets that it sends or forwards. For every packet that a node forwards it watches that the next hop node forwards the packet properly. The monitor looks for inconsistent behaviour and triggers an alarm to the reputation system if misbehaviour is discovered.

c. Trust Management

The trust management system manages the alarm messages. The alarm messages are generated by each node’s monitoring system and exchanged between other nodes as to build and maintain a local rating list. The trust management manages the input of alarm messages and assigns more influence to alarm messages that come from trusted nodes and less influence from other nodes. CONFIDANT assumes pre-existing relationships between a selection of nodes called friends [Buchegger & Boudec, 2002], friend nodes are highly trusted nodes. Local rating lists are exchanged as well and their influence managed by the trust management system.

d. Reputation System

The reputation system manages and maintains the local rating list. This list contains node identities and corresponding rating. A rating will correspond to the amount of misbehaviour a node has displayed. The ratings will be updated from alarm messages and direct observations.

e. Path Management

Paths are selected based on a rating threshold, local rating lists and a blacklist containing all untrusted nodes. A node is blacklisted when its rating is below the rating threshold *t*. The path manager removes the paths in the network which contain the blacklisted node. The path manager manages the route discovery process by reacting to route requests from blacklisted nodes or route requests that have passed through a blacklisted node.

f. Analysis

CONFIDANT is an exclusively reputation based routing protocol. Local rating lists of node's behaviour is recorded and used during the route discovery process. The authors note that only negative evidence is gathered against nodes so nodes can only be identified as less trusted as the network continues. Like most reputation based schemes a counter system is employed where each rating list entry has an associated counter. When the counter reaches zero the rating is reset to the default state of null misbehaving accusations. The CONFIDANT protocol assumes the existence of prior trust relationships between a selected number of nodes called friends [Buchegger & Boudec, 2002].

2.8 Discussion

Several different secure routing protocols are presented in Section- 2.2 they differ in the areas of security and operational requirements. The diversity of their design makes it difficult to compare their success but this section outlines the diverse characteristics of the presented protocols.

a. Security Analysis

The proposals can be categorized by the security techniques which include the asymmetric, symmetric and hybrid cryptographic security approaches. The last category is the reputation based solutions. The security mechanism of each protocol is presented and the attacks which the protocol protects against are highlighted. A summary of the evaluation is presented in Table -2.

Symmetric Cryptography

The symmetric cryptographic approaches include the Secure Efficient Ad Hoc Distance Vector Routing protocol (SEAD) and the Ariadne protocol. Hash functions and hash chains like SHA-1 [Publications F IPS, 2008] and MD5 [Rivest, 1992] are used for authentication purposes usually for the hop count variable. The hash function is lightweight compared to asymmetric security techniques.

The SEAD approach uses a hop-by-hop authentication technique. SEAD authenticates the sequence number and hop count of routing packets protecting the routing procedure from resource consumption attacks for example denial of service attacks, route table poisoning, and replay attacks.

The Ariadne protocol is proposed by the same authors of SEAD. Ariadne uses message authentication code (MAC) to provide end-to-end authentication between communication nodes. Ariadne protects against similar attacks to SEAD but uses end-to-end authentication.

Protocol	Security Approach	Techniques	Attack Prevention
SEAD [Hu et al, 2002]	Symmetric Cryptography	• Hop-by-hop authentication • Hash chains	• Resource consumption • Denial of Service • Route table attack • Replay
Ariadne [Hu et al, 2005]	Symmetric Cryptography	• End-to-end authentication • Hash chains	• Resource consumption • Denial of Service • Route table attack • Replay
ARAN [Sanzgiri et al, 2002]	Asymmetric Cryptography	• End-to-end authentication • Certificate Authority	• Route table attack • Replay attacks
SAODV [Zapata, 2002]	Hybrid Cryptography	• End-to-end authentication • Hash chains • Digital Signatures	• Denial of Service • Route table attack • Replay
SLSP [Papadimitratos & Hass, 2003]	Hybrid Cryptography	• Secure neighbour discovery • Authenticated link state updates	• Denial of Service • Route table attack • Replay
ODSRP [Awerbuch et al, 2008]	Reputation Based	• Path specific reputation lists • Digital Signatures	• Denial of Service • Route table attack • Replay • Byzantine Failures
CONFIDANT [Buchegger & Boudec, 2002]	Reputation Based	• Node specific reputation lists	• Black Hole • Replay

Table 2. Summary of security analysis for secure routing in ad hoc networks

Asymmetric Cryptography

The only solely asymmetric cryptographic approach investigated is the Authenticated Routing for Ad hoc Networks protocol (ARAN). Asymmetric cryptographic is computationally costly compared to symmetric cryptography and it requires the existence of a trusted third party or self organized key management system.

ARAN provides end-to-end authentication for an on-demand mobile ad hoc network. ARAN provides authentication and protecting from replay attacks and unauthorized routing table attacks. ARAN is vulnerable to flooding attacks. A malicious node can flood nodes with fake routing packets signed with illegitimate keys this will result in many unsuccessful verifications and ultimately denial of service and resource consumption.

Hybrid Cryptography

The SAODV and SLSP protocols are hybrid solutions which employ both asymmetric cryptography and symmetric cryptography. The common approach is for all the mutual fields to be digitally signed and the immutable fields, like the hop count, to be protected using hash chains. The Secure Ad Hoc On-demand Distance Vector protocol (SAODV) employs this tactic to provide end-to-end authentication but at the cost of extending the routing packet header. SAODV protects against impersonation attacks on the routing protocol. It also helps prevent replay and denial of service attacks.

Secure Link State Routing Protocol (SLSP) provides secure neighbour discovery and authenticated link state updates but lack a secure data transmission protocol. The Neighbour Location Protocol of SLSP protects against impersonation type attacks where malicious nodes adopting conflicting IP and MAC addresses would want to corrupt the routing table. Furthermore flooding attacks, which result in resource consumption and a denial of services, are prevented by a priority ranking scheme.

Reputation Based

Reputation based or conduct based systems allow for a nodes behaviour in the network to affect its assigned security or trustworthiness. Reputation based protocol can be computationally costly because they usually require packet monitoring systems and the proactive exchange or behavioural evidence between nodes. The On-demand Secure Routing Protocol Resilient to Byzantine Failures (ODSBR) and the CONFIDANT protocol are reputation based systems.

ODSBR uses reputation based system to select the most secure routes. A fault detection method monitors the success of each packet transmission and faults are logged against specific paths. Reputation is path specific in ODSBR. ODSBR couples with asymmetric cryptographic approach to provide end-to-end authentication along the selected secure path. The CONFIDANT uses exclusively reputation based techniques to provide security. Similarly to ODSBR only negative evidence is considered. Nodes monitor every packet which they forward and maintain a local rating list. Reputation or ratings are node specific unlike ODSBR. Both CONFIDANT and ODSBR monitor packet forwarding this will protect the system from black hole attacks. Replay attacks which use the method flooding are prevented using path reputation and node reputation in ODSBR and CONFIDANT respectively. A disadvantage of the negative reputation approach for ODSBR and CONFIDANT is that black list nodes or faulty path entries have an expiration time after which their confidence is reinstated. This allows malicious nodes to continue disrupting the network until they are caught again.

b. Operational Requirements

The presented secure ad hoc routing protocols have certain assumptions that each makes to realize its design. Furthermore protocols are designed specifically for operation in specific routing environments. This section summarizes the operational requirements of the presented secure ad hoc routing protocols. Table -3 summarizes this discussion.

The symmetric cryptographic approaches do not rely on a public key infrastructure but still require some kind of key management in the ad hoc network. The SEAD protocol is designed for a table-driven routing protocol and is based on the DSVD routing protocol. SEAD requires a key management mechanism to distribute an authenticated initial hash element. Ariadne is a DSR based on-demand protocol which assumes there are shared secrets between each communication pair. The shared keys are used in TESLA authentication and a key management system is assumed present to distribute the keys. TESLA authentication also requires time synchronization between each node. This is difficult without the presence of an online TTP.

ARAN, SAODV, SLSP and ODSBR use asymmetric cryptography and key management is simply assumed for each of these protocols. ARAN assumes that an online TTP is present that acts as a certificate authority (CA). Prior shared secrets are assumed between all participating nodes and the CA. ARAN is an on-demand protocol. SAODV protocol is based on the AODV on-demand routing protocol and assumes the presence of a key management system to distribute keys. SLSP assumes that nodes enter the network with asymmetric key pairs and a key management scheme is present to certify the keys in the network. SLSP is a table based routing solution. ODSBR is based on DSR routing and authenticates its routing packets with digital signatures and a public key infrastructure is assumed to manage the keys. Shared keys are also assumed to allow for authentic acknowledgement message communication between source and probe nodes.

Protocol	**Routing**	**Assumption**
SEAD [Hu et al, 2002]	Table Driven DSDV based	• Key management system to distribute an authenticated element for hash chaining
Ariadne [Hu et al, 2005]	On-Demand DSR based	• Time synchronized network • Shared secret key between each node pairs for MAC • Key management system to manage TESLA keys
ARAN [Sanzgiri et al, 2002]	On-Demand Not protocol specific	• TTP acting as a certificate authority (CA) • Prior shared CA public key
SAODV [Zapata, 2002]	On-Demand AODV based	• Key management scheme • Secure IP public key binding
SLSP [Papadimitratos & Hass, 2003]	Table Driven Not protocol specific	• Nodes have existing asymmetric key pair • Key management system
ODSRP [Awerbuch et al, 2008]	On-Demand DSR based	• Key management system • Shared keys between source and probe nodes
CONFIDANT [Buchegger & Boudec, 2002]	On-Demand Not protocol specific	• Pre-existing relationships between a selection of nodes called friends

Table 3. Operational requirements of the present secure routing protocols

CONFIDANT does not use cryptographic techniques and does not require the existence of a key management scheme. CONFIDANT does assume pre-existing relationships between a small number of nodes called friends. The CONFIDANT solution is designed for on demand routing.

From this discussion, the conclusion is made that most secure ad hoc routing protocols assume the existence of a key management system to certify, authenticate, and distribute keying information. Mobile ad hoc networks cannot assume the existence of a TTP and must address the problem of key management.

3. Proposed security scheme: Direct Indirect Trust Distribution (DITD)

A security establishment scheme is proposed for a mobile ad hoc network. Key management is central to the establishment of trust in these networks. The proposal focuses on key management. The proposal is for a mobile ad hoc network, which operates in a self-organized and fully distributive network environment. These networks allow for nodes to join and exit the network, unrestricted. These networks find application in the military and commercial filed. For example application can be found in, tactical positional networks for military based communication or personal area networks for secure peer-to-peer data and file sharing. The proposed protocol is planned for these self-organized distributive networks. It is noted that these networks do not allow for rigorous access control. The proposed protocol can be extended to allow access control services.

In a self-organized mobile ad hoc network, there is no presence of a separate authority member, such as a trusted third part or certificate authority. Instead each node that enters the network is considered the security authority of its own domain. Security is established by nodes creating and issuing certificates which bind nodal identities to their respective public keys. These certificates are issued and distributed in order to realize secure communication. A bi-directional security association is made between nodes *A* and *B,* when node *A* holds a certificate binding the public key of *B* and *B*'s identity; and *B* holds the certificate binding the public key of *A* and *A*'s identity. Malicious adversary nodes that wish to disrupt communication will target the network layer, and more specifically the routing mechanism, as identified in the previous chapter. The network layer is identified as the sphere of design.

The problem is then to provide secure communication, which is implemented on the network layer. Secure communication is achieved when node *A* is able to set up a secure communication channel, where no other entity can interrupt or eavesdrop on its communication with node *B*. The question as to whether node *B* is worthy of trust, is not the concern. That question must be decided by the nodes themselves, based on available trust evidence. The proposal made on the network layer aims to provide the most secure route between *A* and *B*, preventing malicious adversaries from sabotaging communication.

The term trust is defined as the "belief by a trustor with respects to the competence, honesty, security and dependability of a trustee within a specific context." [Grandison, 2003]. There are two trust variables: direct trust and indirect trust. Direct trust is a result of independent or local trust evaluation between two immediate nodes. Indirect trust is evaluated using the advice from other nodes. In the context of certificate base trust, direct trust is defined as trust between local neighbours. Indirect trust is created by certificate chaining.

A hybrid trust model is proposed, uniting certificate and conduct based trust to provide a more secure communication. A key management model is also proposed. This model supports an existing routing protocol. The proposed scheme is called Direct, Indirect Trust Distribution (DITD) and it follows the following procedure: direct trust is established by requesting that all nodes involved in the route discovery stage, share their self certificates with each others' one-hop neighbours involved in the route discovery phase. Indirect trust is further established by requesting that the sender's self certificate propagates with the route request towards the destination. The routing messages trigger certificate distribution allowing direct trust relationships between one-hop neighbours. These trust relationships are then chained together providing a trusted route to the destination node. Keying material is allowed to be propagated along these chains of trust. A disadvantage of the self-organized nature of these networks is that the established security of trust chains will rely on transitive trust [Capkun et al, 2003]. The DITD model proposes coupling the security provided by the certificates with a conduct based trust analysis model. Conduct trust is affix, which allows for more secure communication. This is achieved by calculating the trust of routes, based on the conduct of the nodes involved, and selecting the most trusted route for communication.

3.1 Related work

A detailed survey was presented on key management schemes for mobile ad hoc networks in the previous chapter. Section-2 focused on the network layer and presented a survey of existing secure routing protocols. This section provides work directly related to the DITD model.

The authors of [Capkun et al, 2003] propose a completely self organized public key system for mobile ad hoc wireless networks. This is a PGP based solution which provides key management in ad hoc networks without the presence of an off-line or on-line authority, like a CA, TTP or server. Each node distributes its self certificates and maintains its own certificate repositories. Nodes participating in the network share their certificate repositories and repository updates are preformed in a proactive manner. Certificates are reciprocally authenticated and trust chains formed linking remote nodes to each other. Security is realized on the application layer.

Zapata [Zapata, 2006] addresses the issue of verification delays in secure mobile ad hoc networks. Zapata proposes a protocol to optimize the number of verifications made in a single secure route discovery phase. Once a route is established only then are the shared certificates verified. This helps in reducing the computational overhead of verifications on multi-hop paths. By reducing the total number of verifications made in a network's life time there is a resultant end-to-end delay upon the delivery of routes.

Theodorakopoulos *et al.* [Theodorakopoulos & Baras, 2006] proposes a fully distributive conduct based trust model which has PGP characteristics. This model operates on the application layer and allows for trust to be established without the presence of a central authority member. PGP models share certificates to establish trust while the work proposed in [Theodorakopoulos & Baras, 2006] allows for other trust evidence, like conduct and location, to influence the trust establishment. Trust is fully distributed in a proactive manner allowing all nodes to give trust opinions about other nodes.

Semiring mathematics presented in [Kscischang et al, 2001] has more recently been used to model trust calculations in [Theodorakopoulos & Baras, 2006]. Trust opinions are mathematically aggregated along a path and trust decisions are mathematically represented. The work in [Theodorakopoulos & Baras, 2006] uses Dijsktra's extended algorithm proposed by Mohri [Mohri, 2002] to include trust. This finds the most trusted path between two remote nodes in a proactive manner.

The majority of literature mentioned function in a proactive manner for application layer solutions. The DITD model is designed on the network layer for a reactive, fully distributive, self organized, mobile ad hoc network environment. The ideas of some of these protocols have inspired the creation of the DITD model and the impact of these protocols is discussed in Section-4.

3.2 Proposed security scheme

The aim is to design and investigate a security mechanism to specifically provide: *public key certificate distribution*, *optimal verification*, and a *conduct trust model* to optimize trust decisions.

The security mechanism is to provide secure communication in a mobile ad hoc network environment while satisfying the following requirements based on environment and functionality.

a. Design Requirements

Environment

- **Network layer design:** The security mechanism is to be implemented on the network layer protecting these dynamic networks from attacks and avoiding multi-layer design.
- **Self organised:** Nodes are responsible for their own security services, including the distribution of keying information.
- **Fully distributive:** The certificate distribution scheme is to be designed in a fully distributive manner where all nodes participate in the operation and implementation.
- **On-demand:** The DITD model is to be design in an on-demand environment optimizing the limited resources of ad hoc networks. On-demand models provide security to nodes upon request. The proactive approach provides security to an entire network at once and requires computationally taxing periodic updates.

Functionality

- **Distribution of keying material:** DITD is to provide direct and indirect trust relationships between local and remote nodes by efficiently distributing self certificates between nodes.
- **Minimize the overhead:** DITD aims to minimize the overheads upon the network routing performance while still providing trust establishment. DITD aims to avoid alterations to the routing control packets and strives for independence between routing and trust establishment.
- **Provide secure communication from the start:** Secure communication is requested from the start to the end of the network lifetime unlike the model proposed in [Tanabe & Aida, 2007] which is flawed by an initial setup phase with weak security.
- **Trust evaluation mechanism:** Security should be supported by a trust evaluation mechanism allowing for more secure routes to be established and ensuring the secure distribution of certificates.
- **Robust in the presence of topology change:** The DITD model should be robust to poor connectivity and routing failure due to changing mobility which is an inherent characteristic of a mobile ad hoc network.

b. System Model

To fulfil the constraints given in above, we assume the following system model. There is no pre-existing infrastructure and no online trusted third party present during communication. The model is a fully distributive network of wireless nodes using an ad hoc on-demand routing mechanism. It is assumed that nodes have their own keying material before joining the network generated by a fully self organized mobile ad hoc network [Capkun et al, 2003], or by an off-line authority issuing keying material before a node enters the network for example in [Capkun et al, 2006]. Each node is assumed to have a public and private key pair; a self certificate binding the public key and user identification of the node; and a set of network security parameters common to all nodes in the network. Secure communication is requested from the start to the end of the network's lifetime. Users can join and leave the network without any restrictions. Any user with the correct keying material may participate in the network. It is assumed that conduct information is available to each node from node monitors [Tseng et al, 2003].

The DITD model uses certificate based trust coupled with conduct based trust to develop a hybrid trust protocol maximizing trust in the network. The DITD model addresses the issues of *key exchange*, *verification protocol* and *conduct trust evaluation*. It is designed on the network layer accompanying an on-demand routing protocol. Figure 2 describes the high-level system model.

The DITD scheme performs the task of *key exchange*, exchanging self certificates on the network layer following an on-demand routing mechanism. Direct trust is established by self certificate exchanges among one-hop neighbours, triggered by the route discovery process. This allows for a bi-directional security association to be made between immediate nodes, we refer to this relationship as direct trust. Direct trust associations are chained together creating trusted paths and allowing for two nodes, out of each other's communication range, to exchange self certificates. This describes the *key exchange* element of the DITD model. It is divided into two parts: the exchange of direct and indirect trust relations. Figure 3 illustrates the direct, indirect trust relations established by DITD.

An on-demand route discovery phase will flood the network with route requests in search of a destination. This will couple with the *key exchange* mechanism of DITD. The result of this is a flood of self-certificate exchanges. Verification of these certificates is optimized by the DITD model. Trust chains will have an accumulative certificate verification delay because possibly each direct trust association will need to be verified. The DITD model proposes a *verification protocol* which optimizes and manages the verification process.

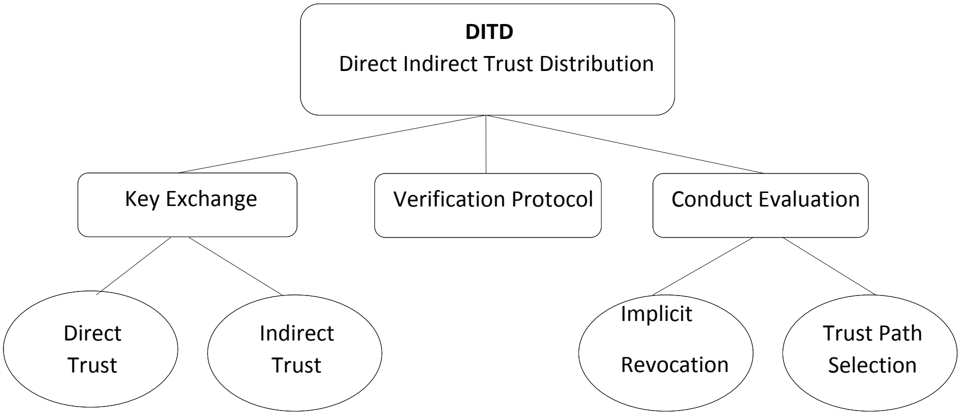


Fig. 2. High level system model

DITD uses the existence of conduct trust evidence to maximize the security provided by public key certificates. Trust is aggregated in a reactive manner using semi-ring mathematics and a reactive shortest path algorithm. The most trusted routes are selected by a *conduct evaluation protocol* which includes an implicit revocation mechanism and trust evaluation metric. Direct and indirect trust establishment is strengthened by DITD's conduct trust evaluation.

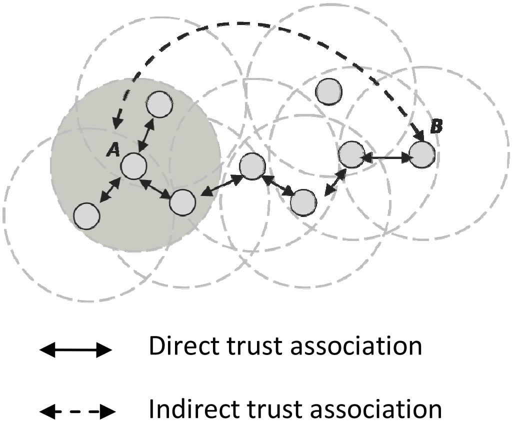


Fig. 3. Direct and Indirect trust establishment in DITD model

c. Key exchange

A certificate trust model is used to create trust between nodes. As an example, a secure bidirectional communication path can be setup between node *A* and *B* if both nodes have exchanged each other's self certificates as seen in Figure 4. The self certificate binds the user *ID* and the public key. So the exchange of self certificates allows keys to be exchanged authentically. Node *A* can trust node *B* if node *A* has a certificate verifying the identity of node *B*, and similarly concerning node *B* with respects to communication with node *A*.

The DITD certificate trust model appends an existing on-demand mobile ad hoc routing protocol. Its principals can be applied to any on-demand routing scheme. Knowledge of the operation of an ad hoc routing mechanism will help visualise the explanation of DITD.

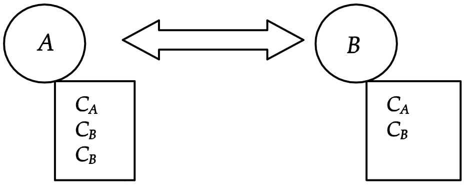


Fig. 4. Secure bidirectional communication between node A and B

The application of an ad hoc on demand routing procedure is briefly revisited to aid the DITD explanation. The AODV [Perkins et al, 2003] routing procedure follows three stages during route discovery:

1. Sending of request message (*RREQ*)
2. Receiving of request message (*RREQ*)
2. Sending reply message (*RREP*)

In stage one, the source node *A* request communication with destination node *B* by broadcasting a routing request $RREQ_{AB}$ into the network. This request is forwarded and propagated through the network to *B*. The *RREQ* message may have been sent by *A* or forwarded by an intermediate node P_i. When the *RREQ* is received by an intermediate node

P_i, stage two begins. At stage two a reverse route to *A* is then set up and *P_i* checks if it is the destination *B* or has a fresh route to the destination node *B*. If not then the *RREQ* is further broadcast by *P_i* and propagates until the destination is found. When the destination or a fresh route to the destination is found stage three commences. At stage three a reply message *RREP* is propagated along the reverse route until it reaches the source node *A* establishing the communication route.

When a node receives a routing control packet, certificate requests are triggered and sent using separate unicast messages. The certificate distribution is added at stage two and stage three, the receiving of a route request and the sending of a reply message respectfully. In Table-4 we define the symbols we used next in our explanation.

P_i	intermediate node *i* receiving the *RREQ*
P_{i-1}	previous node P_{i-1} who forwards *RREQ* to its neighbour P_i
A	originator node of RREQ message
B	destination node of RREQ message
$Cert_i$	certificate of node P_i

Table 4. Definition of symbols

At stage two upon receiving a route request packet, before this packet is processed and the routing table updated, direct trust and indirect trust establishment is set up.

Direct Trust

At stage two, direct trust relationships are made by sharing neighbouring nodes self certificates. When intermediate node P_i receives a route request *RREQ* it first checks its certificate repository for the certificate of the neighbour who forwarded the request, P_{i-1}. If it does not possess such a certificate, $Cert_{i-1}$, a local self certificate exchange is done between node P_i and its previous hop neighbour P_{i-1} as follows: a *unicast* message is sent from P_i to P_{i-1} with P_i's self certificate $Cert_i$ appended; P_{i-1} receives the message, updates its certificate repository and replies with a unicast message to P_i containing P_{i-1}'s certificate. If the certificate $Cert_{i-1}$ is found in P_i*'s* certificate repository there is no need for a self certificate exchange. This procedure follows the *RREQ* as it floods the network in search of a route to the destination node. Direct trust establishment is illustrated in Figure 5. This, as it is expected, causes an increase in control packet overhead as the DITD model transmits additional certificate packets into the network.

A second direct trust establishment approach is proposed, which exploits the *HELLO* packets of the AODV routing protocol. The AODV protocol uses periodic one-hop broadcasts packet to maintain and establish communication between neighbouring nodes. The DITD model proposes that direct trust can be established independent of the route discovery process by including a certificate query in the *HELLO* packets. When a $HELLO_A$ packet is broadcast by node *A* and received by node *B*, the receiver *B* checks its certificate repository for the certificate $cert_A$. In a similar way to the first approach if no certificate is found a localized certificate exchange is performed. The certificate exchange messages are independent from the *HELLO* packets. The second approach allows for direct trust establishment with the least amount of dependence on the routing procedure.

Indirect Trust

Still at the stage two (receiving the routing request) indirect trust is established between remote nodes *A* and *B* by the exchange of remote self certificates. Similarly to the direct trust set up, before node P_i processes the received routing request *RREQ* a certificate search and exchange is made. Node P_i searches for the source *A*'s certificate, $Cert_A$. If the certificate is not found, P_i sends a separate unicast certificate request for $Cert_A$ to the previous node P_{i-1}, whose address can be found at the next hop on the reverse route in the routing table. This addition allows for the source's certificate $Cert_A$ to be propagated towards the destination *B*. It is noted that by not appending the certificate to the route requests this reduces dependency between the route establishment and certificate trust establishment.

For indirect trust to be complete between nodes *A* and *B*, the source *A* is required to possess the destination's certificate, $Cert_B$. Further additions to stage three, sending the reply message, are required to complete the indirect trust establishment. A reply is sent is sent under two conditions. Firstly when the destination node is found and secondly when a fresh route to the destination node is found.

For the first condition, the reverse route to the source *A* is already setup with localized direct trust existing between nodes on the route. Therefore a trusted chain of nodes is available from *B* toward the originator node *A*. All that is required is for the certificate of the destination node, $Cert_B$, to be piggy backed on the routing reply message *RREP* toward *B*. Each intermediate node stores $Cert_B$ and updates its certificate repository and the forward route to *B*.

For the second condition, if a fresh route to *B* is found at P_i, there exists a route from the intermediate node P_i to the destination *B* and a route from P_i to the source *A*. Both routes have localized direct trust existing already. Two *RREP* messages are then propagated, one toward *B* with $Cert_A$ appended and one toward *A* with the $Cert_B$ appended. Indirect trust is therefore set up by certificate chaining as illustrated in Figure 5.

c. Verification Protocol

For trust to be established between two entities they must not only share the certificates but the certificates must be verified for the users to be authenticated. Ideally verification will take place immediately after a certificate is received but a single verification can take up to 1ms delay [Stephan Eichler, 2006] on 1024-bit RSA key, and even more for a ECC key. These verifications can accumulate across multi hop routes. For application specific networks that are time dependent like audio applications and military automation networks a delay of milliseconds is critical. A requirement of DITD is for the security additions not to cripple or delay the existing routing mechanism.

Verification for direct trust establishment can be done immediately without incurring a delay upon the routing mechanism. This is because the localised certificate messages are separate and independent from the request messages. Furthermore during route discovery, request messages (*RREQ*) can be forwarded without waiting for verification to be processed [Zapata, 2006] as verification can be confirmed on the reply route. Delayed confirmation of verification is not possible for the reply message (*RREP*) because the exchange of the destination node's certificate, $Cert_B$, follows the *RREP* message. The $Cert_B$ certificate must be verified before the *RREP* message can be securely forwarded and trusted routes established. This means that a certificate trust chain will have an accumulative processing delay due to verifications. Therefore the problem is that the verification of the destination certificate $Cert_B$ may cause a delay in route establishment because $Cert_B$ is distributed with the *RREP* message.

A solution to this is that if any intermediate node has $Cert_B$, it can distribute $Cert_B$ to the reverse route, during *RREQ* message propagation. When a *RREQ* message is forwarded a

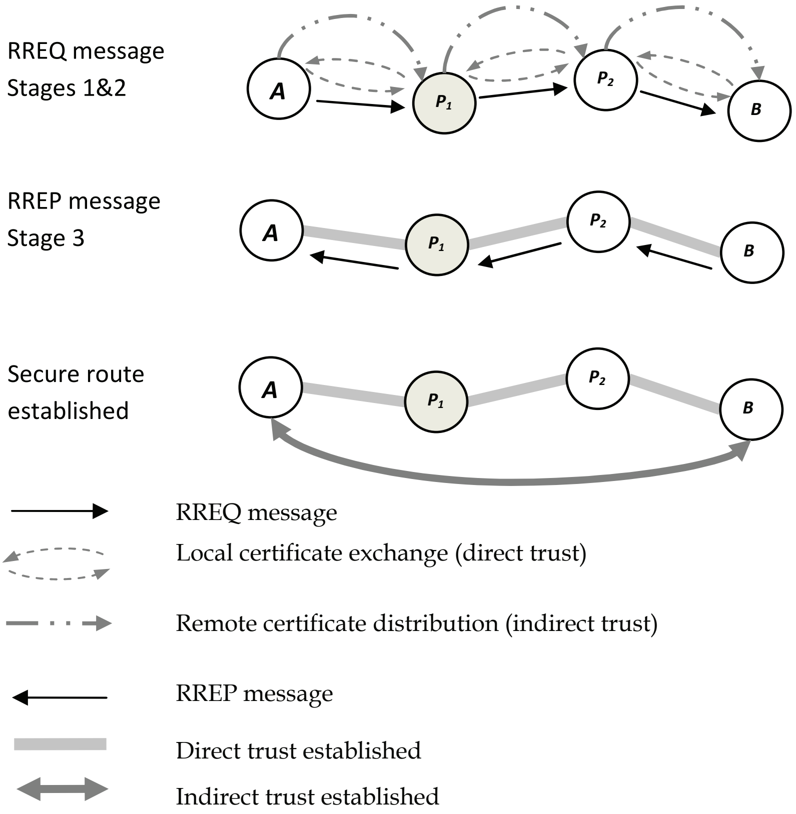


Fig. 5. Illustrating the certificate exchange protocol

flag is appended identifying if the forwarder has the destination certificate $Cert_B$. Intermediate node *P* receives the *RREQ* message and updates the reverse route entry with $flag_{cert}$ indicating if the previous hop has $Cert_B$. *P* checks if it has $Cert_B$ in its certificate repository and assigns an appropriate value to $flag_{cert}$ before forwarding the *RREQ* message. If *P* has $Cert_B$ and the reverse route variable $flag_{cert}$ indicates that the previous hop does not have $Cert_B$ then *P* sends a unicast certificate message containing $Cert_B$ to the previous hop, whose identity can be found from the reverse route in the routing table. The $Cert_B$ is propagated along the reverse route by checking the routing table entry $flag_{cert}$ and responding in a similar fashion. This allows the destination certificate $Cert_B$ to be distributed during the route discovery phase independent from route establishment. The verification protocol is illustrated in Figure 6 where source *A* sends a *RREQ* for destination *B* and intermediate node P_2 has $Cert_B$ but no route to *B* itself. In this case while route discovery continues $Cert_B$ is transmitted to the nodes in the reverse route which do not have $Cert_B$, these nodes are indicated by the $flag_{Cert}$ variable. Certificate verification is done concurrently with route discovery therefore minimising the amount of verifications that delay the route discovery. Outstanding verifications are done following the *RREP*. Verification checks are preformed with the *RREP* message. Figure 6's example allows for three less verification delays at P_2, P_1 and *A* because of back tracked verification.

The condition under which the above will be most effective is when a node has received $Cert_B$ during a previous route establishment but it no longer has an existing route in its routing table for *B*. Such an occurrence is a result of a node previously involved in a route to *B* but due to route expiry, loss of connectivity or node mobility the node is no longer part of such a route. Therefore the benefits of the verification protocol are most evident in ad hoc networks with moderate to high speeds. The DITD model implements verification optimization to reduce the delay incurred on the routing mechanism by verification.

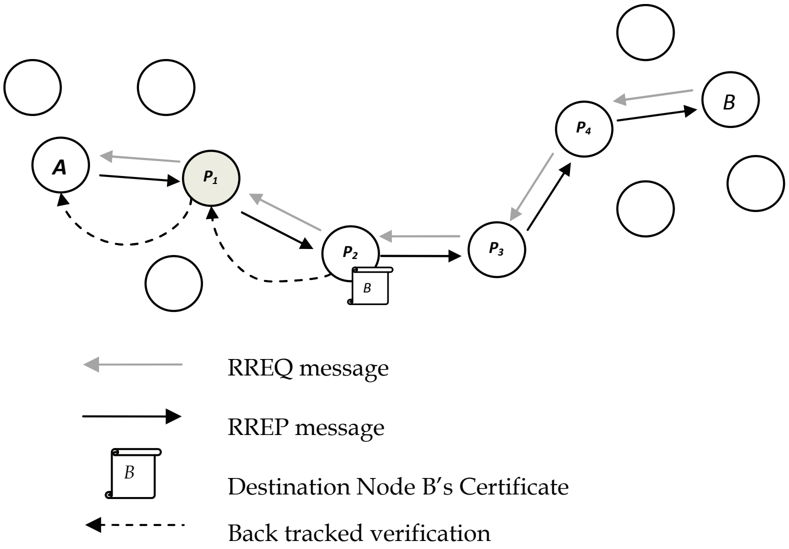


Fig. 6. Illustration of verification protocol

The authentication protocols ARAN, SAODV, SEAD or Ariadne discourage intermediate nodes with a route to the destination to reply to route requests. If the DITD model would be used in conjunction with such a protocol then although intermediate node with a validate destination certificate are unauthorized to reply DITD maximizes the availability of the destination certificate. The verification protocol would be used to distribute the destination's certificate it along the reverse path and perform verification checks so lesser time delay is incurred from the route reply message.

Direct and indirect trust establishment is realised through the route establishment phase of the ad hoc routing scheme. During the initial stage of route establishment the network is flooded with routing requests and in turn certificate exchange messages. It can be expected that there will be a large packet overhead as a result of additional certificate packets.

Mobility produces erratic connectivity problems and unexpected routing failure. Multi-hop routes are vulnerable to failure under increased mobility while localised one-hop route connections are less vulnerable. If the proposed solution was dependent upon such multi-hop routes, like [Capkun et al, 2006] is, it would suffer severely from inherit link breakages common to highly mobile networks. The proposed solution prevents the certificate exchange procedure from using multi-hop routes by exchanging certificates in a strictly localized manner. This allows the DITD certificate distribution scheme to operate in ad hoc networks with varied mobility's and changing connectivity without the worry of routing failure interfering with security.

d. Conduct Trust Evaluation

Providing conduct based trust enhances the trust decision made by nodes and therefore effect keying decisions: "Conduct trust influences decisions like access control, choice of public keys, etc. It could be useful as a complement to a public key infrastructure (PKI), where an entity would accept or reject a public key according to the trustworthiness of the entities that vouch for it; this is the idea behind PGP web of trust [Abdul-Rahman, 1997]. It also provides trust influence at the network layer allowing for routes to be selected based on trust. Trust Establishment incorporates the following functions: specification of evidence, generation, distribution, discovery and evaluation of trust evidence. The scope of the work focuses upon trust evaluation rather than the collecting of trust evidence from the network and the semantics of such trust evidence. These issues are still important, and need to be addressed in a complete system.

Trust Representation

The DITD model represents trust on a weighted trust graph *G(V,E)* by a trust opinion. The trust opinion is a numeric trust variable which is a function of the available confidence and trustworthiness evidence.

$$trust_i(t_{evidence}, c_{evidence}) = t_i \in [0,5] \tag{1}$$

A high trust opinion means that the node is a good node, or that the node provides highly accurate location information, or that the certificate issued by the node is highly trusted. Trust is further influenced by network operation confidence. This includes the duration of a node's participation in the network, or the lack of negative evidence against the node.

The trust function, *trust*, computes the available evidence ($t_{evidence}$ and $c_{evidence}$) into a semantic numeric representation of trust. Trust is represented at each node or vertex of the trust graph. The work focuses upon the evaluation of routes and the assignment of trust to individual nodes is assumed to be taken care of by a network monitoring system.

A trust variable, t_i, will be assigned and stored at each node or vertex of the trust graph. Each node entering the network with a valid self-certificate is provided with a default trust value t_d. The DITD model can be extended to include access control and allow for a trusted outside member to assign trust values to nodes entering the network. This would allow for a more secure system with limited or specified users and still maintain the self organized nature of the network.

Trust is assigned to the established routes including both one-hop neighbouring routes and multi-hop routes. In an on-demand routing environment, nodes maintain a routing table storing the routes to each known node. A trust variable t_{AB} will be assigned to each of these routes representing the aggregated trust from the source node *A* to node *B*. The duration of time for which these routes are maintained securely will influence the weight of trust assigned to the edges of the trust graph. The trust of nodes (t_i) and trust of routes (t_{AB}) will change as the network progresses and new trust evidence is made available. The representation of trust is illustrated on weighted trust graph *G(V,E)* in Figure 7.

Certificate based trust provides the user with binary trust, i.e. when two nodes share certificates they trust each other; otherwise they don't trust each other. DITD represents trust with a range from 0 to 5. Where 0 represents a malicious node or a node not worthy of trust and 5 represent a full confidence in the certificate and trustworthiness of the node. This gives the trust graph system some flexibility.

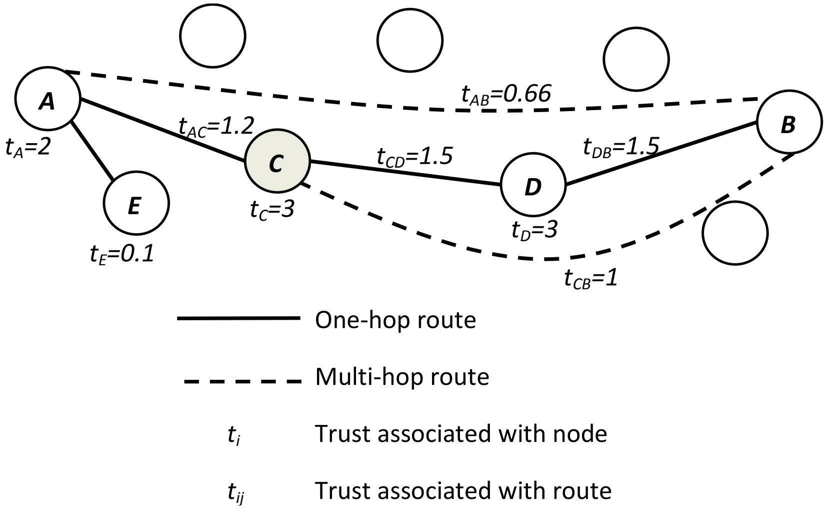


Fig. 7. Weighted Direct Trust Graph

When inconsistent data is shared then a trust accusation may be made against offenders reducing the trust of the node and the routes in which it participates. A proposal is made to use the route maintenance mechanism implemented by the on-demand routing protocol, to help establish the confidence and trust variables. The purpose of the route maintenance is to maintain the routes and to share neighbourhood information. This allows for provided trust evidence to be shared in a localized manner. This maintenance protocol allows nodes to "monitor" their neighbours and when inconsistent data is shared then a trust accusation may be made against the offender.

The DITD model inherits aspects of the semiring mathematical trust representation following semiring properties which are used for aggregating trust opinions along and across paths. The distance semiring operators $\oplus$ and $\otimes$ are applied to optimize trust accumulation. The $\otimes$ operator is used to add trust values along a trusted certificate chain. The $\otimes$ operator allows for a final trust value to be calculated representing a chain of nodes with different trust values. Trust values will be aggregated along a path like parallel resistors would be summed i.e. $\frac{1}{R_T}=\frac{1}{R_1}+\frac{1}{R_2}+...+\frac{1}{R_n}$. The trust will decrease along the path and the final trust of the path can be no larger than the lowest trust value. This aligns with the description of a trust chain which states that a chain is as strong as its weakest link. The distant semiring approach is based on Eiser proposal [Hu et al, 2002]. Figure 5 illustrates how trust is aggregated along a path and stored representing the trust of a specific route between node *A* and *B*.

In summary, methods are proposed to allow for a trust semantic but this is not the focus of the work. The assumption is made that trust evidence is available and that each weighted vertex has been assigned a trust value.

Trust Evaluation

The conduct based proposal compliments the certificate exchange and verification mechanism forming a hybrid security model which embraces certificate trust establishment as well as conduct based trust. Ideas from the modified proactive generic-single-source-shortest-distance algorithm [Theodorakopoulos & Baras, 2006] [Mohri, 2002] are inherited and we propose to apply this semiring mathematical formulae to the reactive on-demand trust path discovery phase of the routing protocol. The generic-single-source-shortest-distance algorithm calculates the shortest path from a source node to all nodes in the network, working in a proactive manner. The DTID proposal is a reactive path specific model. DITD will have a hand in the selection of the multi-hop routes therefore its operation will lie in the network layer. The DITD model modifies and optimizes the shortest path algorithm on the network layer. The following modifications are made to the trust path discovery phase as to compliment the certificate based model with a conduct based model evaluating trust along a path.

1. Trust is aggregated along the *RREQ* path

The distance semiring mathematic operator ⊗ [Mohri, 2002] is used to allow for trust to be calculated for a route from source node *S* through intermediate nodes *1* to *n* toward destination node *D*. Trust for this path is a function of the participating nodes.

$$trust_{SD}(t_S, t_1, t_2, t_3 ..., t_n, t_D) = t_S \otimes t_1 \otimes t_2 \otimes t_3 \otimes ... \otimes t_n \otimes t_D = t_{SD}$$
$$= \left(\frac{1}{\frac{1}{t_S} + \frac{1}{t_1} + \frac{1}{t_2} + \frac{1}{t_3} + ... + \frac{1}{t_n} + \frac{1}{t_D}} \right) \quad (2)$$

Trust is aggregated along the path that the *RREQ* propagates. The trust of the route is updated at every hop and the trust value is stored in the routing table of the intermediate nodes with respects to the level of trust of the reverse path to the source.

2. Trust is aggregated along the *RREP* path

Similarly to 1 the trust from the destination to the source is aggregated using the distance semiring formulae.

$$trust_{DS}(t_D, t_{n-1}, t_{n-2}, t_{n-3} ..., t_1, t_S) = t_D \otimes t_{n-1} \otimes t_{n-2} \otimes t_{n-3} \otimes ... \otimes t_1 \otimes t_S = t_{DS} \quad (3)$$

Although the total trust between the source and destination is already calculated after *RREQ*'s propagation, this step is necessary to provide appropriate trust values for the forward path recorded in the intermediate node's routing table.

The aggregation of trust is illustrated in Figure 8 where source node, *S*, sends a *RREQ* message to destination node *D*, and at each hop of the *RREQ* message the trust is calculated and stored as a trust value associated with the reverse route to *S*. The trust associations are t_{SP1} , t_{SP2} and finally t_{SD} which is the trust for the route between *S* and *D*. Figure 8 also follows the *RREP* message calculating the trust associated with the forward routes stored in the routing table. Figure 8 shows that trust is route specific and trust must be aggregated along both the *RREP* and *RREQ* paths.

3. Implicit revocation: Filter trust path discovery participation

The nodes that participate in the trust path discovery process must all have an acceptable value of trust. This therefore eliminates untrusted nodes from participating in the multi-hop routes; this also eliminates low level trust paths from being discovered. Before a *RREQ* message is processed and forwarded, the aggregated trust of the propagating route request is compared to a trust threshold t_{thresh}. If the trust is lower than the t_{thresh} then the *RREQ* message is discarded.

$$t_{RREQ} < t_{thresh} \tag{4}$$

This procedure will act as an implicit revocation mechanism for DITD. A trust chain is as weak as its weakest link, therefore if the weakest links are not considered then their corresponding weak trust chains are not considered either. This modification helps find the most trusted path and reduces unnecessary network computation and message propagation.

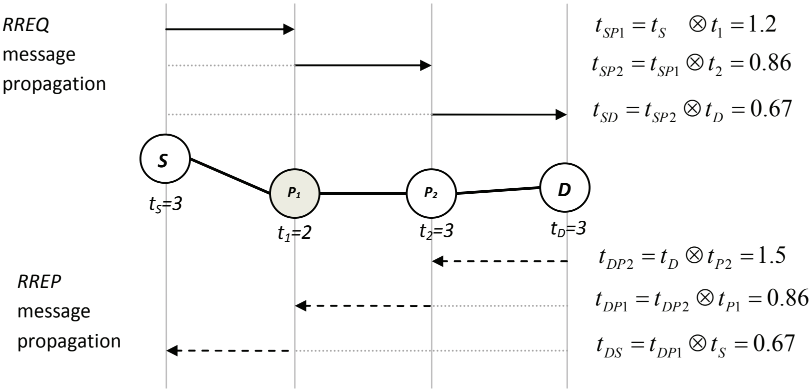

RREQ
message
propagation
$t_{SP1} = t_S \otimes t_1 = 1.2$
$t_{SP2} = t_{SP1} \otimes t_2 = 0.86$
$t_{SD} = t_{SP2} \otimes t_D = 0.67$
S
P1
P2
D
t_S=3
t_1=2
t_2=3
t_D=3
$t_{DP2} = t_D \otimes t_{P2} = 1.5$
$t_{DP1} = t_{DP2} \otimes t_{P1} = 0.86$
$t_{DS} = t_{DP1} \otimes t_S = 0.67$
RREP
message
propagation

Fig. 8. Illustration of trust aggregated along RREQ and RREP path

4. Filter the most trusted path

Figure 9 illustrates this step. When the *RREP* is propagated back to the source node, it is very possible that multiple routes are found therefore an intermediate node may receive more than one *RREP* message. In this case the first *RREP* is forwarded and successive *RREP*'s are only forwarded based on their sequence number or total trust value, effectively filtering the most recent and most trusted routes to the destination. The generic-single-source-shortest-distance algorithm would unicast *RREQ* messages in order of trust to their neighbours. By doing this, cyclic paths are avoided and the procedure of discovering the most trust path is maximised. The possibility of unicasting *RREQ* messages instead of broadcasting them is unfeasible for mobile ad hoc networks due to resource limitations. Instead DITD's proposal of filtering the routes by sequence number and trust will effectively realizes the relaxation process of the Dijkstra's shortest path algorithm in a reactive rather than a proactive manner.

The four additives to the trust path discovery phase allow for conduct trust evaluation to be added to the on-demand routing protocol of the ad hoc network increasing the security of trust chains created during indirect trust establishment. The conduct model is explained with an example.

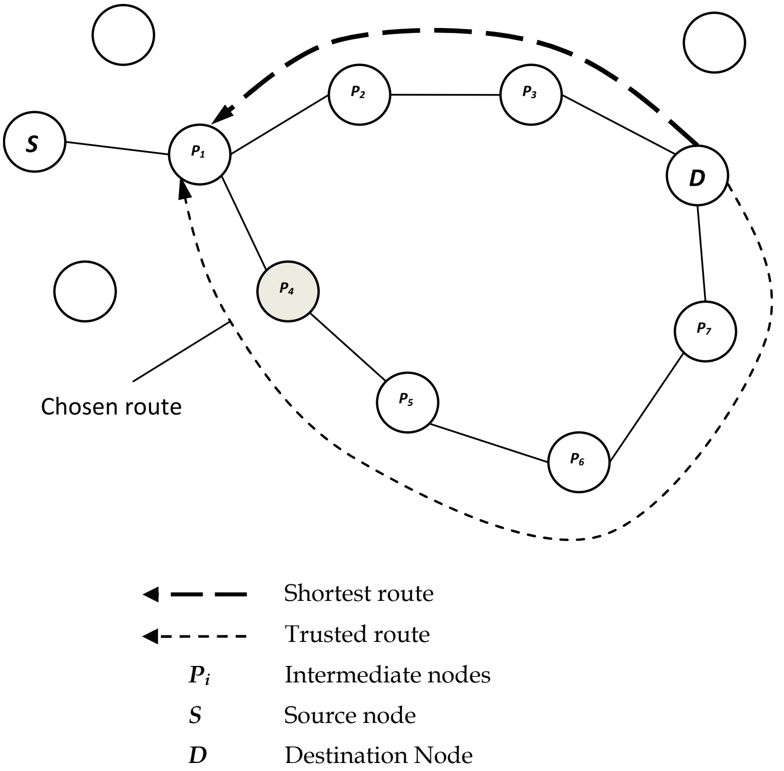


Fig. 9. Illustration of the filtering the most trusted route

To summarise this section, we can say that the section showed how the proposed hybrid trust scheme is incorporated into an ad hoc on-demand routing scheme with a low level complexity. Direct and indirect trust is established by localized one-hope certificate exchanges in a reactive manner and conduct trust is appended by aggregating trust along paths. The following section discusses the performance of the proposed scheme with use of simulations.

4. Performance and simulation study of the proposed DITD Model

There are two main approaches to evaluate routing applications for mobile ad hoc networks: simulations and real test beds [Kiess & Mauve, 2007] [Ke et al, 2000]. Real test beds can provide realistic results. However, they are impractical to set up. A real test bed, for a large network of nodes would requires 50 nodes in operation which is considerably costly. It is also difficult to compare different protocols because of the difficulty in repeating test conditions, such as mobility and erratic wireless connectivity. Therefore, real test beds are logistical unfeasibly. Currently, simulations are widely used to compare proposed routing protocols. Simulation packages like ns2 [http://,2007] and GloMoSim [Zeng et al, 1998] provide an environment to design and compare proposed and existing protocols. The majority of literature on this subject use ns2 as its enhanced functionality is suitable for wireless scenarios. The ns2 network simulator was selected to perform a simulation study for the DITD model.

This section presents the effects of adding the security functionality, proposed by the DITD model, to the AODV routing protocol. This functionality includes a certificate distribution mechanism and a trust evaluation mechanism. The environment investigated is a large mobile ad hoc network which uses an on-demand routing algorithm.

We use subsection 4.1 to: describe the simulation environment, discuss the simulation scenario, and introduce the traffic and mobility models. Subsection 4.2 describes the performance metrics used to analyze the simulated routing protocols. The focus of this section is found in Subsection 4.3 where a comprehensive simulation study is presented. This is done by comparing the proposed DITD model with the AODV routing protocol. Results are presented in simple line graphs and discussed accordingly.

4.1 Simulation setup

The goal of the simulation experiments is to measure the proposed routing protocol's performance to a changing network topology and network conditions. To measure this, protocols are simulated at varied mobility conditions. A comprehensive simulation study is presented of the proposed security scheme for mobile ad hoc networks implemented on the network layer. A summary of the simulation set used in our study is given in Table-5.

a. Simulation Scenario

The network was set up with 50 wireless nodes allowing data communication to occur in a peer-to-peer manner. Nodes are mobile in a rectangular space of 1500m x 300m and the simulation is run for 900 seconds. A rectangular area is preferred to a square area as longer routes can be expected. Nodes were configured to use the 802.11b standard communicating over wireless channels with a two-ray ground radio propagation model with a bandwidth of 2Mbps and a nominal transmission range of 250m.

Simulation Scenario	
Physical and MAC model	IEEE 802.11b standard
Nominal bit rate	2Mbps
Transmission Range	250m
Number of nodes	50 nodes
Simulation duration	900 seconds
Simulation area	1500m x 300m
Traffic Model	
Traffic type	CBR
Data packet size	64 byte
Traffic rate	4 packets per second
Traffic started	0 – 180 seconds
Number of connections and sources	30 and 20
Mobility Model	
Model	Random Waypoint
Max speed	0.1 , 1, 5, 10, 20, 30 m/s
Pause time	0 and 250 seconds

Table 5. Simulation Setup for varied topology

b. Traffic Model

Traffic was simulated using a constant bit rate (CBR) traffic generator which models UDP traffic. TCP traffic was not used because it uses its own flow control mechanism which schedules data packets based on the network's ability to carry them. CBR traffic is more useful for a routing protocol analysis as it allows the routing protocol to manage the flow of traffic. All traffic is started within the first 180 seconds of the simulation. Simulations were performed with data packets sizes of 64, 256, 512 and 1024 bytes. At higher data packet sizes traffic congestion causes a few nodes to drop most of their received packets, this was observed from test simulation runs. A data packet size of 64 bytes was selected for the simulation analysis. The focus of the simulation study is to compare the performance of routing protocols against changing topology and as no load balancing is employed in any simulated protocol, congestion is factored out by selecting a lower data packet size. The traffic analysis model is consistent with routing protocol analysis in [Broch, 1998].

For topology analysis the traffic load is fixed with a rate of 4 packets per second. The maximum number of connections is set to 30 connections with a traffic model with 20 sources.

c. Mobility Model

A modified "random waypoint" mobility model was used to prevent mobility concerns highlighted in [Navidi, 2004]. The modified random waypoint model improves upon the standard model by selecting a speed which is between 10% and 90% of the given maximum speed. This addition provides a more balanced mobility and prevents extreme drops in speed during simulation.

Changing network topology is simulated based on network participant speed. The maximum speed was varied from 0 to 30m/s with 6 different mobility patterns (0.1, 1, 5, 10, 20 and 30m/s) for two different pause time scenarios, 0 and 250 seconds, representing a network with continuous motion and a partially stable network.

4.2 Performance metric

The following quantitative metrics are used to analyze the performance of the routing protocols in mobile ad hoc networks.

a. Packet Delivery Ratio

The packet delivery ratio (PDR) represents the percentage of data packets that are successfully received by their intended destination. This metric is also known as throughput and is considered a measurement of the effectiveness of a routing protocol. The equation for PDR is:

$$PDR\% = \frac{\sum_1^n CBRrec}{\sum_1^n CBRsent} \times 100$$

where $\sum_1^n CBRrec$ and $\sum_1^n CBRrec$ are the number of CBR data packets received and sent respectively.

b. Routing Overhead

A routing protocol uses control packets to establish routes on which data packets are transmitted. Control packets are separate from data packets but share the same communication channel. Due to the lack of channel capacity in mobile ad hoc networks a large number of control packets can result in poor network performance. Key management would require additional control packets to achieve key management functionality this will be reflected in the simulations. The routing overhead is also known as a routing protocol's internal efficiency and will represent the number of control packets used for a given protocol.

c. Average End-to-End Delay

This is a qualitative measurement of the delay of data packets. The average end-to-end delay of a data packet is the time from which it is created at the source and when it arrives at the intended destination. The delay includes propagation and queuing delay. Delay can be caused by a high number of control packets propagating in the network or a high computational overhead for the given protocol. The average end-to-end delay is calculated as follows,

$$End\ to\ End\ Delay = \frac{\sum_1^n (CBRsendtime - CBRrecvtime)}{\sum_1^n CBRrec}$$

where *CBRsendtime* and *CBRrecvtime* represent the record times that a CBR data packet was sent and received.

4.3 DITD simulation

a. Implementation

A linux based server was set up to run the Network Simulator ns-2.31 [http://,2007]. A routing protocol was designed in C++ based on the AODV routing protocol available in the ns-2.31 package. The routing protocol DITD is programmed as a routing agent class. The routing agent handles the establishment of routes, certificate distribution and trust evaluation. Modifications are made to the AODV routing agent at the *RecvRequest, SendRequest, RecvReply*, and *SendReply* functions. These modifications allow for the distribution of separate certificate packets, triggered by the routing packets. The routing agent's packet header was modified to include a certificate control packet *CertS*. The size of the certificate included is 450 bytes which correlates with experiments in [Zapata, 2006]. The size of the certificate control packets is increased resulting in an effective delay in communication simulating the transfer of actual certificates. The authors of [Awerbuch et al, 2008] use a similar approach to simulate the effect of security processing. A certificate table is included at each node *CertTable* which is updated by certificate control packets. The certificate table is linked to the routing table and each node is responsible for managing its own certificate table.

The trust evaluation scheme assumes that monitoring trust evidence is available. Routing control packets are modified to include an associated trust variable. As each routing packet propagates through the network, the trust of the specific route is calculated and stored in the routing table of each node. Implicit trust revocation and trust path selection is performed at *RecvRequest* and *RecvReply* functions respectively.

A simulation *tcl* file is written to setup the mobile ad hoc network's desired simulation scenario, traffic and mobility model. The trace support files in ns-2.31 were modified to support the DITD routing agent allowing the inclusion of certificate control packets and trust information. As a result the output trace and *nam* files reflect the operation of the DITD routing agent. Figure 10 shows a sample output of the *nam* simulation file and Figure

11 shows a sample trace file output. AWK, an extremely versatile programming language for unix based systems, was used to write script files to analyze the trace data and provide the measured performance metrics. Finally unix based shell script files were written to allow for multiple iterations and simulation scenarios to be run simultaneously resulting in over 1000 simulation runs and 430 Gb of data analyzed and presented in simple line graphs.

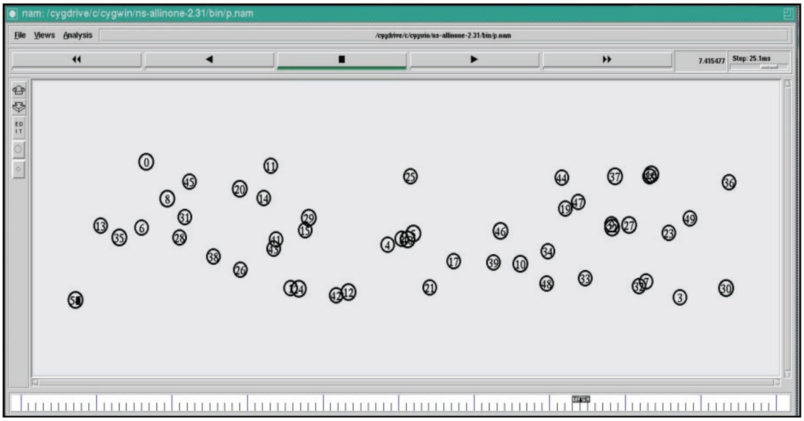

nam: /cygdrive/c/cygwin/ns-allinone-2.31/bin/p.nam
File Views Analysis
/cygdrive/c/cygwin/ns-allinone-2.31/bin/p.nam
7.415477
Step: 25.1ms

Fig. 10. Sample *nam* simulation file illustrating typical network topology

```
r 19.867 _9_ RTR  --- 0 AODV 60 [0 ffffffff 16 800] --- [22:255 -1:255 26 0] [0x2 5 1 [19 0] [17
20]] 20 (REQUEST)
r 19.867 _43_ RTR  --- 0 AODV 60 [0 ffffffff 16 800] --- [22:255 -1:255 26 0] [0x2 5 1 [19 0] [17
20]] 20 (REQUEST)
r 19.868 _25_ RTR  --- 0 AODV 508 [13a 19 27 800] --- [39:255 25:255 1 25] [0x14 [25 32] [39] 10
[0 0] 4 0 1] (CERT_R)
r 19.871 _13_ RTR  --- 125 cbr 84 [13a d 11 800] --- [17:3 19:0 30 13] [0] 1 3
f 19.8711 _13_ RTR  --- 125 cbr 84 [13a d 11 800] --- [17:3 19:0 29 39] [0] 1 3
r 19.8724 _26_ RTR  --- 0 AODV 508 [13a 1a 28 800] --- [40:255 26:255 1 26] [0x14 [26 28] [40] 10
[0 0] 4 0 1] (CERT_R)
r 19.8733 _39_ RTR  --- 125 cbr 84 [13a 27 d 800] ---[17:3 19:0 29 39] [0] 2 3
f 19.8733 _39_ RTR  --- 125 cbr 84 [13a 27 d 800] --- [17:3 19:0 28 19] [0] 2 3
r 19.877 _48_ RTR  --- 0 AODV 508 [13a 30 12 800] --- [18:255 48:255 1 48] [0x12 [48 30] [18]
10.00 [18 17] 1] (CERT_S)
```

Fig. 11. Sample trace file output for DITD simulation run

b. DITD Performance Results

The DITD model is compared with the AODV routing protocol. Further comparisons are presented against a conventional approach to key distribution. The simulation scenario used is described in Section 2.4.2 which is used throughout the simulation study. The traffic model simulates a moderate traffic load at a rate of 4 packets per second. The effects of changing topology are investigated by varying the node speed for a continuously moving network and a partially stable network. The simulation results were averaged over 10 speeds per scenario, resulting in a total of 360 iterations for the speed analysis.

Packet delivery

The packet delivery results for the AODV and DITD routing protocols are presented in Figure 12 and Figure 13. Figure 12 represents a simulation environment with a pause time of 0 seconds. This represents a network of nodes that are continually moving, while Figure 13 represents a partially stable network. The observation is made that as the speed increases both protocols throughput decreases. At high speeds the network topology changes rapidly causing breakages in routing links. The reduction in packet delivery at high speeds is because both protocols will drop data packets as a result of increased routing breakages. The curves for the AODV and DITD packet delivery ratio have similar shapes. This is expected because the DITD model is based on the AODV model. In Figure 12 the DITD model shows a 0–10% reduction gap in packet delivery when compared to the AODV model. The gap increases uniformly as the speed increases leveling at 10% for speeds of 20 m/s and higher. Similarly for the more stable network, presented in Figure 13, there is a reduction in packet delivery ratio of 0-5% when compared to the AODV model. The stable network in Figure 13 shows better performance at higher speeds because the number of route link breakages is reduced as a result of a larger pause time. A large pause time represents a network that will move at a given speed then pause in a fixed location for a set amount of time. During this time routing link breakages are not expected until movement commences again. The reduction in packet delivery ratio of DITD, when compared to AODV, can be attributed to the additional certificate packets being distributed and handled by the routing agent. The packet queue for the routing protocol has a limited capacity and when it is overloaded, packets are dropped. This will cause a resultant drop in throughput. The DITD model optimizes its throughput by processing the routing and certificate control packets independently of each other.

A certificate distribution scheme would expect a severe reduction in performance due to an excessive number of packets being transmitted in the network or the additional size of the control packet. A conventional certificate distribution scheme, suggested as a possible solution in [Buchegger & Boudec, 2002], simply includes the source certificate in the request packets *RREQ* and the destinations certificate in the reply packets *RREP*. This method was implemented as a separate routing agent *AODVcert* in ns2. A similar method is suggested in [Papadimitratos & Hass, 2002]. Implementation includes increasing the packet size of the

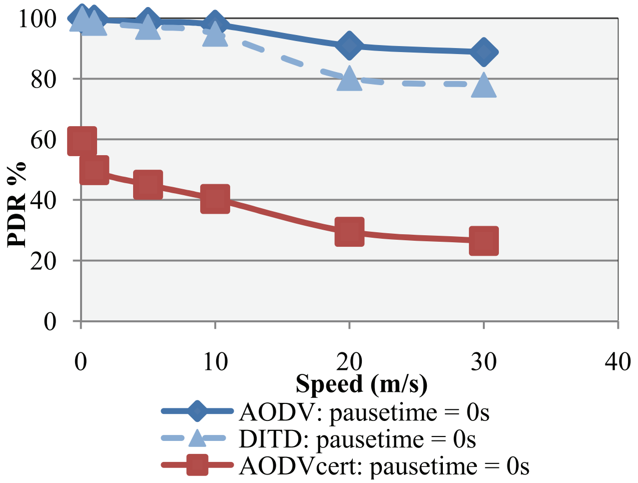


Fig. 12. Packet Delivery Ratio for highly mobile network (0 second pause time)

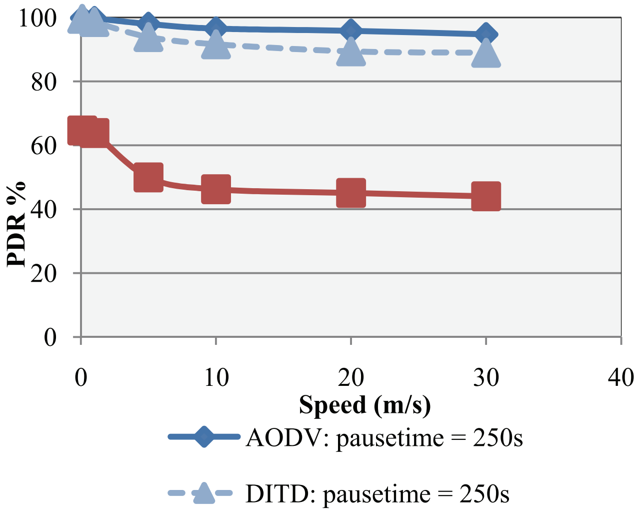


Fig. 13. Packet Delivery Ratio for partially stable network (250 second pause time)

routing control packets to include a 450 byte certificate. This effectively increased the regular 56 byte AODV route control packets to 506 bytes. Such an approach would result in the simplest method of certificate distribution but transmitting 450 bytes more data per control packet would severely reduce the network performance.

The *AODVcert* routing agent was simulated under the same simulation conditions as AODV and DITD, and the packet delivery ratio is presented in Figure 30 and Figure 31. It can be observed that the packet delivery ratio is severely less than both the AODV and DITD model. For a pause time of 0 seconds, there is an average gap of 55% between *AODVcert* and AODV and an average gap of 49% between *AODVcert* and DITD. Similar results are observed for the stable network in Figure 31. This simulation shows that DITD optimizes the distribution of certificates by sending them as separate certificate control packets independent of the route control packets. The certificate control packets are processed independently of the routing packets, allowing concurrent processing in a fully distributive system. The operation of DITD allows for certificate distribution with minimal effect upon the routing procedure.

During this time routing link breakages are not expected until movement commences again. The reduction in packet delivery ratio of DITD, when compared to AODV, can be attributed to the additional certificate packets being distributed and handled by the routing agent. The packet queue for the routing protocol has a limited capacity and when it is overloaded packets are dropped. This will cause resultant drop in throughput. The DITD model optimizes its throughput by processing the routing and certificate control packets independent of each other.

A certificate distribution scheme would expect a severe reduction in performance due to an excessive number of packets being transmitted in the network or the additional size of control packet. A conventional certificate distribution scheme, suggested as a possible solution in [Zapata, 2002], simply includes the source's certificate in the request packets *RREQ* and includes the destination's certificate in the reply packets *RREP*. This method was implemented as a separate routing agent *AODVcert* in ns2. A similar method is suggested in [Sanzgiri et al, 2002]. Implementation includes increasing the packet size of the routing control packets to include a 450 byte certificate. This effectively increased the regular 56 byte AODV route control packets to 506 bytes. Such an approach would result in the simplest method of certificate distribution but the result of transmitting 450 bytes more data per control packet would severely reduce the network performance. The *AODVcert* routing agent was simulated under the same simulation conditions as AODV and DITD and the packet delivery ratio is presented in Figure 12 and Figure 13. It can be observed that the packet delivery ratio is severely less than

Figure 14 shows that the DITD model has a 10% reduction in throughput for high speed mobile ad hoc networks. A high speed network is described by a maximum node speed of 20 and 30 m/s. This simulates mobile units travelling at a maximum speed of 70–100km/h which is typical of mobile military vehicles. Mobility aids the distribution of certificates as nodes come in close contact with each other and are able to establish direct trust relations reducing end-to-end certificate distribution. These benefits are similar to Capkun's solution which relies upon mobility to establish trust in a localized manner [Capkun et al, 2006]. Capkun's solution is aided by mobility but is also dependent upon mobility for trust relations to be established. Because of this dependency, a period of weakened security is expected as nodes exchange certificates. DITD does not only distribute certificates in a localized manner but Figure 30 shows that the DITD model has a 0 - 3% reduction in throughput for low speed mobile ad hoc networks where nodes move at a maximum speed of 0–10 m/s. This type of networks is typical of infantry units or a *nam* the ground scenario. DITD allows for mobility to aid the distribution of certificates but not relying upon mobility for throughput success. This allows DITD to operate successfully in slow moving and stationary type networks. The packet delivery ratio results show that DITD provides certificate distribution at a low performance cost for high speed networks and for low speed networks.

Control Packet Overhead

The control packet overhead presents a comparison between the AODV and DITD models. The overhead is presented in terms of the number control packets. The AODV model will have only routing control packets while the DITD model will have both routing and certificate packets. The results are presented in Figure 32 and Figure 15 for a highly mobile network with pause time of 0 seconds and a partially stable network with pause time of 250 seconds. The DITD model aims to distribute certificates while routes are discovered and a resultant packet overhead is expect. AODV and DITD are similar in shape and it is observed that the number of control packets increases as the speed increases. As the speed increases the topology of the network changes more rapidly causing routing link breakages and forcing nodes requesting communication to re-establish routes by send new route request messages. For a partially stable network presented in Figure 15 the effects of speed are reduced. This confirms that a larger pause time provides a more stable network. Figure 14 and Figure 15 show a consistent control packet overhead for the DITD model. It is observed that the gradient of DITD's packet overhead decreases as speed increases. This is because mobility aids certificate distribution and as the speed increases less certificate control packets are required. For example in Figure 14 at the low speed of 1 m/s there is a 132% increase in the number packets when compared to the AODV protocol. This overhead decreases for higher speeds showing a comparative 38% and 33% packet overhead for speeds of 20 m/s and 30 m/s respectively. This confirms that mobility aids certificate distribution.

A standard AODV request message is 48 bytes and a reply message is 44 bytes. The DITD model uses request message of 60 bytes and reply messages of 56 bytes. Therefore, DITD increases the routing control packet size by 12 bytes. DITD's routing control packets contain trust associated variables and flags to trigger back-tracked certificate distribution. The DITD certificate control packets are 508 bytes in size as they included a 450 byte certificate. It is noted that making the routing and certificate control packets separate and independent from each other has a greater impact on reducing the per byte packet overhead. This independency allows for concurrent processing of packets which is optimal in a fully distributive ad hoc network.

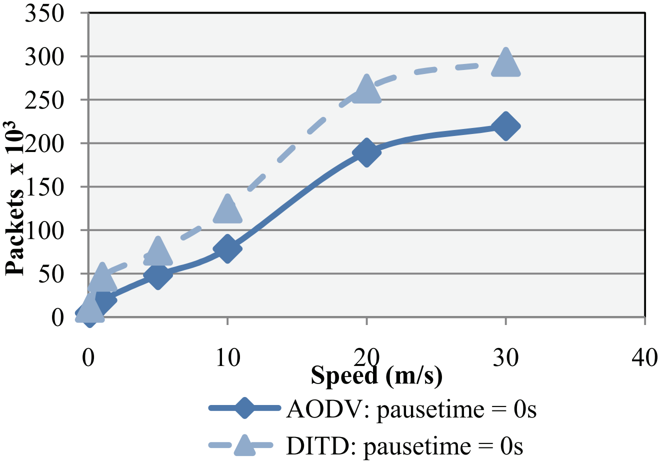


Fig. 14. Control packet overhead for highly mobile network (0 second pause time)

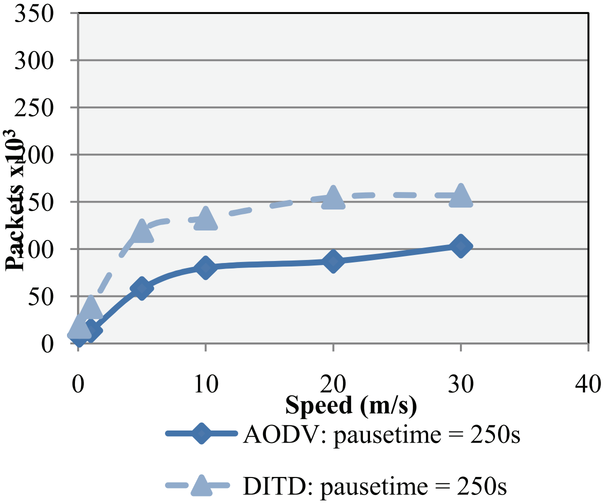


Fig.15. Control packet overhead for partially stable network (250 second pause time)

End-to-End Delay

The average end-to-end delay results are presented in Figure 16 and Figure 17. It is observed that the DITD model delivers packets with more delay than AODV. The additional delay is attributed to the transmission delay, the packet queuing delay, and the processing delay of additional certificate control packets. The processing delay includes verification. A conventional certificate distribution scheme that follows the route discovery process would require that certificates be verified before the routing packets are forwarded. DITD performs verifications independent of the routing procedure. The request route is established following the route request message *RREQ* to the destination and DITD performs verifications independently without hindering the propagation of the *RREQ* message.

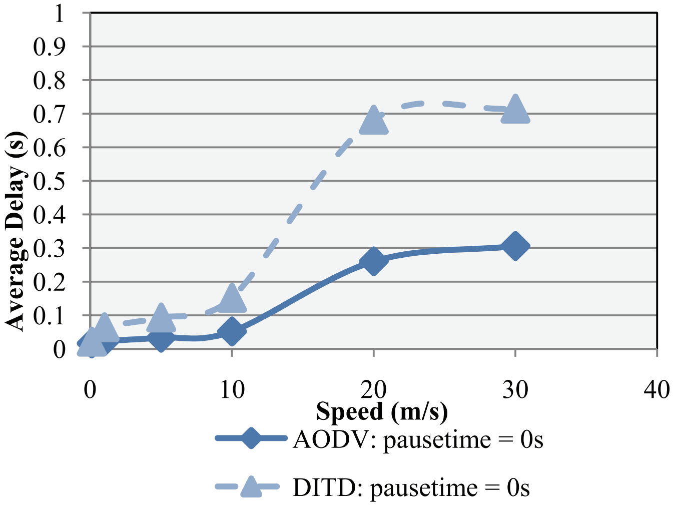


Fig. 16. Average end-to-end delay for highly mobile network (0 second pause time)

DITD uses back-track verification to minimize the number of verifications performed on the reply route which follows the reply message *RREP* toward the source. Hass and Pearlman [Haas & Pearlman, 2001] propose a solution which performs all verifications on the reply route. This method minimizes the nuns performed in a networks lifetime but results in delayed establishment of routes. If ECC (elliptic curve cryptography) type keys are used the verification process could take up to 16 ms per verification [Zapata, 2006] such a delay is unrealistic for multi hop routes requiring verification. DITD's approach attempts to minimize the delay incurred.

c. Trust Evaluation Results

In order to test the performance of the security evaluation scheme, a black hole attack was simulated to show that DITD's security evaluation scheme excludes malicious nodes from trust and route establishment protecting the network from black hole type attacks. A black hole adversary model was designed on the ns-2.31 link layer (LL) which lies below the routing layer. Modifications were made to the link layer agent *ll.cc* to simulate a black hole attack. Each packet sent by the routing layer is checked at the link layer, the adversary model silently drops all data packets while still allowing routing packets to be passed. This creates the affect of a black hole attack. A second black hole adversary model was implemented which includes a rushing type attack. The rushing attack was implemented by allowing adversary nodes to forward routing packets immediately, removing the small jitter delay that AODV implements. AODV uses this small delay to reduce the number of collisions and ensure the shortest path is selected. The rushing attack gives an adversary node a time advantage over normal nodes resulting in the adversary node becoming part of considerably more routes.

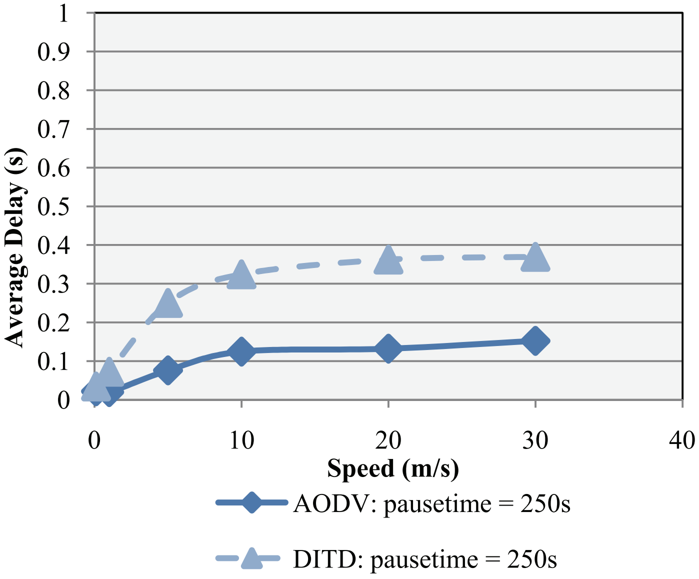


Fig. 17. Average end-to-end delay for partially stable network (250 second pause time)

The same simulation scenario and traffic model was used to analyse the black hole attack. The mobility was fixed with a pause time of 0 seconds and three speeds were investigated (0.1m/s, 5m/s and 20m/s). A 50 node network was simulated with 6 different attack scenarios. The attack scenarios were created by varying the number of black hole adversary nodes added by 0 to10. Figure 18 shows the *nam* simulation file for a simulation scenario with 10 adversary nodes. Each scenario was averaged over 10 seeds resulting in 720 iterations for the security evaluation scheme analysis. The black hole attack aims to drop data packets and reduce the networks throughput. The effects of a black hole and rushing attack are analysed using the packet delivery ratio performance metric.

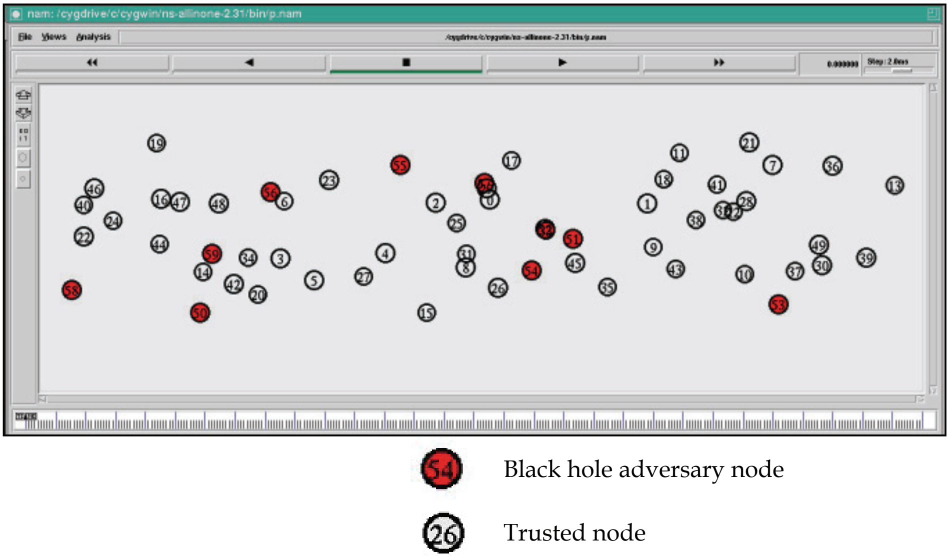


Fig. 18. Sample *nam* simulation of black hole network simulation

Packet delivery

A black hole type problem is implemented to simulate the success of DITD's security evaluation scheme. The scenario assumes weighted nodes carry a security metric which identifies fault detection or data transmission errors carried out by a monitoring system at each node. An example of such a system is found in [Buchegger & Boudec, 2002]. The weighted nodes are used to establish a weighted trust graph where each edge or route carries a trust calculated by DITD's security evaluation scheme. The effects of the black hole attack upon AODV and DITD are compared in Figure 37 and Figure 38. It is observed that as the number of adversary nodes increases the packet delivery ratio for the AODV model decreases. The AODV model is vulnerable to black hole attacks and in the presence of 10 adversary nodes the packet delivery ratio is below 65%. The reduction in throughput is expected as more data packets will be dropped by the presence of many adversary nodes. DITD avoids the adversary nodes by implicitly excluding these nodes during route establishment. The success of the protocol at low speeds is presented in Figure 19 and it is observed that even in presence of 10 adversary nodes the packet delivery ratio is not less than 90%. Figure 38 presents the success of the DITD model at a higher mobility of 20m/s. The DITD model prevents the severe effects of black hole attacks showing better results when 4 and greater than 4 adversary nodes are present. There is approximately a 10% decrease in packet delivery ratio when compared to the low mobility scenario in Figure 19. This reduction in packet delivery ratio is attributed to the increase in link breakages apparent at higher speeds and the overhead incurred from the certificate exchange protocol. The results of DITD in Figure 20 correlate to the packet delivery ratio at 20m/s in Figure 12.

A rushing attack was included for the simulations presented in Figure 21 and Figure 22. An adversary node equipped with a rushing type attack will participate in more routes maximising the effect of its attack. Figure 21 and Figure 22 show that when adversary nodes employ a rushing attack the effects of the black hole attack are maximised. The packet delivery ratio of the AODV protocol is dropped to 40% when 10 adversary nodes are present. This is considerably less when compared to the 60-65% packet delivery ratio that AODV experiences under the same conditions with a standalone black hole attack. The results of DITD under rushing attacks are unnoticeable when compared to DITD with no rushing attacks. For low speeds, DITD provides a throughput rate of above 90% even in the presence of 10 adversary nodes.

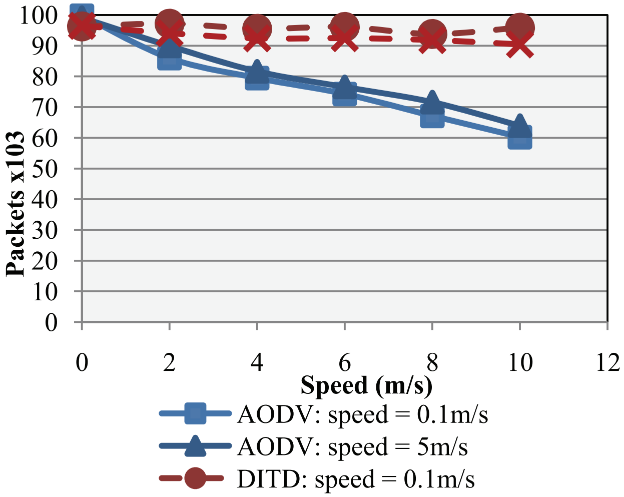


Figure 19: Packet Delivery Ratio for slow moving network under black hole attack

DITD provides a security scheme that excludes malicious nodes from participating in trusted routes, therefore preventing black hole attacks and a number of other attacks targeting the network layer. The inclusion of this trust evaluation scheme allows the distribution of certificates to operate in the most trusted routing environment.

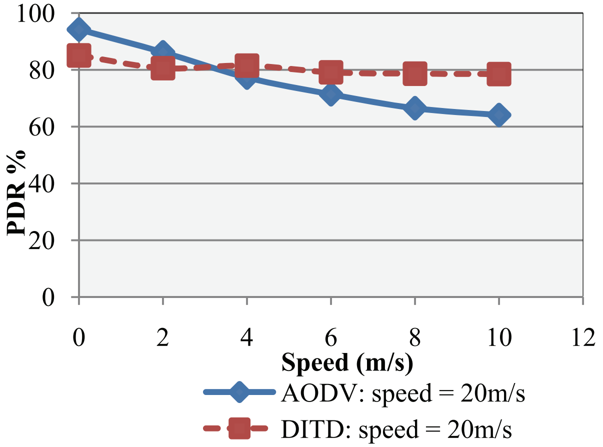


Fig. 20. Packet Delivery Ratio for fast moving network under black hole attack

4.4 Design verification

The DITD model, in relation to the design requirements stated in Section-2.3, will now be discussed. These requirements are based on the environment and functionality. The design requirements are briefly revisited throughout the discussion that follows.

a. Environment

The DITD model is required to operate on the network layer in an on-demand, fully distributive, self-organized manner. Implementation was performed on the network layer, which avoided multi-layer design problems. The simulation environment is set-up with no TTP member. This is similar to the way in which a certificate authority and network nodes are responsible for their own routing and trust establishment. The successful operation of DITD in the given environment is proven through simulation results, as presented in Section-6.

DITD is self-organized in nature. However, it is noted that DITD assumes the nodes are able to create their own keying material prior to joining the network. Self-certificates provide a strong binding between a user's key and a unique identity. The generation of keying material without the presence of a TTP is a complex problem. Solutions exist based on identity-based key generation [Shamir, 1984] [Weimerkirch & Westhoff, 2003]. The author suggests that further research in this area is carried out.

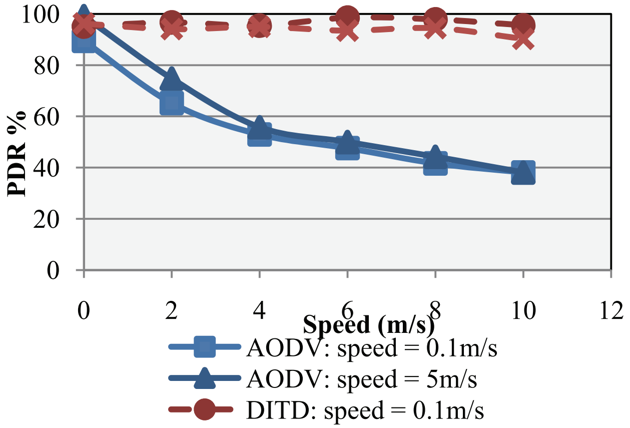


Fig. 21. Packet Delivery Ratio for slow moving network under black hole rush attack

b. Functionality

Certificate distribution is a requirement of the DITD model. DITD provides the distribution of keying material in the form of self-certificates. Local certificate exchanges are made between one-hop neighbors, which create direct trust relations. These direct trust relations are chained together to share certificates across multi-hop channels.

The DITD model assumes the existence of a weighted conduct value at each node. This allows the initial direct trust relations to have meaning. If this information is not available, direct trust relationship need to be established over a location-limited channel to ensure security, similar to infrared. Proximity based solutions are used in [Capkun et al, 2006] [Scannell et al, 2009]. DITD's simulation model assumes the availability of conduct information. Certificates are observed in the trace table as they are successfully transmitted to their desired destinations.

A second design requirement is that DITD must minimize the network overhead. The DITD model distributes certificates which use separate unicast certificate control packets. The certificates are triggered by the routing control packets. In comparison to AODV, DITD has an approximate 38% increase in control packets for highly mobile, high speed networks. The routing control packet size is increased by 12 bytes to include trust information and certificate control packets are 508 bytes in size. These packets result in a serve control packet overhead. The effects upon performance are reduced by: independency; concurrent processing; and back-track verification. Despite the significant control packet overhead, DITD merely reduces the packet delivery ratio by a 0-10% gap when compared to AODV. This reduction is notable if compared to a convention certificate distribution method, which increases the routing control packets by 450 bytes and results in over 50% reduction in packet delivery ratio. The performance of DITD is improved with more stable networks which have a higher pause time.

Simulations show that as the speed of nodes increase, the network performance decrease, as a result of a rapidly changing topology and increased link breakages. Simulations also show that mobility aids certificate distribution. However, DITD is not reliant on mobility and can still successfully operate in low speed and stationary type networks. This allows DITD to meet the requirement to provide secure communication at the start of the network lifetime. Solutions in [Capkun et al, 2006] [Tanabe & Aida, 2007] depend on mobility to establish trust and expect an initial time delay before trust is established. DITD provides secure communication in a reactive manner without a significant time delay. DITD is not limited by mobility, as it shows high throughput rates for low speed and stationary network environments.

DITD is required to be robust in spite of changing topologies. The simulations presented in Section- 6 were performed under varied pause times and speeds. This helped the investigation of the performance of DITD under varying topology environments. The simulation results show that DITD is robust in the presence of changing mobility, which will inherently have frequent routing failures. As mentioned above, DITD only reduces the throughput by a 0-10% gap across for changing topologies. It was observed that the DITD

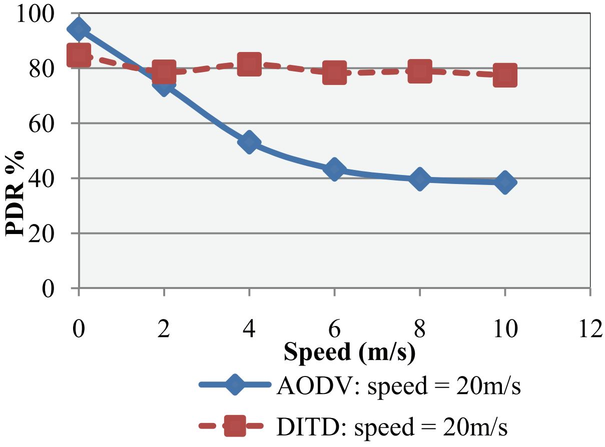


Fig. 22. Packet Delivery Ratio for fast moving network under black hole rush attack

model has an approximate 0.7 second end-to-end delay (0.4 seconds greater than AODV) for high speed, highly mobile networks. This indicates that DITD is not feasible to use for audio application, in highly mobile network environments. DITD's average end-to-end delay is reduced to 0.35 seconds (0.2 more than AODV) in a more stable network environment, which is within acceptable limits for audio application.

The last functional requirement was the inclusion of trust evaluation scheme. The trust evaluation scheme allows for the most trusted route to be selected and for malicious nodes to be excluded from route participation. The success of the scheme is present in its prevention against black hole attacks. Simulations show that a black hole attack of 10 adversary nodes causes a 35-40% reduction in packet delivery for the AODV routing protocol. DITD avoids black hole and rushing attacks by excluding malicious nodes. In low speed networks DITD achieves a 90-95% throughput rate in the presence of 10 adversary nodes.

5. Contribution and future work

5.1 Summary of contribution

Mobile ad hoc networks allow for a new set of applications that benefit from the dynamic, autonomous, and spontaneous mobile nature, inherent to these networks. However, the very qualities that make these networks so attractive also provide designers with new security challenges.

The focus of this work is upon trust establishment in mobile ad hoc network. This work contributes to the body of work in the following ways:

- Background knowledge on mobile ad hoc networks is presented. Their application in the military and commercial arena is investigated. A review of security attacks is present. Such attacks include: black hole attacks; wormhole attacks; eavesdropping attacks; byzantine attacks; resource consumption attacks; and routing table poisoning. The author identifies that mobile ad hoc networks are most vulnerable to network layer attacks and focus is placed on trust establishment on the network layer.
- Providing a comprehensive survey on the existing key management solutions for mobile ad hoc networks. The solutions are intended for different types of ad hoc networks and therefore their comparison is difficult. The solutions that are investigated are:
    - Off-line Trusted Third Party Models
    - Partially Distributed Certificate Authority
    - Fully Distributed Certificate Authority
    - Cluster based Model
    - Proximity-based Identification
    - Self Issued Certificate Chaining

    A discussion of the functionality and characteristics of each approach is presented. The self-issued certificate model is identified as providing the lowest level of pre-configuration and off-line trusted third party (TTP) involvement.
- A secure ad hoc routing survey. This work is vital to understanding trust establishment on the network layer. The following solutions are presented:
    - SEAD: Secure Efficient Ad Hoc Distance Vector Routing Protocol
    - Ariadne: A secure on-demand routing protocol for ad hoc networks

  - ARAN: Authenticated Routing for Ad Hoc Networks
  - SAODV: Secure Ad hoc On-demand Distance Vector (SAODV)
  - SLSP: Secure Link-state routing
  - ODSBR: On-Demand Secure Routing Byzantine Resilient Routing Protocol
  - CONFIDANT: Reputation based solution

  A comparative summary is presented focusing upon the security analysis and operational requirements of each solution. The Ariadne, ARAN, SAODV, OSRP and CONFIDANT are designed for on-demand ad hoc routing. All the protocols investigated, except the CONFIDANT protocol, assumption pre-existing key relationships or the presence of a key management system to perform the tasks of key distribution and maintenance. The CONFIDANT protocol avoids key management by establishing trust based solely on conduct. This part of the dissertation identifies an open research field in area of key management on the routing layer of mobile ad hoc networks.

- Presenting a novel security solution for mobile ad hoc networks. The solution is called Direct Indirect Trust Distribution (DITD) and is designed for an on-demand, fully distributive, self-organized, mobile ad hoc network. The scheme provides key distribution in the form of separate unicast certificate exchanges. The certificate exchange packets are independent from the routing control packets allow route establishment to operate concurrently but independently from trust establishment. A trust evaluation scheme is proposed that allows conduct based trust to influence to selection of routes and implicitly exclude malicious attacking nodes. This scheme allows the keying information to be distributed in a more secure manner.
- A comprehensive simulation study compares the performance of DITD and AODV, the protocol on which DITD is based. Simulation results show that under changing topologies DITD provides successful certificate distribution and trust evaluation with a minimal throughput reduction of 0-10%. Simulations show that DITD does not rely on mobility to distribute certificates and still performs in low speed communication networks. A black hole and rushing attack adversary model is designed on the link layer. Simulations show that DITD is successful in excluding malicious nodes from participating in route and trust establishment. The work simulation results and the discussions show that the proposed model can be implemented with low complexity and provides the functionality of key distribution and security evaluation with trivial effects on the network performance.

5.2 Future work

Future development will be made to enhance the DITD protocol, to further minimise the performance overhead. Future work includes the implementation of a load balancing agent to compliment and optimize the efficiency of DITD's key management.

The proposed model is not a standalone security solution. Future work includes the integration of the DITD scheme with a secure ad hoc routing protocol to realize a complete security system.

The key management tasks are key distribution, key generation, key maintenance and key revocation [Menezes et al, 1996b]. The DITD model addresses key distribution assuming that keys are generated by participating nodes. The generation of a secure certificate binding between a node and its public key is difficult without the presence of a trusted third party.

Furthermore, the effects adversary nodes with multiple identities performing Sybil attacks is a problem that is difficult to solve.

Trust evaluation schemes require that trust evidence be made available. Trust establishment is made up of the following services: gathering, generation, discovery and evaluation of trust evidence. This dissertation focuses upon the trust evaluation. Future work includes the gathering and interpreting of trust evidence by using local network monitors.

Mobile ad hoc cluster based networks has found increasing application in the military sector. Efficient and secure cluster based key management is a open research area to be investigated in the future.

6. References

[Abdul-Rahman, 1997] A. Abdul-Rahman, "The PGP trust model," *EDI-Forum: The Journal of Electronic Commerce,* vol. 10, pp. 27-31, 1997.

[Aram et al, 2003] K. Aram, K. Jonathan, and A. A. William, "Toward Secure Key Distribution in Truly Ad-Hoc Networks," in *Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03 Workshops)*: IEEE Computer Society, 2003.

[Awerbuch et al, 2002] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens, "An on-demand secure routing protocol resilient to byzantine failures," in *Proceedings of the 1st ACM workshop on Wireless security* Atlanta, GA, USA: ACM, 2002.

[Awerbuch et al, 2008 B. Awerbuch, R. Curtmola, D. Holmer, C. Nita-Rotaru, and H. Rubens, "ODSBR: An on-demand secure Byzantine resilient routing protocol for wireless ad hoc networks," *ACM Trans. Inf. Syst. Secur.,* vol. 10, pp. 1-35, 2008.

[Basagni et al, 2001] S. Basagni, K. Herrin, D. Bruschi, and E. Rosti, "Secure pebblenets," in *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking \&amp; computing* Long Beach, CA, USA: ACM, 2001.

[Broch, 1998] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking* Dallas, Texas, United States: ACM, 1998.

[Bruce, 2003] S. Bruce, *Beyond Fear: Thinking Sensibly about Security in an Uncertain World*: Springer-Verlag New York, Inc., 2003.

[Buchegger & Boudec, 2002] S. Buchegger and J.-Y. L. Boudec, "Performance analysis of the CONFIDANT protocol," in *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking \&amp; computing* Lausanne, Switzerland: ACM, 2002.

[Capkun et al., 2003] S. Capkun, L. Butty, and J.-P. Hubaux, "Self-Organized Public-Key Management for Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing,* vol. 2, pp. 52-64, 2003.

[Capkun et al, 2006] S. Capkun, L. Buttyan, and J.-P. Hubaux, "Mobility Helps Peer-to-Peer Security," *IEEE Transactions on Mobile Computing,* vol. 5, pp. 43-51, 2006.

[Chor et al, 1985] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch, "Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract)," *proc. 26th IEEE Annual Symposium on Foundations of Computer Science,* October, 21-23 1985.

[Davis, 2004] C. R. Davis, "A localized trust management scheme for ad hoc networks. ," *In: 3rd International Conference on Networking (ICN'04)*, pp. 671–675, 2004.

[Desmendt & Jajodia, 1997] Y. Desmedt and S. Jajodia, "Redistributing Secret Shares to New Access Structures and Its Applications," Department of Information and Software Engineering, School of Information Technology and Engineering, George Mason University, Technical ReportJuly 1997.

[Douceur, 2002] J. R. Douceur, "The Sybil Attack," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*: Springer-Verlag, 2002.

[Eschenauer & Gligor, 2002] L. Eschenauer and V. D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks," *proc. 9th ACM Conf. on Computer and Communication Security (ACM CCS'02)*, November, 17-21 2002.

[Frankel et al, 1997] Y. Frankel, P. Gemmell, D. MacKenzie, and M. Yung, "Optimal resilience proactive public key cryptosystems," *proc. 38th Annual Symposium on Foundations of Computer Science (FOCS '97)*, October, 19-22 1997.

[Grandison, 2003] T. Grandison, "Trust Management for Internet Applications," Imperial College London, 2003.

[Haas & Pearlman, 2001] Haas Z.J. and M. R. Pearlman, "The performance of query control schemes for the zone routing protocol," *IEEE/ACM Trans. Netw.*, vol. 9, pp. 427-438, 2001.

[Hu et al, 2002] Hu Y.C., D. B. Johnson, and A. Perrig, "SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks," in *Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications*: IEEE Computer Society, 2002.

[Hu et al, 2003b] Hu Y.C., A. Perrig, and D. B. Johnson, "Packet leashes: a defense against wormhole attacks in wireless networks," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*. vol. 3, 2003, pp. 1976-1986 vol.3.

[Hu et al, 2005] Hu Y.C., A. Perrig, and D. B. Johnson, "Ariadne: a secure on-demand routing protocol for ad hoc networks," *Wirel. Netw.*, vol. 11, pp. 21-38, 2005.

[http://2007] "The Network Simulator," *ver 2.31, Available at http://isi.edu/nsnam/ns/,* 2007.

[Johnson et al, 2001] Johnson D.B., D. A. Maltz, and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks," in *In Ad Hoc Networking, edited by Charles E. Perkins, Chapter 5*, 2001, pp. 139-172.

[Ke et al, 2000] Ke Q., I. David, D. Maltz, and D. B. Johnson, "Emulation of Multi-Hop Wireless Ad Hoc Networks," in *in The 7th International Workshop on Mobile Multimedia Communications (MoMuC*, 2000.

[Kiess&Mauve, 2007] Kiess W. and M. Mauve, "A survey on real-world implementations of mobile ad-hoc networks," *Ad Hoc Netw.*, vol. 5, pp. 324-339, 2007.

[Kscischang et al, 2001] Kschischang F.R., B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory,* vol. 47, pp. 498-519, 2001.

[Menezes et al, 1996a] Menezes A., P. van Oorschot, and S. Vanstone, *Handbook in Applied Cryptography*: CRC Press, 1996.

[Menezes et al, 1996b] Menezes A.J., S. A. Vanstone, and P. C. V. Oorschot, *Handbook of Applied Cryptography*: CRC Press, Inc., 1996.

[Mohri, 2002] Mohri M., "Semiring frameworks and algorithms for shortest-distance problems," *J. Autom. Lang. Comb.*, vol. 7, pp. 321-350, 2002

[Navidi, 2004] Navidi W., "Stationary Distributions for the Random Waypoint Mobility Model," *IEEE Transactions on Mobile Computing,* vol. 3, pp. 99-108, 2004.

[Papadimitratos & Hass, 2002] Papadimitratos P. and Z. J. Haas, "Secure Routing for Mobile Ad Hoc Networks," in *proc. SCS Communication Network and Distributed System Modeling and Simulation Conf. (CNDS'02)*, 2002.

[Papadimitratos & Hass, 2003] Papadimitratos P. and Z. J. Haas, "Secure Link State Routing for Mobile Ad Hoc Networks," in *Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03 Workshops)*: IEEE Computer Society, 2003.

[Perkins & Bhagwat, 1994] Perkins C.E. and P. Bhagwat, "Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers," *SIGCOMM Comput. Commun. Rev.*, vol. 24, pp. 234-244, 1994.

[Perkins et al, 2003] Perkins C., E. Belding-Royer, and S. Das, *Ad hoc On-Demand Distance Vector (AODV) Routing*: RFC Editor, 2003.

[Perrig et al, 2001] Perrig A., R. Canetti, D. Song, and D. Tygar, "Efficient and Secure Source Authentication for Multicast," Network and Distributed System Security Symposium (NDSS'01), 2001.

[Publications FIP, 2008] F. I. P. S. Publications, "Secure Hash Standard (SHS)," National Institute of Standards and TechnologyOctober 2008.

[Rivest, 1992] Rivest R., *The MD5 Message-Digest Algorithm*: RFC Editor, 1992

[Sanzgiri et al, 2002] Sanzgiri K., B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer, "A Secure Routing Protocol for Ad Hoc Networks," in *Proceedings of the 10th IEEE International Conference on Network Protocols*: IEEE Computer Society, 2002.

[Scannell et al, 2009] Scannell A., A. Varshavsky, A. LaMarca, and E. D. Lara, "Proximity-based authentication of mobile devices," *Int. J. Secur. Netw.*, vol. 4, pp. 4-16, 2009.

[Shamir, 1984] Shamir A., "Identity-Based Cryptosystems and Signature Schemes," in *proc. Advances in Cryptology: Crypto'84*, 1984.

[Stalling, 2003] Stallings W., *Cryptography and Network Security: Principles and Practices*: Prentice Hall, 2003.

[Stephan Eichler, 2006] Stephan Eichler C.R., "Challenges of Secure Routing in MANETs: A Simulative Approach using AODV-SEC," in *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, 2006.

[Tanabe & Aida, 2007] Tanabe M. and M. Aida, "Secure communication method in mobile wireless networks," in *Proceedings of the 1st international conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications* Innsbruck, Austria: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007.

[Theodorakopoulos & Baras, 2006] Theodorakopoulos G. and J. S. Baras, "On Trust Models and Trust Evaluation Metrics for Ad-Hoc Networks," *IEEE Journal on Selected Areas in Communications,* vol. 24, pp. 318-328, 2006 2006.

[Tseng et al, 2003] Tseng C.Y., P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, and K. Levitt, "A specification-based intrusion detection system for AODV," in *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks* Fairfax, Virginia: ACM, 2003.

[Weimerkirch & Westhoff, 2003] Weimerskirch A. and D. Westhoff, "Identity Certified Authentication for Ad-hoc Networks," in *proc. 1st ACM workshop on Security of ad hoc and sensor networks*, 2003.

[Zapata, 2002] Zapata M.G., "Secure ad hoc on-demand distance vector routing," *SIGMOBILE Mob. Comput. Commun. Rev.,* vol. 6, pp. 106-107, 2002.

[Zapata, 2006] Zapata M.G., "Key management and delayed verification for ad hoc networks," *J. High Speed Netw.,* vol. 15, pp. 93-109, 2006.

[Zeng et al, 1998] Zeng X., R. Bagrodia, and M. Gerla, "GloMoSim: a Library for Parallel Simulation for Large-scale Wireless Networks," in *proc. 12th Workshop on Parallel and Distributed Simulations (PADS '98)*, 1998.

26

Data Delivery in Delay Tolerant Networks: A Survey

Shyam Kapadia[1], Bhaskar Krishnamachari[2] and Lin Zhang[3]
[1]Cisco Systems Inc., San Jose, CA
[2]Department of Computer Science, Department of Electrical Engineering, University of Southern California, Los Angeles, CA
[3]Department of Electronic Engineering, Tsinghua University, Beijing
[1,2]USA
[3]China

1. Introduction

Delay-Tolerant Networks (Fall (2003)), also called disruption tolerant networks (DTNs), represent a fairly new networking paradigm that allows inter-connection between devices that current networking technology cannot provide. There are a wide variety of networks where an end-to-end connection between a given source and destination may never be present. Consequently, traditional routing protocols cannot be directly applied in these scenarios for delivering data. However, if one were to take the graph formed by the nodes based on their connectivity dictated by their radio range and consider the overlap not only over space but also time then there is a high likelihood that the network will appear as a single connected component. So while at any given instant, the network may not be connected, it may still be possible to route data from a source to a destination. DTNs are sometimes also called Intermittently-Connected Mobile Networks (ICMNs). The primary goal in such networks is to *get* the information from a source to the destination; these networks can tolerate a relatively higher delay.

A wide variety of "challenged" networks fall under this category ranging from outer-space networks, under-water networks, wireless sensor networks, vehicular networks, sparse mobile ad-hoc networks etc. Students moving about in a college campus (Hsu & Helmy (2006)), or buses moving about in a small metropolitan area (Burgess et al. (2006)), or a wireless sensor network with some mobile nodes (Shah et al. (2003); Juang et al. (2002)) acting as relays to assist in the data-collection phase provide representative examples of DTNs.

This chapter strives to provide a survey of some of the most relevant studies that have appeared in the domain of data delivery in delay tolerant networks. First, we introduce some fundamental challenges that are unique to DTNs. Then we present the major parameters of interest that various proposed routing solutions have considered, examples include end-to-end delay, throughput, mobility model of the nodes, energy efficiency, storage etc. Subsequently, we provide a classification of various approaches to routing in DTNs and pigeon-hole the major studies that have appeared in the last few years into the classified categories.

2. Challenges

In Delay-tolerant networks, at any given time instant, the network may not be connected. Data is delivered in a DTN using a store-carry-forward model. Nodes in the network relay data from source to the destination, where existing nodes in the network relay the data from the source to the destination, in one or more hops, such that each node along the path receives the data from the previous node and stores it locally. This node then carries the data for a while, and upon contact with other nodes, forwards the data. In this way, the data is finally delivered to the destination.

Whenever two nodes are in the vicinity of one another, they may exchange data, such an opportunity is termed as a *contact* or *encounter*. In other words, a link is established between these pair of nodes. This link is time-sensitive in that it is only valid for the duration when the nodes are in range of one another. If one or both nodes move away, then this link is broken. Moreover, at a time, there can be multiple links between a pair of nodes. For example, in case of 2 cell phones in vicinity, there can be a high-bandwidth peer-to-peer link (WiFi, IEEE 802.11 a/b/g) as well as a low bandwidth (EDGE/GPRS) link present simultaneously. In that sense, the connectivity of a DTN can be modeled as a time-varying multigraph. In the following, we enlist some of the unique challenges present in DTNs as compared to traditional networks.

2.1 Encounter schedule

In order to deliver data from a given source to a destination, the source node can wait till it encounters the destination node and then deliver the data directly to it. However, depending on the particular setting, this may take a long time and may not even happen. If the source node was an oracle and *a priori* it had information about the encounters between every pair of nodes, then it can pre-calculate and determine the best path or best set of nodes to forward its information in order to reach the destination node (Jain et al. (2004); Ghandeharizadeh et al. (2006)). In most practical scenarios, the schedules of encounters may not be known *a priori*. Even if the schedules are known to some extent, there may be errors and consequently, routing should be able to adapt and still deliver data to the destination. In the extreme case, where the mobility pattern of the nodes is random leading to memoryless encounter schedules, no assumptions can be made about the node contact pattern. Hence, the mobility model of the nodes is an important parameter that determines how the nodes will encounter one another. While a random walk based mobility model has been considered in a number of DTN studies due to its amenability to analysis, DTNs comprising vehicles or students have been shown to follow a community-based mobility model (Hsu & Helmy (2006)).

2.2 Network capacity

In general, the duration of an encounter as well as the link bandwidth dictate the amount of data that can be exchanged between a pair of nodes. Another factor is contention in the presence of multiple nodes trying to send data during a given encounter. This may also determine whether a message from a source to a destination needs to be fragmented.

2.3 Storage

During an encounter, nodes may decide to exchange all their information. However, if the nodes are storage-constrained, eventually, the node buffer will be exceeded resulting in data loss. Consequently, the naive approach of exchanging all data on an encounter may not scale or be applicable in all application settings. Intelligent schemes that restrict the number of copies of a given data item in the DTN, as well as schemes that trigger deletion of stale data (data already delivered to the destination of interest) are needed to efficiently utilize node storage. If the network is formed of nodes that have heterogenous capacities where some nodes are more powerful and less resource-constrained compared to others then this can be leveraged to design a better data delivery strategy for such a DTN.

2.4 Energy

DTNs span a wide spectrum of application settings. Transmission and reception of data as well as computation incurs power. In some settings, such as battery operated wireless sensor networks, the resources may be highly constrained where it is important to take into account the residual energy of a node while determining whether to exchange data during an encounter. However, in other settings, such as vehicular networks, the constraints on power may not be as severe. Data delivery techniques for DTNs should be able to adapt to such a wide range of scenarios.

3. Metrics of interest

The vast majority of the routing schemes for delay tolerant networks aim at optimizing a few metrics that affect their system performance. These are summarized below.

3.1 Message delivery ratio

This metric captures the number of successful deliveries in a DTN. In other words, how many packets (or messages) generated by various sources were delivered to their intended destinations in the network setting under consideration. Note that a message may be associated with a delivery deadline. If this message is not delivered within an acceptable amount of time specified by this deadline then it is considered a failed delivery. A modified definition of the delivery ratio is the fraction of the messages correctly delivered to their destinations within a specified period.

3.2 Delay

While the applications are able to tolerate larger delays in a DTN, as long as packets are delivered to their intended destinations, this is a metric of interest which should be optimized. Most DTN routing approaches aim to optimize both the delivery ratio as well as the delay. Consider an example scenario in a college campus where a professor wishes to broadcast a change in the timing of a lecture to all students or an executive trying to communicate the change in the time of an upcoming meeting. In both cases, the message is only valid if communicated before the start of the event (lecture or meeting). Consequently, while the delay in DTNs does not need to be instantaneous, the goal should be to keep it as short as possible subject to resource constraints.

3.3 Number of replicas

The efficiency of a data delivery mechanism generally improves as additional copies of a packet are generated and transported by various relays. However, the increase in the probability of data delivery comes at the cost of increase in the storage requirement at the individual nodes of a DTN. Hence, the number of replicas is an auxiliary metric that accompanies the delay and packet delivery ratio to provide an all-round indication of the performance of a given data delivery mechanism in a DTN.

3.4 Energy/Power

Usually the energy expended to achieve a given data delivery ratio and average delay is a function of the total number of transmissions and receptions incurred by all the participating nodes. This should include the energy expended due to idle receptions as well as computation (for example, aggregation etc.). Most studies employ the number of packet transmissions as an indicator of this metric. This metric is sometimes difficult to quantify especially in cases where nodes have heterogenous resources. Also, energy may not be a big concern in some application scenarios such as in the case of vehicular networks.

4. Data delivery mechanisms

In this section, we have classified routing schemes for DTNs into a small number of categories based on their characteristics.

4.1 Epidemic routing schemes

One of the earliest and probably the simplest protocols proposed for data delivery in DTNs is epidemic routing (Vahdat & Becker (2000)). The idea is whenever two nodes encounter one another they will exchange all the messages they currently carry with each other. At the end of the encounter, both will possess the same set of messages. As this process continues, eventually, every node will be able to send information to every other node. So the packets are basically flooded through the network much like the spread of a viral epidemic. This represents the fastest possible way in which information can be disseminated in a network with unlimited storage and unlimited bandwidth constraints. This scheme requires no knowledge about the network or the nodes. However, in most practical scenarios, such a scheme will result in inefficient use of the network resources such as power, bandwidth, and buffer at each node. Moreover, messages may continue to exist in the network even after they have been delivered to the destination. Epidemic routing serves as the baseline for comparison for most of the DTN routing schemes.

Davis et al. (2001) improved the basic epidemic scheme with the introduction of adaptive dropping policies. They restrict the size of the buffer at each node so that it can only store the top *K* packets that are sorted in accordance with a dropping policy. They explore four types of drop strategies, including Drop-Random (DRA), Drop-Least-Recently-Received (DLR), Drop-Oldest (DOA) and Drop-Least-Encountered (DLE). Their simulation results show that DLE and DOA yield the best performance. DLE seeks to drop packets based on information about node location and movement while DOA drops packets that have been in the network the longest relying on the premise that the globally oldest packets are the ones that are likely to have already been delivered to their intended destinations.

Harras et al. (2005) propose a set of strategies for controlled flooding in DTNs. These include schemes that have a Time-To-Live (TTL) as well as an expiry time associated with every message. In addition, once a message is delivered to the destination, a healing process is started to ′cure′ the network of the stale copies of this message. This is similar to the concept of ″death certificates″ proposed earlier in the context of replicated database maintenance

(Demers et al. (1987)). All these improvements reduce the resource consumption of epidemic routing while having little impact on the average delivery delay. An aggressive death certification scheme has been shown to reduce the storage required at each node (Small & Haas (2005)) but the tradeoff is that such a scheme will consume more transmissions (Harras & Almeroth (2006)) although it can be used to provide a notion of reliable message delivery in DTNs.

4.2 Direct-contact schemes

This data delivery scheme is one of the simplest possible where a source delivers a packet to a destination when it comes in direct-contact. In other words, the source waits till it comes in radio range of the destination and then directly delivers the packet to the same. This scheme does not consume any additional resources and makes no additional copies of the data. However, the major limitation is that the delivery delay can be extremely large and in many cases the source and the destination may never come in direct-contact of each other.

Perhaps the earliest incarnation of direct-contact based delivery schemes for DTNs is the well-known infostation model (Frenkiel et al. (2000)). The idea is that infostations are deployed at certain locations providing smaller “islands” of coverage which service the needs of data-intensive mobile nodes as they pass by. This approach serves to maximize the capacity of wireless data systems while reducing the cost of the services provided. The authors present a capacity-delay-cost trade-off for the infostation model for both one-dimensional and two-dimensional systems. In wireless sensor networks, a wide variety of application scenarios involve mobile sink nodes collecting sensed data from sensors deployed in a field. The sensors themselves may be static or mobile and are independent sensing entities. In ZebraNet (Juang et al. (2002)), data sensed by sensors attached to zebras is collected by humans as they drive by in a vehicle. In the context of vehicular networks, Kapadia et al. (2009) have also employed direct-contact based data delivery. They present comparative performance of a family of replication strategies that determine the number of replicas for a given data item based on its popularity.

Shared Wireless Infostation Model (SWIM Small & Haas (2003)), represents a hybrid scheme that extends the concept of an infostation through information sharing between nodes. The idea is that the nodes, in this case sensors attached to whales, collect data that is shared among themselves via replication and diffusion employing an epidemic routing like scheme when two sensors are in the vicinity of one another. Subsequently, when the whales come to the surface, the collected data is relayed to a small number of static on-shore base-stations. By allowing the sensor nodes to share data, the capacity requirements at the individual nodes goes up; however, the delay until one of the replicas reaches an infostation reduces. The authors examine this fundamental capacity delay tradeoff in the context of a real-world application.

4.3 One-hop relay schemes

In this scheme, the source delivers a packet to an intermediate node, aka relay, which in turn delivers the same to the destination. Compared to direct-contact, this scheme only incurs an overhead of one additional copy of a packet. A large number of application scenarios have employed this scheme for successful data delivery. The mobility of the relay node may be controlled or random. With Data Mules (Shah et al. (2003)), intermediate carriers that follow a random walk mobility model are used to carry data from static sensors to base-stations.

The individual sensor nodes transfer their data to the mule when it comes in radio range and the collected data is in turn delivered to the sinks. The study shows that by increasing the buffer capacity of the mules, fewer mules can service a sensor network albeit at the cost of a higher data delivery delay.

In DakNet (Pentland et al. (2004)), vehicles loaded with Mobile Access Points (MAPs) are used to transport data between village kiosks and centralized internet hubs. This represents one of the earliest practical applications of deploying wireless technology, specifically IEEE 802.11, also documented as the first national e-governance initiative in India related to computerizing land records in rural areas. Message Ferries (Zhao & Ammar (2003)) capture a more generalized scenario where the movement of the ferries can be controlled to carry data from a source node to a destination node. The initial proposal for ferries assumed that the nodes had limited resources, were stationary, and consequently were not burdened with the routing functionality. However, in follow-up works, the authors (Zhao et al. (2004; 2005)) extend the scheme to networks with mobile nodes and multiple ferries. This scheme requires online collaboration between the ferries and mobile nodes. The nodes need to proactively move so as to intersect with the path chosen by the ferries to transfer data to the latter. This assumption in turn was relaxed in a recent study (Bin Tariq et al. (2006)) where the message ferry routes were designed based on the mobility model of the nodes and probabilistic node locations.

4.4 Routing based on knowledge oracles

Jain et al. (2004) present a family of algorithms for routing in delay tolerant networks based on the presence of knowledge oracles. They model the DTN as a directed multigraph with time-varying edge costs, based on propagation delay and edge capacity. The various knowledge oracles considered provide information about the following (a) all future contacts of nodes such as time of contact, duration of contact, bandwidth available for information exchange during contact, (b) the future traffic-demand of the nodes, (c) the instantaneous queue sizes at each node. Using information from one or more oracles, various algorithms have been designed to send data from a source to a destination along a single path using either source-routing or local-per-hop routing. The authors have extended Dijkstra's shortest path algorithm to use time-varying edge costs. The performance of algorithms has been evaluated via simulations using a discrete-event simulator. The authors also present a linear programming formulation that uses all the oracles to determine the optimal routing for minimizing average delay in the network. The solution to this optimization serves a base-line optimum. The results indicate that as algorithms are fed more knowledge from the oracles, they provide better performance. However, in most practical settings, where the future traffic demand and global instantaneous queue knowledge may not be easily available, algorithms making per-hop decisions based on local knowledge can route around congestions and provide a good performance.

In reality, complete knowledge of contact schedules may not always be available. Additionally, the schedules may be imprecise and unpredictable. Jones et al. (2005) extend some of the algorithms presented above to compute the edge costs based on a sliding window of observed connectivity. They argue that an approach that defers the routing decision as late as possible thereby allowing forwarding based on the most recent information is better suited for DTNs. They introduce the concept of per-contact routing where nodes frequently recompute their routing table, similar to a traditional link-state routing protocol, whenever contact is made with another node. This routing information is then redistributed through the network using an epidemic routing like protocol thereby allowing nodes to take advantage of opportunistic connectivity and recompute routing for each message stored in the message buffer. The authors show that this scheme shows superior performance compared to epidemic routing as well as other schemes employing wireless LAN traces of a student population collected from a college campus.

A variant of the earliest-delivery algorithm proposed above, has been employed in the context of data delivery in vehicular networks by the Zebroids (Ghandeharizadeh et al. (2006)) study. The idea is that the source has knowledge of the contacts between the vehicles for a certain limited duration in the near future and based on this schedule, it determines the delivery path of the packet via one or more carrier vehicles. The vehicles themselves have storage constraints. Consequently while accepting a packet from its predecessor, if the vehicle's buffer is full, it employs a replacement policy to determine which packet must be evicted to accommodate the new one. The authors evaluate a wide variety of replacement policies and conclude that a policy that decides eviction candidates randomly provides competent performance. This study also validates the performance of the proposed scheme based on real-world encounter traces gathered from a small bus network in and around a college campus (Burgess et al. (2006)).

Approximate knowledge of the trajectory of the nodes has also been employed to deliver data in dynamic disconnected ad-hoc networks (Li & Rus (2000)). Given this information, the authors present an algorithm to pro-actively change the trajectory of intermediate nodes in order to deliver data between hosts. The goal is to minimize trajectory modifications while getting the message across as fast as possible. The authors present an analytical framework to prove the optimality of their proposed optimal relay path calculation algorithm.

4.5 Location-based schemes

In certain scenarios, the nodes may be aware of their location which can be used for opportunistic forwarding in DTNs. The location information may be known in either a physical (for example, from GPS devices attached to nodes or through a location service) or a virtual coordinate space (designed to represent network topology taking obstacles into account). On an encounter, a node forwards data to another node only if it is closer to the destination. Hence, location-based routing is a form of greedy, geographical-based routing (Takagi & Kleinrock (1984)). This minimal information is enough to perform routing and deliver data to the destinations. Hence, location-based schemes are fairly efficient in that they avoid the need to maintain any routing tables or exchange any additional control information between the nodes. These schemes have a well-known limitation where they suffer from a local minima phenomenon. Approaches such as perimeter forwarding (Karp & Kung (2000)) have been suggested to address this limitation.

The MoVe scheme (LeBrun et al. (2005)) employs information about the motion vectors of the mobile nodes in addition to the location information to perform routing in DTNs. Given the location and relative node velocity information, the scheme calculates the closest distance a mobile node is predicted to get to the destination when following its current trajectory. So a node only forwards to a neighbor if the neighbor is predicted to be moving toward the destination and getting closer to the destination than itself. The location-based routing algorithms are shown to outperform others based on realistic mobility traces obtained from GPS data collected from buses in the San Francisco MUNI system.

Leguay et al. (2006; 2005) propose a framework for routing in DTNs, called MobySpace, where each node is represented by a point in a multi-dimensional Euclidean virtual space.

Routing is done by forwarding messages toward nodes that have mobility patterns that are more and more similar to the mobility pattern of the destination. The authors demonstrate the feasibility of this framework through an example in which each dimension represents the probability for a node to be found in a particular location. Real world mobility traces (Henderson et al. (2004); Balazinska & Castro (2003)) of users show that the distribution of the probabilities of visit to locations as well as session durations generally follow a power law distribution. This property can be efficiently utilized by such a routing scheme. The results show that this scheme can bring benefits in terms of enhanced message delivery and reduced communication costs when compared with epidemic routing.

4.6 Gradient-based schemes

In gradient-based routing, the message follows a gradient of improving utility functions toward the destination thereby delivering the packet with a low delay and using minimal system resources. One of the early proposals, PROPHET (Lindgren et al. (2003)), employed probabilistic routing using history of encounters of the node and transitivity. This strategy was designed to take advantage of the non-random mobility behavior of the nodes as is the case in typical real-world scenarios. The idea is that each node is associated with a metric that represents its delivery predictability for a given destination. When a node carrying a message encounters another node with a better metric to the destination, it passes the message to it. The metrics are positively updated based on recent node encounters and metrics for sparsely encountered nodes are appropriately aged. The connectivity information is exchanged periodically among the nodes thereby allowing nodes to maintain meaningful metrics. As nodes run out of memory, the eviction candidate is selected based on a FIFO strategy although more intelligent eviction strategies have also been studied. The PROPHET strategy has been shown to have superior performance as compared to epidemic routing in case of a community mobility model.

Other researchers have proposed similar strategies in the case of ad-hoc networks using other kinds of information to calculate the gradient metric such as age of last encounter (Grossglauser & Vetterli (2003)), history of past encounters and the encounter rate (Nelson et al. (2009)), etc. Gradient based routing is also sometimes called adaptive routing (Musolesi et al. (2005)) since the metrics used for routing decisions essentially capture the context information of the nodes such as the rate of change of connectivity of a host (i.e., the likelihood of it meeting other hosts) and its current energy level (i.e., the likelihood of it remaining alive to deliver the message). Context is defined as a set of attributes that describe the aspects of the system that can be used to optimize the process of message delivery. The authors have introduced a generic method that uses Kalman filters to combine and evaluate the multiple dimensions of the context of the nodes to take routing decisions.

The Shortest Expected Path Routing (SEPR) is another scheme based on the link probability calculated from the history of node encounters (Tan et al. (2003)). Each message in a nodes cache is assigned an effective path length (EPL) based on the link probabilities along the shortest path to the destination. A smaller EPL value indicates higher delivery probability. When two nodes meet, they first exchange the link probability table and employ Dijkstra algorithm to get expected path length to all other nodes in the network. This novel EPL metric is employed for message forwarding as well as replacement when node buffer is full. This algorithm is similar to a traditional link state routing protocol in that nodes update their local tables on an encounter and in this way connectivity information is maintained in the network in a distributed manner. Simulation results confirm that SEPR achieves a higher delivery rate employing fewer message copies as compared to epidemic routing.

Gradient-based routing schemes suffer from a slow-start phase. Sufficient number of encounters must happen before the nodes develop meaningful metrics for each destination. In addition, this information needs to be propagated through the network. One solution to address this shortcoming is the Seek and Focus scheme (Spyropoulos et al. (2004)). This scheme initially forwards the message picking a neighbor at random until the metric utility value reaches a certain threshold. Thereafter a gradient-based approach may be employed to deliver the message to the destination.

4.7 Controlled replication schemes

Compared to traditional epidemic routing based schemes and its variants that rely on reducing the consumption of network resources, Spray and Wait (Spyropoulos et al. (2005)) presents a novel way to achieve efficient routing in DTNs. The idea is that it reduces the number of copies of a given message, and hence the number of transmissions for a given message, to a fixed value L that can be tuned in accordance with the delivery delay requirement. The scheme 'sprays' a number of copies of a message into the network to L distinct relays and then 'waits' till one of these relays meets the destination. A number of heuristics are presented about how the L copies are sprayed, for example, the source is responsible for spraying all L copies or more optimally, each progressive node encountered by a source or relay is handed over the responsibility to distribute half of the remaining copies (called Binary Spray and Wait). This scheme requires no knowledge of the mobility of the nodes. The expected delay of this scheme is analytically computed for the case of mobile nodes performing random walks on the surface of a 2-dimensional torus and compared with the optimal delay. This delay is independent of the size of the network and only depends on the number of nodes. The scheme is shown to posses robust scalability as the node density goes up.

A variant of this scheme called Spray and Focus (Spyropoulos et al. (2007)) provides further improvements by taking advantage of the mobility information in the wait-phase. The idea is that once the spray phase is over, each relay can then forward the packet further using a single-copy utility based scheme instead of naively waiting to meet the destination. Hence, this scheme combines the advantages of controlled replication along with those of gradient-based schemes presented earlier. Simulation results with a variety of mobility models such as random walk, random way-point, community-based etc. show significant improvements in the delivery delay.

4.8 Network coding based schemes

As opposed to the traditional model of forwarding in DTNs where nodes may forward the entire copy of the message to encountered relays, an alternate approach is to employ network coding based schemes. In (Wang et al. (2005)), the authors provide an erasure-coding based approach to forward data in DTNs. The idea is that the source node encodes a message and generates a large number of code blocks guided by a replication factor r. The generated code blocks are then equally split among the first $k \cdot r$ relays, for some constant k, and those relays must deliver the coded blocks to the destination directly. The original message can be decoded once $1/r$ coded blocks have been received. In other words, the message can be decoded as soon as k relays deliver their data to the destination. Such a scheme is more robust to failures of a few relays or some bad forwarding choices. The authors demonstrate via simulation evaluation with both synthetic and real world traces that this scheme achieves better worst-case delay performance that existing approaches with a fixed overhead.

Study	Scheme	Mobility Model	Energy	Delay	Copies created	Storage
Drop Oldest (Davis et al. (2001))	Epidemic Routing	Random Waypoint		X	Many	X
Infostations (Frenkiel et al. (2000))	Direct	Highway		X	None	
Message Ferries (Zhao & Ammar (2003))	One-hop Relay	Nonrandom Pro-active	X	X	One	
Zebroids (Ghandeharizadeh et al. (2006))	Knowledge Oracles	Random with predictions		X	Many	X
MoVe (LeBrun et al. (2005))	Location	Bus movement		X	One	X
Seek and Focus (Spyropoulos et al. (2004))	Gradient	Random Walk	X	X	One	
Spray and Wait (Spyropoulos et al. (2005))	Controlled Replication	Random Walk	X	X	Many	
Erasure Coding (Wang et al. (2005))	Source Coding	Animal movement based	X	X	Many	

Table 1. Related studies on intermittently connected networks.

Compared to the scheme proposed earlier that employs source coding, Widmer & Le Boudec (2005) propose a network coding based protocol for routing in DTNs. The idea is that intermediate nodes send out packets based on some linear combination of previously received information. In this way, a receiver reconstructs the original message once it receives enough encoded messages. A packet received by a node is considered innovative if it increases the ″rank″ of the set of received packets at this node. A parameter controls with which probability the reception of innovative packets causes a node to send a packet. The authors incorporate a mechanism of information aging in their protocol so that efficient network coding can still be achieved with little available memory. The process of determining how many and which messages will be coded together poses significant challenges especially if this is to be done in a distributed manner.

On the basis of the classification introduced in this section, we provide a small summary of DTN routing schemes in Table 1 depicting their representative characteristics.

5. Conclusions and future work

In this chapter, we have presented a survey of some of the most promising approaches proposed for data delivery in DTNs. Our survey and classification has concluded that there is no universal scheme that will be applicable in all scenarios. Depending on the particular scenario in question, either one or more likely a combination of schemes will be applicable to satisfy the needs of the application. A couple of other surveys for routing in delay tolerant networks that compliment this study have also appeared in recent literature (Spyropoulos et al. (2010); Jones & Ward (2006)). However, with so many choices available, some form of industry-wide agreement on standardization of a subset of these techniques as well as a

DTN architecture is necessary. The Delay Tolerant Network Research Group (DTNRG) is one such effort where an architecture for messaging in DTNs has been proposed (Cerf et al. (2007)).

Delay Tolerant Networks are a reality. With a large amount of different devices such as the smart-phones, netbooks, thin-clients etc. available in the market today, DTN routing has become even more challenging since it has to adapt to a vast set of heterogeneous nodes with different capabilities and networking technologies. Additionally, it has become increasingly clear that DTNs must be able to reach the global Internet. One proposal that enables communication between DTNs and the Internet is the Tetherless Communication Architecture (Seth et al. (2005)). More and more real-world deployments of DTNs at different scales that practically demonstrate the utility of the routing schemes and show how they can be employed to either alleviate or solve practical problems will allow researchers to drive the adoption of DTNs.

Finally, an important consideration for DTNs relates to issues of security, privacy, anonymity, and trust. For DTN routing to function, intermediate nodes must cooperate and agree to carry content of other users. In addition, the content must be transported securely and possibly encrypted to protect the information as well as prevent man-in-the-middle kind of attacks. The routing schemes themselves must have in-built mechanisms that address all these issues. While there have been independent proposals to address some of these aspects (Farrell & Cahill (2006); Seth & Keshav (2005); Kate et al. (2007)), a framework that integrates all these aspects and provides a holistic solution for DTNs is still missing.

6. References

Balazinska, M. & Castro, P. (2003). Characterizing mobility and network usage in a corporate wireless local-area network, *MobiSys '03: Proceedings of the 1st international conference on Mobile systems, applications and services*, ACM, New York, NY, USA, pp. 303–316.

Bin Tariq, M. M., Ammar, M. & Zegura, E. (2006). Message ferry route design for sparse ad hoc networks with mobile nodes, *MobiHoc '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, ACM, New York, NY, USA, pp. 37–48.

Burgess, J., Gallagher, B., Jensen, D. & Levine, B. (2006). MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networking, *Proc. of IEEE Infocom*.

Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K. & Weiss, H. (2007). Delay-tolerant networking architecture.
URL: *http://www.rfc-editor.org/rfc/rfc4838.txt*

Davis, J., Fagg, A. & Levine, B. (2001). Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks, *Wearable Computers, 2001. Proceedings. Fifth International Symposium on*, pp. 141 –148.

Demers, A., Greene, D., Hauser, C., Irish, W., Larson, J., Shenker, S., Sturgis, H., Swinehart, D. & Terry, D. (1987). Epidemic algorithms for replicated database maintenance, *PODC '87: Proceedings of the sixth annual ACM Symposium on Principles of distributed computing*, ACM, New York, NY, USA, pp. 1–12.

Fall, K. (2003). A delay-tolerant network architecture for challenged internets, *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, ACM,* New York, NY, USA, pp. 27–34.

Farrell, S. & Cahill, V. (2006). *Delay- and Disruption-Tolerant Networking*, Artech House, Inc., Norwood, MA, USA.

Frenkiel, R., Badrinath, B., Borres, J. & Yates, R. (2000). The infostations challenge: balancing cost and ubiquity in delivering wireless data, *Personal Communications, IEEE* 7(2): 66 –71.

Ghandeharizadeh, S., Kapadia, S. & Krishnamachari, B. (2006). An evaluation of availability latency in carrier-based wehicular ad-hoc networks, *MobiDE '06: Proceedings of the 5th ACM international workshop on Data engineering for wireless and mobile access*, ACM, New York, NY, USA, pp. 75–82.

Grossglauser, M. & Vetterli, M. (2003). Locating nodes with EASE: last encounter routing for Ad Hoc networks through mobility diffusion, *Proc. IEEE Infocom*, Vol. 3, pp. 1954–1964.

Harras, K. A. & Almeroth, K. C. (2006). Transport layer issues in delay tolerant mobile networks, *IN IFIP NETWORKING*.

Harras, K. A., Almeroth, K. C. & Belding-royer, E. M. (2005). Delay tolerant mobile networks (dtmns): Controlled flooding schemes in sparse mobile networks, *In IFIP Networking*.

Henderson, T., Kotz, D. & Abyzov, I. (2004). The changing usage of a mature campus-wide wireless network, *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, ACM, New York, NY, USA, pp. 187–201.

Hsu, W. & Helmy, A. (2006). On modeling user associations in wireless lan traces on university campuses, *In Proceedings of the Second Workshop on Wireless Network Measurements (WiNMee)*.

Jain, S., Fall, K. & Patra, R. (2004). Routing in a delay tolerant network, *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications,* ACM, New York, NY, USA, pp. 145–158.

Jones, E. P. C., Li, L. & Ward, P. A. S. (2005). Practical routing in delay-tolerant networks, *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, ACM, New York, NY, USA, pp. 237–243.

Jones, E. P. & Ward, P. A. (2006). Routing strategies for delay-tolerant networks, *Submitted to ACM Computer Communication Review (CCR)* .

Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L. & Rubenstein, D. (2002). Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet, *SIGARCH Computer Architecture News* .

Kapadia, S., Krishnamachari, B. & Ghandeharizadeh, S. (2009). Static replication strategies for content availability in vehicular ad-hoc networks, *Mob. Netw. Appl.* 14(5): 590–610.

Karp, B. & Kung, H. T. (2000). Gpsr: greedy perimeter stateless routing for wireless networks, *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, ACM, New York, NY, USA, pp. 243–254.

Kate, A., Zaverucha, G. M. & Hengartner, U. (2007). Anonymity and security in delay tolerant networks, *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on,* pp. 504–513. URL: http://www.cacr.math.uwaterloo.ca/techreports/2007/cacr2007-12.pdf

LeBrun, J., Chuah, C.-N., Ghosal, D. & Zhang, M. (2005). Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks, *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, Vol. 4, pp. 2289 – 2293 Vol. 4.

Leguay, J., Friedman, T. & Conan, V. (2005). Dtn routing in a mobility pattern space, *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, ACM, New York, NY, USA, pp. 276–283.

Leguay, J., Friedman, T. & Conan, V. (2006). Evaluating mobility pattern space routing for dtns, *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pp. 1 –10.

Li, Q. & Rus, D. (2000). Sending messages to mobile users in disconnected ad-hoc wireless networks, *Proc. of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom)*, New York, NY, USA, pp. 44–55.

Lindgren, A., Doria, A. & Schelén, O. (2003). Probabilistic routing in intermittently connected networks, *SIGMOBILE Mob. Comput. Commun. Rev.* 7(3): 19–20.

Musolesi, M., Hailes, S. & Mascolo, C. (2005). Adaptive routing for intermittently connected mobile ad hoc networks, *WOWMOM '05: Proceedings of the Sixth IEEE International Symposium on World of Wireless Mobile and Multimedia Networks*, IEEE Computer Society, Washington, DC, USA, pp. 183–189.

Nelson, S., Bakht, M. & Kravets, R. (2009). Encounter-based routing in dtns, *INFOCOM 2009, IEEE*, pp. 846 –854.

Pentland, A., Fletcher, R. & Hasson, A. (2004). DakNet: Rethinking Connectivity in Developing Nations, *Computer* 37(1): 78–83.

Seth, A., Darragh, P., Liang, S., Lin, Y. & Keshav, S. (2005). An architecture for tetherless communication, *in* M. Brunner, L. Eggert, K. Fall, J. Ott & L. Wolf (eds), *Disruption Tolerant Networking*, number 05142 in *Dagstuhl Seminar Proceedings*, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, Dagstuhl, Germany.
URL: *http://drops.dagstuhl.de/opus/volltexte/2005/351*

Seth, A. & Keshav, S. (2005). Practical security for disconnected nodes, *Secure Network Protocols, 2005. (NPSec). 1st IEEE ICNP Workshop on*, pp. 31 – 36.

Shah, R., Roy, S., Jain, S. & Brunette, W. (2003). Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks, *Elsevier Ad Hoc Networks Journal* 1.

Small, T. & Haas, Z. J. (2003). The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way), *Proc. of the 4th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc)*, ACM Press, New York, NY, USA, pp. 233–244.

Small, T. & Haas, Z. J. (2005). Resource and performance tradeoffs in delay-tolerant wireless networks, *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, ACM, New York, NY, USA, pp. 260–267.

Spyropoulos, T., Psounis, K. & Raghavendra, C. (2004). Single-Copy Routing in Intermittently Connected Mobile Networks, *Proc. of IEEE SECON.*

Spyropoulos, T., Psounis, K. & Raghavendra, C. S. (2005). Spray and wait: an efficient routing scheme for intermittently connected mobile networks, *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, ACM, New York, NY, USA, pp. 252–259.

Spyropoulos, T., Psounis, K. & Raghavendra, C. S. (2007). Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility, *PERCOMW '07: Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops*, IEEE Computer Society, Washington, DC, USA, pp. 79–85.

Spyropoulos, T., Rais, R. N. B., Turletti, T., Obraczka, K. & Vasilakos, A. (2010). Routing for disruption tolerant networks: Taxonomy & design, *Wireless Networks* .

Takagi, H. & Kleinrock, L. (1984). Optimal transmission ranges for randomly distributed packet radio terminals, *Communications, IEEE Transactions on* 32(3): 246 – 257.

Tan, K., Zhang, Q. & Zhu, W. (2003). Shortest path routing in partially connected ad hoc networks, *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, Vol. 2, pp. 1038 – 1042 Vol.2.

Vahdat, A. & Becker, D. (2000). Epidemic routing for partially-connected ad hoc networks, *Technical report*, Department of Computer Science, Duke University. Wang, Y., Jain, S., Martonosi, M. & Fall, K. (2005). Erasure-coding based routing for opportunistic networks, *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, ACM, New York, NY, USA, pp. 229–236.

Wang, Y., Jain, S., Martonosi, M. & Fall, K. (2005). Erasure-coding based routing for opportunistic networks, WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, ACM, New York, NY, USA, pp. 229–236.

Widmer, J. & Le Boudec, J.-Y. (2005). Network coding for efficient communication in extreme networks, *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delaytolerant networking*, ACM, New York, NY, USA, pp. 284–291.

Zhao, W. & Ammar, M. (2003). Message ferrying: proactive routing in highly-partitioned wireless ad hoc networks, *Distributed Computing Systems, 2003. FTDCS 2003. Proceedings. The Ninth IEEE Workshop on Future Trends of*, pp. 308 – 314.

Zhao, W., Ammar, M. & Zegura, E. (2004). A message ferrying approach for data delivery in sparse mobile ad hoc networks, *Proc. of the 5th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc)*, ACM Press, New York, NY, USA, pp. 187–198.

Zhao, W., Ammar, M. & Zegura, E. (2005). Controlling the mobility of multiple data transport ferries in a delay-tolerant network, *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, Vol. 2, pp. 1407 – 1418 vol. 2.

27

Broadcasting in Mobile Ad Hoc Networks

Sangwoo Lee and Chaewoo Lee
Ajou University
Republic of Korea

1. Introduction

Mobile ad hoc networks (MANETs) are self-organizing and the constituent mobile nodes communicate with each other as autonomous hosts in the absence of a fixed infrastructure. Recently, MANETs are deployed to places where the network is required to be promptly established such as military operations and disaster relief. However, the mobile nodes merely operate with limited resources such as processing, communication, and energy. The nodes further have the characteristic of high mobility. Thus, MANET has the properties of frequently route breakage and unpredictable topology changes.

Clearly, these properties make the transmission methods widely used in fixed infrastructures inappropriate for MANET. Broadcasting is an alternative which is a one-to-all transmission method, namely a packet or a message generated by a node, called the source, is sent to all other nodes in the network. Moreover, broadcasting is an important operation in applications performing route discovery (Johnson & Maltz, 1996; Park & Corson, 1997; Pearlman & Haas, 1999; Perkins & Royer, 1999), updating the network knowledge, or sending an alarm signal. However, it seems greedy and excessive in aspect of resource limitation, especially energy which is a major concern in MANET, since the nodes transmit packets in a multi-hop communication manner. Therefore, the energy cost of broadcast packet transmission (i.e., the number of transmissions) should be minimized to conserve the energy of the mobile nodes.

Blind flooding is the most straightforward approach to broadcasting. Specifically, every node in the network forwards the broadcast packet exactly once. It ensures the full coverage of all the network: all the nodes in the network are guaranteed to receive the broadcast packet in case that the network is static and the occurrence of collision and error is not considered during propagation. However, flooding may generate excessive redundant transmissions which cause a critical problem, referred to as the *broadcast storm* problem (Ni et al., 1999), introducing communication contention and collision due to sharing wireless resources and overlapping coverage areas among nodes.

The broadcast storm problem can be readily avoided by reducing the number of retransmissions. In order to alleviate the broadcast storm problem, probability-based, area-based, and neighbor knowledge approaches control the amount of traffic, that is, each node determines whether or not to retransmit the broadcast packet. The probability-based approach controls message flood with a predefined probability or received packet count. Obviously, it resembles blind flooding when the probability that a node retransmits the broadcast packet equals to one. In the area-based approach, each node determines whether or not to rebroadcast the packet with evaluation of its additional coverage area by rebroadcasting. If the additional coverage is less than the threshold, the node abandons retransmitting. This method relies on location or distance information of nodes to determine rebroadcasting. The neighbor knowledge approach utilizes one or two hop neighbor information obtained via periodical hello packets to reduce redundant rebroadcasting. This approach allows retransmitting only when it results in any additional neighbor to be reached.

According to the methods controlling message flood, network overhead can be significantly reduced. However, some problems can occur such as end-to-end delay or latency and unreliability. Each node requires a certain waiting time to examine whether or not to rebroadcast a packet. In the area-based approach, for example, a node sets a random waiting time when a previously unseen packet arrives and it observes the duplicate packet arriving during the waiting time. Since nodes hold a packet for waiting times, the time spent on propagation from the packet origination to reach a node, namely end-to-end delay, increases.

Reliability is considered in a network that nodes are fully connected to others in a single- or multi-hop fashion and the network is static. When a node determines to discard a packet with an examination of the necessity of rebroadcasting, some one-hop neighbors may not receive the packet. Furthermore, the packet is unreachable to nodes which have the sole connection through the neighbors.

More precisely, a perfectly reliable broadcasting with minimizing redundancy is defined as a problem finding the minimum connected dominating set (MCDS) where a connected dominating set states that each node either belongs to the set or has a neighbor which belong to the set and is fully connected to others. Unfortunately, the problem of finding the MCDS is classified as NP-complete even if the global topology information is given (Lim & Kim, 2001; Lou & Wu, 2002).

Some broadcasting schemes form a conjunction of area-based and neighbor knowledge approaches, called hybrid broadcasting schemes, to efficiently resolve redundant transmission, unreliability, and latency. Based on that outer nodes from the sender are prone to have more additional coverage than inner nodes (i.e., area-based approach), the outer takes higher priority of retransmission than the inner: when a node receives a previously unseen packet, it first sets a waiting time determined in inverse proportion to the distance to the sender. Instead of computation of additional coverage to make a determination of packet drop, each node examines whether all its neighbors receive the packet (i.e., neighbor knowledge) to resolve a potential unreliability.

The remainder of this chapter is organized as follows. Section 2 introduces issues in broadcasting. Section 3 reviews previously published broadcasting methods. Section 4 describes hybrid broadcasting schemes. Section 5 concludes this chapter and gives some possible future works.

2. Issues in broadcasting

2.1 The broadcast storm

As mentioned above, flooding is the simplest solution to broadcasting. The fundamental idea behind flooding is that every node participates in transmission of a packet exactly once

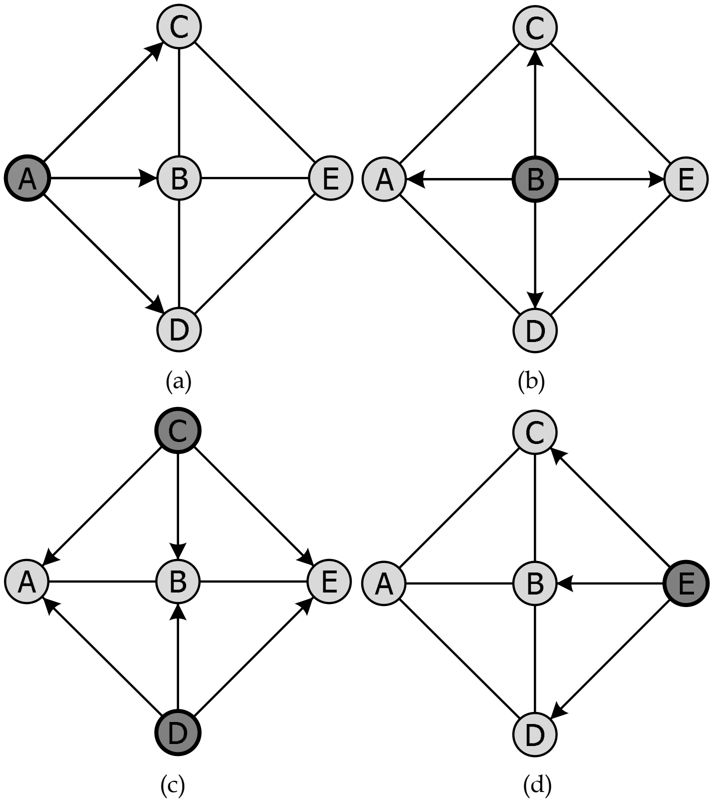


Fig. 1. A sample network with five nodes: (a) Broadcasting by source *A* (b) Optimal broadcasting (c) Redundant broadcasting and hidden node problem without RTS/CTS handshake (d) Redundant broadcasting

to deliver it throughout the network in a multi-hop fashion. Hence, intermediate nodes have the obligation to retransmit the packet. This leads to *n* transmissions in a network of *n* hosts for a single packet. It achieves perfectly reliable broadcasting if the communication channel is error-free with no collision.

Unfortunately, redundancy, contention, and collision can be observed which are referred to as the broadcast storm problem in (Ni et al., 1999). The main reason for redundancy is that the coverage of a node may overlap with others nearly placed. In other words, several intermediate nodes perform redundant transmission in case that all neighbors within the transmission range of the node have already received the packet. Additionally, because of the lack of bandwidth, wireless resource sharing, and the absence of collision detection, redundant transmissions are prone to trigger noticeable contention and collision. The broadcast storm problem seriously worsens if the size of the network increases and nodes are densely distributed. Figure 1 illustrates a network with five nodes. When a broadcast packet is generated and forwarded by source *A*, the packet reaches to all the nodes by node *B* as shown in figure 1(b). Figures 1(c) and 1(d) illustrates redundant broadcasting by nodes *C*, *D*, and *E*. In addition to the redundancy in figure 1(c), the hidden node problem can be found in the network based on contention-based protocols including ALOHA (Abramson, 1970), slotted ALOHA (Saadawi & Ephremides, 1981), CSMA (carrier sense multiple access), and IEEE 802.11.

The broadcast storm problem can be avoided by merely reducing the number of retransmissions of the packet (i.e., packet drop). Thus, the way to drop the packet efficiently is a substantial goal in broadcasting. Several enhancements to flooding are discussed in section 3.

2.2 Unreliability and latency

Apparently, packet drop becomes an essential function in broadcasting to reduce the broadcast redundancy, which is implicit in resource usage, by judging necessity of transmission. The packet drop effectively resolves the broadcast storm problem and even reduces resource usage. Unfortunately, end-to-end unreliability and latency can arise due to packet drop by an incorrect judgement.

Because of the properties of MANET, flooding rarely guarantees perfect reliability. For instance, some nodes are often isolated from the network. In spite of difficulties, a perfectly reliable broadcasting is necessary in some applications (e.g., the localization in (Doherty et al., 2001; Niculescu & Nath, 2001; Shang & Ruml, 2004)). The underlying assumption here is that the network is seen as static or a snapshot of mobile networks with the error-free channel. In this network, flooding ensures the perfectly reliable delivery. However, some nodes in the network may not receive the packet when flooding is implemented with packet flood control. As far as a node on standby for transmission determines the necessity of transmission without global topology information or local announcements representing packet reception from its neighbors, the node is prone to make a poor determination. In general, packet loss occurs by collision and it results in poor reliability as well.

Latency is also introduced from the packet drop which is the time spent from when a packet is originated until it reaches to a node. Each node waits for a certain time to make a determination of retransmission in packet flood control. It is primarily required for appropriate flood control. Furthermore, packet drop is likely to block the shortest path of a packet in sparse networks. In other words, the packet makes a detour to nodes in specific regions.

Typically, many researches aim at minimizing the number of transmission while attempting to ensure the full coverage of the network at the same time. Recently, since end-to-end delay becomes a major issue in designing networks, rapid spread of a broadcast packet is indeed considered as well.

3. Broadcasting in MANET

We now describe some previously published works for the broadcast problem. The broadcasting methods can be classified as blind flooding, probability-based, area-based, and neighbor knowledge approaches (Williams & Camp, 2002).

Blind flooding (Ho et al., 1999; Jetcheva et al., 2001) requires each node to rebroadcast a broadcast packet to all its neighbors and this continues until all nodes retransmit the packet at least once. In order to alleviate the inefficiencies of blind flooding, probability-based methods assign a probability to a node to determine whether or not to rebroadcast. Area-based methods allow a node to rebroadcast a packet if its transmission range sufficiently covers additional area. Neighbor knowledge methods decide the retransmission of the packet based on local neighbor lists through hello packets. A performance comparison of some of the schemes can be found in (Williams & Camp, 2002).

In this section, we cover probability-based, area-based, and neighbor knowledge approaches which are enhancements to blind flooding.

3.1 Probability-based approach

A probability-based approach attempts to resolve redundancy on the basis of that more duplicate packet receptions promptly deteriorates additional coverage via retransmission. Intuitively, multiple nodes are prone to share similar coverage in dense networks. Figure 2 presents additional coverage by retransmission at node *A*. We assume here that transmission range of each node is identical and omnidirectional. A circle centered at a node represents its transmission range. The additional coverage of node *A* is denoted by shaded area. A scenario is shown in figure 2(a) where node *A* receives a packet from neighboring node *B* and forwards the packet. As seen in figure 2(b), node *A* receives the same packet from both nodes *B* and *C*. In this case, the additional coverage is reduced because coverage of node *A* is covered by nodes *B* and *C* in advance.

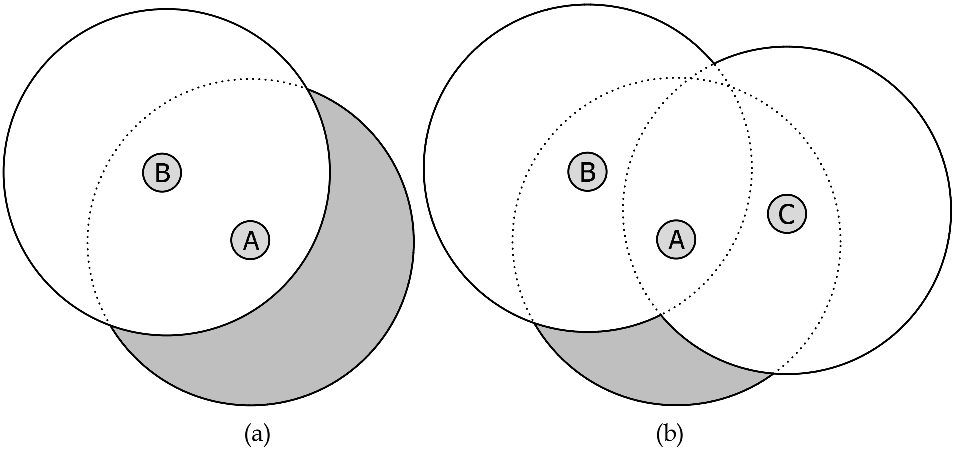


Fig. 2. Additional coverage area by node *A*: (a) A single packet reception from node *B* (b) Duplicate packet reception from nodes *B* and *C*

This approach is classified into two categories: probabilistic and counter-based schemes (Ni et al., 1999). The probabilistic scheme controls packet flood with a predetermined probability or dynamic probability based on derived conditions such as node density and distribution (Ryu et al., 2004; Tseng et al., 2001). If the probability is set to a far small value, reliability will decrease in sparse networks and even in dense networks where nodes are randomly distributed. On the other hand, redundant transmission will be introduced if the probability is set far large. Incidentally, the probabilistic scheme resembles blind flooding when the probability is set to one.

The counter-based scheme allows a node to retransmit a packet unless the node has received the redundant packet more than a predefined threshold over a waiting time called random assessment delay (RAD). Upon receiving a previously unseen packet, the node respectively initiates a counter and a timer (RAD) to one and zero. During the RAD, the counter is incremented by one for receiving each duplicate packet. When the RAD expires, if the counter reaches a threshold or over, the node drops the packet. Otherwise, it rebroadcasts the packet to neighbors.

The probability-based approach is simple to embody with local topology information. Thus, it effectively reduces the broadcast redundancy while the full coverage may not be ensured in dense networks; whereas all nodes are likely to rebroadcast in sparse networks. However, it is difficult to take into consideration the factors of rebroadcasting and reliability to derive a proper probability and the optimal threshold.

3.2 Area-based approach

Typically, additional coverage of a node is dependent on either distance to the sender or locations of the nodes. Suppose two nodes have bidirectional link with identical and omnidirectional transmission range as seen in figure 3. The overlapped coverage area increases as two nodes are more closely located. In other words, additional coverage of a node is maximized when the node is located at the boundary of the coverage of the sender. In area-based approach, a node computes additional coverage based on the coverage of its neighbors and it drops a broadcast packet when the additional coverage is less than a preset threshold. The area-based approach is categorized as distance-based and location-based schemes.

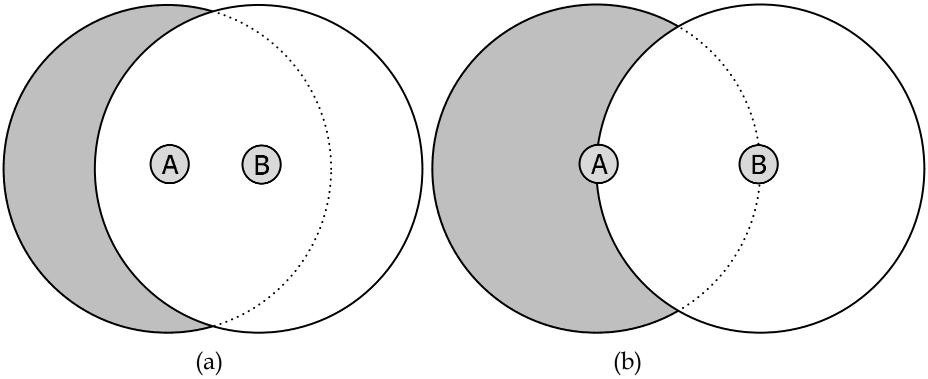


Fig. 3. Additional coverage area by node *A*: (a) When node *A* is closely placed to node *B* (b) When node *A* is placed at the boundary of transmission range of node *B*

The distance-based scheme determines if packet forwarding is necessary through ranging, that is, a process measuring the distance between nodes based on received signal strength, time of arrival, or time difference of arrival. A node initiates a RAD when a previously unseen packet arrives. During the RAD, the node measures the distance to each sender transmitting the packet and compares the distance and a threshold distance. If any of the distance measurements is closer than the threshold, the node stops forwarding the packet.

In the location-based scheme (Ni et al., 1999), each node has the physical location information through global positioning system (GPS) (Getting, 1993) or manual configuration. A node receiving a previously unseen packet initiates a RAD and it accumulates the coverage by the senders of the packet based on the locations of the senders for the RAD. When the RAD expires, the node rebroadcasts the packet if the accumulated coverage is narrower than a threshold.

Geoflood is a location-based scheme introduced in (Arango et al., 2004). Each node sets a RAD when the first packet arrives and defines quadrants with own position as the origin. Upon receiving the duplicate packets for the RAD, the node records which quadrants are covered. When a node receives the duplicates from all quadrants, it decides that retransmission is unnecessary and discards the packet. Otherwise, the node forwards the packet. Let d be the distance from the sender. The RAD $W(d)$ is randomly chosen over a range which is given by

$$W(d) = rand(\max(h(d) - W_{off}, 0), h(d) + W_{off}) \tag{1}$$

where W_{off} is the maximum random offset and

$$h(d) = W_{\max} - \frac{d \times W_{\max}}{R}$$

with the predefined maximum waiting time $W_{\max}$ and the range R of the network. Since outer nodes cover much additional coverage and help the packet propagate faster than inner nodes, shorter times are allocated to outer nodes than inner nodes. Geoflood may seem able to achieve perfectly reliable delivery while reducing the number of retransmissions at the same time. However, there is high possibility to make the packet unreachable to some nodes. Figure 4 depicts the worst case that some nodes may not receive the packet. In this figure, node A receives the identical packet sequentially from nodes B, C, D, and E. Since node A has received from all quadrants, it drops the packet. Thus, nodes in the shaded area are ignored by node A.

The area-based approach attempts to achieve the expected coverage of the network while reducing redundancy through computation of the local coverage area. However, additional devices are requires for ranging or localization in the determination of retransmission. These are likely to be expensive and power-intensive. In aspect of energy usage, this approach may lead to reducing network lifetime that should be primarily considered in MANET.

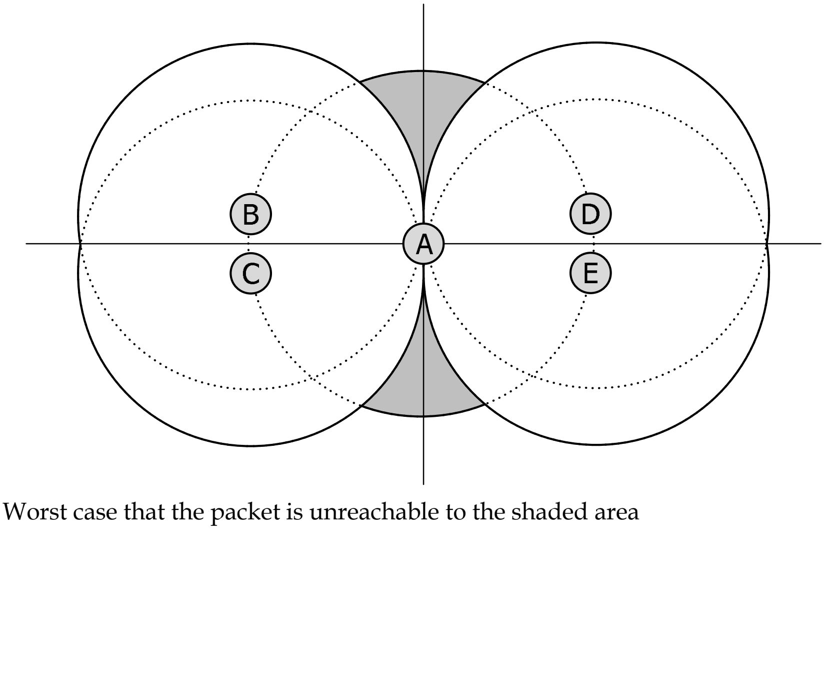


Fig. 4. Worst case that the packet is unreachable to the shaded area

3.3 Neighbor knowledge approach

A neighbor knowledge approach alleviates the broadcast redundancy based on the local topology information in one-hop or two-hop. Additionally, when the network topology is static, it achieves the perfectly reliable delivery. In order to obtain knowledge of one- or two-hop neighbor, it is necessary that each node periodically emits a hello packet to inform its existence to the neighbor.

(Lim & Kim, 2001) proposed two simple schemes called *self pruning* and *dominant pruning*. Self pruning employs one-hop neighbor knowledge for broadcasting. Each node manages the neighbor list via a periodic hello packet. The node piggybacks the list in the broadcast packet when it performs retransmission. The neighbor, which receives the forwarded packet including the neighbor list of the sender, examines whether there is a difference between its neighbor list and the sender's list. In other words, $N(A)- N(B)-\{B\} = \emptyset$ where $N(A)$, $N(B)$ respectively represent the list of sender A, the list of neighbor B. If the examination says true, the neighbor renounces broadcasting since this indicates that there is no additional node covered by the neighbor. Otherwise, the packet is forwarded with updating the neighbor list. Whereas self pruning only exploits the knowledge of neighbors directly connected to the sender, dominant pruning requires the knowledge of neighbors within two-hop from the sender. In addition to the knowledge extension, the authority to make a determination of retransmission for the neighbors is given to the sender in dominant pruning. The sender selects some or all of one-hop neighbors, which covers all other nodes with the minimum number of transmissions. It then marks the IDs of the selected nodes in the forward list of the packet for retransmission. Since each node belonging to the forward list broadcasts the packet once, the minimized forward list states the minimum number of transmissions. Figure 5 shows the sets of nodes within two-hop from the sender A. In this figure, node B is the former sender before the current sender A. The sets $N(A)$ and $N(B)$ present respectively neighbors within the coverage range of nodes A and B. The set of neighbors within two-hop from node A is denoted by $N(N(A))$. Thus, in dominant pruning, the sender A should determine the forward list from the set $N(A)$ to cover the set

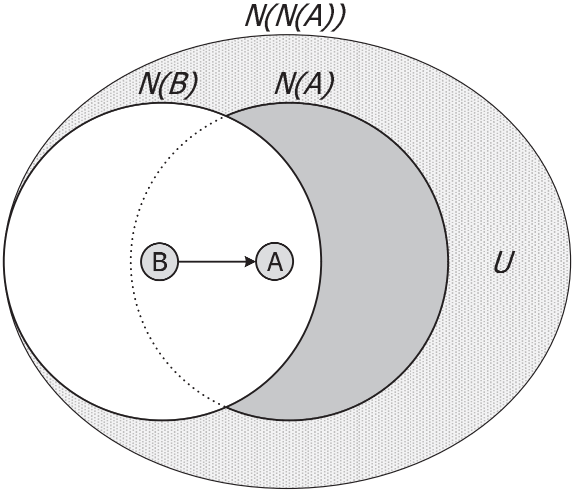


Fig. 5. Dominant pruning

$U = N(N(A)) - N(B) - N(A)$. As the former sender B sets the forward list that helps the packet reachable to two-hop distance from node B, namely $N(N(B))$, the forward list for the current sender A is selected into the set $N(A) - N(B)$ denoted by the shaded area in figure 4. This is the set cover problem, which is NP-complete, to minimize a set F of nodes subject to $\bigcup_{C\in F}(N(C) \cap U) = U$ and $F \subseteq N(A) - N(B)$. The set F is approximately obtained with the greedy set cover algorithm.

(Peng & Lu, 2000) described *scalable broadcast algorithm* (SBA) in which each node forwards the broadcast packet including the neighbor list covered by the transmission. Once node A receives the broadcast packet m from node B, node A discards the packet promptly if $N(A) \subseteq N(B)$. If not, the node initiates a covered node set $N_m(A) = N(B) \cup \{B\}$ and sets a waiting time W, which is randomly chosen from uniform distribution $U(x)$ between 0 and x, given by

$$W = U(\Delta \times W0) \tag{2}$$

where Δ is a small constant delay and

$$W0 = \frac{1 + d_m(A)}{1 + d(A)}$$

with the maximum degree $d_m(A)$ of neighbors of node A and the degree $d(A)$ of node A. During the waiting time, upon receiving the duplicate packet from node C, node A updates the covered node set such that $N_m(A) = N_m(A) \cup N(C) \cup \{C\}$. When the waiting time expires, it examines whether $N(A) \subseteq N_m(A)$. If the set $N(A)$ either belongs or equals to the covered node set $N(A)$, node A determines that retransmission is unnecessary and discards the packet. In other words, all the neighbors placed in the coverage area of node A have already received the broadcast packet. Moreover, a random waiting time is dynamically determined according to the degree of a node: a node with more neighbors waits shorter.

(Qayyum et al., 2002) described *multipoint relay* (MPR) where each node selects a small subset of neighbors, namely MPR set, using knowledge of neighbors within two-hop distance from itself. If a node receives the packet and belongs to MPR set of the sender, the node rebroadcasts the packet. The optimal MPR set covers all one- and two-hop neighbors with a minimum number of one-hop neighbors. Since the network topology dynamically changes, each node requires to reselect appropriate MPRs when a change of the local topology is detected through periodic hello packets. Denote the set of one-hop neighbors of node A by $N(A)$ and the set of two-hop neighbors of node A by $N(N(A))$. Let the selected MPR set of node A be $MPR(A)$. The heuristic selection for MPR set is stated as follows:

1. Initiate an empty multipoint set, namely $MPR(A) = \varnothing$
2. Select nodes in $N(A)$ as MPRs which are only neighbors in $N(N(A))$ and include these MPRs in $MPR(A)$
3. While there still exist uncovered nodes in $N(N(A))$ by $MPR(A)$, add a node of $N(A)$ which covers the maximum number of the uncovered nodes to $MPR(A)$
4. Repeat steps (2) ~ (3) until all two-hop neighbors are covered by $MPR(A)$

The neighbor knowledge approach resolves the broadcast redundancy acquiring the local topology information. Thus, this approach efficiently reduces excessive redundant transmissions while attaining the goal of reliable broadcasting. Unfortunately, since each node periodically emits a hello packet to acquire the local topology, more overhead is generated than other approaches.

4. Hybrid broadcasting scheme

This section covers two broadcasting schemes based on a hybrid concept of area-based and neighbor knowledge approaches. These schemes requires the local knowledge of one-hop neighbors from the sender. Therefore, each node emits a hello packet periodically to directly connected neighbors to inform its presence. The underlying assumptions are that each node knows its physical location and all the nodes have an identical, omnidirectional antenna: a pair of nodes has a bidirectional link when the nodes are placed within the transmission range of each other.

Specifically, in the area-based approach, outer nodes have shorter waiting times than inner nodes to enlarge coverage area and to spread the packet rapidly. Geoflood, for example, allocates the minimum waiting time to the farthest node from the sender. However, the farthest node should hold the packet for the waiting time although it becomes the sender earlier than any other candidate after the previous sender. Further, other candidates should wait for more than necessity. Thus, high latency occurs in sparse networks and even in dense networks.

(Lee & Ko, 2006) introduced *flooding based on one-hop neighbor information and adaptive holding* (FONIAH) to reduce latency and unreliability. Each node accumulates hello packets from all its neighbors and makes the neighbor list in advance. When a node receives a previously unseen packet, it initiates a waiting time which is inversely proportional to the distance from the sender. During the waiting time, the node holds the packet and eliminates the IDs of neighbors transmitting the duplicate packet from the neighbor list. When the waiting time expires, the node discards the packet if the neighbor list is an empty set; whereas it forwards the packet for the rest.

In FONIAH, a waiting time $W(d, d_{\max})$ is stated as a simple linear function:

$$W(d, d_{\max}) = W_{\max} - \frac{d \times W_{\max}}{d_{\max}} \tag{3}$$

where $W_{\max}$ is the predetermined, constant maximum waiting time. d is the distance from the sender. $d_{\max}$ is referred to as the maximum distance, which is the distance from the sender to the farthest one-hop neighbor of the sender. When the sender forwards the packet embedding the maximum distance, all its one-hop neighbors become candidates with different waiting times. Therefore, the candidates have relative waiting times. In other words, the farthest candidate from the sender rebroadcasts immediately as the packet arrives wherever it is placed.

Generally, the additional coverage of a node is dependent on both the distance from the sender and the location of the node. Therefore, although two nodes are located at the same distance from the sender as seen in figure 6, one covers more area than the another. In this figure, two nodes *C* and *D* receive the identical broadcast packet after retransmissions of nodes *A* and *B*. Since nodes *C* and *D* are placed at the boundary of the range of node *B* (i.e., the sender), both nodes have the same waiting time in FONIAH. That is, collision between nodes *C* and *D* occurs. Thus, one has to wait for more than its waiting time to help another's retransmission successful. This raises a question of which node should rebroadcast first.

In (Jaegal & Lee, 2008), *efficient flooding algorithm for position-based ad hoc networks* (EFPA) gives an answer to this question to attain the goal of the area-based approach, namely to enlarge the coverage area with less retransmissions. In figure 6, although nodes *C* and *D*

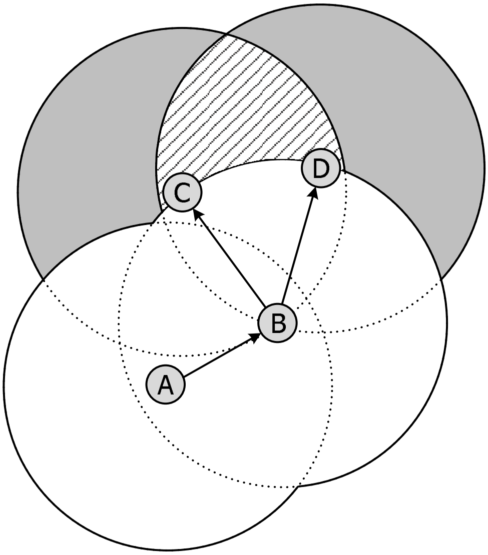


Fig. 6. A scenario where two nodes are placed at the boundary of the sender

have the same distance to the sender *B*, each additional coverage is different. The sender *B* has already covered the same amount of the coverage of nodes *C* and *D*, whereas the coverage of node *C* overlaps with the coverage of the former sender *A*. Thus, a node covers more additional area as it is farther away from the former sender. In other words, a node which corresponds to the direction of packet transmission has more additional coverage than other nodes.

EFPA utilizes the concept of a direction of packet transmission to reduce redundancy and to spread a packet rapidly throughout the network. EFPA proceeds in three phases: priority node selection, waiting time allocation, and packet drop. Once a node receives a packet with priority given by the sender, it immediately retransmits the packet to all its neighbors. Otherwise, a node with no priority examines the necessity of retransmission during a waiting time.

Each node collects the locations and the list of all one-hop neighbors via hello packets in advance. When a node (i.e., the current sender) decides to retransmit the packet, it calculates the distance to each one-hop neighbors and the direction of packet transmission from the former sender. The current sender embeds the ID of a node, which is the farthest towards the direction of packet transmission, in the packet. Figure 7 depicts the direction of packet transmission and priority node selection. Nodes *A* and *B* are the former and the current senders, respectively. The direction is merely determined as the line extension between the former and the current senders. As can be seen in the figure, node *C* is closer to the boundary of the range of the sender *B* than node *D*. However, the priority is given to node *D* since it is reached the farthest towards the direction of packet transmission.

Figure 8 shows an example to describe the difference of packet delivery between two cases where one considers the direction of packet transmission and another does not. These two cases are primarily based on the distance between nodes to determine waiting times: a waiting time is inversely proportional to the distance. In the case where nodes consider a direction of packet transmission, node n_1 which receives a packet from the source

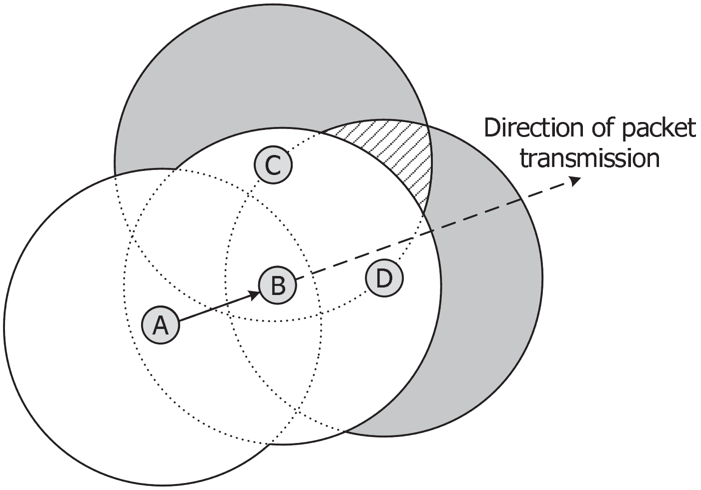


Fig. 7. The direction of packet transmission and priority node selection

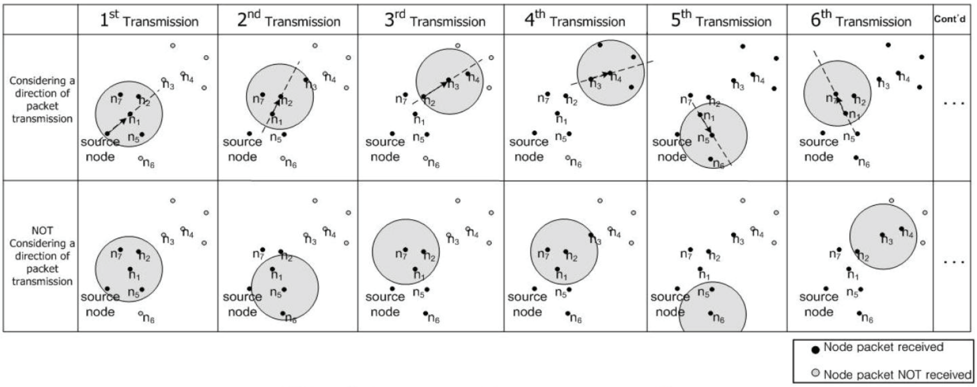


Fig. 8. Comparison of transmission procedure by whether a node considers a direction of packet transmission or not [taken from (Jaegal & Lee, 2008)]

rebroadcasts it to all its neighbors n_2, n_5, and n_7. The source receives the packet as well, but here it is neglected since it has already transmitted once. After retransmission of node n_1, nodes n_2, n_3, and n_4 as priority nodes deliver the packet in order. Nodes n_5 and n_7 retransmit the packet after their waiting times expire. On the other hand, in the case where nodes do not consider a direction of packet transmission, node n_5 becomes the sender after node n_1 because it has the shortest waiting time among the candidates, that is, node n_5 is the farthest node from node n_1. The rest of nodes also deliver the packet after own waiting time.

The priority is embedded in a packet by the sender. When candidates except for the priority node receive the packet, they set their own waiting times determined in inverse proportion to the distance from the sender as follows.

$$W(d,R) = W_{\max} - \frac{d \times W_{\max}}{T} \tag{4}$$

where $W_{\max}$ is the predetermined, constant maximum waiting time. d is the distance from the sender. T denotes the transmission range of the sender. Typically, the transmission range T of each node is assumed as identical. Thus, a waiting time is merely a function of the distance from the sender.

During the waiting time, each node examines the necessity of retransmission based on neighbor knowledge. Specifically, a packet is transmitted with the list of its one-hop neighbors of the sender. A node receiving the packet eliminates nodes of its neighbor's list that are included in the list of the sender. If all its neighbors are removed from the list, the node discards the packet since all the neighbors have already received the packet. Let $N(S)$ be the neighbor set of the sender S and $N(R)$ be the neighbor set of a receiver R. The packet drop is stated as follows:

1. If node R receives the duplicate packet from node S, the neighbor list of node R updates the list, namely $N(R) = N(R) - N(S) - \{S\}$
2. If node R has the empty neighbor set $N(R) = \emptyset$, it drops the packet
3. Whereas repeat steps (1) ~ (2) until the waiting time expires

In (Jaegal & Lee, 2008), the performance comparison of blind flooding, geoflood, FONIAH, and EFPA was well researched with metrics: average number of transmissions, average delay, and flooding completion time were used to capture broadcast redundancy, end-to-end latency, and time spent on packet delivery throughout the network, respectively. The simulation was performed using (ns-2, 1999) with GPSR (greedy perimeter stateless routing) (Karp & Kung, 2000). Nodes are randomly deployed in an experiment region of 2000*m* × 2000*m*. Each node transmits a packet with IEEE 802.11 MAC protocol and has the identical transmission range of 250*m*. Figures 9, 10, and 11 depict the performance of blind flooding, geoflood, FONIAH, and EFPA. Irrespective of the maximum waiting time, blinding flooding requires a node to rebroadcast a packet without a waiting time. Thus, this simplest approach has less delay than other schemes in spite of the heaviest overhead. An area-based scheme (geoflood) and its enhancement (FONIAH) have similar reduction on the number of transmission with increase of node where $W_{\max}$ = 0.05*sec* and $W_{\max}$ = 0.35*sec* When the maximum waiting time is set to 0.05*sec*, the difference between the time spent (including average delay and completion time) of geoflood and FONIAH is small, whereas noticeable

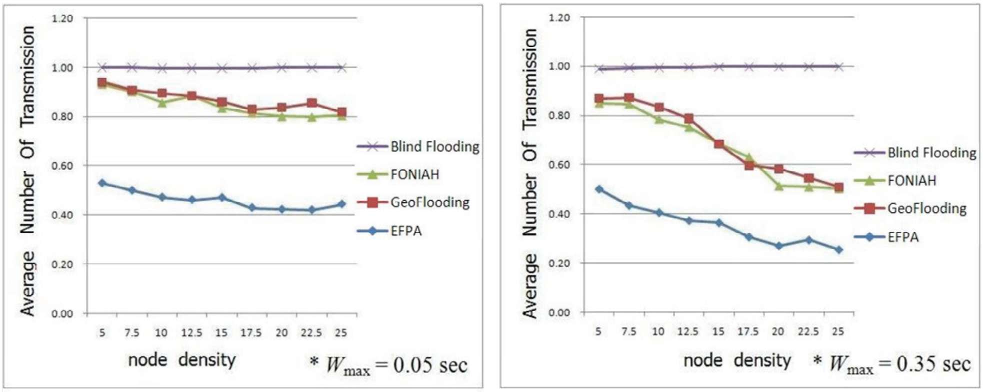


Fig. 9. Average number of transmissions with varying the maximum waiting time as 0.05*sec* and 0.35*sec* [taken from (Jaegal & Lee, 2008)]

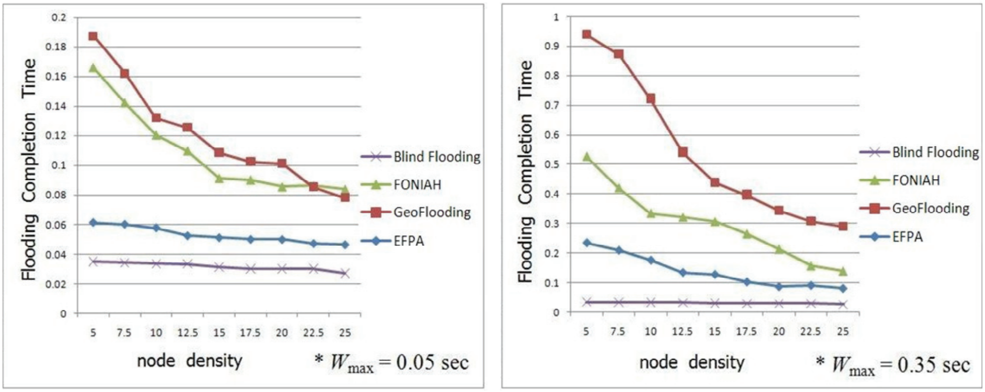


Fig. 10. Average delay with varying the maximum waiting time as 0.05*sec* and 0.35*sec* [taken from (Jaegal & Lee, 2008)]

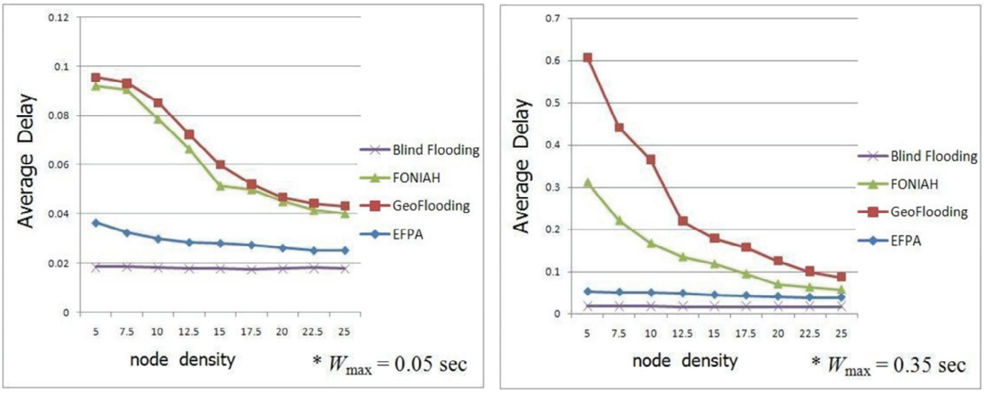


Fig. 11. Flooding completion time with varying the maximum waiting time as 0.05*sec* and 0.35*sec* [taken from (Jaegal & Lee, 2008)]

difference occurs in case when $W_{\max} = 0.35$*sec*. Latency and completion time of FONIAH are considerably lower than those of geoflood as nodes are densely deployed. EEPA achieves rapid delivery throughout the network as fast as blind flooding and it noticeably alleviates the number of transmissions.

5. Conclusion

In this chapter, we discussed the issues behind supporting efficient broadcasting for MANET and previously published broadcasting schemes. The key argument of efficiency in broadcasting is reducing the amount of overhead introduced during the propagation of a packet to nodes in the network. The reason is that MANET is one of resource-constrained networks such as mobile networks and wireless sensor networks. More precisely, collision and contention are likely to occur due to wireless resource sharing under the condition that the resource is strictly limited. In addition to the problems, energy consumption is an important consideration.

The optimal reliable broadcasting is known as NP-complete even if each node has the global topology information. Hence, many of the broadcasting schemes require each node to listen to redundant packets during a short waiting time to examine the necessity of transmission. Since the waiting time may be a factor increasing end-to-end delay, some broadcasting schemes employs the concept of a hybrid approach to alleviate delay granting a priority to help a node rebroadcast immediately.

With the enhancement of the broadcasting approach, the performance has been considerably improved. Unfortunately, most broadcasting schemes presented here barely ensure the feasibility and practically in the real world because of the underlying assumptions such as static network model and error-free communication. Moreover, using extra devices such as ranging measurements and GPS is costly and power-intensive. Therefore, significant research effort is needed with consideration of high mobility and energy conservation.

6. References

Abramson, N. (1970). The Aloha System-Another Alternative for Computer Communications, *Proceedings of the Fall Joint Computer Conference (AFIPS'70)*, pp. 281-285, Montvale, Nov. 1970

Arango, J.; Degermark, M.; Efrat, A. & Pink, S. (2004). An Efficient Flooding Algorithm for Mobile Ad-hoc Networks, *Proceedings of IEEE Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt'04)*, Cambridge, Mar. 2004

Doherty, L.; Pister, K. S. J. & Ghaoui, L. E. (2001). Convex Position Estimation in Wireless Sensor Network, *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1655-1663, ISBN 0-7803-7016-3, Anchorage, Apr. 2001

Getting, I. A. (1993). Perspective/navigation-The Global Positioning System. *IEEE Spectrum*, Vol.30, Iss.12, Dec. 1993, pp. 36-47, ISSN 0018-9235

Ho, C.; Obraczka, K.; Tsudik, G. & Viswanath, K. (1999). Flooding for Reliable Multicast in Multi-hop Ad Hoc Networks, *Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M'99)*, pp. 64-71, Seattle, Aug. 1999

Jaegal, C. & Lee, C. (2008). An Efficient Flooding Algorithm for Position-based Wireless Ad Hoc Networks, *Proceedings of the International Conference on Convergence and Hybrid Information Technology*, pp. 13-20, ISBN 978-0-7695-3407-7, Busan, Nov. 2008

Jetcheva, J. G.; Hu, Y.; Maltz, D. A. & Johnson, D. B. (2001). A Simple Protocol for Multicast and Broadcast in Mobile Ad Hoc Networks. *Internet Draft of the Internet Engineering Task Force (IETF): draft-ietf-manet-simple-mbcast-01.txt*, Jul. 2001

Johnson, D. B. & Maltz, D. A. (1996). Dynamic Source Routing in Ad Hoc Wireless Networks, In: *Mobile Computing*, Tomasz Imielinski & Henry F. Korth, pp. 153-181, Kluwer Academic Publishers, ISBN 0792396979, Boston

Karp, B. & Kung, H. T. (2000). GPSR: Greedy Perimeter Stateless Routing for Wireless Networks, *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00)*, pp. 243-254, Boston, Aug. 2000

Lee, S. & Ko, C. (2006). An Efficient Neighbor Based Broadcasting for Mobile Ad Hoc Networks, *Proceedings of the International Conference on Computational Science (ICCS'06)*, pp. 1097-1100, ISBN 978-3-540-34381-3, Reading, May 2006

Lim, H. & Kim, C. (2001). Flooding in Wireless Ad Hoc Networks. *Computer Communications Journal*, Vol.24, No.3, Feb. 2001, pp. 353-363, ISSN 0140-3664

Lou, W. & Wu, J. (2002). On Reducing Broadcast Redundancy in Ad Hoc Wireless Networks. *IEEE Transactions on Mobile Computing*, Vol.1, Iss.2, Apr.-Jun. 2002, pp. 111-122, ISSN 1536-1233

Ni, S.; Tseng, Y.; Chen, Y. & Sheu, J. (1999). The Broadcast Storm Problem in a Mobile Ad Hoc Network, *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99)*, pp. 151-162, ISBN 1-58113-142-9, Seattle, Aug. 1999

Niculescu, D. & Nath, B. (2001). Ad-hoc Positioning System (APS), *Proceedings of IEEE GLOBECOM 2001*, pp. 2926-2931, ISBN 0-7803-7206-9, San Antonio, Nov. 2001

Park, V. D. & Corson, M. S. (1997). A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks, *Proceedings of the 16th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'97)*, pp. 1405-1413, ISBN 0-8186-7780-5, Kobe, Apr. 1997

Pearlman, M. R. & Hass, Z. J. (1999). Determining the Optimal Configuration for the Zone Routing Protocol. *IEEE Journal on Selected Areas in Communications*, Vol.17, Iss.8, Aug. 1999, pp. 1395-1414, ISSN 0733-8716

Peng, W. & Lu, X. (2000). On the Reduction of Broadcast Redundancy in Mobile Ad Hoc Networks, *Proceedings of the 1st Annual International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'00)*, pp.129-130, ISBN 0-7803-6534-8, Boston, Aug. 2000

Perkins, C. E. & Royer, E. M. (1999). Ad-hoc On-demand Distance Vector Routing, *Proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90-100, ISBN 0-7695-0025-0, New Orleans, Feb. 1999

Qayyum, A.; Viennot, L. & Laouiti, A. (2002). Multipoint Relaying for Flooding Broadcast Messages in Mobile Wireless Networks, *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, pp. 3866-3875, ISBN 0-7695-1435-9, Big Island, Jan. 2002

Ryu, J.; Kim, M.; Hwang, S. & Han, K. (2004). An Adaptive Probabilistic Broadcast Scheme for Ad-Hoc Networks, *Proceedings of the 7th IEEE International Conference on High Speed Networks and Multimedia Communications (HSNMC'04)*, pp. 646-654, ISBN 978-3-540- 22262-0, Toulouse, Jun.-Jul. 2004

Saadawi, T. & Ephremides, A. (1981). Analysis, Stability, and Optimization of Slotted ALOHA with a Finite Number of Buffered Users. *IEEE Transactions on Automatic Control*, Vol.26, Iss.3, Jun. 1981, pp. 680-689, ISSN 0018-9286

Shang, Y. & Ruml, W. (2003). Improved MDS-based Localization, *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'04)*, pp. 2640-2651, ISBN 0-7803-8355-9, Hong Kong, Mar. 2004

Tseng, Y.; Ni, S. & Shih, E. (2001). Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Network, *Proceedings of the 21st International Conference on Distributed Computing Systems*, pp. 481-488, ISBN 0-7695-1077-9, Mesa, Apr. 2001

Williams, B. & Camp, T. (2002). Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks, *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 194-205, ISBN 1-58113-501-7, Lausanne, Jun. 2002

THE CMU MONARCH GROUP. Wireless and Mobility Extensions to ns-2.
http://www.monarch.cs.cmu.edu/cmu-ns.html, Oct. 1999

28

Energy Efficient Resource Allocation in Cognitive Radio Wireless Ad Hoc Networks

Song Gao[1], Lijun Qian[1], and D.R. Vaman[2]
[1]*Prairie View A&M University,*
[2]*CeBCom Research Center*
U.S.A

1. Introduction

Recent technological advances have resulted in the development of wireless ad hoc networks which are envisioned to provide rapid on-demand network deployment due to their self-configurability and lack of pre-deploy infrastructure requirements. These devices generally have small form factors, and have embedded storage, processing and communication ability I. F. Akyildiz (2009). With the growing proliferation of such wireless devices, the spectrum is increasingly getting congested. However, it has also been pointed out in several recent measurement reports that the spectrum are highly under-utilized FCC (2002). In order to achieve much better spectrum utilization and viable frequency planning, Cognitive Radios (CRs) are under development to dynamically capture the unoccupied spectrum J. Mitola (1999). Many challenges arise with such dynamic and hierarchical means of accessing the spectrum, especially for the dynamic resource allocation of CR users by adapting their transmission and reception parameters to the varying spectrum condition while adhering to power constraints and diverse quality of service (QoS) requirements (see, for example, S. Tao (2006); Q. Zhao (2007)).

In this chapter, an energy constrained wireless CR ad hoc network is considered, where each node is equipped with CR and has limited battery energy. One of the critical performance measures of such networks is the network lifetime. Additionally, due to the infrastructureless nature of ad hoc networks, distributed resource management scheme is desired to coordinate and maintain communications between each transmitting receiving pair. In this context, the present chapter provides a framework of distributed energy efficient spectrum access and resource allocation in wireless CR ad hoc networks that employ orthogonal frequency division multiple access (OFDMA) K. Fazel (2003); A. Pandharipande (2002) at the physical layer. OFDMA is well suited for CR because it is agile in selecting and allocating subcarriers dynamically and it facilitates decoding at the receiving end of each subcarrier J. Bazerque (2007). In addition, multi-carrier sensing can be exploited to reduce sensing time I. F. Akyildiz (2006).

Each emerging CR user will select its subcarriers and determine its transmission parameters individually by solving an optimization problem. The optimization objective is to minimize its energy consumption per bit[1] while satisfying its QoS requirements and power limits.

[1] which is defined as the ratio of the total transmission and reception power consumption over available subcarrier set to its achieved throughput

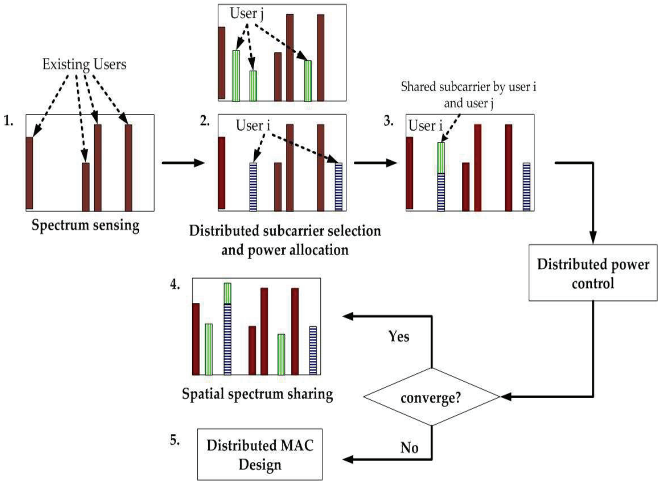


Fig. 1. Block diagram of the proposed distributed resource allocation algorithm

Compared with the power minimization with respect to target data rate constraints S. Tao (2006) or throughput maximization under power upper bound Q. Zhao (2007), this objective function, which measures the total energy consumed for reliable information bits transmitted, is particularly suitable for energy constrained networks where the network lifetime is a critical metric.

Although the emerging CR users will not cause harmful interference to the existing users, they may choose the same subcarriers in the same time slot independently, and thus co-channel interference may be introduced. In this work, we allow multiple new users to share the same subcarriers as long as their respective Signal-to-Interference-and-Noise-Ratio (SINR) is acceptable. This may be achieved by distributed power control R. Yates (1995), which converges very fast. The flow chart of the proposed distributed energy efficient spectrum access and resource allocation scheme is highlighted in Fig. 1, where step 2 corresponds to the constrained optimization performed by each emerging user individually. More detailed illustrations of the flow chart are given in section 4.

Resource allocation problem in wireless ad hoc networks has been extensively investigated in the literature. In G. Kulkarni (2005), the resource allocation problem is explored for OFDMA-based wireless ad hoc network by directly adopting distributed power control scheme for the power and bits allocation on all subcarriers to improve power efficiency. A greedy algorithm is proposed for best subcarrier selection in CR networks employing multicarrier CDMA Q. Qu (2008), and distributed power control is performed thereafter to resolve co-channel interference. An Asynchronous Distributed Pricing (ADP) scheme is proposed in J. Huang (2006), where the users need to exchange information indicating the interference caused by each user to others. In the context of CR enabled wireless sensor network (WSN) S. Tao (2006), a two-step algorithm is proposed to tackle the allocation problem: channel assignment with objective of minimizing transmission power and channel contention to reserve the subcarrier set for transmission by intended transmitters, while the interference spectrum mask is assumed to be known a priori. The authors of Q. Zhao (2007) address the opportunistic spectrum access (OSA) problem in WSN, in which a distributed channel allocation problem is modeled by a partially observable Markov decision process framework (POMDP) while assuming the transition probability of each channel is known. In Y. T. Hou (2007), the CR spectrum sharing problem is formulated in multi-hop networks with objective to minimize the *space-bandwidth product* (SBP). However, the transmission power allocated on each subcarrier is assumed to be the same which may lead to significant performance loss. The effect of power control is analyzed in a subsequent work Y. Shi (2007). Dynamic Frequency Hopping Community (DFHC) is proposed in W. Hu (2007) for the spectrum sharing in CR based IEEE 802.22 wireless regional area networks (WRANs) to ensure QoS satisfaction and reliable protection to licensed users.

In this chapter, a new constrained optimization problem is formulated and solved that minimizing energy per bit across users subject to QoS and power constraints in a multi-user ad hoc network. A novel concept, "*energy-efficient* waterfilling", is given in this section that is fundamentally different from the *rate-adaptive* waterfilling or *margin-adaptive* waterfilling[2]. In this case the optimal point is located in the constraint interval rather than on the boundary. In fact, the *rate-adaptive* and *margin-adaptive* waterfilling can be considered as special cases of the *energy-efficient* waterfilling presented in this work. The results obtained provide a valuable insight that the optimal solution of energy efficient resource allocation is *not* best subcarrier selection for multiple transmitting receiving pairs in an OFDMA network S. Gao (2008). The proposed distributed subcarrier selection and power allocation scheme provides an *efficient* and *practical* solution for dynamic spectrum access in CR wireless ad hoc networks employing OFDMA. By combining the optimal resource allocation of individual users and distributed power control, the proposed method guarantees fast convergence speed, computational efficiency and implementation simplicity. Motivated by iterative waterfilling (IWF) algorithm in W. Yu (2002), another distributed solution may be obtained by solving the multi-user distributed channel and power allocation problem iteratively. However, it may take many steps for the iterative algorithm to converge if it converges at all and the delay may be too large to be tolerable. The cost of the additional computation complexity is high. On the contrary, the proposed optimal resource allocation of individual users is easy to obtain and distributed power control algorithm has well-known fast convergence speed. Furthermore, it will be shown that the proposed distributed algorithm performs closely to the global optimal point.

2. System model

We consider an energy constrained CR OFDMA network of N communicating pairs. Both transmitter i and receiver j is indexed by $\mathcal{N} := \{1, 2, ..., N\}$. If $j = i$, receiver j is said to be the intended receiver of transmitter i. The transmission system is assumed to be a time-slotted OFDMA system with fixed time slot duration T_S. Slot synchronization is assumed to be achieved through a beaconing mechanism. Before each time slot, a guard interval is inserted to achieve synchronization, perform spectrum detection as well as resource allocation (based on the proposed scheme). Inter-carrier interference (ICI) caused by frequency offset of the side lobes pertaining to transmitter i is not considered in this work (which can be mitigated by windowing the OFDM signal in the time domain or adaptively deactivating adjacent subcarriers T. Weiss (2004)).

A frequency selective Rayleigh fading channel is assumed at the physical layer, and the entire spectrum is appropriately divided into M subcarriers to guarantee each subcarrier

[2] The optimal allocation strategy with objective to minimize power or maximize throughput is named *margin-adaptive* and *rate-adaptive* waterfilling over frequency channels, respectively.

experiencing flat Rayleigh fading S. Kondo (1996). We label the subcarrier set available to the transmitter receiver pair i after spectrum detection by $\mathcal{L}_i \subset \{1,2,...,M\}$. Let $\mathbf{G} := \left\{G_{i,j}^k, i,j \in \mathcal{N}, k \in \mathcal{L}_i\right\}$ denote the subcarrier fading coefficient matrix, where $G_{i,j}^k$ stands for the sub-channel coefficient gain from transmitter i to receiver j over subcarrier k. $G_{i,j}^k = |H_{i,j}^k(f)|^2$, where $|H_{i,j}^k(f)|$ is the transfer function. It is assumed that $\mathbf{G}$ adheres to a block fading channel model which remains invariant over blocks (coherence time slots) of size T_S and uncorrelated across successive blocks. The noise is assumed to be additive white Gaussian noise (AWGN), with variance $\sigma_{i,k}^2$ over subcarrier k of receiver i. We define $\mathbf{P} := \left\{p_i^k, p_i^k \geq 0, i \in \mathcal{N}, k \in \mathcal{L}_i\right\}$ as the transmission power allocation matrix for all users in $\mathcal{N}$ over the entire available subcarrier set $\bigcup_{i \in \mathcal{N}} \mathcal{L}_i$, where p_i^k is the power allocated over subcarrier k for transmitter i. For each transmitter i, the *power vector* can be formed as

$$\boldsymbol{p}_i := [p_i^1, p_i^2, ..., p_i^M]^T \tag{1}$$

If the k^{th} subcarrier is not available for transmitter i, $p_i^k = 0$. Each node is not only energy limited but also has peak power constraint, i.e., $\sum_{k \in \mathcal{L}_i} p_i^k \leq p_i^{max}$. The set of all feasible power vector of transmitter i is denoted by $\mathcal{P}_i$

$$\mathcal{P}_i := \left\{\boldsymbol{p}_i \subset \prod_{k \in \mathcal{L}_i} [0, p_i^{max}], \sum_{k \in \mathcal{L}_i} p_i^k \leq p_i^{max}\right\} \tag{2}$$

The signal to interference plus noise ratio (*SINR*) of receiver i over subcarrier k (γ_i^k) can be expressed as

$$\begin{aligned} \gamma_i^k(p_i^k) &= \alpha_i^k(p_j^k) \cdot p_i^k \\ \alpha_i^k(p_j^k) &= \frac{G_{i,i}^k}{\sum_{j \neq i, j \in \mathcal{N}} G_{j,i}^k \cdot p_j^k + \sigma_{i,k}^2} \end{aligned} \tag{3}$$

where α_i^k is defined as the channel state information (CSI) which treats all interference as background noise. α_i^k can be measured at the receiver side and is assumed to be known by the corresponding transmitter through a reciprocal common control channel.

When all users divide the spectrum in the same fashion without coordination, it is referred to as a *Parallel Gaussian Interference Channel* which leads to a tractable inner bound to the capacity region of the interference model. The achievable maximum data rate for each user (Shannon's capacity formula) is

$$\frac{c_i(\boldsymbol{p}_i)}{B_i^k} = \sum_{k \in \mathcal{L}_i} \frac{c_i^k\left(p_i^k\right)}{B_i^k} = \sum_{\substack{k \in \mathcal{L}_i, \\ p_i^k \in \mathcal{P}_i}} \log_2\left(1 + \alpha_i^k(p_j^k) \cdot p_i^k\right) \tag{4}$$

where B_i^k is the equally divided subcarrier bandwidth for transmitter i. Without loss of generality, B_i^k is assumed to be unity in this work. The noise is assumed to be independent of the symbols and has variance σ^2 for all receivers over entire available subcarrier set. Furthermore, all communicating transmitter and receiver pairs are assumed to have diverse

QoS requirements specified by $\sum_{k\in\mathcal{L}_i} c_i^k \geq r_i^{tar}$, where r_i^{tar} is the target data rate of transmitter i.

In an energy constrained network (such as a wireless sensor network), reception power is not negligible since it is generally comparable to the transmission power. We denote the receiving power as p_i^r which is treated as a constant value for all receivers[3].

Aiming at achieving high energy efficiency, the energy consumption per information bit for transmitter receiver pair i in each time slot is

$$e_i(\boldsymbol{p}_i, c_i) := \frac{\sum_{k\in\mathcal{L}_i} p_i^k + p_i^r}{\sum_{k\in\mathcal{L}_i} c_i^k} \tag{5}$$

Let $\mathcal{S}_i(\boldsymbol{p}_i, c_i)$ denote the set of all power and rate allocations satisfying QoS requirements and power limit constraints for transmitter i, and it is given by

$$\mathcal{S}_i(\boldsymbol{p}_i, c_i) = \left\{\boldsymbol{p}_i, c_i : \boldsymbol{p}_i \in \mathcal{P}_i,\ c_i \geq r_i^{tar},\ i \in \mathcal{N}\right\} \tag{6}$$

Given the above system assumptions and the objective defined in (5), we end up with the following constrained optimization problem.

$$\begin{aligned} &\min_{p_i^k, c_i^k \in \mathcal{S}_i} e_i(\boldsymbol{p}_i, c_i) \\ s.t.\quad & c_i(\boldsymbol{p}_i) \geq r_i^{tar}, \forall i \in \mathcal{N} \\ & \sum_{k\in\mathcal{L}_i} p_i^k \leq p_i^{max}, \forall i \in \mathcal{N} \end{aligned} \tag{7}$$

3. Energy efficient resource allocation algorithm

The problem (7) is a combinatorial optimization problem and the objective function is not convex/concave. Constrained optimization techniques can be applied here but with considerable computational complexity. Hence, a two-stage algorithm is proposed in this section to decouple the original problem into an unconstrained problem in order to reduce the search space. After the optimal solution for the unconstrained problem is obtained in stage 1, the power and data rate constraints will be examined in search of the final optimal solution. It should be noted that the solution of the unconstrained problem provides the optimal operating point which can be taken as the benchmark for the system design.

3.1 Unconstrained energy efficient resource allocation

We define the unconstrained energy per bit function as

$$f(\hat{\boldsymbol{p}}_i, \boldsymbol{\alpha}_i) := \frac{\sum\limits_{k\in\mathcal{L}_i} \hat{p}_i^k + p_i^r}{\sum\limits_{k\in\mathcal{L}_i} \log_2\left(1 + \alpha_i^k \cdot \hat{p}_i^k\right)} \tag{8}$$

[3] In this work, we consider an energy constrained CR ad-hoc wireless network where the throughput requirement is usually not as high as the throughput demanding networks such that the baseband symbol rate is not very high. Thus this baseband power consumption is quite small compared with the power consumption in the RF circuitry. Hence, we neglect the energy consumption of baseband signal processing blocks to simplify the model, and the receiving power equals to the power consumption in the RF circuitry and can be treated as a constant S. Cui (2005)

where ˆ is used to represent the variables in the unconstrained optimization domain and $\boldsymbol{\alpha}_i = [\alpha_i^1, \alpha_i^2, \ldots, \alpha_i^k]$. It is assumed $f(\hat{\boldsymbol{p}}_i, \boldsymbol{\alpha}_i)$ is a continuous function in $\mathbb{R}_M^+$. We define the unconstrained optimal energy per bit for transmitter i of (8) as $\hat{\zeta}_i^* = \min f(\hat{\boldsymbol{p}}_i, \boldsymbol{\alpha}_i)$.

3.1.1 Energy efficient waterfilling

Theorem 1 *Given the channel state information* $\boldsymbol{\alpha}_i$ *and noise power, power allocation* $\hat{\boldsymbol{p}}_i^* = [\hat{p}_i^{1*}, \hat{p}_i^{2*}, \ldots, \hat{p}_i^{k*}, k \in \mathcal{L}_i]$ *is defined as the* unconstrained optimal power allocation *by satisfying*

$$f(\hat{\boldsymbol{p}}_i^*, \boldsymbol{\alpha}_i) \leq f(\hat{\boldsymbol{p}}_i, \boldsymbol{\alpha}_i), \ \forall \hat{\boldsymbol{p}}_i \subset \mathbb{R}^{M+} \tag{9}$$

Then the unconstrained optimal power allocation *can be obtained by solving the following equations:*

$$\begin{aligned} \hat{p}_i^{k*} &= \max\left\{\log_2 e \cdot \hat{\zeta}_i^* - \frac{1}{\alpha_i^k}, 0\right\} \\ \hat{\zeta}_i^* &= \frac{\sum\limits_{k\in\mathcal{L}_i} \hat{p}_i^{k*} + p_i^r}{\sum\limits_{k\in\mathcal{L}_i} \log_2\left(1 + \alpha_i^k \cdot \hat{p}_i^{k*}\right)} \end{aligned} \tag{10}$$

Proof: Differentiating $f(\hat{\boldsymbol{p}}_i, \boldsymbol{\alpha}_i)$ with respect to $\hat{p}_i^k$ (which stands for the power allocated for transmitter i on subcarrier k), we obtain the equations (10). The details of the derivation are given in **Appendix A.** ■

The value of $\hat{\zeta}_i^*$ can be obtained by using a numerical method which will in turn determine $\hat{\boldsymbol{p}}_i^*$. It is observed that $\hat{\boldsymbol{p}}_i^*$ has similar type of *rate-adaptive* / *margin-adaptive* waterfilling results, and we name it *energy-efficient* waterfilling. Whereas, the fundamental difference among them lies in the positions of their respective optimal points. The *rate-adaptive* waterfilling maximizes the achievable data rate under power upper bound, and *margin-adaptive* waterfilling minimizes the total transmission power subject to a fixed rate constraint W. Yu (2002), both of which achieve their optimality at the boundary of the constraints. On the contrary, the proposed *energy-efficient* waterfilling selects the most energy-efficient operating point (in other words, it selects the optimal data rate that minimizes the energy consumption per information bit) while adhering to the QoS requirements and power limits. In this case, optimality is usually obtained in the constraint interval rather than on the boundary. In fact, the *rate-adaptive* and *margin-adaptive* waterfilling can be considered as special cases of the *energy-efficient* waterfilling solved. If we set $\sum_{k\in\mathcal{L}_i} p_i^k = p_{con} \leq p_i^{max}$ or $\sum_{k\in\mathcal{L}_i} c_i^k(p_i^k) = r_i^{tar}$, the *energy-efficient* allocation problem is reduced to the well explored *rate-adaptive* or *margin-adaptive* waterfilling problem.

3.1.2 Feasibility region

The existence of the solution for the unconstrained optimization ($\min f(\hat{\boldsymbol{p}}_i, \boldsymbol{\alpha}_i)$) depends on the subcarrier condition α_i^k if we assume other system parameters (e.g. bandwidth, maximal

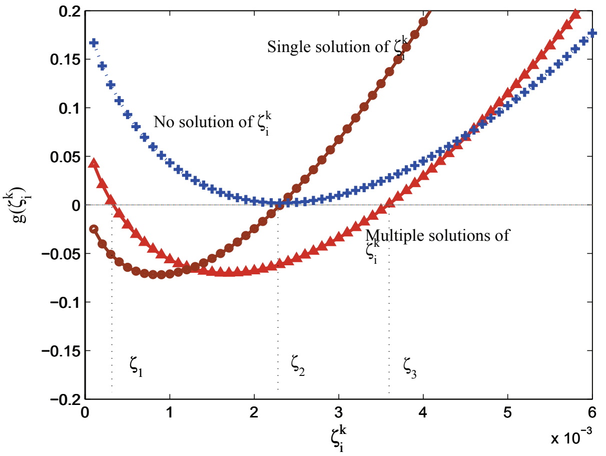


Fig. 2. Feasible solution vs. subcarrier condition

power, etc.) are fixed. From (10), if we take $\hat{\boldsymbol{p}}_i$ into the expression of $\hat{\zeta}_i^*$, we can get

$$
\begin{aligned}
\hat{\zeta}_i &= \frac{\Gamma(\hat{\boldsymbol{p}}_i^*)\cdot \log_2^e \cdot \hat{\zeta}_i - \sum\limits_{k\in\mathcal{L}_i} \frac{1}{\alpha_i^k}\cdot I(\hat{p}_i^{k*}) + p_i^r}{\Gamma(\hat{\boldsymbol{p}}_i^*)\cdot \log_2(\log_2^e \cdot \hat{\zeta}_i) + \sum\limits_{k\in\mathcal{L}_i} \log_2(\alpha_i^k)\cdot I(\hat{p}_i^{k*})} \\
I(\hat{p}_i^{k*}) &= \begin{cases} 1, & \hat{p}_i^{k*} > 0 \\ 0, & \hat{p}_i^{k*} \le 0 \end{cases}
\end{aligned}
\tag{11}
$$

where $\Gamma(X)$ is defined as the cardinality of nonzero elements in vector X. The optimal solution $\hat{\zeta}_i$ can be determined by solving equation (11), and the existence of the optimal solution is influenced by the subcarrier condition α_i^k. This is illustrated in Fig.2. A unique optimal solution ($\hat{\zeta}_{i,2}^*$) is obtained when the subcarrier condition is good; while no feasible solution exists when the subcarrier condition is bad. Multiple solutions may be obtained when the subcarrier condition is in the middle range. In this case, only the larger solution ($\hat{\zeta}_{i,3}^*$) is the feasible solution, and this can be verified by checking the corresponding power allocation, i.e., all the allocated power should be non-negative.

The feasibility condition of the unconstrained optimization problem is given in the following theorem.

Theorem 2 *Denote the maximal optimal solution of* $\hat{\zeta}_i^*$ *as* $\hat{\zeta}_i^{max}$ *and the channel gain of the* best *subcarrier as* α_i^{τ}, $\alpha_i^{\tau} = \max\{\alpha_i^k, \forall k \in \mathcal{L}_i\}$. *The feasibility condition for the existence of the optimal solution of the* energy efficient *waterfilling (10) is given by* $\alpha_i^{\tau} \ge \frac{\ln 2}{\hat{\zeta}_i^{max}}$.

Proof: 1) Necessity: From the optimal solution of *energy efficient* waterfilling (10), it is observed the amount of allocated power is determined by the subcarrier condition α_i^k, specifically, more power should be allocated on better subcarrier. Thus, if the optimal solution exists, at least the power allocated on the *best* subcarrier should be non-negative, i.e., $\hat{p}_i^{\tau} = \log_2^e \cdot \hat{\zeta}_i^{max} - \frac{1}{\alpha_i^{\tau}} \ge 0 \Longrightarrow \alpha_i^{\tau} \ge \frac{\ln 2}{\hat{\zeta}_i^{max}}$.

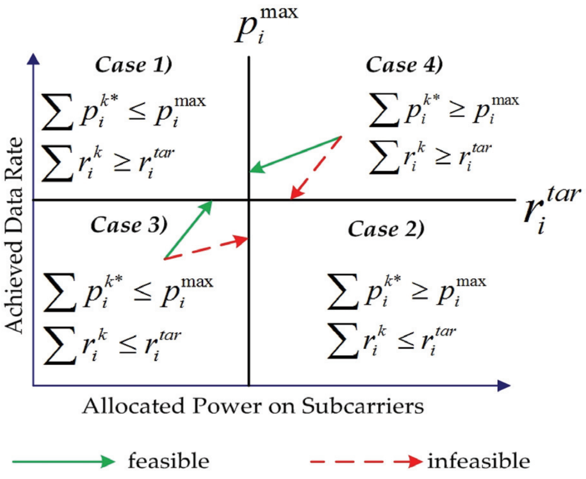

$p_i^{\max}$
Case 1)
$\sum p_i^{k*} \le p_i^{\max}$
$\sum r_i^k \ge r_i^{tar}$
Case 4)
$\sum p_i^{k*} \ge p_i^{\max}$
$\sum r_i^k \ge r_i^{tar}$
r_i^{tar}
Case 3)
$\sum p_i^{k*} \le p_i^{\max}$
$\sum r_i^k \le r_i^{tar}$
Case 2)
$\sum p_i^{k*} \ge p_i^{\max}$
$\sum r_i^k \le r_i^{tar}$
Achieved Data Rate
Allocated Power on Subcarriers
feasible
infeasible

Fig. 3. Partition of the solution space of the constrained optimization problem

2) Sufficiency: We prove this part by contradiction. If $\alpha_i^\tau \ge \frac{\ln 2}{\hat{\zeta}_i^{max}}$ and still no optimal solution exists, which implies that the power allocated on the entire subcarrier set is negative, i.e., $\hat{p}_i^{k*} < 0, \forall k \in \mathcal{L}_i$, then $\hat{\zeta}_i^{max} - \frac{\ln 2}{\alpha_i^\tau} < 0 \Longrightarrow \alpha_i^\tau < \frac{\ln 2}{\hat{\zeta}_i^{max}}$, which contradicts the condition $\alpha_i^\tau \ge \frac{\ln 2}{\hat{\zeta}_i^{max}}$. This completes the proof. ■

Theorem 2 suggests that it is sufficient to check the best available subcarrier in order to determine the feasibility of the unconstrained optimization problem.

3.2 Constrained energy-efficient allocation algorithm

Given the unconstrained optimal solution $\hat{\boldsymbol{p}}_i^*, i \in \mathcal{N}$, the previous section offers the optimal operating point with best energy efficiency of each individual user. However, some users may not satisfy their respective data rate and/or power constraints when operating at this point. In this section, we partition the solution space of the constrained optimization problem (7) into four sub-spaces based on the power and data rate constraints, as highlighted in Fig.3.

1. $\sum\limits_{k \in \mathcal{L}_i} \hat{p}_i^{k*} \le p_i^{max}$ and $\sum\limits_{k \in \mathcal{L}_i} c_i^k(\hat{p}_i^{k*}) \ge r_i^{tar}$.
   In this case, the unconstrained optimal solution $\hat{\boldsymbol{p}}_i^*$ of (10) satisfies the sum-power and rate requirement constraints. Apparently $\hat{\boldsymbol{p}}_i^*$ is the optimal solution of the original problem (7).

2. $\sum\limits_{k \in \mathcal{L}_i} \hat{p}_i^{k*} \ge p_i^{max}$ and $\sum\limits_{k \in \mathcal{L}_i} c_i^k(\hat{p}_i^{k*}) < r_i^{tar}$ or $\sum\limits_{k \in \mathcal{L}_i} \hat{p}_i^{k*} > p_i^{max}$ and $\sum\limits_{k \in \mathcal{L}_i} c_i^k(\hat{p}_i^{k*}) \le r_i^{tar}$.
   In this case, the allocated power has already exceeded the sum-power constraint, but the rate requirement is still not met, even under the optimal subcarrier selection and power allocation. Therefore, there is no feasible solution for the original problem (7).

3. $\sum\limits_{k \in \mathcal{L}_i} \hat{p}_i^{k*} < p_i^{max}$ and $\sum\limits_{k \in \mathcal{L}_i} c_i^k(\hat{p}_i^{k*}) < r_i^{tar}$.
   If both the power allocated on all subcarriers does not reach the maximal power bound and the data rate requirement is not met, the power should be increased to achieve data rate requirement under the maximal power bound.

Based on (7) and (10), we can modify the original problem as

$$\begin{aligned}
&\min_{\hat{p}_i^k+\triangle p_i^k\in\mathcal{S}_i} \frac{\sum\limits_{k\in\mathcal{K}_i}\left(\hat{p}_i^{k*}+\triangle p_i^k\right)+p_i^r}{\sum\limits_{k\in\mathcal{K}_i}\log_2\left(1+\alpha_i^k\cdot\left(\hat{p}_i^{k*}+\triangle p_i^k\right)\right)}\\
&s.t. \sum_{k\in\mathcal{K}_i} c_i^k(\hat{p}_i^{k*}+\triangle p_i^k)\geq r_i^{tar}, \forall i\in\mathcal{N}\\
&\sum_{k\in\mathcal{K}_i}\left(\hat{p}_i^{k*}+\triangle p_i^k\right)\leq p_i^{max}, \forall i\in\mathcal{N}
\end{aligned} \tag{12}$$

where $\mathcal{K}_i$ is defined as the selected subcarrier set through the optimal *energy efficient* waterfilling solution, $\mathcal{K}_i\subset\mathcal{L}_i$. If we increase the power on any one of the subcarriers, such as the k^{th} subcarrier, the corresponding constrained energy consumption per bit can be expressed as

$$\begin{aligned}
\zeta_i^k &= \frac{\triangle p_i^k+\sum\limits_{k\in\mathcal{K}_i}\hat{p}_i^{k*}+p_i^r}{\sum\limits_{k\in\mathcal{K}_i}\log_2\left(1+\alpha_i^k\cdot\hat{p}_i^{k*}\right)+\triangle c_i^k}\\
\triangle c_i^k &= \log_2\left(\frac{1+\alpha_i^k\cdot\left(\hat{p}_i^{k*}+\triangle p_i^k\right)}{1+\alpha_i^k\cdot\hat{p}_i^{k*}}\right)
\end{aligned} \tag{13}$$

From (10), $\triangle c_i^k$ can be simplified to $\triangle c_i^k=\log_2\left(1+\frac{\triangle p_i^k}{\log_2^e\cdot\hat{\zeta}_i^*}\right)$. It is observed that given the increased power $\triangle p_i^k$ on subcarrier k, the increased data rate does not rely on its subcarrier condition α_i^k, since $\log_2^e\cdot\hat{\zeta}_i^*$ is a constant value for the entire selected subcarrier set. In other words, for any two subcarrier $k,l\in\mathcal{K}_i$ of transmitter $i\in\mathcal{N}$, if $\triangle p_i^k=\triangle p_i^l$, then $\triangle c_i^k=\triangle c_i^l$. And the constrained energy consumption per bit ζ_i^k and ζ_i^l will not vary due to different chosen subcarriers. If we presume, in order to reach the data rate requirement r_i^{tar}, the additional required power $\triangle p_i$ over the selected subcarrier set is known and denoted as $\triangle p_i=\sum_{k\in\mathcal{K}_i}\triangle p_i^k$. Then, problem (12) is equivalent to

$$\begin{aligned}
&\min_{\hat{p}_i^k+\triangle p_i^k\in\mathcal{S}_i} \frac{\triangle p_i+\sum\limits_{k\in\mathcal{K}_i}\hat{p}_i^{k*}+p_i^r}{\sum\limits_{k\in\mathcal{K}_i}\log_2\left(1+\alpha_i^k\cdot\hat{p}_i^{k*}\right)+\sum\limits_{k\in\mathcal{K}_i}\triangle c_i^k}\\
&s.t. \sum_{k\in\mathcal{K}_i} c_i^k(\hat{p}_i^{k*}+\triangle p_i^k)\geq r_i^{tar}, \forall i\in\mathcal{N}\\
&\sum_{k\in\mathcal{K}_i}\left(\hat{p}_i^{k*}+\triangle p_i^k\right)\leq p_i^{max}, \forall i\in\mathcal{N}
\end{aligned} \tag{14}$$

If we assume $\triangle p_i$ has been pre-determined, in order to minimize energy consumption per bit ζ_i, $\sum_{k\in\mathcal{K}_i}c(\hat{\boldsymbol{p}}_i)+\sum_{k\in\mathcal{K}_i}\triangle c_i^k$ need to be maximized. In other words, maximizing

$\sum_{k\in\mathcal{K}_i} \triangle c_i^k$ will result in a classical *rate-adaptive* waterfilling problem.

$$\max \sum_{k\in\mathcal{K}_i} \log_2\left(1+\frac{\triangle p_i^k}{\log_2^e \cdot \hat{\zeta}_i^*}\right)$$
$$s.t. \sum_{k\in\mathcal{K}_i}\left(\hat{p}_i^{k*}+\triangle p_i^k\right) \le p_i^{max}, \forall i \in \mathcal{N} \tag{15}$$

Because $\log_2^e \cdot \hat{\zeta}_i^*$ is a constant value for the entire selected subcarrier set $\mathcal{K}_i$, the solution of the above water filling problem implies that the optimal solution $\triangle p_i^k$ for (15) should be the same for all chosen subcarriers. In other words, given the total required additional power $\triangle p_i$, the power should be equally allocated on all subcarriers, $\triangle p_i^k = \frac{\triangle p_i}{\Gamma(\mathcal{L}_i)}$. Thus, problem (12) can be rewritten as

$$\min \frac{\triangle p_i + \sum\limits_{k\in\mathcal{K}_i} \hat{p}_i^{k*} + p_i^r}{\sum\limits_{k\in\mathcal{K}_i} \log_2\left(1+\alpha_i^k \cdot \hat{p}_i^{k*}\right) + \sum\limits_{k\in\mathcal{K}_i} \triangle c_i^k}$$
$$s.t. \sum_{k\in\mathcal{K}_i} \hat{p}_i^{k*} + \triangle p_i \le p_i^{max}, \forall i \in \mathcal{N} \tag{16}$$

where $\sum \triangle c_i^k = \Gamma(\mathcal{K}_i)\cdot \log_2\left(1+\frac{\triangle p_i}{\Gamma(\mathcal{K}_i)\log_2^e \cdot \hat{\zeta}_i^*}\right)$. Given the unconstrained optimal solution $\hat{\boldsymbol{p}}_i^*$ from stage 1, (16) can be considered as an objective function in terms of variable $\triangle p_i$ bounded by $p_i^{max} - \sum_{k\in\mathcal{K}_i} \hat{p}_i^{k*}$.

Lemma 1 *The constrained energy consumption per bit of problem (16) which is denoted as ζ_i is always worse than the unconstrained optimal energy efficiency $\hat{\zeta}^*$ with respect to the power increase $\triangle p_i^k$, i.e. $\zeta_i \ge \hat{\zeta}^*, \forall \triangle p_i \in \mathbb{R}^+$.*

The proof of Lemma 1 is given in **Appendix B**. Due to the optimality of the unconstrained solution $\hat{\boldsymbol{p}}_i$, the minimal deviation from $\hat{\zeta}_i$ will result in the optimal energy efficiency. Thus, the optimal power increase to satisfy the target data rate will be the minimal required additional power as illustrated in Fig.4. Therefore, the optimal required additional power ($\min \triangle p_i$) to satisfy the data rate requirement r_i^{tar} can be calculated as $\min \triangle p_i = \sum_{k\in\mathcal{K}_i} \triangle p_i^{k*}$.

The minimal required additional power $\triangle p_i^{min} = \min \triangle p_i$ can be derived by

$$\log_2\left(1+\frac{\triangle p_i^{min}}{\Gamma(\mathcal{K}_i)\cdot \log_2^e \cdot \hat{\zeta}_i^*}\right) = \frac{r_i^{tar} - \sum\limits_{k\in\mathcal{K}_i} c_i^k(\hat{p}_i^{k*})}{\Gamma(\mathcal{K}_i)} \tag{17}$$

From (17), the optimal power increase on kth subcarrier $\triangle p_i^{k*}$ is given by

$$\frac{\triangle p_i^{k*}}{\log_2^e \cdot \hat{\zeta}_i^*} = \exp\left(\frac{r_i^{tar} - \sum_{k\in\mathcal{K}_i} c_i^k(\hat{p}_i^{k*})}{\log_2^e \cdot \Gamma(\mathcal{K}_i)}\right) - 1 \tag{18}$$

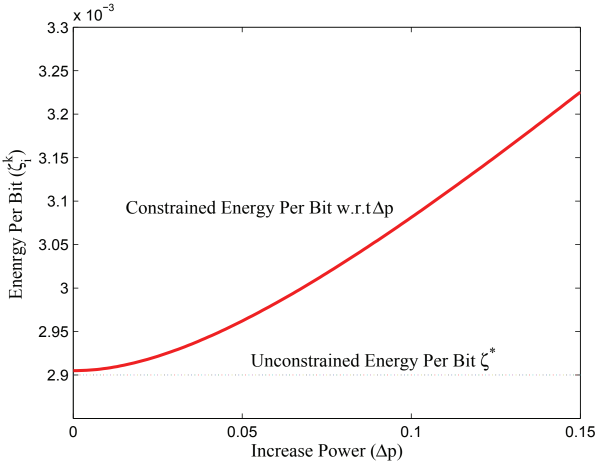


Fig. 4. Illustration of constrained ζ_i^k with respect to $\triangle p_i$

If $\triangle p_i^{min}$ exceeds the remaining power, i.e., $\sum_{k\in\mathcal{K}_i}\hat{p}_i^{k*}+\triangle p_i^{min}\geq p_i^{max}$, there is no feasible solution for (7). If $\sum_{k\in\mathcal{K}_i}\hat{p}_i^{k*}+\triangle p_i^{min}\leq p_i^{max}$, the optimal solution for the original problem (7) is

$$p_i^{k*}=\hat{p}_i^{k*}+\triangle p_i^{k*} \tag{19}$$

4. $\sum\limits_{k\in\mathcal{L}_i}\hat{p}_i^{k*}>p_i^{max}$ and $\sum\limits_{k\in\mathcal{L}_i}c_i^k(\hat{p}_i^{k*})>r_i^{tar}$.
   In this case, the data rate requirement is satisfied but the allocated power exceeds the limit. In order to obtain a feasible solution, the allocated power should be decreased. The derivation of the optimal solution follows similar procedures as given in case 3) and is available in S. Gao (2008).

The inter-relationship and evolvement of the four cases partitioned by the power and data rate constraints are highlighted in Fig.3. Excellent and terrible subcarrier conditions will lead to case 1) (feasible) and case 2) (infeasible), respectively. When the subcarrier conditions are "good", the solid lines from case 3) and case 4) lead the problem into the feasible region case 1) of the constrained optimization problem when it reaches the maximal power and target data rate bounds, respectively. Whereas, the dashed lines suggest that the problem enters the infeasible region case 2) when the current subcarrier condition cannot accommodate the target data rate under the maximal power limits.

4. Distributed power control

In the previous section, each emerging new user obtains its optimal subcarrier selection and power allocation individually without considering other new users. Although no interference will be introduced to the existing users, due to the non-cooperative behavior of each user, multiple new users may choose the same subcarriers and co-channel interference will be introduced among themselves. In order to maintain user's QoS, we propose an iterative and distributed algorithm for reaching an equilibrium point among multiple transmitter and receiver pairs based on the distributed power control scheme R. Yates (1995). The distributed power control algorithm is given by

$$p_i^k(t+1) = \min\left\{ \frac{\gamma_i^{k*}}{\gamma_i^k(t)} p_i^k(t), p_i^{max} \right\} \tag{20}$$

where γ_i^{k*} is the individual target *SINR* of the i^{th} transmitter receiver pair over each subcarrier k, which is determined by the constrained optimal solution $\boldsymbol{p}^*$, $\gamma_i^{k*} = \exp(\ln 2 \cdot c(p_i^{k*})) - 1$. In the power control stage, each node only needs to know its own received *SINR* (γ_i^k) at its designated receiver to update its transmission power. This is available by feedback from the receiving node through a control channel. As a result, the proposed scheme is fully distributed. Convergence properties of this type of algorithms were studied by Yates R. Yates (1995). An interference function $I(P)$ is standard if it satisfies three conditions: positivity, monotonicity and scalability. It is proved by Yates R. Yates (1995) that the standard iterative algorithm $P(t+1) = I(P(t))$ will converge to a unique equilibrium that corresponds to the minimum use of power. The distributed power control scheme (20) is a special case of the standard iterative algorithm.

In summary, the proposed energy efficient spectrum access and resource allocation scheme includes the following steps, as highlighted before in Fig. 1.

Distributed Energy Efficient Spectrum Access and Resource Allocation

1. *Initialization*

   – Each transmitter receiver pair obtains their respective available subcarrier set $\mathcal{L}_i$ through spectrum detection.

2. *Individual Energy Efficient Resource Allocation*

   – Each transmitter receiver pair derives its respective unconstrained optimal solution from equation (10).

   – Based on the power limit and data rate constraint, each transmitter receiver pair adjusts its power allocation according to the constrained optimal solution given in Section III. B.

   – Each transmitter receiver pair also calculates its corresponding optimal target *SINR* γ_i^{k*} based on the constrained optimal solution.

3. *Multiuser Distributed Power Control*

   – Through a control channel, each transmitter acquires the measured *SINR* $\gamma_i^k(t)$ from the designated receiver.

   – If $\gamma_i^k(t) \neq \gamma_i^{k*}$, the transmission power will be updated according to (20).

   – If $|\gamma_i^k(t) - \gamma_i^{k*}| \leq \epsilon$, $\forall i$, where ϵ is an arbitrary small positive number, the power control algorithm converges to a unique equilibrium point. Otherwise, it is infeasible to accommodate all the new users in the current time slot.

During the power control stage, if the target *SINR* γ_i^{k*} cannot be maintained when transmitter i hits its power bound p_i^{max}, the network is unable to accommodate all the new users. In this case, a multi-access control (MAC) scheme is required to guarantee the fairness among the users. This will be one of our future efforts.

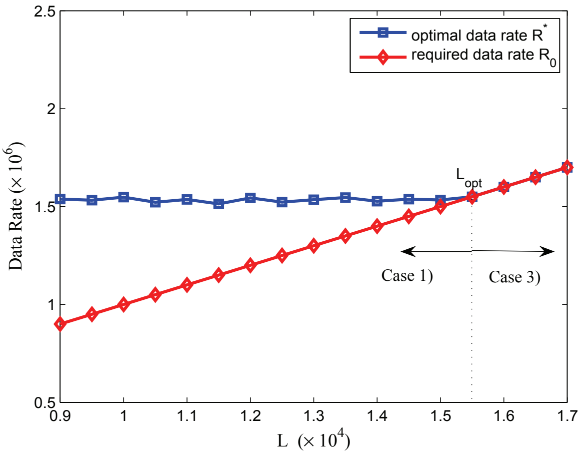


Fig. 5. Impact of L to optimal data rate

5. Numerical results

In this section, we evaluate the performance and convergence of the proposed distributed energy efficient channel selection and power allocation algorithm. A wireless ad hoc network with cognitive radio capability is considered. Specifically, the parameters of mica2/micaz Berkeley sensor motes G. Anastasi (2004) are employed. The sensor motes operate on 2 AA batteries and the output of each battery is about 1.5 volts, 25000 mAh. The channel gains are assumed to be sampled from a Rayleigh distribution with mean equals to $0.4d^{-3}$, where d is the distance from the transmitter to the receiver. The power bound for the transmission power is 150 mW. The entire spectrum is equally divided into subcarriers with bandwidth 100 kHz. The duration of each time slot T_S is assumed to be $10ms$ in which L bits need to be transmitted. Thus, the target data rate is assumed to be $r_i^{tar} = L/T_S$. The thermal noise power is assumed to be the same over all subcarriers and equals to 10^{-8}W.

For each individual user, we first investigate the impact of the target data rate on energy efficiency. We consider a transmitter receiver pair with available subcarrier set $\Gamma(\mathcal{L}_i) = 18$, the required data rate $r_i^{tar} = L/T_S$ ranges from 9×10^5 bps to 1.7×10^6 bps. In Fig.5, the *squared* line represents the optimal data rate allocation with the increase of r_i^{tar}, while the *diamond* line represents the required data rate r_i^{tar}. It can be observed from Fig. 5 that the optimal rate and power allocation remains approximately[4] unchanged given the channel conditions of the available subcarriers as long as $r_i^{tar} < r_i^{opt} = 1.55 \times 10^6$. After the two lines converge at $L_{opt} = 15500$ bits, the optimal data rate coincides with r_i^{tar}, i.e., the required rate can only be obtained at the cost of lower energy efficiency. It is noticeable that L_{opt} is an important system design parameter, and its optimal value can be pre-calculated given the channel conditions.

Fig. 6 illustrates the effect of L (thus the target data rate r_i^{tar} for fixed T_S) on energy efficiency. We define $E_i = \zeta_i^* \times L$ as the energy consumption per time slot which is jointly determined by ζ_i^* and L. It is observed that in case 1) with the increase of L, E_i increases linearly with respect to L and the energy consumption per bit remains approximately unchanged. When the system

[4] due to numerical round-off errors

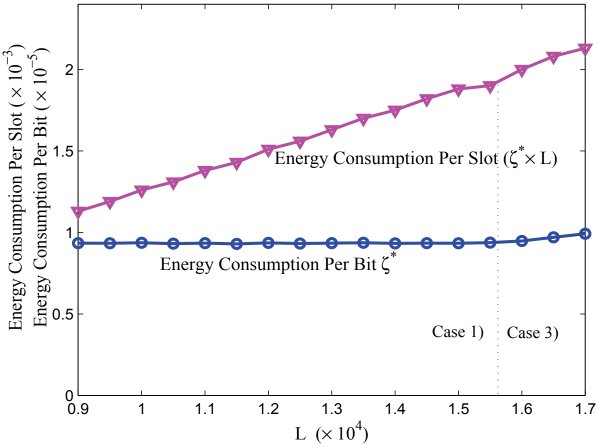


Fig. 6. Impact of L to energy efficiency

enters case 3) due to the increase of r_i^{tar}, ζ_i^* degrades which suggests that the required data rate r_i^{tar} is satisfied with the expense of energy efficiency.

The impact of the number of available subcarriers on energy efficiency is plotted in Fig. 7. It is shown that the increase of the number of available subcarriers ($\Gamma(L_i)$) improves energy efficiency by providing more available bandwidth. In fact, the total optimal allocated power to satisfy a fixed target data rate is reduced with the increase of $\Gamma(L_i)$. It can be seen in Fig. 7 that the dashed circle line (which represents the unconstrained optimal energy consumption) converges with the constrained energy consumption (solid circle line) when the number of available subcarriers reaches 28. It implies that when the available subacarriers are less than 28, the unconstrained optimal solution corresponds to case 3) in Section 3.2. The system will enter case 1) when $\Gamma(L_i) \geq 28$.

The performance of the proposed *energy-efficient* waterfilling with respect to network lifetime (which is a critical metric for energy constrained CR ad hoc networks) is investigated. Assuming uniform traffic patterns and persistent traffic flow across the network, we define the network lifetime as $T_l = E_{max}/(L \times \zeta_i^*)$, where E_{max} is the maximal energy source of each transmitter. Compared with *rate-adaptive* and *margin-adaptive* waterfilling (for transmitting the same amount of information bits in the network), it is observed in Fig.8 that the proposed scheme outperforms the other two allocation schemes in terms of network lifetime. As the optimal allocated rate approaches the target data rate, *energy-efficient* waterfilling will converges with *margin-adaptive* waterfilling as expected. However, since the target data rate in a typical energy constrained ad hoc network is usually low, it is expected that the proposed scheme will improve network lifetime in most applications.

After each new user obtains its optimal subcarrier selection and power allocation independently, distributed power control (20) may be triggered to manage the co-channel interference if multiple new users happen to choose the same subcarriers. The convergence of allocated power is shown in Fig. 9 (including the total required power and the power allocated on two randomly chosen subcarriers of two randomly chosen Tx-Rx pairs). It is observed that the convergence occurs in 3-4 steps.

In this part of the simulation (Fig. 10), the performance of the proposed distributed scheme is

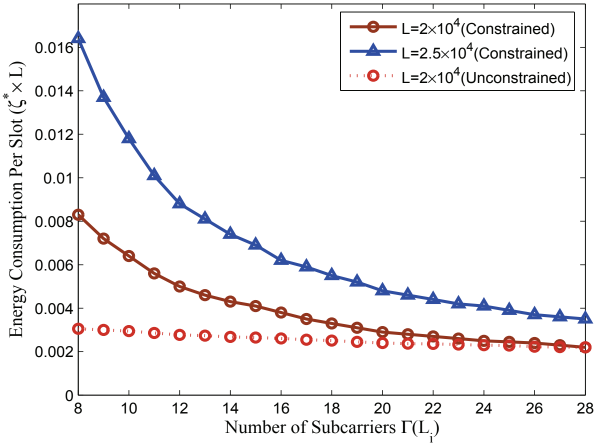


Fig. 7. Impact of number of available subcarriers to energy efficiency

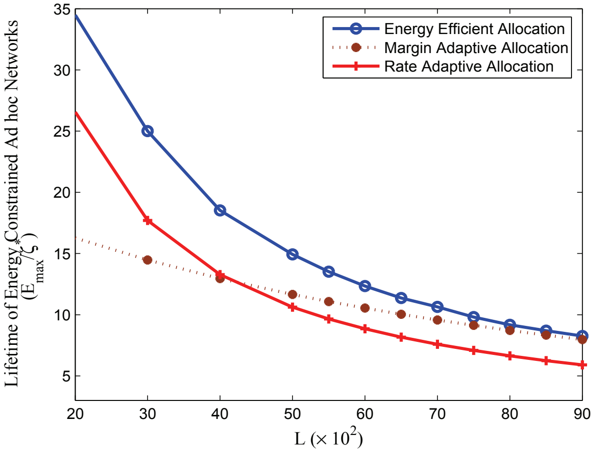


Fig. 8. Performance comparison among different allocation schemes

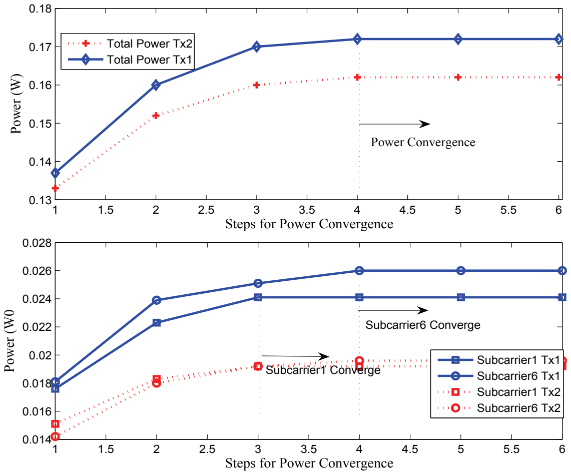


Fig. 9. Convergence of distributed power control

compared with the centralized optimal solution, where it is assumed that a central controller collects all the $M \times N^2$ channel gain information from all the N new users, and calculates the global optimal solution by considering all the co-channel interference. The case for 8 users and each user with 16 available subcarriers is investigated here S. Gao (2008). It is observed that the proposed distributed scheme (the upper two lines) performs closely to the centralized optimal solution (the middle line). In addition, the competitive optimal solution is also shown in Fig. 10, where each user calculates its own solution *without* considering co-channel interference (thus optimistic).

6. Conclusion

In this section, a framework of distributed energy efficient resource allocation is proposed for energy constrained OFDMA-based cognitive radio wireless ad hoc networks. A multi-dimensional constrained optimization problem is formulated by minimizing the energy consumption per bit over the entire available subcarrier set for each individual user while satisfying its QoS constraints and power limit. A two-step solution is proposed by first decoupling it into an unconstrained problem, and a constrained partitioning procedure is applied thereafter to obtain the constrained optimal solution by branching the solution space according to power and rate constraints. Co-channel interference may be introduced by concurrent *new* users and the distributed power control scheme may be triggered to manage the interference and reach the equilibrium point in the multiuser environment.

The proposed spectrum sharing plus resource allocation scheme provide a practical distributed solution for a CR wireless ad hoc network with low computational complexity. It is important to point out that the proposed algorithm for CR networks can be easily modified and applied to multi-channel multi-radio (MC-MR) networks which can be considered as a special case of the CR based wireless networks Y. T. Hou (2007).

In this work, it is assumed that the subcarrier detection is perfect. The effects of detection errors will be investigated in our future work.

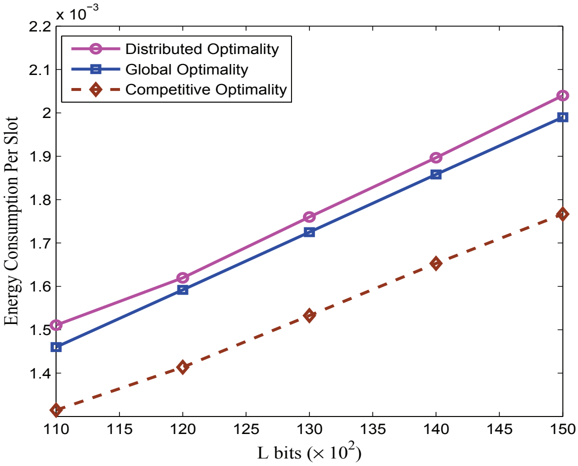


Fig. 10. Performance comparison between distributed scheme and global optimality

7. Appendix A

The unstrained optimization problem (8) is

$$f(\hat{\boldsymbol{p}}_i,\boldsymbol{\alpha}_i) := \frac{\sum\limits_{k\in\mathcal{L}_i} \hat{p}_i^k + p_i^r}{\sum\limits_{k\in\mathcal{L}_i} \log_2\left(1+\alpha_i^k\cdot\hat{p}_i^k\right)} \tag{21}$$

The first order derivative of (21) with respect to $\hat{p}_i^k$ can be derived as

$$\begin{aligned} \frac{\partial f(\hat{\boldsymbol{p}}_i,\boldsymbol{\alpha}_i)}{\partial \hat{p}_i^k} &= \frac{1}{\log_2 e}\cdot\left(\frac{\partial\Phi(\hat{\boldsymbol{p}}_i,\boldsymbol{\alpha}_i))}{\partial \hat{p}_i^k}\right) \\ \Phi(\hat{\boldsymbol{p}}_i,\boldsymbol{\alpha}_i) &= \frac{\hat{p}_i^k + \sum\limits_{l\in\mathcal{L}_i, l\neq k} \hat{p}_i^l}{\ln\left(1+\alpha_i^k\hat{p}_i^k + p_i^r\right) + \sum\limits_{\substack{l\in\mathcal{L}_i\\ l\neq k}} \ln\left(1+\alpha_i\hat{p}_i^k\right)} \end{aligned} \tag{22}$$

If $k \neq l$, $c_i(\hat{p}_i^l)$ is taken as constant with respect to $\hat{p}_i^k$ since the mutual interference between subcarriers is not considered in this work. Therefore, (22) can be expressed as

$$\frac{\partial\Phi(\hat{\boldsymbol{p}}_i,\boldsymbol{\alpha}_i)}{\partial \hat{p}_i^k} = \frac{\sum\limits_{k\in\mathcal{L}_i} \frac{c_i^k(\hat{p}_i^k)}{\ln 2} - (\sum\limits_{k\in\mathcal{L}_i} \hat{p}_i^k + p_i^r)\left(\frac{\alpha_i^k}{1+\alpha_i^k\hat{p}_i^k}\right)}{\left[\sum\limits_{k\in\mathcal{L}_i} \ln\left(1+\alpha_i^k\cdot\hat{p}_i^k\right)\right]^2} \tag{23}$$

We assume the data rate $\sum_{k\in\mathcal{L}_i} c_i^k(\hat{p}_i^k) \geq 0$ in this work, thus for $\frac{\partial f(\hat{\boldsymbol{p}}_i,\boldsymbol{\alpha}_i)}{\partial \hat{p}_i^k} = 0$, (23) can be reduce to

$$\frac{\alpha_i^k}{1+\alpha_i^k \cdot p_i^k} = \frac{\sum\limits_{k\in\mathcal{L}_i} \ln\left(1+\alpha_i^k \cdot \hat{p}_i^k + p_i^r\right)}{\sum\limits_{k\in\mathcal{L}_i} \hat{p}_i^k} \tag{24}$$

From (24), we can derive the unconstrained optimal power allocated for transmitter i over subcarrier k as

$$\hat{p}_i^k = \frac{\sum\limits_{k\in\mathcal{L}_i} \hat{p}_i^k + p_i^r}{\sum\limits_{k\in\mathcal{L}_i} \ln\left(1+\alpha_i^k \cdot \hat{p}_i^k\right)} - \frac{1}{\alpha_i^k} \tag{25}$$

From the definition of unconstrained energy consumption per bit $\hat{\zeta}_i$, the first term of (25) is in the similar type of $\hat{\zeta}_i$. If we assume the optimal solution of (A1) does exist (the subcarrier condition resides in the feasible region), there must be a corresponding optimal value of energy per time slot $\hat{\zeta}_i^*$ with respect to $\hat{\boldsymbol{p}}_i$. Then (25) can be expressed in terms of $\hat{\zeta}_i^*$ as

$$\hat{p}_i^{k*} = \log_2 e \cdot \hat{\zeta}_i^* - \frac{1}{\alpha_i^k} \tag{26}$$

8. Appendix B

The proof of *Lemma* 1 is given in this appendix. We first define $f(\triangle p_i)$ as

$$f(\triangle p_i) = \frac{\triangle p_i + \sum\limits_{k\in\mathcal{K}_i} \hat{p}_i^{k*} + p_i^r}{\sum\limits_{k\in\mathcal{K}_i} \log_2\left(1+\alpha_i^k \cdot \hat{p}_i^{k*}\right) + \sum\limits_{k\in\mathcal{K}_i} \triangle c_i^k} \tag{27}$$

Where $\sum_{k\in\mathcal{K}_i} \triangle c_i^k = \Gamma(\mathcal{K}_i)\log_2\left(1+\frac{\triangle p_i}{\Gamma(\mathcal{K}_i)\log_2^e \cdot \hat{\zeta}_i^*}\right)$. We denote

$$h(\triangle p_i) = \frac{\triangle p_i}{\sum\limits_{k\in\mathcal{K}_i} \triangle c_i^k} = \frac{\triangle p_i/\Gamma(\mathcal{K}_i)}{\log_2\left(1+\frac{\triangle p_i/\Gamma(\mathcal{K}_i)}{\log_2^e \cdot \hat{\zeta}_i^*}\right)} \tag{28}$$

Due to $x \geq \ln(1+x) \forall x \geq 0$, we can obtain

$$\ln 2 \cdot \frac{\triangle p_i/\Gamma(\mathcal{K}_i)}{\hat{\zeta}_i^*} \geq \ln\left(1+\frac{\triangle p_i/\Gamma(\mathcal{K}_i)}{\hat{\zeta}_i^*}\right) \tag{29}$$

Since $\triangle p_i$,$\Gamma(\mathcal{K}_i)$, and $\hat{\zeta}_i^* \geq 0$ in this work, we get

$$h(\triangle p_i) = \frac{\triangle p_i/\Gamma(\mathcal{K}_i)}{\log_2\left(1+\frac{\triangle p_i/\Gamma(\mathcal{K}_i)}{\log_2^e \cdot \hat{\zeta}_i^*}\right)} \geq \hat{\zeta}_i^* \tag{30}$$

Based on the definition of $f(\triangle p_i)$, summing $h(\triangle p_i)$ and unconstrained optimal energy per bit $(\hat{\zeta}_i^*)$, we have

$$f(\triangle p_i) \geq \hat{\zeta}_i^*, \forall \triangle p_i \in \mathbb{R}^+ \tag{31}$$

Thus, we can conclude the increasing power deteriorates the energy efficiency.

9. Acknowledgment

This research work is supported in part by the U.S. Army Research Office under Cooperative Agreement No. W911NF-04-2-0054. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U. S. Government.

10. References

A. Pandharipande, "Principles of OFDM", *IEEE Potentials Magazine*, Vol. 21, Issue 2, pp. 16-19, Apr. 2002.

Federal Communication Commission, "Spectrum Policy Task Forc", *Rep. ET Docket no.02-135*, Nov. 2002.

G. Kulkarni, S. Adlakha, and M. Srivastava, "Subcarrier allocation and bit loading alogrithm for OFDMA-based wireless networks", *IEEE Trans. on Mobile Computing*, Vol. 4, No. 6, pp. 652-662, Nov. 2005.

G. Anastasi, A. Falchi, M.Conti and E.Gregori, "Performance Measurements of Motes Sensor Netoworks", *ACM MSWiM 2004-Proceedings of the Seventh ACM Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp 174-181, 2004.

I. F. Akyildiz, Won-Yeol Lee, and Kaushik R. Chowdhury. "CRAHNs: Cognitive Radio Ad Hoc Networks", *Ad Hoc Networks*, vol. 7, pp 810 - 836, 2009.

I. F. Akyildiz, W. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless network: A survey.", *Computer Networks*, vol. 50, no. 13, pp. 2127 - 2159, Sep. 2006.

J. Mitola et al, "Cognitive radio: making software radios more personal", *IEEE Personal Communications*, vol. 6, no. 4, pp. 13 - 18, Aug. 1999.

J. Huang, R. A. Berry, and M. L. Honig, "Distributed Interference Compensation for Wireless Networks", *IEEE JSAC*, Vol. 24, No. 5, pp.1074-1084, May 2006.

J. Bazerque, and G. B. Giannakis, "Distributed Scheduling and Resource Allocation for Cognitive OFDMA Radios", *IEEE International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, 2007.

K. Fazel and S. Kaiser, *Multi-Carrier and Spread Spectrum Systems*, New York: Wiley, 2003.

Q. Zhao, L. Tong, A. Swami, and Y. Chen, "Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks", *IEEE Journal on Selected areas in Communications*, vol. 25, no. 3, pp. 589 - 600, Apr. 2007.

Q. Qu, L. Milstein, D.R. Vaman, "Cognitive Radio Based Multi-User Resource Allocation in Mobile Ad Hoc Networks Using Multi-Carrier CDMA Modulation", *IEEE JSAC*, Vol. 26, No. 1, pp.70-82, Jan 2008.

R. Yates, "A Framework for Uplink Power Control in Cellular Radio Systems", *IEEE Journal on Selected areas in Communications*, vol. 13, no. 7, pp. 1341 - 1348, Sept. 1995.

S. Gao, L. Qian, and D. R. Vaman, "Centralized Energy Efficient Spectrum Access in Wireless Cognitive Radio Networks", *CebCom Center Tech. Report*, March 2008. [Online]. Available: http://nsf-rise.pvamu.edu/webpage/files/papers/GaoTechCR2008.pdf.

S. Tao, S. Cui, and M. Krunz, "Medium Access Control for Multi-Channel Parallel Transmission in Cognitive Radio Networks", *IEEE Global Comm. Conference*, pp. 1 - 5, Nov. 2006.

S. Cui, A. J. Goldsmith, and A. Bahai, "Energy-constrained Modulation Optimization", *IEEE Transactions on Wireless Communications*, vol. 4, no. 5, pp. 2349 - 2360 Sept. 2005.

S. Kondo, and B. Milstein, "Performance of multicarrer DS CDMA systems", *IEEE Trans. on Communicatons*, Vol. 44, Issue 2, pp. 238-246, Feb. 2001.

T. Weiss, J. Hillenbrand, A. Krohn, and F. K. Jondral, "Mutual Interference in OFDM-based Spectrum Pooling Systems", *IEEE 59th Vehicular Technology Conference, VTC*, Vol. 4, pp. 1873-1877 , May 2004.

W. Yu, G. Ginis, and J. M. Cioffi, "Distributed Multiuser Power Control for Digital Subcriber Lines", *IEEE Journal on Selected areas in Communications*, Vol. 20, No. 5, pp. 1105-1115, June 2002.

W. Hu, et.al, "Dynamic Frequency Hopping Communities for Efficient IEEE 802.22 Operation", *IEEE Communications Magazine*, pp.80-87, May 2007.

Y. T. Hou, Y. Shi, and H. D. Sherali, "Optimal Spectrum Sharing for Multi-Hop Software Defined Radio Networks", *IEEE International Conference on Computer Communications (INFOCOM)*, pp.1-9, May 2007.

Y. Shi, Y. T. Hou, "Optimal Power Control for Multi-Hop Software Defined Radio Networks", *IEEE International Conference on Computer Communications (INFOCOM)*, pp.1694-1702, May 2007.

29

Theory and Applications of Ad Hoc Networks

Takuo Nakashima
Tokai University
Japan

1. Introduction

In ad hoc mobile networks (MANET), the mobility of the modes is a complicated factor that significantly affects the effectiveness and performance of ad hoc routing protocols. In addition, MANET requires the quality of data transmission. Improvement of a routing protocol between communication links provide high quality data transmission. The routing protocol of wireless network exchanges the route information to establish the communication link.

The routing algorithms exchanging the path construction and maintenance messages generate a connectivity related dynamic graph representing the topology of the network by a series of messages passes. Using such a data structure, messages can be transmitted over a number of intermediate nodes that interconnect the source with the destination, also known as routing paths or routes. These routing protocols can be classified into reactive or on-demand protocols (1) such as Ad hoc on-demand distance vector (AODV) (8) and proactive or table-driven protocols (2), such as Dynamic source routing protocol (DSR) (9Proactive protocols always maintain a route to every possible destination, while reactive protocols are considered to discover and maintain a route to a destination only when a route is demanded. AODV routing protocol uses an on-demand approach for finding routes, that is, a route is established only when it is required by a source node for transmitting data packets. It employs destination sequence numbers to identify the most recent path. The major difference between AODV and DSR stems out from the fact that DSR uses source routing in which a data packet carries the complete path to be traversed. However, in AODV, the source node and the intermediate nodes store the next-hop information corresponding to each flow for data packet transmission. The major difference between AODV and other on-demand routing protocols is that it uses a destination sequence number to determine an up-to-date path to the destination.

In a mobile environment, reactive routing protocols have more advantages than proactive routing protocols since reactive routing protocols exchange routing information only when a path is required by a node to communicate with a destination. On the contrary, proactive routing protocol exchanges routing information in order to maintain the global topology information whenever one of path information is required to update triggered by the node movement.

The performance analysis for proactive and reactive routing protocols has been explored in the last decade. To realize the real environment, the selection of the mobile pattern and the size of nodes are key element of simulation. Marinoni et al. (3) discussed routing protocol performance in a realistic environment. New mobility model has introduced and installed in ns-2 simulator. The discussion, however, is limited in DSR protocol, and no comparison to other protocols appeared. To investigate the truth, Zhang et al. (4) researched the performance of routing protocol in very large-scale mobile area allocating about 50,000 mobile nodes in GTNetS simulator. As the traffic patterns are constructed by randomly selected sources and destinations, it is hard to separate communication overload and routing overload. Mbarushimana et al. (5) explored comparative study of reactive and proactive routing protocol performances. The parameter of simulation, however, is not clearly described, and mobility is not precisely evaluated in this research.

Various routing protocols have been compared in different conditions. In the radio models, Yang et al. (13) analysed the performance of three Ad Hoc routing protocols: AODV, DSR and DSDV considering two radio models TwoRayGround and Shadowing. This research concluded that the shadowing phenomena reduce the mean distance among nodes and increase the latency of packet transmission. The Data link and Physical layers have the potential of the technology improvements. One of our aims is to find the optimal parameter sets of each routing protocols. We focus the performances of routing protocols and the transport layer. The researches for the performance evaluation (14), (15), (16), (17) between the proactive and reactive protocols have found the features for different type routing protocols. These results, however, are obvious for the original design and aim of each protocol.

One approach to enhance the AODV protocol introduces the probability to forward the messages. The researches (18), (19) have proposed the probability based messaging mechanism especially in which (19), each node communicates to its' surrounding nodes/routers with calculated probability, and reduces the overhead of the routing protocols. The routing tables are maintained based on this probability leading to select the most optimal path for transmitting the data packets by sensing the breaking of route. Therefore this approach is effective to avoid the broken route. This method has the more significant meaning than hop count about adjacency between two nodes. The probability, however, is not utilized by the RREQ flooding mechanism meaning that this method does not reduce the RREQ flooding packets. The main degradation of AODV routing depends on the RREQ flooding leading that this method does not improve the end-to-end performance significantly.

The other enhancing approach is to introduce the awareness of the accessibility of the neighbour nodes (20). Nodes acquire the accessibility information of the other nodes through routine routing operations and keep it in their routing table. Later, this information worked to enhance the routing operations. The mobility, however, will change dynamically and induce the huge consumption of memory resources holding the access state for the accessibility prediction. The experiments have been conducted over a small number of mobile nodes. The experiments over the huge number of mobile nodes should be executed to verify the performance improvement.

If mobile nodes are equipped with a digital compass, each mobile node recognizes the direction of messaging. El-azhari et al. (21) proposed the routing protocol based on the direction angle and hop count. This approach easily expands to other devices such as a GPS leading the position based routing protocol. We will focus our discussion on the normal mobile node without any special devices.

In this research, firstly, we focus on the performance of reactive routing protocols such as DSR and AODV, and the throughput of TCP for different routing protocols in the variable, the number of intermediate mobile nodes and the speed of mobile nodes. Secondly, the new metric, the node density, will be introduced to evaluate the routing performance. Thirdly, we focus on the AODV control messages such as RREQ (Route Request) and RREP (Route Reply), and evaluate the messaging overhead between adjacent nodes. Fourthly, the RREQ flooding patterns generated from both the source node and intermediate nodes are analyzed. Finally, we control the communication power to change the communication distance, and discuss the flooding features of the AODV protocol.

The final goal of our research is to improve the TCP performance over the ad hoc wireless network. Before the discussion of TCP congestion and flow control mechanism, the performance features of the routing protocol should be extracted to evaluate accurate performance. Our previous research indicated the AODV routing protocol had shown the high performance for the mobile node with high speed and the over the dense condition of intermediate nodes. Motivation of this research explores accrete flooding performance of AODV under different mobility models. We will split the overload performance to distinguish the total communication performance depended on the routing performance or TCP performance.

This paper is organized as follows; first, mobile Ad Hoc Networks are explained in Section 2 followed by the comparison of different type of routing protocols in Section 3. The performance of the AODV protocol are discussed in Section 4 followed by the RREQ flooding features of the AODV protocol in Section5. Section 6 the propagation features of the AODV protocol under different mobility models followed by the flooding features of the AODV protocol under the different communication distances in Section 7. Section 8 gives the summary and discussion for future work.

2. Mobile ad hoc networks

2.1 Reactive type routing protocols

Proactive type routing protocols decide the route direction based on the routing table maintained constantly to communicate between adjacent routers. These protocols are activated after preparing the routing table, so proactive type is called table-driven routing protocols. On the other hand, reactive type routing protocols decide the path when IP packets demand transferring to the destination. These protocols prepare the routing table after demanding the path, so reactive type is called on-demand routing protocols. Unlike the table-driven routing protocols, on-demand routing protocols execute the path-finding process and exchange routing information only when a path is required by a node to communicate with a destination. This subsection briefly explores two reactive type routing protocols.

2.1.1 Dynamic Source Routing Protocol

Dynamic Source Routing Protocol (DSR) is an on-demand protocol designed to restrict the bandwidth consumed by control packets in ad hoc wireless networks by eliminating the periodic table-update messages required in the table-driven approach. The major difference between this and other on-demand routing protocols is that it is beacon-less and hence does not require periodic hello packet (beacon) transmission, which are used by a node to inform its neighbors of its presence. The basic approach of this protocol during the route construction phase is to establish a route by flooding Route Request packets in the network. The destination node, on receiving a Route Request packet, responds by sending a Route Reply packet back to the source, which carries the route traversed by the Route Request packet received. In addition, the TTL field constrains the reachable distance, and the sequence number filed prevents double transmission and loops.

2.1.2 Ad hoc On-demand Distance Vector

Ad hoc on-demand distance vector (AODV) routing protocol uses an on-demand approach for finding routes, that is, a route is established only when it is required by a source node for transmitting data packets. It employs destination sequence numbers to identify the most recent path. The major difference between AODV and DSR stems out from the fact that DSR uses source routing in which a data packet carries the complete path to be traversed. However, in AODV, the source node and the intermediate nodes store the next-hop information corresponding to each flow for data packet transmission. The major difference between AODV and other on-demand routing protocols is that it uses a destination sequence number to determine an up-to-date path to the destination.

2.2 Mobility models

In this section, we shortly introduce three mobility models that have been proposed for the performance evaluation of an ad hoc network protocol. The detailed discussion of the mobility model is described by Camp et. al.(11).

2.2.1 Random walk

In Random Walk Mobility Model, an Mobile Node (MN) moves from its present location to a new one by randomly choosing a direction and speed in which to travel. The new speed and direction are both chosen from pre-defined range, [*min_speed*, *max_speed*] and [0, 2π] respectively. Each movement in the Random Walk Mobility Model occurs in either a constant time interval *t* or a *d* traveled in constant distance, at the end of which a new direction and speed are calculated. If an MN which moves according to this model reaches a simulation boundary, it bounces off the simulation border with an angle determined by the incoming direction. The Random Walk Mobility is a widely used mobility model. On the other hand, if the specified time (or specified distance) an MN moves in the Random Walk Mobility Model is short, then the movement pattern is a random roaming pattern restricted to a small portion of the simulation area.

2.2.2 Random waypoint

The Random Waypoint Mobility Model includes pause times between changes in direction and/or speed. An MNbegins by staying in one location for a certain period of time. Once this time expires, the MN chooses a random destination in the simulation area and a speed that is uniformly distributed between [*min_speed*, *max_speed*]. The MN then travels to the newly chosen destination at the selected speed.

2.2.3 Random Direction

The Random Direction Mobility Model (12) was created to overcome density waves in the average number of neighbors by the Random Waypoint Model. A density wave is the clustering of nodes in one part of the simulation area. In the case of the Random Waypoint Mobility Model, this clustering occurs near the center of the simulation area. In the Random Waypoint Mobility Model, the probability of an MN choosing a new direction located in the center of the simulation area, or a destination which requires travel through the middle of the simulation area, is high. In the Random Direction Mobility Model developed by (12), MNs choose a random direction in which to travel similar to the Random Walk Mobility Model. An MN then travels to the border of the simulation area in that direction. Once the simulation boundary is reached, the MN pauses for a specified time, choose another angular direction (between o and 180 degrees) and continues the process.

2.3 Propagation model

The propagation signal power between adjacent nodes is defined using the following the equation of the Two-Ray-Ground propagation model.

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L} \tag{1}$$

where $P_r(d)$ be the receiving signal power on the distance d (meter), P_t be the sending signal power, G_t be the gain of sending antenna, G_r be the gain of receiving antenna, h_t be the height of the sending antenna, h_r be the height of the receiving antenna, d be the distance and L be the system loss.

2.4 Flooding feature of RREQ and performance features of RREP

We introduce the new metrics to characterize the routing performance based on the message communication. The average number of RREQ (Route Request) adjacent nodes to evaluate the degradation of RREQ flooding performance and the average propagation rate of RREP(Route Reply) should be defined as follows.

$$\textit{Average number of RREQ adjacent nodes} = \frac{\textit{The total number of RREQ receiving packets}}{\textit{The number of RREQ sending packets}} \tag{2}$$

This metric indicates how many average nodes are reachable on each node. As the RREQ is the broadcast message, average number of RREQ adjacent nodes is induced by the number of RREQ receiving packets divided by the number of RREQ sending packets.

$$\textit{Average propagation rate of RREP} = \frac{\textit{The total number of RREP receiving packets}}{\textit{The number of RREP sending packets}} \tag{3}$$

The other metric of the average propagation rate of RREP indicates the percentage that the RREP packets could reach from the source node to the destination node on unicast message without loss.

2.5 Simulation model

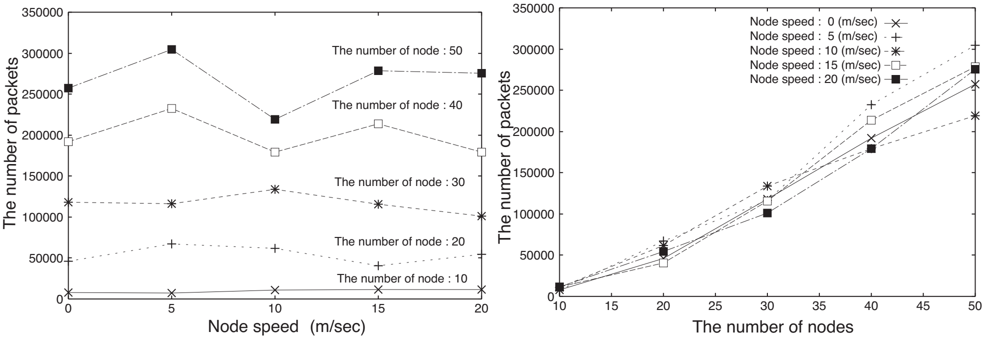
We adopted the LBNL network simulator (ns) (6) to evaluate the effects of performance of ad hoc routing protocols. The ns is a very popular software for simulating advanced TCP/IP algorithms and wireless ad hoc networks. Our experiments configure two stationary end nodes and intermediate mobile nodes in the ad hoc networks. The TCP connection will be limited between two stationary end nodes to generate one TCP traffic through an appropriate route controlled by ad hoc routing protocols. While mobile nodes are located randomly on the 500 (m) * 400 (m) square area, and move through this area with the same speed. In our simulations, the speed of mobile node will be changed at 5, 10, 15 and 20 (m/sec) to emulate a bicycle and a normal car speed. The direction of movement changes after elapsing 10 simulation seconds with in a random manner. The number of intermediate mobile nodes varies from 10 to 50 with 10 nodes interval. MAC and physical layer consist of 802.11 (7) with omniantenna model.

To discuss separately in terms of two communication nodes and intermediate nodes, we classify the mobility pattern into three types. Firstly, two communication nodes are still stationary and other intermediate nodes randomly move. Secondly, two communication nodes randomly move with stationary intermediate nodes. Finally, all nodes move randomly. To verify the efficiency of routing protocol, we focus on the first pattern in our simulation.

3. Comparison of different type of routing protocols

Firstly, we focus on the end-to-end performance varying the node speed for different routing protocols. The accurate evaluation for the sensitivity of the speed requires the uniform speed, and that for the end-to-end performance requires the stationary nodes for TCP communication. From these requirements, we adopted the random walk mobility model with uniform speed, and classify nodes into two types, two communication stationary nodes and intermediate mobile nodes under the random walk mobility model.

3.1 AODV control packets



(a) Varying the node speed (b)Varying the number of nodes

Fig. 1. The number of Route Request (RREQ) packets.

Figure 1(a) shows the number of Route Request (RREQ) packets varying the node speed. Main significant result is that the number of RREQ packets does not depend on speed. The total number of RREQ packet does not change whether intermediate nodes stand still or move with 20 (m/sec). This simulation results is contrary to the prediction that if the node speed increases, then the connectivity is damaged, to be followed by the control packet to establish the connectivity increase. The unexpected result is caused by the following two reasons. Firstly, the reachable area of wireless MAC protocol covers the wide range. Secondly, the communication packet on the intermediate nodes is limited to the routing packet meaning that packets are smoothly transferred on the intermediate nodes and the path is decided faster than the moving speed. We can conclude that the constant change of moving direction does not follow the increase of the total number of RREQ packets since the reachable area covers the moving area.

To clarify the property between the number of node and the total number of RREQ packets, Figure 1(b) illustrates the number of RREQ packets varying the number of intermediate nodes. The number of RREQ linearly increases following the number of intermediate nodes, which means the number of RREQ linearly depends on the density of intermediate nodes. In addition, we confirm that the increase of the total number of RREQ is proportional to the increase of the number of reachable node for RREQ packets. This increase enhances the availability to communicate the destination node. In general, the enhancement of availability for path selection can improve the performance overload with path data saved in cache area in such a way that the relevant path is selected, while generating fluctuated path selection in this case.

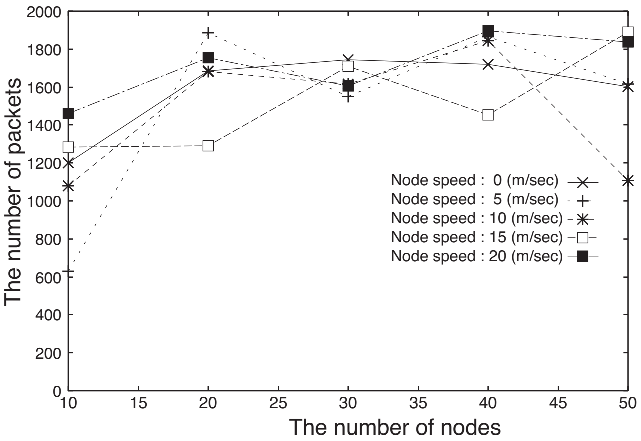


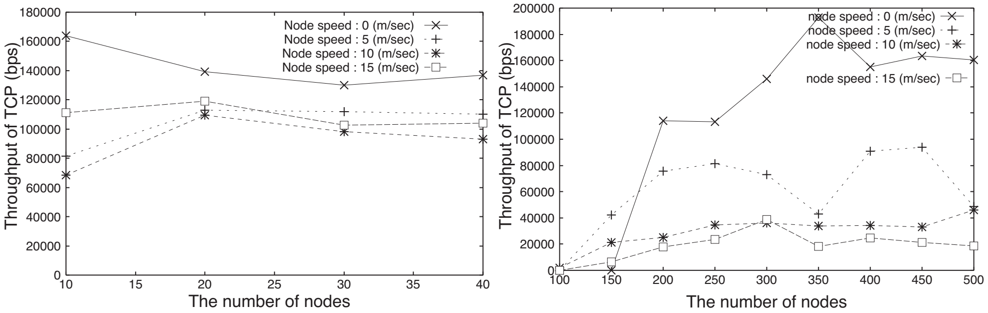
Fig. 2. The number of RREP packets varying the number of nodes.

So far, we discussed the performance of the RREQ packets transferred to the forwarding direction from source node so far, and then discuss the Route Reply (RREP) packets transferred to the backwarding direction. Figure 2 illustrates the number of RREP packets varying the number of nodes with different node speeds. In contrast to the number of RREQ packets, the number of RREP packets does not change apparently for each the number of node and each node speed. This indicates that the number of selection for paths does not change for the each number of nodes and each node speed, for one path selection generates the one RREP packet. As the AODV is a reactive type protocol, the path is decided when the request occurs. At that time, the RREP packet is generated on the destination node to notify the path information to the source node even if the intermediate node does not move. This property leads to generate some amount of RREP packets when the node speed is zero. The same amount of RREP packets in both cases - the node speed is zero and node moves with some speed – indicates that path selection is rarely rearranged during the communication, and the data is transferred on the first selected path regardless of the movement in intermediate nodes. Compared to the moving speed which are considered to be relevant to real environment, the data transmission time is very short. In addition, considering with the linear increase of RREQ packets, flooding based RREQ packets are affected by the number of intermediate nodes, but the number of RREP packets is stable after an appropriate path selection.

3.2 TCP performance

In this subsection, we evaluate the end-to-end performance on TCP, focusing on the throughput of TCP.

3.2.1 TCP performance on AODV



(a) Varying the number of nodes (b)varying the number of nodes in wide area

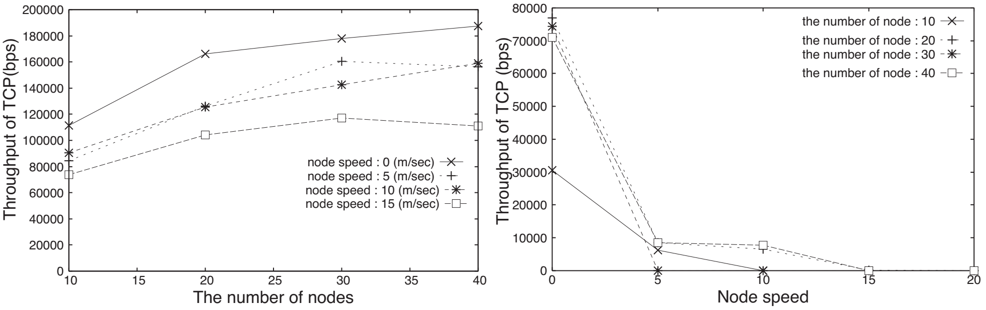
Fig. 3. The throughput of TCP.

Figure 3(a) shows the throughput of TCP varying the number of nodes in different node speeds. Each throughput has no significant difference except for the number of node being 10 meaning that throughput does not depend on the large number of nodes. In addition, the TCP throughput does not depend on the speed of mobile nodes. Compared to the stationary intermediate nodes, the performances of throughput of mobile nodes degrade about 20 % independently of the large number of nodes and node speeds. The coarse distribution of intermediate nodes such as ten nodes causing the loss of data transmission which results in performance deterioration. As the RREQ control packets equally flow in the case of ten nodes compared to other cases as previously mentioned, the frequency of route change can equally occur in this case. The coarse density, however, causes the performance deterioration. On the other hand, the performance gain for stationary ten intermediate nodes is influenced by the quick establishment of connection without wasted routing time.

To clarify the effect of coarse condition of intermediate nodes, we experimented in a wide area environment. Figure 3 (b) shows the fluctuations of TCP throughput, when the mobile area is enlarged to 3,000 (m) * 2,400 (m) with the number of intermediate nodes increased 500. If 40 intermediate nodes exist in 500 (m) * 400 (m) then the density is 20 nodes in 100 square meters. While intermediate nodes exist in this experiment, the density is 0.7 nodes in 100 square meters representing a very coarse condition. In a coarse condition, the throughput is significantly influenced by the node speed. When the node speed is zero, these nodes can not communicate with each other, i.e. the throughput value is zero under the condition of 150 intermediate nodes. However, if the number of intermediate nodes increases to greater than 200, the throughput indicates good performance for the stationary intermediate nodes. Then when the number of intermediate nodes exceeds 300, i.e. the density exceeds 0.4 nodes in 100 square meters, the stable performance of TCP is obtained.

On the other hand, when these intermediate nodes move with the speed such as 5, 10 and 15 (m/sec), the TCP performances enhance linearly for each speed until the number of nodes reaches 200. The throughput in dense nodes is stable and depends on the node speed. The throughput performance drops drastically from zero to 10 (m/sec) node speeds meaning that the throughput performance does not linearly degrade in terms of the node speed, but exponentially degrades until 10 (m/sec) node speeds.

3.2.2 TCP performance on DSR and DSDV



(a)DSR (b) DSDV

Fig. 4. The throughput varying the number of nodes.

The throughput performance for DSR which is another typical reactive type routing protocol is shown in Figure 4 (a). Compared to AODV protocol, there are two distinct features. Firstly, the throughput increases in fine node condition representing that the communication path quickly connected both end nodes. Secondly, the throughput degrades for high-speed intermediate nodes indicating that the performance depends on the node speed, degrading to half the performance in the 15 (m/sec) node speed compared to the condition with zero speed. The dependency of the node speed is one of the disadvantages of DSR, which are caused by path changes and path management mechanism which hold total path information on only source and destination nodes in contrast to the fact that AODV can replay RREP packets on intermediate nodes.

We have previously discussed the performance of reactive type routing protocols. In this subsection, we focus on one typical proactive type routing protocol, DSDV (Destination Sequenced Distance-Vector Routing Protocol), and compare the throughput performances. Figure 4 (b) illustrates the throughput variation created by the node speed. All throughputs for each number of node drastically degrade when the node speed exceeds 5 (m/sec). In addition, the coarse node condition such as 10 intermediate nodes causes the poorer performance even if intermediate nodes are stationary. These results show that proactive type routing protocol is weak in terms of the node speed, and the coarse density node condition causes the performance degrade. The proactive type routing protocol is not a realistic selection in mobile communication.

3.3 Summary

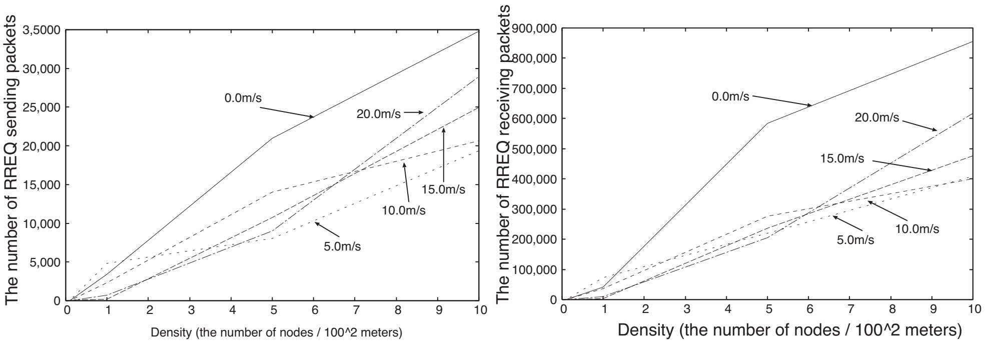
Our simulation results explored two performance features for the AODV control packets as follows. Firstly, route stability is more sensitive to the number of intermediate nodes than the speed of intermediate nodes. Secondly, the number of RREQ packets linearly increases in terms of the number of node. In addition, we captured the following three features of TCP throughput performances for different protocols. Firstly, the TCP throughput on AODV with mobile intermediate nodes degrades about 20 % compared to stationary intermediate nodes in dense node condition, and exponentially degrades until 10 (m/sec) node speeds in a coarse node condition. Secondly, the TCP throughput on DSR degrades in accordance with the increase of node speed meaning that it is sensitive to the node speed. Thirdly, the TCP throughput on DSDV drastically degrades in the mobile environment.

4. Performance of the AODV protocol

In this section, mobile nodes are located randomly on the 1,000 (m) * 1,000 (m) square area with four different density such as 0.1, 1, 5 and 10 nodes / 100 square meters, and move through this area with the same speed. The direction of movement changes after elapsing 10 simulation seconds with in a random manner.

4.1 AODV control packets

In this experiment, we separate the sending and receiving packets to clarify the differences between active (sending) nodes and non-active (receiving) nodes.



(a) Sending packets (b) Receiving packets

Fig. 5. The number of RREQ varying the node speed.

Figure 5 (a) shows the number of Route Request (RREQ) sending packets varying the node speed. As a standard property, when the density of node becomes denser, the number of sending packets increases in proportion to the node density. More wide-spread reachable areas of wireless network make the covering area wider in proportion to the node density. We captured two main properties; first, stationary condition such as the zero speed generates RREQ sending packets most frequently, and secondly, the condition with high speed mobility that creates RREQ sending packets more frequently than that with low speed mobility.

Node mobility makes it possible for a message to reach larger number of nodes in a wider area compared to a message in stationary condition. This leads to the conclusion that the number of sending nodes in mobile condition is smaller than in stationary condition. On the other hand, when a node density is as close to the density of 10 nodes in 100 square meters, high speed nodes can reestablish the route and can consume RREQ sending packets.

Compared to the sending packets in Figure 5 (a), the number of receiving packets is 20 times larger in the case of 2 nodes in 100 square meters as shown in Figure 5 (b). This figure illustrates two identical properties of sending packets; firstly, stationary condition generates

RREQ receiving packets most frequently, secondly, the condition with high speed mobility creates RREQ receiving packets more frequently than that with low speed mobility. The receiving node here does not mean the number of activated nodes, but mean the number of existing node of sending packets in reachable area.

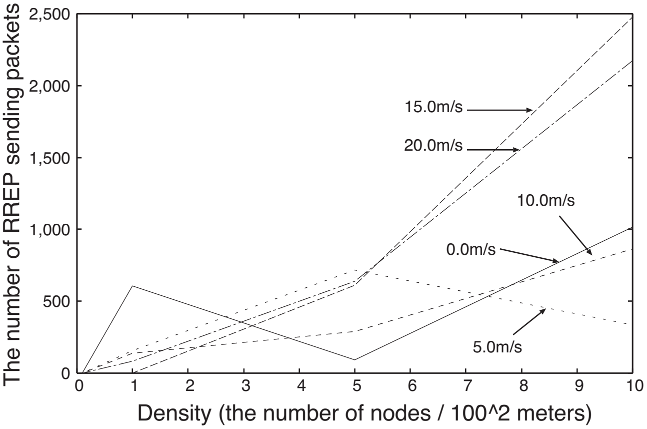


Fig. 6. The number of Route Reply (RREP) sending packets varying the node speed.

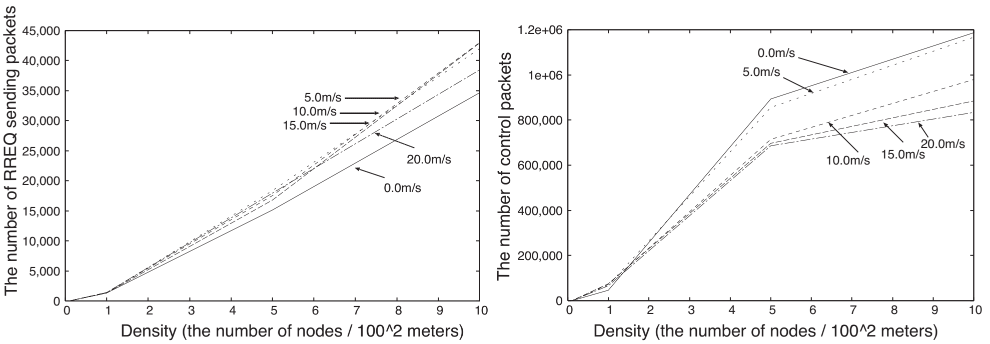
The number of RREQ packets indicates the extent of reachable area. On the other hand, the number of RREP packets indicates the stability of the path connection. The large number of RREP packets shows that the path is under the unstable condition and disconnected for the mobile activity of intermediate nodes.

Figure 6 shows the number of Route Reply (RREP) sending packets varying the node speed. We captured two properties; firstly, when the node speed is under 10 m/sec, then the number of RREP is restrained to small values over every density, secondly, when the node speed exceeds 15 m/sec, then the number of RREP suddenly increases, where the node density is as close to 10 nodes in 100 square meters. These results show that the stable and mobile nodes under 10 m/sec speed can communicate using the same path even if each node moves with 10 m/sec. When the speed, however, exceeds 15 m/sec, then the established paths are disconnected by the node mobility causing the increase of the number of RREP packets. The node density effects to increase of the number of RREP packets, since dense condition creates the path with long hops. The number of RREP sending packets and RREP receiving packets, which is not shown, vary with no significant differences.

4.2 DSR and DSDV control packets

Figure 7(a) illustrates the number of DSR RREQ packets varying the node speed. The number of RREQ increases in proportion to the node density, since that the reachable area spread widely in accordance with the increase of density. The number of DSR RREQ packets increases by 40 % compared to that of AODV meaning that AODV protocol behaves more effectively to establish the path than DSR protocol over every density.

As an example of Proactive type protocol, the number of control packets of DSDV is shown in Figure 7 (b). When the node density exceeds 1, control packets to maintain the routing table suddenly increase and reach the huge amount of packets for every speed. The DSDV protocol shows the inefficiency in terms of the node speed and the dense node condition.



(a) DSR Route Request (RREQ) packets (b) DSDV control packets

Fig. 7. The number of control packets varying the node speed.

4.3 Summary

Our simulation results explored following four performance features. Firstly, the high speed degrades in terms of AODV RREQ sending packets merely on the density of 10 nodes per 100 square meters. Secondly the high speed exceeding 15 m/s creates more RREP sending packets over dense node conditions. Thirdly, the number of DSR RREQ sending packets increases by 40 % compared to that of AODV. Finally, the DSDV protocol shows the inefficiency in terms of the node speed and the dense node condition.

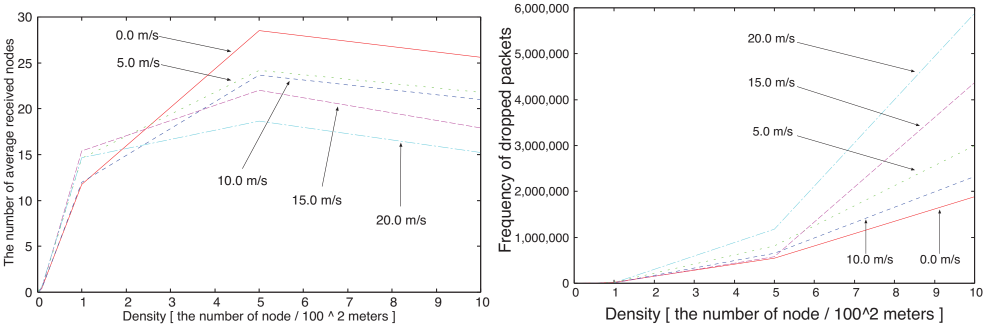
5. RREQ flooding features of the AODV protocol

In this section, mobile nodes are located randomly on the 1000 (m) * 1000 (m) square area, and move around this area with the same speed. The direction of movement changes after elapsing 10 simulation seconds with in a random manner. The density of intermediate mobile nodes varies with 1, 5 and 10 (nodes / 100^2 meters).

5.1 Flooding feature of RREQ

Figure 8 (a) and (b) shows degradation features of RREQ, such as the average number of RREQ adjacent nodes and the frequency of DROP packets in different conditions respectively. "DROP" means the sign indicated by the ns simulator. In Figure 8 (a), the average number of RREQ adjacent nodes increases linearly at the density = 5. In a denser condition, the increase of RREQ sending packets leads to packet collisions and finally packet drops. These conditions maintain almost the same number of RREQ receiving packets, and keep or decrease the average number of RREQ adjacent nodes. Figure 8 (b) shows the frequency of DROP packets. If the node speed is 15 or 20 (m/s), the DROP packets on density = 10 is five times more frequent that on density = 5. On the other hand, at such slow node speed as 0, 5 or 10 (m/s), the frequency of DROP packets on density = 10 is around two times, meaning that packet DROP is dependable on the node speed.

In more detailed analysis for trace data, the cause of DROP packets is mostly collisions on the simultaneous broadcasting. Contrary to unicast packets such as the RREP packets, RREQ flooding causes the performance degradation.

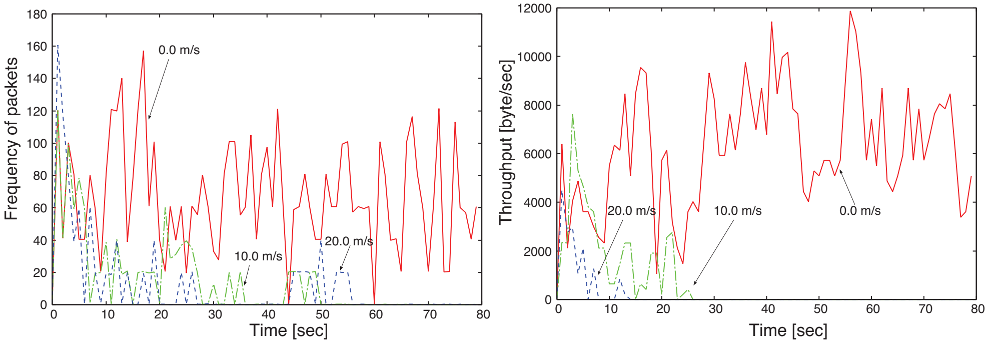


(a) Average number of RREQ adjacent nodes in different conditions

(b) Frequency of DROP packets in different conditions

Fig. 8. Degradation features of RREQ.

5.2 Time sequence features of RREQ flooding



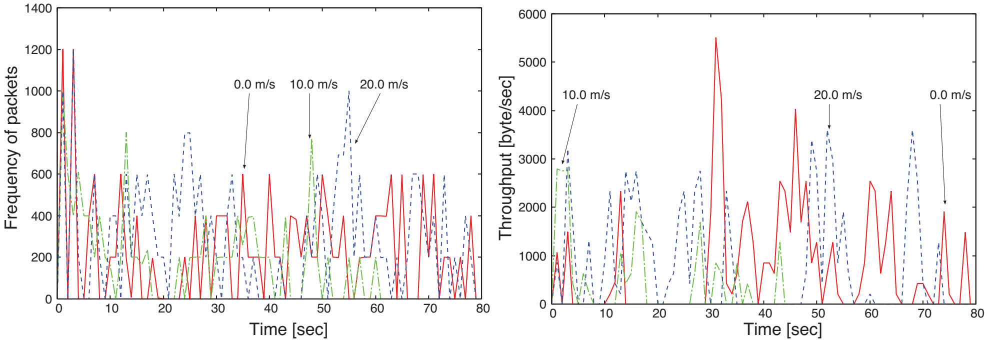
(a) RREQ sending packets

(b) TCP throughput

Fig. 9. Time sequence of frequency on the condition of density = 1.

Firstly, we examined the time sequence of frequency for RREQ sending packets and TCP throughput on the condition of density = 1 in Figure 2 (a) and (b) respectively varying the node speed on 0.0, 10.0 and 20.0 (m/s). Figure 9 (a) illustrates that the frequency of RREQ packets on the speed 0.0 keeps some amount of packets. On the other hand, when nodes have some speed such as 10.0 or 20.0 (m/s), the fluctuations of frequency gradually decrease and finally reach to zero traffic within 60 simulation seconds. These trends appear in the upper layer TCP as early as within 30 simulation seconds, and the TCP throughputs rapidly decrease to zero on 10.0 and 20.0 (m/s) case in Figure 9 (b). This performance degradation is caused by the unreachability between adjacent nodes on this course node condition.

On the dense condition such as the density = 10, Figure 10 (a) and (b) show the time sequence of frequency of RREQ sending packets and TCP throughput respectively. Each frequency of RREQ with different speeds spread over all simulation time, and the frequency with high speeds keep higher values than that with low speeds, meaning that the high speed movement requires more RREQ packet to maintain the route path. TCP throughput time sequences in Figure 10 (b) illustrates that high speed movement such as 20.0 (m/s) creates more efficient TCP throughput than the case of 10.0 (m/s), even if RREQ packets for 20.0 (m/s) are sent more frequently meaning that the RREQ packets contribute to establish the path between two end nodes and the TCP throughput.



(a) RREQ sending packets

(b) TCP throughput

Fig. 10. Time sequence of frequency on the condition of density = 10.

5.3 RREQ flooding patterns

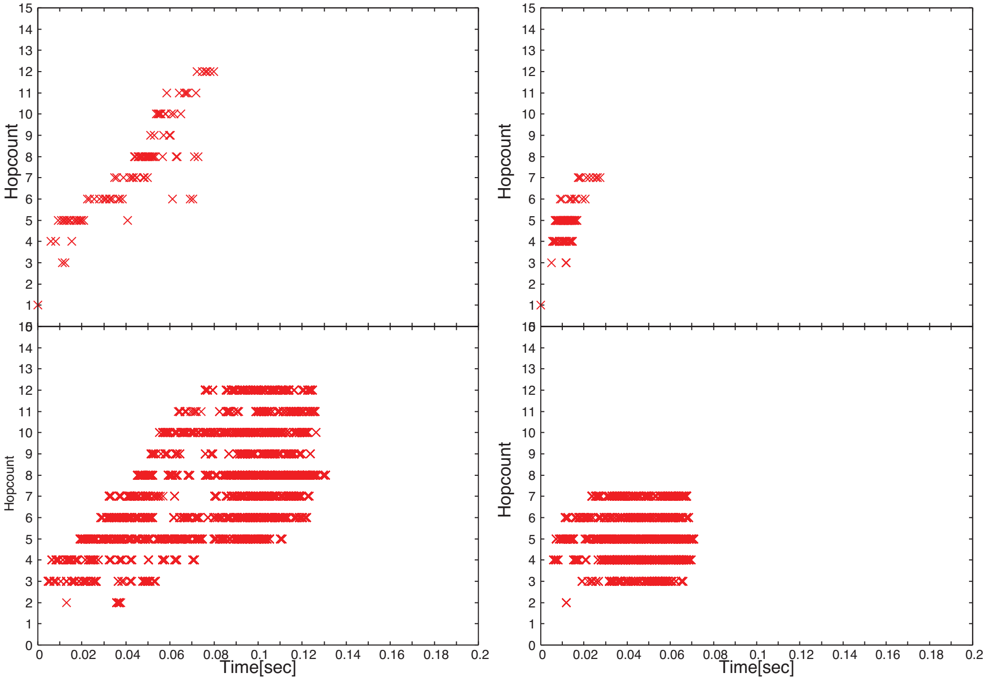
In this sub section, we examined the experiments and discussed flooding patterns showing the time sequences of flooding with the hop count. These experiments were carried out under the congested condition with density = 10 and node speed = 20.0 (m/s). As the result of experiments, we confirmed the flooding patterns visually using the nam, followed by classification into two patterns, "starting from the source node" and "starting from the intermediate node" Firstly, "starting from the source node" patterns are classified to two patterns depicted in Figure 11 (a) and 11 (b). The x-axis is the simulation time starting the first RREQ and first DROP, the y-axis is the hop count of RREQ and DROP packets. These properties are listed in Table 1 describing that main difference between the properties in Figure 11(a) and (b) is the average hop count.

In Table 1, "sending elapsed time" means the elapsed time from the start time of first RREQ packet to the last time of last RREQ packet over one RREQ flooding. "DROP elapsed time" also means the elapsed time from first DROP to last DROP.

We extracted the following features of RREQ flooding from Figure 11 (a). Firstly, delayed RREQ packets appear on hop count = 5 or 6 or 7. In general, flooding propagates sequentially from a node to an adjacent node. This propagation, however, is delayed causing the performance degradation. Secondly, long continuous DROPs appear after the

Properties	Pattern 1	Pattern 2
Maximum hop count	12	7
Average hop count	7.48	4.90
Sending elapsed time	0.079	0.029
DROP elapsed time	0.125	0.064

Table 1. Properties of the starting from source node pattern.



(a) Long hop count

(b) middle hop count

Fig. 11. Starting from the source node.

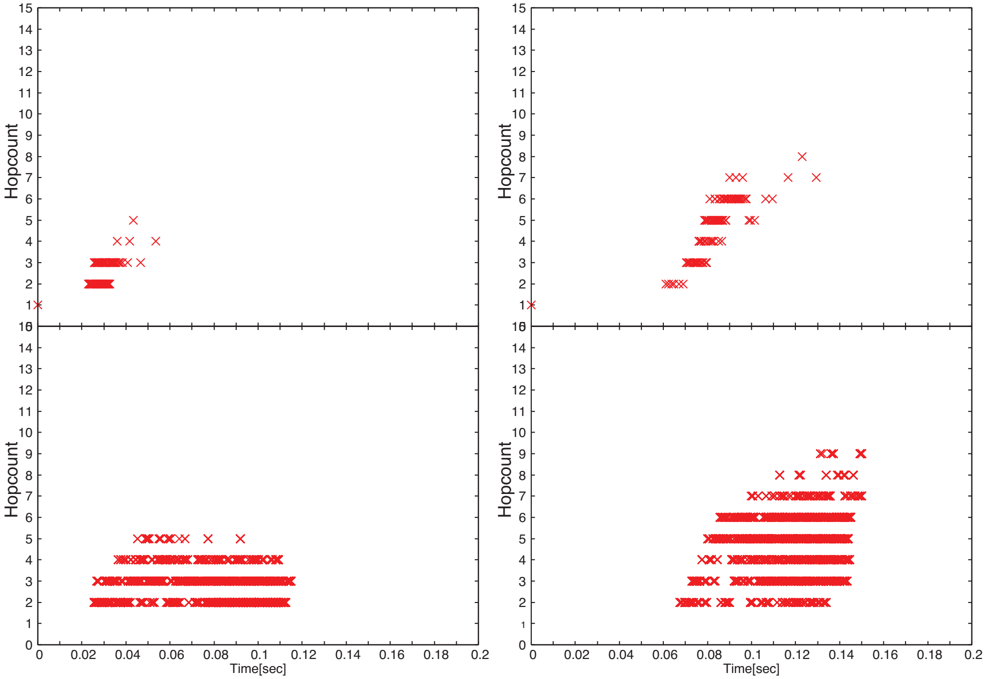
hop count = 5. Considering the first feature, the reason of these DROPs seem to be generated by the delayed RREQ propagation.

The flooding features of RREQ in Figure 11 (b) show the different propagation pattern. In this case, the continuous time of RREQ flooding and DROP packets are very short, meaning that the shorter path seems to be selected in this flooding pattern.

The flooding patterns of the other category, “start from the intermediate node”, are shown in Figure 12 (a)and (b) respectively. Both patterns (a) and (b) have the short and middle hop count respectively, since the start node is the intermediate node, and have the long propagation delay at the initial RREQ flooding. Table 2 shows that the case in Figure 12(b) has the long sending elapsed time. The reason of both long sending elapsed time and middle hop count is explained by the position of the starting intermediate node. The position of the intermediate node in Figure 12 (a) is in the middle of the moving area. On the other side, the position of the intermediate node in Figure 12 (b) is on the edge part of total communication area.

Properties	Pattern 1	Pattern 2
Maximum hop count	5	9
Average hop count	2.50	4.70
Sending elapsed time	0.053	0.133
DROP elapsed time	0.089	0.082

Table 2. Properties of the starting from intermediate node pattern.



(a) Short hop count

(b) middle hop count

Fig. 12. Starting from the intermediate node.

5.4 Summary

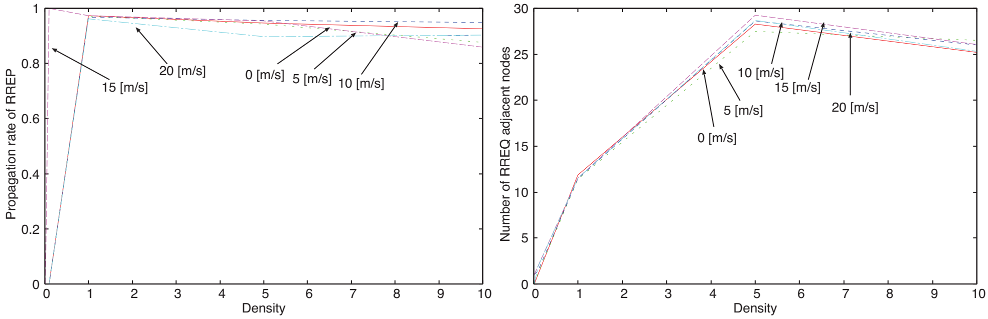
Our simulation results explored the following degradation features of RREQ flooding. Firstly, the main degradation of RREQ flooding is caused by the high density node condition and the high speed movement. Secondly, the reason of DROPs is generated by the delayed RREQ propagation on the case of RREQ flooding from the source node. Finally, the delayed start of RREQ flooding from the intermediate node generates the degradation of performance.

6. Propagation features of the AODV under different mobility models

In this section, mobile nodes are located randomly on the 1000 (m) * 1000 (m) square area to move around this area with the same speed. MNs start at 1.5 simulation seconds toward the selected direction. After the reach at the destination, MNs consume the pause time, and select the next direction. The pause time varies with 0 second, within 1 and 5 seconds, or within 1 and 10 seconds. The density of intermediate mobile nodes varies within 1, 5 and 10 (nodes / 100^2 meters).

6.1 Variation of each metric for different conditions

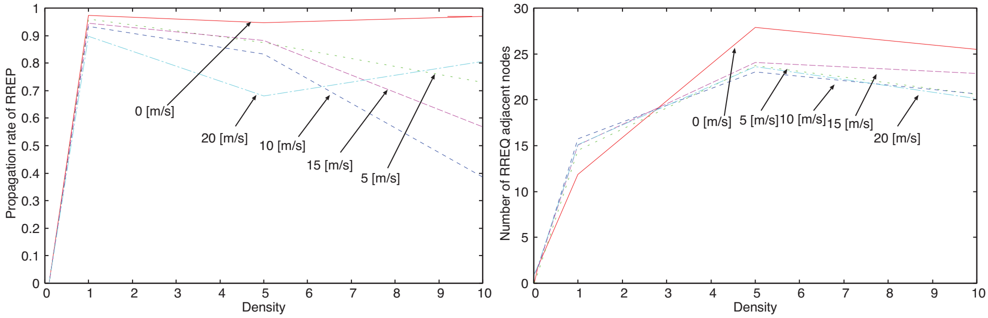
Figure 13 (a) and (b) respectively show the average propagation rate of RREP and the average number of RREQ adjacent nodes under the Random Direction Mobility Model with



(a) Average propagation rate of RREP (b) Average number of RREQ adjacent nodes

Fig. 13. Variation of each metric under the Random Direction Mobility Model.

zero pause time. Both figures represent these plotting patterns which are independent on speed and density. Compared to Figure 14, the plotted lines of both Figure 13 (a) and (b) maintain higher values than that in Figure 14 meaning that the lossless traffics are generated and MNs have reachable adjacent MNs with high density. The Random Direction Mobility Model has an advantage to move widely in the simulation area.



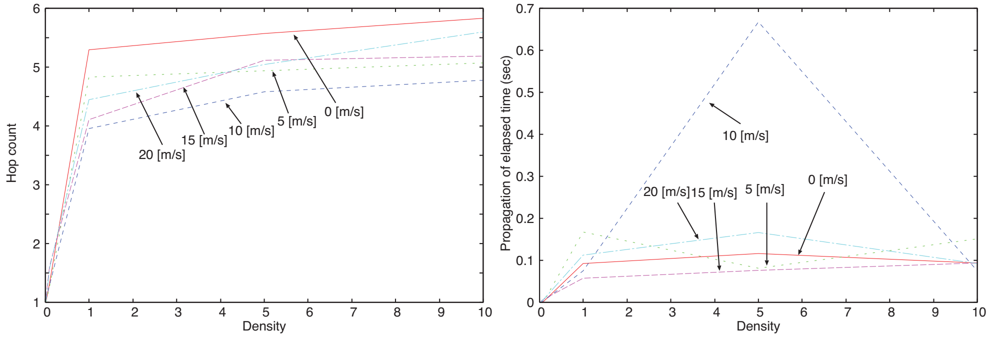
(a) Average propagation rate of RREP (b) Average number of RREQ adjacent nodes

Fig. 14. Variation of each metric under the Random Waypoint Mobility Model.

Figure 14 (a) and (b) show the same metrics in Figure 13 (a) and (b) with the RandomWaypoint Mobility Model with zero pause time. This model with zero pause time causes the largest degradation of values for each speed, making the condition of the density waves fall quickly - typically in a condition where the density is greater than 5 (nodes / 100*100 meters).

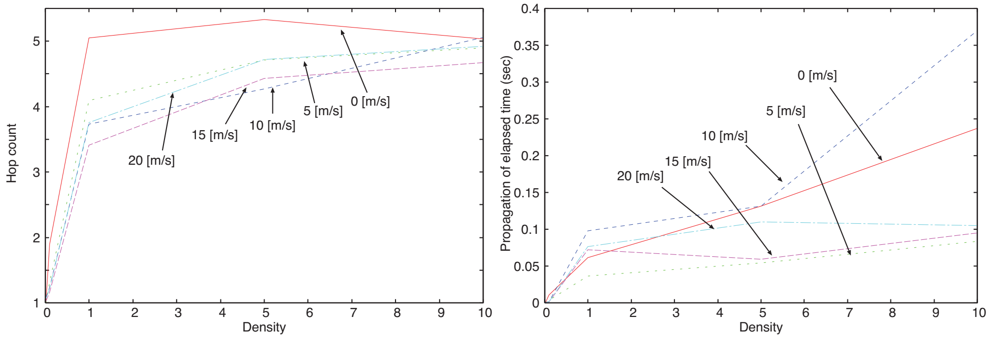
6.2 Flooding features of RREQ

Figure 15 (a) and (b) respectively shows the average hop count and the elapsed time of propagation under the Random Direction Mobility Model. On the other hand, Figure 16 illustrates the same metrics under the Random Waypoint Mobility Model with zero pause time. Compared to respective hop counts in Figure 16 (a), those in Figure 15 (a) maintain higher values meaning that these MNs widely spread over the simulation area and require



(a)Average hop count (b) Elapsed time of propagation

Fig. 15. Flooding features under the Random Direction Mobility Model.



(a)Average hop count (b) Elapsed time of Propagation

Fig. 16. Flooding features under the Random Waypoint Mobility Model.

additional hops. The hop counts of MNs are smaller than those of stationary nodes indicating that the mobility efficiently causes the short hop transmission from source to destination node. The elapsed time of propagation under the specific speed with the density = 5 (nodes / 100*100 meters) takes the transmission time intensively under the Random Direction Mobility Model in Figure 15 (b). Other additional experiments confirmed that the elapsed time of propagation was sensitive and fluctuated over the middle density = 5 for some MNs with the speed = 5, 10, 15 (m/s). It means that the mobile condition with middle density is unstable for the middle speeds. Except for an irregular phenomenon like this, the elapsed times of propagation are restrained in small values meaning that the effective transmissions occur under the Random Direction Mobility Model. On the other hand, an ascending pattern appears in a dense condition under the Random Waypoint Mobility Model in Figure 16 (b). This degradation is likely to be caused by the density waves.

6.3 Summary

Our simulation results explored the following features. Firstly, the Random Direction Mobility Model provides the widely distributed position for MNs over the simulation area leading to the loss-less propagations of RREP and reachable transmissions of RREQ for both speed and density. Secondly, flooding features of RREQ shows the degradation of the elapsed time of propagation caused by the density waves under the Random Waypoint Model.

7. Flooding features of the AODV under the different communication distances

In this section, mobile nodes are located randomly on the 1000 (m) * 1000 (m) square area to move through this area with the same speed. MNs start at 1.5 simulation seconds toward the selected direction. After reaching the destination, MNs consume the pause time to select the next direction. The pause time varies with 0 second, within 1 and 5 seconds, or within 1 and 10 seconds. The density of intermediate mobile nodes varies within 1, 5 and 10 (nodes / 100^2 meters). In addition, we examined the simulation varying the communication distance (radius) to 50, 100, 150, 200 and 250 meters.

7.1 RREQ adjacent nodes and propagation rate

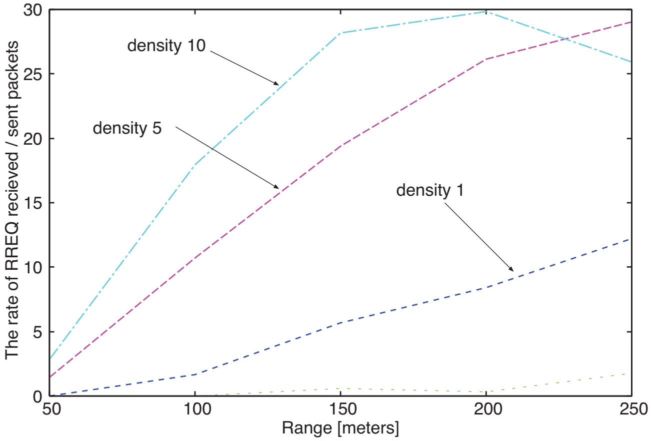


Fig. 17. The average number of RREQ adjacent nodes.

Figure 17 shows the average number of RREQ adjacent node as the function of the communication distance (meter) with three different distances = 1, 5 and 10. The average number of RREQ adjacent node in the sparse condition less then density = 5 linearly increases in terms of the communication distance meaning that the communication area widely spreads with no much RREQ packets and the reachable adjacent node increases. On the other hand, the adjacent node in the dense condition over density = 10 has the peak point meaning that many RREQ packets frequently generate the packet collision causing the decrease of the adjacent node. These different ascending patterns in terms of the density value are caused by the spread of communication area. The long distance communication assists the establishment of end-to-end communication in the sparse mobile node condition, while degrading the communication performance caused by the packet collision.

The average propagation rate of RREP as the function of the communication distance is illustrated in Figure 18. Each line for different density shows the reachable area under the specific communication distance. In addition, the average propagation rate of the density =

10 decreases when the communication area exceeds 200 meters. These results indicate that the communication distance of each density has the effective range. If we control the communication distance of mobile node, the performance could be upgraded.

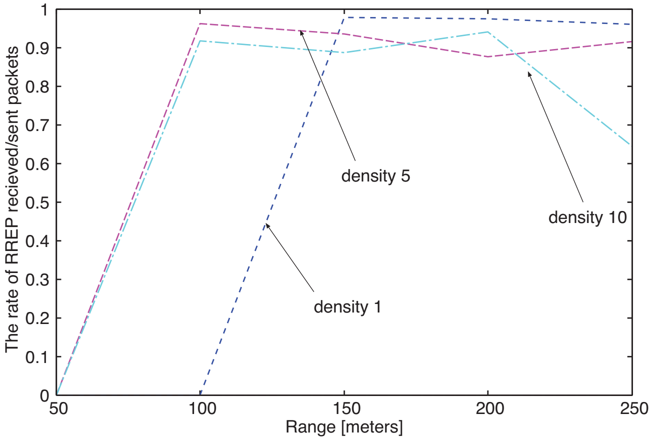
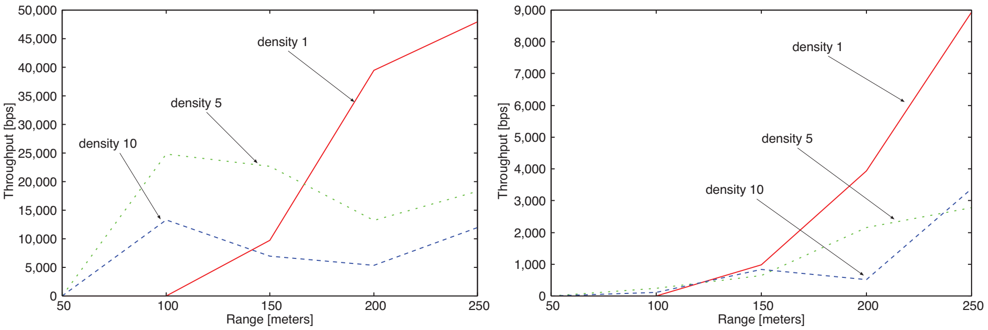


Fig. 18. The average propagation rate of RREP.

7.2 TCP throughput

Figure 19 (a) shows the TCP throughput as the function of the communication range (meter) with the three different density = 1, 5 and 10. In this stationary condition, the TCP performance with each density has the most effective value of the communication range for each other. For example, the TCP throughput with density = 5 shows the highest value under the communication range around 150 meters. The value of TCP performance could be kept highest in density under stationary condition, given the communication distances enabled to be tuned.

Figure 19 (b) shows the TCP throughput as the function of the communication range under the mobile node with the speed = 20 (m/s). Compared to the stationary condition in Figure 19 (a), Figure 19 (b) indicates that all TCP throughputs for the different density gradually



(a) Speed = 0 (m/s)

(b) Speed = 20 (m/s)

Fig. 19. TCP throughput.

increase in terms of the communication range. The results means if the communication range spread wider area leading to the expansion of the reachable area, the TCP throughput upgrades for every density. Controlling the communication range should enables us to tune maximum range to generate the high TCP performance regardless of the increase of RREQ packets and occurrence of packet collisions.

7.3 Summary

Our simulation results explored the following features. Firstly, the long distance communication assists the establishment of end-to-end communication in the sparse mobile node condition, while degrading the RREQ communication performance caused by the packet collision. Secondly, the communication distance of each density has the effective range in terms of the average propagation rate of RREP. Thirdly, tuning the communication distance keeps the TCP performance highest in the density under stationary condition. Finally, the maximum range to generate the high TCP performance should be tuned under mobile condition regardless of the increase of RREQ packets and occurrence of packet collisions.

8. Conclusion

In ad hoc mobile networks (MANET), the message exchange of the ad hoc routing protocol over the intermediate nodes is a significant factor that affects the effectiveness and performance. To focus on the performance of the routing protocols, mobility nodes are restricted to the intermediate nodes while communication nodes are restricted to two stationary end nodes in this section where the intermediate nodes operate merely as the routers for the application users. We especially focus on the AODV protocol and TCP throughput performance for stationary end nodes over the mobile intermediate nodes using the ns-2 network simulator. The Propagation properties of control packets, such as Route Request (RREQ) and Route Reply (RREP) as well as flooding properties of RREQ are extracted in our experiments.

In the comparison of different type of routing protocols, our simulation results explored two performance features. Firstly, route stability is more sensitive to the number of intermediate nodes than the speed of intermediate nodes for the AODV. Secondly, the number of RREQ packets of AODV linearly increases in terms of the number of node. In addition, we captured the following three features of TCP throughput performances for different protocols. Firstly, the TCP throughput on AODV with mobile intermediate nodes degrades about 20 % compared to stationary intermediate nodes in dense node condition, and exponentially degrades until 10 (m/sec) node speeds in a coarse node condition. Secondly, the TCP throughput on DSR degrades in accordance with the increase of node speed meaning that it is sensitive to the node speed. Thirdly, the TCP throughput on DSDV drastically degrades in the mobile environment.

In the performance of the AODV protocol, our simulation results explored following four performance features. Firstly, the high speed degrades in terms of AODV RREQ sending packets merely on the density of 10 nodes per 100 square meters. Secondly the high speed exceeding 15 m/s creates more RREP sending packets over dense node conditions. Thirdly, the number of DSR RREQ sending packets increases by 40 % compared to that of AODV.

Finally, the DSDV protocol shows the inefficiency in terms of the node speed and the dense node condition.

In the RREQ flooding features of the AODV protocol, our simulation results explored the following degradation features of RREQ flooding. Firstly, the main degradation of RREQ flooding is caused by the high density node condition and the high speed movement. Secondly, the reason of DROPs is generated by the delayed RREQ propagation on the case of RREQ flooding from the source node. Finally, the delayed start of RREQ flooding from the intermediate node generates the degradation of performance.

In the propagation features of the AODV under different mobility models, our simulation results explored the following features. Firstly, the Random Direction Mobility Model provides the widely distributed position for MNs over the simulation area leading to the loss-less propagations of RREP and reachable transmissions of RREQ for each speed and density. Secondly, flooding features of RREQ shows the degradation of the elapsed time of propagation caused by the density waves under the Random Waypoint Model.

In the flooding features of the AODV under the different communication distances, our simulation results explored the following features. Firstly, the long distance communication assists the establishment of end-to-end communication in the sparse mobile node condition, while degrading the RREQ communication performance caused by the packet collision. Secondly, the communication distance of each density has the effective range in terms of the average propagation rate of RREP. Thirdly, tuning the communication distance keeps the TCP performance highest in the density under stationary condition. Finally, the maximum range to generate the high TCP performance should be tuned under mobile condition regardless of the increase of RREQ packets and occurrence of packet collisions.

In the next step, we will propose the new mechanism to restrain the RREQ flooding in AODV protocol, and implement this mechanism in the ns-2 simulator to evaluate the effectiveness for the flooding performance.

9. References

[1] A. Boukerche, J. Linus and A. Saurabah, A performance study of dynamic source routing protocols for mobile and wireless ad hoc networks, *In 8th International Conference on parallel and Distributed Computing (EUROPAR 2002)*, pp.957–964, Spring-Verlag, 2002. Lecture Notes in Computer Science, LNCS 2400, 2002.

[2] C. Perkins, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, *In 18th ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM 1994)*, pp.234–244, 1994.

[3] S. Marinoni and H. H. Kari, Ad Hoc Routing Protocol Performance in a Realistic Environment, Proc. of The International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL'06), pp.96–105, 2006.

[4] X. Zhang and G. F. Riley, Performance of Routing Protocols in Very Large-Scale Mobile Wireless Ad Hoc Networks, *Proc. of 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, pp.115–124, 2005.

[5] C. Mbarushimana and A. Shahrabi, Comparative Study of Reactive and Proactive Routing Protocols Performance in Mobile Ad Hoc Networks, *Proc. of The 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, pp.679–684, 2007.

[6] UCB/LBNL/VINT groups. UCB/LBNL/VINT Network Simulator, http://www.isi.edu/nsnam/ns/, May, 2001.

[7] 802.11-1999 IEEE Standard for Information Technology - LAN/MAN Specific requirements - Part 11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification, 1999.

[8] C. E. Perkins and E. M. Royer, Ad Hoc On-Demand Distance Vector Routing, *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications 1999*, pp.90–100, February, 1999.

[9] D. B. Johnson and D. A. Maltz, Dynamic Source Routing in Ad HocWireless Networks, *Mobile Computing, Kluwer Academic Publishers*, Vol.353, pp.153–181, 1996

[10] T. Nakashima and T. Sueyoshi, A Performance Simulation for Stationary End Nodes in Ad Hoc Networks, International Journal of Innovative Computing, Information and Control, Vol.5, No.3, pp.707-716, March 2009.

[11] T. Camp, J. Boleng and V. Davies: A Survey of Mobility Models for Ad Hoc Network Research, *Wireless Communication and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, Vol.2, No.5, pp.483–502, 2002.

[12] E. Royer, P.M. Melliar-Smith and L. Moser: An analysis of the optimum node density for ad hoc mobile networks, *In Proceedings of the IEEE International Conference on Communications (ICC)* 2001.

[13] T. Yang, M. Ikeda, G. D. Marco and L. Barolli, Performance Behavior of AODV, DSR and DSDV Protocols for Different Radio Models in Ad-Hoc Sensor Networks, *Proc. Of the 2007 International Conference on Parallel Processing Workshops*, pp.1-6, 2007.

[14] C. Mbarushimana and A. Shahrabi, Comparative Study of Reactive and Proactive Routing Protocols Performance in Mobile Ad Hoc Networks, *Proc. of the 21st International Conference on Advanced Information Networking and Applications Workshops*, pp1-9, 2007.

[15] A.Tuteja, R. Gujral and S. Thalia, Comparative Performance Analysis of DSDV, AODV and DSR Routing Protocols in MANET using NS2, *Proc. of the 2010 International Conference on Advances in Computer Engineering*, pp330-333, 2010.

[16] Q. Feng, Z. Cai, J. Yang and X. Hu, A Performance Comparison of the Ad Hoc Network Protocols, *Proc. of the 2009 Second International Workshop on Computer Science and Engineering*, pp.293-297, 2009.

[17] E. Mahdipour, E. Aminian, M. Torabi and M. Zare, CBR Performance Evaluation over AODVand DSDV in RWMobility Model, *Proc. of the International Conference on Computer and Automation Engineering*, pp.238-242, 2009.

[18] N. Mahesh, T.V.P. Sundararajan and A. Shanmugam, Improving Performance of AODV Protocol using Gossip based approach, *Proc. of the International Conference on Computational Intelligence and Multimedia Applications 2007*, pp.448-452, 2007.

[19] N. Mishra, T. Ansari and S. Tapaswi, A Probabilistic based Approach to improve the performance and efficiency of AODV protocol, *Proc. of the Fourth International Conference on Wireless and Mobile Communications*, pp.125-129, 2008.

[20] H. Rehman and L. Wolf, Performance Enhancement in AODV with Accessibility Prediction, *Proc. of the 2007 IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems*, pp1-6, 2007.

[21] M. S. El-azhari, O. A. Al-amoudi, M. Woodward and I. Awan, Performance Analysis in AODV Based Protocols for MANETs, *Proc. of the 2009 International Conference on Advanced Information Networking and Applications Workshops*, pp.187-192, 2009.

30

The Dimensioning of Non-Token-Bucket Parameters for Efficient and Reliable QoS Routing Decisions in Bluetooth Ad Hoc Network

Halabi Hasbullah and Mahamod Ismail
Universiti Teknologi PETRONAS,
Universiti Kebangsaan Malaysia,
Malaysia

1. Introduction

1.1 Bluetooth applications and its technical issues

For specific applications, such as in Wireless Personal Area Network (WPAN), Bluetooth ad hoc network can be suitably deployed as a substitution for other technologies to provide the last meter connectivity solutions. In a WPAN, as specified in IEEE 802.15 Specification, nearby devices can be connected together for short-range communications to form a personal networking setup, for instance between a hand phone and an ear phone, an access point and a PDA, a GPS receiver and a navigator, etc. The other short-range technologies, such as Wireless Sensor Network (WSN), ZigBee, and Radio Frequency IDentification (RFID) may not be able to create such personal networking capability. Bluetooth communications technology was originally designed and intended to replace the cable connectivity between these nearby devices. Now, all the devices within the WPAN are neatly and seamlessly connected together without cable cumbersome, as well as providing mobility support to end users. To some extent, roaming facility is provided to allow a user to move around from one coverage area to another. This roaming service may be applicable in a museum or shopping mall where information about items or products is transmitted transparently to users' or clients' mobile devices, while they are on the move. In this way, the users would be more informative about the services and products offering around them.

However, Bluetooth ad hoc network is constrained by limited resources due to its low-power and short-range communication capability, as described by (Haartsen, 1998). The smallest networking unit of Bluetooth, called piconet, can support up to 8 active mobile devices at a time. A greater number of mobile devices in vicinity can only be supported by creating multiple piconets, which now a scatternet that interconnects multiple piconets is developed. Additionally, its communications range can only cover to a maximum of 100 meters and beyond this range multi-hop communication using multiple relay devices is required. This is in contradiction to WLAN, WiFi and WiMAX setups, where hundreds of mobile devices are handled simultaneously and wider coverage area is obtained by using only a single cell. However, with no exception, Bluetooth ad hoc network is also carrying multimedia, interactive, and real-time data, and the demands for transporting these data types over the network is keep increasing every day. Simply, Bluetooth is a limited-resources network type but to accommodate growing and demanding applications.

Figure 1 depicted a typical Bluetooth scatternet topology, created by a number of interconnected piconets. A piconet is made up of one master node and a maximum of 7 active slave nodes, where the master node is in control of all the slave nodes. Two slave nodes in a piconet cannot directly communicate with each other, but communication is allowed after passing through the master node that controls them. A node that resides in the overlapping coverage area between two or more scatternets is called a bride or relay node. A node may become a slave node in a piconet and at the same time acts as a master node in another piconet. In this case, role switching is required at different time instances to change the role from a master node to a slave node, and vice versa. As a result, switching time is required, thus delay transmission time is introduced.

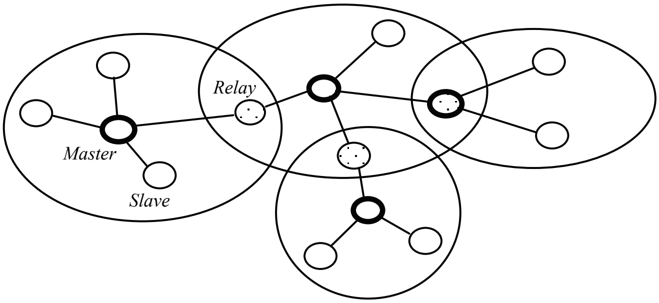


Fig. 1. A typical Bluetooth ad hoc network with a topology of scatternet

A relay node that connects two piconets is likely to behave as a router node in the scatternet, forwarding packets from one piconet to another piconet in the topology. Hence, a router node is in the position of making decision to select one forwarding link from the available outgoing links of that node, based on certain decision criteria. For reliable packets transmission, redundant links may be selected with additional costs. The selected link not only has to satisfy the demand for the required resources, but also to provide certain degree of service guarantee to the requested application, and ultimately to the end users.

Hence, one could guess that the basic technical challenge that appeared in the Bluetooth ad hoc network is its limited communication capabilities, derived from its limited transmission power that it can emit, limited distance that it can cover, and limited device number that it can handle simultaneously. These limitations however, should not hinder the Bluetooth network from providing similar services as that of the other wireless networks have offered. Considering the limitations, therefore providing Quality of Service (QoS) in this network has been a major issue. One of such QoS issues is efficient and reliable routing in the network, by which a level of service guarantee must be provided for every single packet sent. The efficiency can be interpreted as consuming as much low power energy as possible to prolong the network lifetime, i.e. the lesser energy consumed the longer lifetime of the overall network. Reliability can be defined as having low or no packet lost probability in each transmission in order to uphold the data integrity. High packet lost probability is a reflection of unreliable transmissions. To achieve these two QoS goals, every single device in the Bluetooth network, regardless of their role of whether master or slave node, is expected to use the available but limited resources wisely during the course of routing and transmitting. However, their contributions for the transmission efficiency and reliability are determined by several internal and external factors, such as bursty traffic pattern, channel condition, etc.

1.2 Bluetooth and traffic characterization

The characteristic of today's traffic is naturally bursty (Leland et al., 1994). Real-time and interactive applications from web browsing, audio/video conferencing, video-on-demand, forecasting, sensoring, on-line transactions, and multiparty games are generating bursty traffics. Traffic is considered as bursty when there is an unpredictable fluctuations of similar data blocks patterns that appeared at any point of times during its transmission. In its simplest form, burst is defined by (Handle et al., 1996) as a ratio between peak and mean bit rate. In many cases, burst can be triggered from rare events. For example, when several large size file transfers is detected as compared to average size file transfers, several intervals of too long packets arrival times as compared to average packets arrival times, etc. In another form, burst in a traffic stream is determined by its high variability over a time series.

Bursty traffic, as described by (Park et al., 1996), is known to associate with self-similar property. Also, as had been determined by (Leland et al., 1994), bursty traffic induced a direct impact on network performance, which may be in the form of unreliable routing decision at router node. The variability in the input traffic does not allow network performance to be accurately measured, i.e. performance consistency does not appear at different time series. Additionally, the parameters that contribute to the inconsistency may not be identifiable. As a result, network control software could not be designed to effectively and seamlessly manage the router nodes, and ultimately the network system may fail to perform as expected. If this happened, QoS is hard to achieve because the traffic's characteristic cannot be accurately described to Resource Manager for reservation and allocation of resources when routing decision is about to be made at a router node. Obviously, efficiency, reliability, and service guarantee in the routing function will not be achieved in the network. Simply, burst of traffic has an impact on how a mobile device in the scatternet makes its routing decisions, as well as on the overall performance of the Bluetooth network. On other factor, channel quality of a link that connects between two adjacent mobile nodes may also affect the network performance, particularly when QoS routing decision is to be made by one of these nodes. High bit error rate on the selected link may lead to low transmission quality, and high packet lost probability may result in unreliable connectivity.

Hence, if some parameters of the source bursty traffic and the error-prone wireless channel can be accurately dimensioned for resource reservation and allocation to Resource Manager, a better level of QoS can then be granted to user applications. For this purpose, a *traffic-descriptor* is required to describe the characteristics of the input traffic and the current wireless channel condition to the Resource Manager for appropriate QoS requirements. In other words, by having a traffic-descriptor, the QoS demands for a specific routing task can now be more accurately stated. In this way, the achievement toward QoS provisioning in Bluetooth ad hoc network is more promising than before and thus, promoting the Bluetooth technology and its applications to a greater height. Token-Bucket scheme can be used to produce such traffic-descriptor.

The development of a traffic-descriptor so far however, had only considered the Token-Bucket parameters, while the uses of non-Token-Bucket parameters, which they may contribute significantly to QoS provisioning, have been omitted. The prime reason for inclusion of the non-Token-Bucket parameters is to meet the various and flexible QoS demands, while Token-Bucket can only provide standard and fixed parameters with limited usage. Importantly, a parsimonious traffic-descriptor is required, where only least parameters are used in the traffic-descriptor, but it has the ability to efficiently and reliably providing QoS routing decisions at a router node. Hence, if the resource-limited Bluetooth ad hoc network is to be used effectively in supporting the WPAN implementation, a parsimonious traffic-descriptor that also contained the non-Token-Bucket parameters must be developed.

Figure 2 illustrates router nodes of *i* and *k*, at which decisions are to be made to select one forwarding link from the available outgoing links to forward their received packets to the next node, based on certain decision criteria (e.g. the bit rate of the link and length of the link). To execute this routing task, a mapping between the available bit rate of the link and the requested bit rate from the application is performed, and see whether or not this request can be fulfilled. The selected forwarding link not only has to satisfy the demand of the bit rate, but also to provide certain degree of efficiency and reliability for the forwarding task. However, in many nowadays complex and sophisticated application scenarios, there are multiple QoS demands that need to be satisfied. For this reason, the selected forwarding link may only be able to provide an optimal solution to the QoS routing function. Note that a complete route is made up of a set of links. Therefore, a set of routing decisions are to be made along a complete route that connects a pair of source node and sink node in the scatternet topology. Hence, the final result shall be exhibited through an optimal use of the limited resources over the complete route.

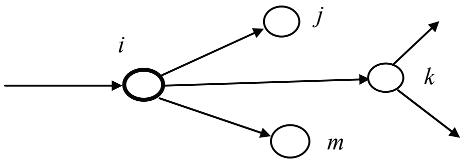


Fig. 2. Routing decision at router nodes *i* and *k*

Therefore, to describe the characteristics of the source traffic and forwarding link for resource reservation and allocation during routing decision, a traffic characterization process is required, such that resources are appropriately reserved and allocated. To support the characterization process, a mathematical model must be developed, which will include the parameters of input traffic, channel condition, and others that are expected to have direct impact on the routing performance, as well as on the network QoS provisioning.

1.3 The issues in traffic characterization

However, in developing an effective traffic-descriptor, a problem has appeared during the formulation of a mathematical model that could parsimoniously incorporate only least possible parameters, but well describing the application's QoS requirements. Specifically, how the non-Token-Bucket parameters, which may contribute significantly to the efficiency and reliability of a routing function can be embedded into the traffic-descriptor. This is critical for Bluetooth ad hoc network for two very basic reasons. First, most of the time, only a single link is provided between any two communicating devices, thus no option available to select any other efficient and reliable links. This situation of only-one-link availability is created in Bluetooth network because of the master-driven communication approach, where nearby slave nodes in a piconet cannot directly communicate, except through a master node. Second, Bluetooth is a resource-limited network provided with low-power and short-range communication capabilities, and hence the available resources must be used efficiently. Hence, with parsimonious traffic-descriptor, the processing time would be faster and the energy usage would be lower. In short, Bluetooth ad hoc network has many constraints with respect to its ability to fulfill the ever-increasing demands for the QoS guarantees.

Therefore, the objective of this work is to develop a parsimonious traffic-descriptor that could incorporate the non-Token-Bucket parameters, together with the standard Token-Bucket parameters so that an efficient and reliable QoS routing decisions could be made at a router node. This requires the task of dimensioning the traffic-desriptor's parameters, i.e. the determination of the parameters and analyze their impacts in the routing decision-making processs at a router node. In order to dimension the parameters, characterization process on the environment of the routing function is required.

For the purpose of producing the such traffic-descriptor, a mathematical model that measures the burstiness level α and the degree of self-similarity H in a traffic stream, as well as the channel quality (via bit error rate) must be developed. Using this deterministic information, the QoS provisioning shall be more guaranteed. However, the exact procedures for resource reservation and allocation are not within the scope of this research work; it shall be given to Resource Manager to manage the complete procedures. Thus, this work is limited to only producing traffic-descriptor for Resource Manager to use.

To verify the usefullness of the developed parsimonious traffic-descriptor in providing supports for QoS routing in Bluetooth ad hoc network, a Matlab simulated router node was developed to measure its performance. It was found that a parsimonious traffic-descriptor could be generated with a promising performance. That is to say, with only few parameters, the traffic-descriptor has been able to meet the required QoS routing demands. With the result, it has promoted Bluetooth ad hoc network to a higher level of usefulness and practical applications, particularly in solving the last-meter connectivity issues.

The rest of this chapter is organized as follows. Section 2 discusses self-similar property that appeared in a traffic burst and the use of Token-Bucket scheme in characterizing the burst. Section 3 explains the methodology to derive a traffic-descriptor. Results and analysis on the performance of the developed traffic-descriptor in supporting QoS routing decision at a router node are presented in Section 4. Finally, conclusion and future work are made in Section 5.

2. Related works

2.1 Self-similar property in the burst

It has been a long belief that network traffic pattern is following Poisson distribution. However, in actual fact it has been proved that it was only applicable to speech data of the telephony system. Discovery by (Leland et al., 1994) has provided evidence that the inter-arrival times for bursty traffic in local area network (LAN) is actually following heavy-tailed distribution of power law. Their study on Ethernet LAN traffics from 1989 to 1992 has established that fractal (or self-similar) property in a traffic stream could not any longer be captured by conventional traffic model. Supported by (Paxson, 1995), it was confirmed that the packets' inter-arrival times have deviated away from exponential distribution, but following a heavy-tailed distribution. Heavy-tailed distribution is characterized by a slowly decaying tail of the hyperbolic graph to infinity. Additionally, a work by (Beran, 1994) has determined that variable bit rate of MPEG video traffic is associated to self-similarity, which is common property for the bursty traffics.

As determined by the Bluetooth Specifications v1.0B, (1999), Bluetooth network implements Segmentation and Reassembly (SAR) protocol at L2CAP layer, by which long message blocks received from upper layers are segmented into smaller packets types of DMx or DHx (M – Medium, H – High, x = 1, 3, 5 slots). Since the network is also handling bursty input traffic, it is anticipated that the SAR protocol execution on MPEG video data might produce the same heavy-tailed distribution with respect to some of its features.

To describe heavy-tailed distribution and self-similar property, second order statistics is required. Heavy-tailed distribution is defined by (Crovella & Lipsky, 1997) as follows. Let X be a random variable with cumulative distribution function (cdf) of $F(x) = P(X \leq x)$ and complementary cumulative distribution function (ccdf) of $\overline{F}(x) = 1 - F(x) = P(X > x)$. A distribution $F(x)$ is said to be heavy-tailed if

$$\overline{F}(x) = P(X > x) \sim cx^{-\alpha} \tag{1}$$

when $x \to \infty$ for positive c value and $0 < \alpha < 2$. In other words, a distribution is heavy-tailed if the ratio of $P(X > x)/x^{-\alpha}$ is approaching 1 when $x \to \infty$ for $\alpha > 0$. The asymptotic form of the distribution is following a power law distribution. One of the simplest heavy-tailed distribution is Pareto distribution with probability distribution function (pdf) of the form $f(x) = \alpha k^{\alpha} x^{-\alpha-1}$, where $\alpha > 0$, $0 < k \leq$ x. Accordingly, the distribution is respectively having cdf of

$$F(x) = P(X \leq x) = 1 - (k/x)^{\alpha} \tag{2}$$

and ccdf of

$$\overline{F}(x) = P(X > x) = (k/x)^{\alpha} \tag{3}$$

where α is a shape parameter and k is a scale parameter.

The mean for Pareto distribution is $\mu = \alpha k/(\alpha - 1)$ and the variance is $\sigma^2 = \alpha k^2/(\alpha - 1)^2(\alpha - 2)$. If $\alpha < 1$, the distribution would have infinite mean; if $\alpha < 2$, the distribution would have infinite variance; if $1 < \alpha < 2$, it would has finite mean and infinite variance; and if $\alpha \geq 2$, both mean and variance are finite. In general, if its variance is infinite, then X would associate to high variability in its distribution.

One important property of the heavy-tailed distribution is that it is self-similar, as have been proved by (Leland et al., 1994) and supported by (Crovella & Bestavros, 1999). Additionally, as claimed by (Taqqu et al., 1997), superimposition of several independent of ON/OFF heavy-tailed traffic sources is just enough to produce a self-similar traffic stream.

Self-similarity is defined by (Fernandes et al., 2003) as follows. Let $X(t)$ be a wide-sense stationary time series with mean μ, variance σ^2, and autocorrelation function $\rho(\tau)$. Let $X^m(t)$ be a newly derived time series from $X(t)$ by averaging a number of m non-overlapping block sizes. Its aggregated series is

$$X^{\mathrm{m}}(t) = (\mathrm{m}^{-1})(X_{\mathrm{tm-m+1}} + X_{\mathrm{tm-m+2}} + \ldots + X_{\mathrm{tm}}) \tag{4}$$

and $\rho^{\mathrm{m}}(\tau)$ is its autocorrelation function. Process $X(t)$ is said to be self-similar if

$$\rho^m(\tau) = \rho(\tau) \quad \text{for } m = 1, 2, 3, \ldots \tag{5}$$

A work by (Park et al., 1996) has proved that self-similar property has a direct impact on network performance. Also, as identified by (Leland et al., 1994), if it is known that the source traffic is bursty, two definite consequences will occur: the increase in buffer requirement and the longer delay experienced.

To characterize the burst property, it is important to identify the level of burstiness α and the degree of self-similarity H in the source traffic stream. The relationship between α and H for Pareto distribution has been derived by (Leland et al., 1994), which is expressed as $H = (3 - \alpha)/2$. As stated by (Hadzi-Velkov & Garrilovska, 1999), α is an indicator of the burstiness level in a traffic stream. For the source traffic to have a heavy-tailed distribution, an interval of $1 < \alpha < 2$ must be obtained, where $\alpha \rightarrow 1$ indicates too bursty traffic. On the other hand, to measure the degree of self-similarity, an interval of $0.5 \leq H < 1$ is to be obtained. $H \rightarrow 1$ indicates a high degree of self-similarity.

Combined with some other characteristics of the traffic stream and/or system, such as packets' efficiency and channel quality, the QoS requirements of an application can be described to Resource Manager in a much more accurate manner. As have been discussed, it is stated in a traffic-descriptor. In this way, much better resource reservation and allocation could be made, and deterministic network performance could be measured and obtained. Then, QoS can be granted to user applications with higher degree of confidence.

2.2 Token-Bucket scheme

There have been many works in the literature describing on the use of Token-Bucket (TB) scheme for traffic regulation, which the usage was based only on its own basic parameters. Work by (Norashidah & Norsheila, 2007) has improvised the TB by including a fuzzy logic component, resulting in a fuzzy logic TB predictor that has the ability to adapt its token rate based on actual traffic requirements. In this way, the actual bandwidth requirement can be predicted and feedback is relayed to Admission Control mechanism. In order to characterize the bursty input traffic, TB scheme may be used in a 2-in-1 combined function as proposed by (Procissi et al., 2001). The first function is to regulate the arriving burst to a more controllable and deterministic form of traffic flow. The second function is to characterize the incoming self-similar traffic so that the conbtributing parameters could be identified and measured. Subsequently, a traffic-descriptor is produced.

Typically, a traffic-descriptor is expressed as (ρ, b), where ρ is the token rate and b is the bucket size, and both are the basic TB parameters. Papers by (Li, 2002) and (Glasmann et al., 2000) have elaborated the production of a traffic-descriptor using TB scheme. However, work by (Yang, 2000) has suggested a traffic-descriptor of the form (ρ, unlimited), by which the bucket size b can be as large as possible. However, as confirmed by queuing theory, the larger bucket size the longer processing delay experienced by the packets in the queue and in other subsequent processing tasks. This claim is supported by a scheduling work of (Garroppo et al., 2001). In summary, study by (Procissi et al., 2001) could be the best piece of work that has taken into account the self-similar property of the source traffic for QoS routing decisions at a router node, by which the other previous works did not.

However, the critical issue in QoS provisioning in Bluetooth ad hoc network is to find an appropriate probabilistic model for the source traffic, in particular if stochastic approach is used as described by (Valaee & Gregoire, 2005). As determined by (Li, 2002), the production of a traffic-descriptor is application-dependent and case-sensitive. Furthermore, as stated by (Glasmann et al., 2000) that until today, a standard procedure to produce a traffic-descriptor is still none existence. Therefore, the development of a parsimonious traffic-descriptor for Bluetooth ad hoc network is still an open issue and research opportunities on it are wide open. Parsimonious is referred to as having only least possible parameters but well describing the system in question, whilst achieving a level of efficiency and reliability in routing. This is because the more parameters to handle, the more resources are required, which may not lead to QoS provisioning. Therefore, parsimonious is very much needed in Bluetooth ad hoc network since it has only limited resources to offer.

However, the TB scheme alone may not be able to completely describe the source traffic and the network characteristics. Therefore, some other components are required to work with the TB, by which these components will provide the necessary additional non-TB parameters. The expected result shall be of a more accurate reservation and allocation of network resources from users applications for each routing decision made at a router node.

3. Methodology

3.1 The system model

In a Bluetooth scatternet topology, the master node is usually the one that makes most routing decisions. However, depending on the established topology, a slave node may also be in the position of making the same routing decisions, especially when the node is a relay/bridge node connecting two or more piconets. The decision must not only accomodating the constrainted resources but also meeting the QoS demands of the applications in terms of its efficiency and reliability. For this reason, only optimal routing solution may be achieved at any point of routing instances. To achieve this QoS routing requirement, Token Bucket (TB) approach is suggested to be used to handle the burst issue.

In this work, TB smoothing scheme is initially used to smooth out the bursty traffic input, as described by (Yang, 2000). This is done by applying buffer queue for the received burst, while the non-burst can easily be processed at normal rate. With the smoothed, shaped, and controlled traffic, the network performance could be measured and its performance parameters could be identified. In this way, controll software could then be developed to handle the burtsy traffic in an automated manner. However, the function provided by the standard TB with its basic parameters, will not be able to handle the issue of self-similar property that come together with the burst. The basic function of TB scheme is only able to smooth out the external bursty property of the input traffic while its internal property of self-similar pattern still remain unexplained. The self-similar property must be handled in different manner, which a statistical tool may suitably be used. Therefore, the TB scheme must also be equipped with a set of non-standard TB parameters that takes into account the performance effects of these parameters when they get involved in routing decision process.

Figure 3 represents a system model of a router node, where routing decisions are to be made at the router node, based on certain decision criteria, to select the best possible forwarding link from the available outgoing links, whilst meeting the QoS demands of the application. Based on this routing model, routing decisions are to be simulated. To support the need to include the non-standard TB parameters for QoS routing decisions, the TB is proposed to work together with a Transmission Controller (TC). As standard TB can only provide basic TB parameters, hence TC is expected to provide the necessary non-TB parameters. In summary, the routing decisions are to be made by the combined function of Token Bucket (TB) and Transmission Controller (TC).

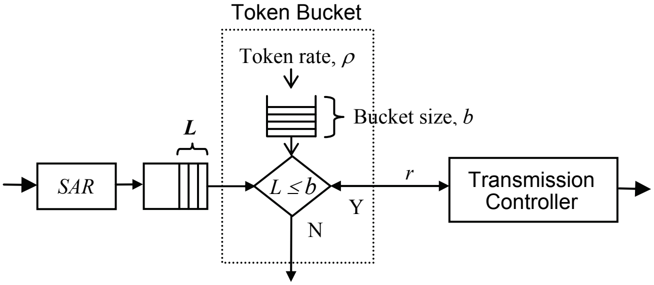


Fig. 3. The system model of a router node in Bluetooth scatternet

At the input point before submission to TB, the Segmentation and Reassembly (SAR) protocol accepts frames from upper layer and segments them into smaller packets of DHx or DMx. The types of packet produced will be determined by the SAR algorithm, which in this work Best-Fit algorithm was chosen. These packets are then put in a buffer queue with length *L*, where the queue management is a simple first-in-first-out (FIFO). A packet of size *L* can only proceed with transmission when there is a bucket size *b* to carry the packet, i.e. $L \leq b$. On the other hand, if $L > b$, the bucket size cannot accommodate the packet size, thus discarded from the system. When this happened, there exists packet loss probability. In a system with bursty traffic source, this loss probability is allowed to occur but to a minimum with proper control mechanism.

TB is used together with TC to produce a traffic-descriptor, which then will be used for resources reservation for QoS routing decisions. Simply, a traffic-descriptor describes the QoS requirement of the input traffic. The main function of a TB is to make routing decisions. However, TB is only parameterized with its basic parameters of ρ (token rate), *b* (bucket size), *p* (peak rate), *m* (minimum controlled unit), and *M* (maximum packet size), as described by (Glasmann et al., 2000). Therefore, in this work, TC is proposed to work with TB for the reason that the basic parameters of TB are not sufficient to completely describe the QoS requirements if other decision criteria are to be considered in the decision model.

The fundamental function of a TC is to control the transmission: if the QoS requirement of the traffic is fulfilled against the available resources, forwarding of packet data over a link is allowed; else select the other links (if available). If other links are not found, just use the only single available link, probably with QoS adaptation. When adaptation is required, for instance when the available resources are not sufficient to fulfill the traffic and QoS requirements, TC will send feedback to TB, by which necessary adjustment is then executed.

3.2 The mathematical model

Based on the system model and the requirement to handle the self-similar input traffic with Pareto distribution (α, k), according to (Li, 2002), the probability that a packet will have a size of length $L > b$ is

$$p = P(L > b) = \int_b^\infty f(x)dx = \int_b^\infty \frac{\alpha k^\alpha}{x^{\alpha+1}}dx = (k / b)^\alpha \qquad (6)$$

where *f(x)* is the pdf for packet size *L*, α is the shape parameter ($\alpha > 1$), and *k* is the scale parameter that limits the *b* value. This equation can also be interpreted as packet lost probability, i.e. the probability a packet will be discarded. It is observed that *p* is a function of *b*. When a graph of *p* versus *b* is plotted, a hyperbolic graph with slow decaying rate is obtained, and it goes to infinity.

On the other hand, transmission delay *d* experienced by a packet from this router node to the next node over a selected forwarding link is expressed by (Garroppo et al., 2001) as

$$d = b / r \tag{7}$$

where *r* is the bit rate of the forwarding link. Since, each packet type of DHx or DMx is having its own maximum bit rate, then *r* is assumed to have the appropriate bit rate for the transmitted packet. As can be seen, *d* is directly related to TB via *b* but indirectly related to TB via *r*, since *r* is not a basic TB parameter. In this case, *r* is to be obtained from the TC, which is a non-TB parameter. When a graph of *d* versus *b* is plotted, a straight-line graph is obtained.

The mathematics of algebra is then used to derive the (*r*, *b*) pair, which is known as a traffic-descriptor. This pair value is the intersection point between the graphs of hyperbolic and straight line, as a result of equating Equation (6) to Equation (7). Given *r*, the required bucket size can be computed as

$$b = (rk^{\alpha})^{1/(1+\alpha)} \tag{8}$$

Therefore, by taking into account the bit rate of the forwarding link suitable for transmission of a specific packet type, the QoS routing decisions can now be expected to be more efficient and reliable. The (*r*, *b*) pair could also be interpreted as optimal routing solution considering the various application requirements and the limited Bluetooth resources.

However, in order to provide a much better level of QoS routing decisions at a router node, further improvement could be made on the above mathematical model. In this work, specific characteristics of the source traffic and the quality of the forwarding link are further investigated. By doing so, improved efficiency and reliability for the QoS routing decisions can be expected.

One of the interesting characteristics from the source traffic is the efficiency of Bluetooth packet. The packet's efficiency is stated in (Kim et al., 2001) as $\varepsilon = \varphi / ((\xi + 1) \text{ x } \delta)$, where φ is maximum bit number for a packet type, $(\xi + 1)$ is the number of slot for a single packet inclusive its acknowledgment slot, and δ is the length of a slot in bit. The efficiency of each packet type has been computed and tabulated in Table 1.

Packet type	**Efficiency, ε**
DH5	0.72
DM5	0.48
DH3	0.59
DM3	0.39
DH1	0.17
DM1	0.10

Table 1. Bluetooth packet efficiency values

Further improvement on efficiency and reliability of QoS routing can also be achieved from the quality of the outgoing link. Typically, the quality of a forwarding link can be measured from its *PER* value, which is determined by the packet type forwarded on that link. They are stated by (Chen et al., 2004) respectively as follows for different packet types,

$$PER = 1 - (1 - BER)^{s} \qquad \text{for DHx} \tag{9}$$

$$PER = 1 - ((1 - BER)^{15} + 15 \text{ x } BER \text{ x } (1 - BER)^{14})^{s/15} \qquad \text{for DMx} \tag{10}$$

where *s* is the maximum packet size (user payload) in bit unit, and *BER* is the bit error rate of the link. It is assumed that a router node has the ability to measure the *BER* of each of its outgoing links. In Bluetooth network, the *BER* value shall not be greater than 10^{-3} for good signal reception. Taking into account the packet efficiency and the channel quality, the effective bit rate $R(X)$ as derived by (Kim et al., 2001) for DHx or DMx packets can now be expressed as

$$R(X) = (1 - PER(X)) \times \varepsilon \times \psi \tag{11}$$

where ψ is the nominal bit rate provided by a Bluetooth network, which is 1 Mbps. Substituting Equation (8) by Equation (11), the bucket size is obtained as

$$b = [(1 - PER(X)) \times \varepsilon \times \psi \times k^{\alpha}]^{1/(1+\alpha)} \tag{12}$$

From Equation (11) and Equation (12), a new traffic-descriptor of the form ($R(X)$, b) is now produced. This traffic-descriptor not only takes into account the properties of the source traffic (i.e. the self-similarity and packet efficiency), but also the quality of the forwarding link (i.e. BER). With these deterministic set of information, network resources are reserved and allocated to the requesting application during a routing decision based on true scenario of the network environment. In this way, much more efficient and reliable QoS routing decisions could be made at each of the router node in the scatternet topology. Very similar to the (r, b) attainment discussed previously, the ($R(X)$, b) traffic-descriptor could also be interpreted as providing only an optimal solution to the routing function, but with more accuracy due to additional parameters being added into it.

3.3 The source traffic

There are three types of traffic normally used by researchers in the study of traffic engineering: on-line experimental traffic, generated traffic, or video traces. In this work, video traces are chosen for the reason that they are readily available for on-line simulation, by which the frame size is directly segmented into packet counts. More importantly, the traffic have been identified by (Beran, 1994) to contain MPEG encoded data, which they are bursty and proved to associate with self-similar property. The other traffic types however, may require some forms of conversion before packet counts is produced and hence, introduced routing and transmission delays.

Specifically, the bursty traffic source is to be generated from *Jurassic Park* and *Soccer* video traces. These two traces, as well as many other traces, can be obtained publicly from website of http://www-tkn.ee.tuberlin.de/research/trace.trace.html. Each trace contained a set of frame numbers, and each frame number has its own frame size (in byte). Both traces have the same frame number of 89,998 but each frame has different byte length. Therefore, there is always a chance for the two traces to be different, particularly with respect to the number of packets that they may produce when the SAR segmentation algorithm is applied on each of the frame.

3.4 The simulation

The performance of the proposed traffic-descriptor needs to be measured to prove its ability in providing support to efficient and reliable QoS routing decision in Bluetooth ad hoc network. For this purpose, a Matlab simulation is developed to simulate the decision-making process made by a router node to select only one forwarding link from a set of available links, which is the optimal link. A complete route consists of a number of links, where each link is connecting two adjacent nodes, and finally a route is connecting a sender node at one end to a receiver node at the other end. Hence, routing information stored at each router node will be collected and summarized to represent the overall routing decision over a complete route. This complete route shall reflect an optimal route.

Figure 4 depicted a simulation area of 100m x 100m where 300 nodes were randomly deployed within the area. At any point of time, a route will be established between a sender node and a receiver node over the scatternet topology. However, the success for link and route creation will be very much dependent on node density and class of transmits power of the node. The higher node density and the longer coverage radius resulted in a better chance for successful link and route creation. In this simulation, each node is assumed to use 10 meters transmit power, and for simplicity, a route is developed from left to right with fix lower left position is the sender node position and random right position is the receiver node position.

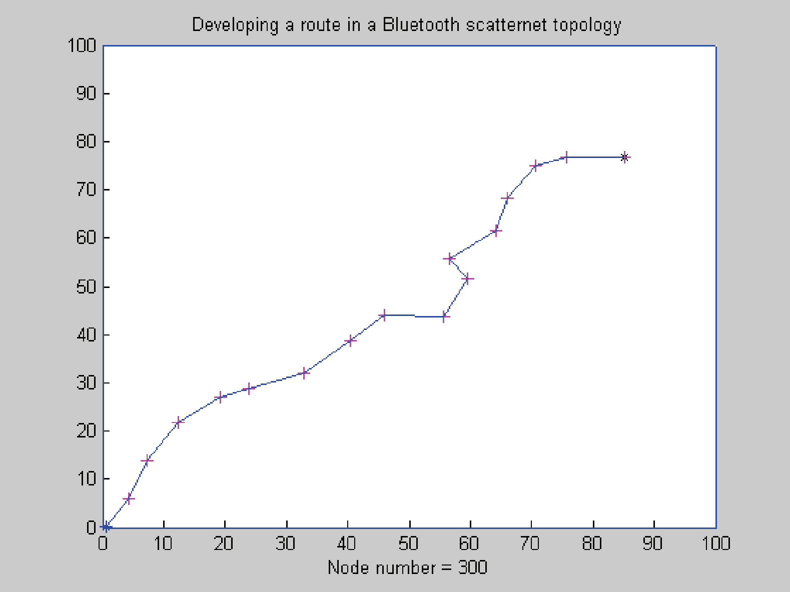


Fig. 4. Screenshot for a route creation connecting a sender and a receiver in a scatternet

The simulation will capture the two important features of the bursty traffic: burstiness level α and degree of self-similarity H. As elaborated by (Leland et al., 1994), the relationship between them for Pareto distribution is expressed as $H = (3 - \alpha)/2$. The H value will be obtained from QQ-plot method for the series of packets that is received by the router node. The QQ-plot is a statistical method used to show the level of self-similarity in a burst, and was well explained by (Dinh et al., 1998). Having the H value, α value is then computed.

Video traces of *Jurassic Park* and *Soccer* are the source of the bursty traffic. Simulation is repeated for a set of different frame ranges: say, starts at 1 to 5,000 as the smallest range size and ends at 1 to 89,998 frame number as the largest range size. Hence, the frame size is getting bigger and as well the time taken to complete the simulation is getting longer for each simulation run. This may produce different H values with better accuracy. The collected H values from each frame range are then averaged out and use to compute the α value for that frame range and for each of the video trace.

Figure 5 illustrates the implementation of *Best-Fit* algorithm of the SAR protocol as proposed by (Das et al., 2001). The protocol is used to segment the received bursty stream into smaller DHx or DMx packet sizes. This algorithm is executed at each router node, and total number of packets types of DHx and DMx were produced. However, the *Best-Fit* algorithm has the tendency to produce packets with higher capacity (such as DH5 and DH3) than to produce packets with lower capacity. For this reason, the simulation only takes DH5 packet type, since it offers higher capacity as compared to the other packet types. This simplicity is expected not to affect the performance of the proposed traffic-descriptor if lower capacity packet types are used.

```
BEGIN
  if frame_size >= 339
    frame_size/339
    remainder_frame_size = mod(frame_size/339)
  else
    if 183 <= remainder_frame_size < 339
      remainder_frame_size/183
      remainder_frame_size = mod(frame_size/183)
    else
      if 27 <= remainder_frame_size <183
        remainder_frame_size/27
        remainder_frame_size = mod(frame_size/27)
      else
        remainder_frame_size = mod(frame_size/27)
      end
    end
  end
END
```

Fig. 5. The SAR Best-Fit algorithm

4. Results and analysis

Based on the system model described in Section 3.1 and the mathematical model in Section 3.2, Table 2 is presented to show the simulation results for the pair of traffic-descriptor of

(R(X), b) when DH5 packet is used in making routing decision at a router node. The α value is calculated from H = (3 - α)/2, by which H value can be obtained from the QQ-plot method for every frame range of *Jurassic Park* and *Soccer* video traces.

Five observations could be made from Table 2 and the resultant graph of Figure 6. These observations reflecting the ability of the proposed traffic-descriptor for providing support to QoS routing function in the Bluetooth ad hoc network. They are explained as follows.

Frame range	α		**(R(X), b)**	
	Jurassic	**Soccer**	**Jurassic**	**Soccer**
1- 5,000	0.910	0.911	719374, 30870	719379, 30887
1- 10,000	0.918	0.918	718976, 30384	718981, 30395
1-15,000	0.926	0.927	718931, 30124	718928, 30134
1-20,000	0.937	0.935	719381, 29849	719389, 29850
1-25,000	0.952	0.953	718714, 29051	718720, 29048
1-30,000	0.967	0.968	719116, 28323	719121, 28332
1-35,000	0.983	0.983	719177, 27552	719170, 27565
1-40,000	0.994	0.994	719356, 27047	719358, 27053
1-45,000	1.001	1.002	718025, 26721	718029, 26717
1-50,000	1.008	1.009	719300, 26453	719296, 26459
1-55,000	1.012	1.012	719196, 26267	719200, 26273
1-60,000	1.019	1.018	715380, 25927	715384, 25930
1-65,000	1.024	1.025	718904, 25778	718900, 25781
1-70,000	1.028	1.029	714636, 25510	714643, 25507
1-75,000	1.032	1.031	718406, 25450	718410, 25459
1-80,000	1.034	1.035	719346, 25349	719349, 25354
1-85,000	1.037	1.037	715426, 25189	715433, 25196
1-89,998	1.038	1.039	719136, 25205	719138, 25200

Table 2. The effect of burst on (R(X), b) for DH5 packet transmission

First, traffic-descriptors of (R(X), b) for *Jurassic Park* and *Soccer* have been produced. This pair values provides the requesting application with information, which can be communicated to the Resource Manager for reservation and allocation of resources. The pair is a parsimonious traffic-descriptor since it contained only least number of parameters but has the ability to describe the required resources accurately for QoS routing decision in the Bluetooth ad hoc network. However, depending on the frame range and the number of packets they produced after segmentation, the resulting values of (R(X), b) may differ from each other.

Second, as can be observed from Table 2, while R(X) values have changed randomly and not following a certain order, b values reduced linearly as α values go higher. This is graphically presented in Figure 6, where both video traces have shown similar graph pattern.

Importantly, the $R(X)$ values are practical values that fluctuate within the acceptable range and come close to the bit rate of 723,200 bps for DH5 packet.

Third, the accuracy of the traffic-descriptor ($R(X)$, b) can be measured from the values of effective bit rate $R(X)$. The mean value for $R(X)$ is 718,380 bps for both *Jurassic Park* and *Soccer*. When this is compared against the maximum allowable bit rate of 723,200 bps for DH5 packet as specified in the (Bluetooth Specification v1.0B, 1999), the mean value is clearly within an acceptable range. The difference between them is only 4,820 bps, which is equivalent to 0.66%. In particular, after taking into account the channel quality at the time when forwarding decision was made at a router node, this $R(X)$ value can be considered as a realistic value suitable for practical implementation.

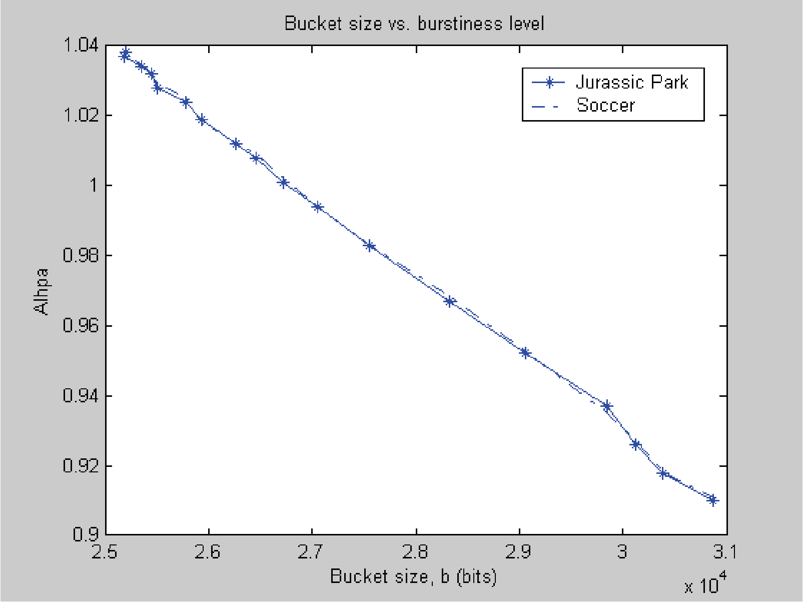


Fig. 6. Burstiness level α versus bucket size b for *Jurassic Park* and *Soccer* video traces

Forth, when α value is deviating away from 1 but approaching to 2 (which implies a less bursty traffic), the bucket size gets smaller. It means that the lower variability in the source traffic, the smaller required bucket size in handling the burst. Subsequently, the smaller bucket size the better routing performance, in term of its transmission delay from the queue. Therefore, $\alpha \to 2$ (or equivalently $H \to 0.5$) is very much needed. In this way, traffic with lower burst level could be produced, by which the performance of a QoS routing control scheme could be more effective and reliable. This is critically needed when limited link (or only a single link) and limited resources of Bluetooth network are available, but to provide a level of service guarantee to the requesting applications.

Fifth, for both video traces, bucket size can be used to control the burstiness level of the input traffic. It can be observed from the graph that bigger bucket size would imply lower variability in the input traffic, and vice-versa. However, careful must be taken to allow only suitable bucket size to operate, i.e. too big bucket size may lead to significantly higher packet lost probability, thus may lead to system collapse. Therefore, it is important to balance between the bucket size to use and the allowable burstiness level for efficient and reliable QoS routing decision at a router node.

5. Conclusions and future work

A traffic-descriptor (r, b) of which contained non-Token-Bucket parameters has been successfully developed. In particular, the developed traffic-descriptor has parsimoniously described the application requirement to the Resource Manager for the required resources. The model has been parameterized with only least number of parameters, but it has the capability to describe very well the QoS routing requirement. Additionally, to achieve better level of service guarantee in routing, ($R(X)$, b) traffic-descriptor has been developed as well. It has dimensioned the parameters of the input traffic and the channel quality using both standard Token-Bucket and non-Token-Bucket parameters for much more accurate resource reservation and allocation. Hence, the QoS routing decisions at a router node are then made to be more efficient and reliable. However, this traffic-descriptor is only applicable to Bluetooth ad hoc network setting. This is because the traffic-descriptor has characterized the source traffic and the channel quality by mean of a mathematical model that is dependent on Bluetooth network setting.

In the case of Bluetooth's DH5 packet transmission, it was found that the traffic-descriptor has accurately described the required effective bit rate $R(X)$ to only a difference of 0.66% from its maximum bit rate, whilst QoS guarantee is granted to user application. Therefore, the developed mathematical model is assumed to represent the system correctly and to work accordingly. The final result shall be exhibited through the optimal usage of the network resources. This is critically needed as to achieve efficient and reliable routing decision-making processes at the router nodes, since Bluetooth network is having only a handful of resources.

Therefore, the contribution of this work was in the development of a parsimonious traffic-descriptor that has incorporated the non-Token-Bucket parameters, in addition to basic Token-Bucket parameters, through the formulation of a mathematical model that is suitable only for use in Bluetooth ad hoc network. Hence, the identified future work is to develop a general-purpose traffic-descriptor, which can be used to dimension any other non-Token-Bucket parameters for use in any types of wireless networks.

6. References

A. Das, A. Ghose, A. Razdan, H. Saran & R. Shorey, (2001). Enhancing performance of asynchronous data traffic over the Bluetooth wireless ad hoc network. *Proceedings of the 20th Annual Joint Conference of IEEE Computer & Communications Society (INFOCOM 2001)*, pp. 591-600.

Bluetooth Specifications v1.0B, (1999). *Bluetooth SIG*. Available at: http://www.bluetooth.com, [Accessed on 2 August 2010].

F.Y. Li, (2002). Local and global QoS-aware Token Bucket parameters determination for traffic conditioning in 3rd generation wireless networks. *Proceedings of the European Wireless'02*, pp. 362-368.

G. Procissi, M. Gerla, J. Kim, S.S. Lee & M.Y. Sanadidi, (2001). On long range dependence and Token Buckets. *Proceedings of the SPECTS'01*.

IEEE 802.15 Specifications. Available at http://standards.ieee.org/getieee802/802.15.html, [Accessed on 2 August 2010].

J. Beran, (1994). *Statistics for Long-Memory Processes*, Chapman and Hall/CRC, 1st edition, New York.

J. Glasmann, M. Czermin & A. Riedl, (2000). Estimation of Token Bucket parameters for videoconferencing systems in corporate networks. *Proceedings of the International Conference on Software, Telecommunications and Computer Networks*.

J. Haartsen, W. Allen, J. Inouye, O.J. Joeressen & M. Naghshineh, (1998). Bluetooth: vision, goals and architecture. *ACM Mobile Computing and Communications Review*, Vol. 1, No. 2, pp. 1-8.

J. Kim, Y. Lim, Y. Kim & J.S. Ma, (2001). An adaptive segmentation scheme for the Bluetooth-based wireless channel. *Proceedings of the IEEE IC3N '01*, pp. 440-445.

K. Park, G. Kim & M. Crovella, (1996). On the relationship between file sizes, transport protocols, and self-similar network traffic. *Proceedings of the IEEE International Conference Network Protocols*, pp. 171-180.

L.J. Chen, R. Kapoor, M.Y. Sanadidi & M. Gerla, (2004). Enhancing Bluetooth TCP throughput via link layer packet adaptation. *Proceedings of the IEEE International Conference on Communications'04*.

M. Taqqu, W. Willinger & R. Sherman, (1997). Proof of a fundamental result in self-similar traffic modeling. *ACM/SIGCOMM Computer Communications Review*, Vol. 27, pp. 5-23.

M.D. Norashidah & F. Norsheila, (2007). Fuzzy logic Token Bucket bandwidth predictor for assured forwarding traffic in a DiffServ-aware MPLS Internet. *Proceedings of the Asia International Conference on Modelling & Simulation (AMS'07).*

M.E. Crovella & A. Bestavros, (1999). Self-similarity in world wide web traffic: evidence and possible causes. *IEEE/ACM Transactions on Networking*, Vol. 5, No. 6, pp. 835-846.

M.E. Crovella & L. Lipsky, (1997). Long-lasting transient conditions in simulations with heavy-tailed workloads. *Proceedings of the Winter Simulation Conference*, pp. 1005-1012.

R. Handle, M. Anber & S. Schroder, (1996). *ATM Networks Concepts, Protocols and Applications,* Addison-Wesley, New York.

R.G. Garroppo, S. Giordano, S. Niccolini & F. Russo, (2001). A simulation analysis of aggregation strategies in WF^{2}Q+ schedulers network, *IP Telephony'01*.

S. Fernandes, C. Kamienski & D. Sadok, (2003). Accurate and fast replication on the generation of fractal network traffic using alternative probability models. *Proceedings of the SPIE 5244,* pp. 154-163.

S. Valaee & J-C. Gregoire, (2005). An estimator of regulator parameters in a stochastic setting. *IEEE/ACM Transaction on Networking*, Vol. 13, No. 6, pp. 1376-1389.

T.D. Dinh, S. Molnar & A. Vidacs, (1998). Investigation of fractal properties in data traffic. *Journal of Communications*, Vol. XLIX, pp. 12-18.

V. Paxson, (1994). Empirically derived analytic models of wide area TCP connections. *IEEE/ACM Transactions on Networking*, Vol. 2, No. 4, pp. 316-336.

W.E, Leland, M.S. Taqqu, W. Willinger & D.V. Wilson, (1994). On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking,* Vol. 2, No. 1, pp. 1-15.

X. Yang, (2000). Designing traffic profiles for bursty Internet traffic. *Proceedings of the IEEE Global Internet*, pp. 2149-2154.

Z. Hadzi-Velkov & L. Garrilovska, (1999). Performance of the IEEE802.11 wireless LANs and influence of hidden terminals. *Proceedings of the Telsiks'99*, pp. 102-105.