UNIT 1

Short Answers

1. Explain about the term MANET?

A Mobile Ad hoc NET work(MANET)is one that comes together as needed, not necessarily with any support from the existing infrastructure any other kind of fixed stations. An ad hoc network as an autonomous system of mobile hosts (also se ving as oute s) connected by wireless links, the union of which forms a communication network modeled in the form of an arbitrary graph. An Adhoc Network is a Multi hop peer-to-peer network.

2. Explain the Characterstics of MANETs

1. Distributed operation : There is no background network for the central control of the network operations, the control of the network is distributed among the nodes. The nodes involved in a MANET should cooperate with each other and communicate among themselves and each node acts as a relay as needed, to implement specific functions such as routing and security.

2. Multi hop routing : When a node tries to send information to other nodes which is out of its communication range, the packet should be forwarded via one or more intermediate nodes.

3. Autonomous terminal: In MANET, each mobile node is an independent node, which could function as both a host and a router.

4. Dynamic topology: Nodes are free to move arbitrarily with different speeds; thus, the network topology may change randomly and at unpredictable time. The nodes in the MANET dynamically establish routing among themselves as they travel around, establishing their own network.

5. Light-weight terminals: In maximum cases, the nodes at MANET are mobile with less CPU capability, low power storage and small memory size.

6. Shared Physical Medium :The wireless communication medium is accessible to any entity with the appropriate equipment and adequate resources. Accordingly, access to the channel cannot be restricted.

3. Explain the challenges of the MANETs?

- **Limited bandwidth :** Wireless link continue to have significantly lower capacity than infrastructure networks. In addition, the realized throughput of wireless communication after accounting for the effect of multiple access, fading, noise, and interference conditions, etc., is often much less than a radio's maximum transmission rate.
- **Dynamic topology :** Dynamic topology membership may disturb the trust relationship among nodes. The trust may also be disturbed if some nodes are detected as compromised.
- **Routing Overhead :** In wireless adhoc networks, nodes often change their location within network. So, some stale routes are generated in the routing table which leads to unnecessary routing overhead.
- **Hidden terminal problem :**The hidden terminal problem refers to the collision of packets at a receiving node due to the simultaneous transmission of those nodes that are not within the direct transmission range of the sender, but are within the transmission range of the receiver.
- **Packet losses due to transmission errors:** Ad hoc wireless networks experiences a much higher packet loss due to factors such as increased collisions due to the presence of hidden terminals, presence of interference, uni-directional inks, frequent path breaks due to mobility of nodes.
- **Mobility-induced route changes:** The network topology in an ad hoc wireless network is highly dynamic due to the movement of nodes; hence an on-going session suffers frequent path breaks. This situation often leads to frequent route changes.
- **Battery constraints:** Devices used in these networks have restrictions on the power source in order to maintain portability, size and weight of the device.
- **Security threats :**The wireless mobile ad hoc nature of MANETs brings new security challenges to the network design. As the wireless medium is vulnerable to eavesdropping and ad hoc network functionality is established through node cooperation, mobile ad hoc networks are intrinsically exposed to numerous security attacks.

4. Briefly discuss the applications of MANETs?

- **Military battlefield:** Ad-Hoc networking would allow the military to take advantage of commonplace network technology to maintain an information network between the soldiers, vehicles, and military information head quarter.
- **Collaborative work:** For some business environments, the need for collaborative computing might be more important outside office environments than inside and
- where people do need to have outside meetings to cooperate and exchange information on a given project.
- **Local level:** Ad-Hoc networks can autonomously link an instant and temporary multimedia network using notebook computers to spread and share information among participants at a e.g. conference or classroom. Another appropriate

- local level application might be in home networks where devices can communicate directly to exchange information.
- **Personal area network and Bluetooth:** A personal area network is a short range, localized network where nodes are usually associated with a given person. Short-range MANET such as Bluetooth can simplify the intercommunication between various mobile devices such as a laptop, and a mobile phone.
- **Commercial Sector:** Ad hoc can be used in emergency/rescue operations for disaster relief efforts, e.g. in fire, flood, or earthquake. Emergency rescue operations must take place where non-existing or damaged communications infrastructure and rapid deployment of a communication network is needed.

5. What are the advantages of MANET?

The advantages of an Ad-Hoc network include the following:
•They provide access to information and services regardless of geographic position.
Independence from central network administration.
Self-configuring network, nodes are also act as routers. Less expensive as compared to wired network.
•Scalable—accommodates the addition of more nodes.
•Improved Flexiblibility.
•Robust due to decentralize administration.
•The network can be set up at any place and time.

6. Briefly discuss about the design challenges of routing protocols in adhoc networks?

- Network topology is highly dynamic due to movement of nodes. hence, an ongoing session suffers frequent path breaks.
- Disruption occurs due to the movement of either intermediate nodes in the path or end nodes .
- Wired network routing protocols cannot be used in adhoc wireless networks because the nodes are here are not stationary and the convergence is very slow in wired networks.
- Mobility of nodes results in frequently changing network topologies x Routing protocols for ad hoc wireless networks must be able to perform efficient and effective mobility management. x Abundant bandwidth is available in wired networks due to the advent of fiber optics and due to the exploitation of wavelength division multiplexing (WDM) technologies.
- In a wireless network, the radio band is limited, and hence the data rates it can offer are much less than what a wired network can offer.
- This requires that the routing protocols use the bandwidth optimally by keeping the overhead as low as possible.
- The limited bandwidth availability also imposes a constraint on routing protocols in maintaining the topological information
- The broadcast nature of the radio channel poses a unique challenge in ad hoc wireless networks.

- The wireless links have time-varying characteristics in terms of link capacity and link-error probability. 10. This requires that the adhoc wireless network routing protocol interact with the MAC layer to find alternate routes through better-quality links.
- Transmissions in ad hoc wireless networks result in collisions of data and control packets.
- Therefore, it is required that ad hoc wireless network routing protocols find paths with less congestion.

7. What is meant by collision detection problem?

- The hidden terminal problem refers to the collision of packets at a receiving node due to the simultaneous transmission of those nodes that are not within the direct transmission range of the receiver, but are within the transmission range of the receiver.

- Collision occurs when both nodes transmit packets at the same time without knowing about the transmission of each other.

8. Explain the characteristics of the routing protocols?

A routing protocol for ad hoc wireless networks should have the following characteristics:

- It must be fully distributed
- It must be adaptive to frequent topology changes caused by the mobility of nodes.
- Route computation and maintenance must involve a minimum number of nodes. Each node in the network must have quick access to routes, that is, minimum connection setup time is desired
- It must be localized, as global state maintenance involves a huge state propagation control overhead
- It must be loop-free and free from state routes.
- The number of packet collisions must be kept to a minimum by limiting the number of broadcasts made by each node. The transmissions should be reliable to reduce message loss and to prevent the occurrence of state routes.
- It must converge to optimal routes once the network topology becomes stable. The convergence must be quick.
- It must optimally use scarce resources such as bandwidth, computing power, memory, and battery power.
- Every node in the network should try to store information regarding the stable local topology only.
- Changes in remote parts of the network must not cause updates in the topology information maintained by the node.
- It should be able to provide a certain level of quality of service (QoS) as demanded by the applications, and should also offer support for time-sensitive traffic.

9. Explain the various ways to implement the protocols in Ad hoc networks?

The routing protocol for ad hoc wireless networks can be broadly classified into 4 categories based on

- Routing information update mechanism.

- Use of temporal information for routing
- Routing topology
- Utilization of specific resource

10. Explain the Different types of Routing protocols in Ad hoc Networks?

The protocols are classified into three categories

- Proactive or table-driven routing protocols:
- Reactive or on-demand routing protocols
- Hybrid routing protocols:

11. Explain about the characteristics of Proactive or table-driven routing protocols?

- Every node maintains the network topology information in the form of routing tables by periodically exchanging routing information.
- Routing information is generally flooded in the whole network.
- Whenever a node requires a path to a destination, it runs an appropriate path-finding algorithm on the topology information it maintains.

12. Explain about the characteristics of Reactive or on-demand routing protocols?

- Do not maintain the network topology information.
- Obtain the necessary path when it is required, by using a connection establishment process.

13. Explain about the characteristics of Hybrid routing protocols?

- Nodes within a certain distance from the node concerned, or within a particular geographical region, are said to be within the routing zone of the given node.
- For routing within this zone, a table-driven approach is used.
- For nodes that are located beyond this zone, an on-demand approach is used.

14. Explain different ways in location service and also forwarding strategies?

These require information about the physical position of the participating nodes in the network and their availability. Commonly, each node determines its own position through the use of GPS or some other type of positioning service. Position based routing is mainly focused on two issues:
1) A location service is used by the sender of a packet to determine the position of the destination and to include it in the packet's destination address
2) A forwarding strategy used to forward the packets.

A location service can be any one of the four:

a) Some for some
b) Some for all
c) All for all
d) All for some.

A forwarding strategy can be like;

a) Greedy forwarding
b) Restricted directional flooding and
c) Hierarchical routing.

15. List out the different protocols in Proactive routing protocols?

1. Destination sequenced distance vector Routing protocol (DSDV)
2. Wireless routing protocol (WRP)
3. Source-tree adaptive routing protocol (STAR) and
4. Cluster-head gateway switch routing protocol (CGRP)

Long Answer Questions:

1. Expalin about the DSDV Routing Protocol?

Destination sequenced distance vector routing protocol

- It is an enhanced version of the distributed Bellman -Ford algorithm where each node maintains a table that contains the shortest distance and the first node on the shortest path to every other node in the network.
- It incorporates table updates with increasing sequence number tags to prevent loops, to counter the count-to-infinity problem, and for faster convergence.
- As it is a table-driven routing protocol, routes to all destinations are readily available at every node at all times.
- The tables are exchanged between neighbors at regular intervals to keep an up -to-date view of the network topology.

The table updates are of two types:

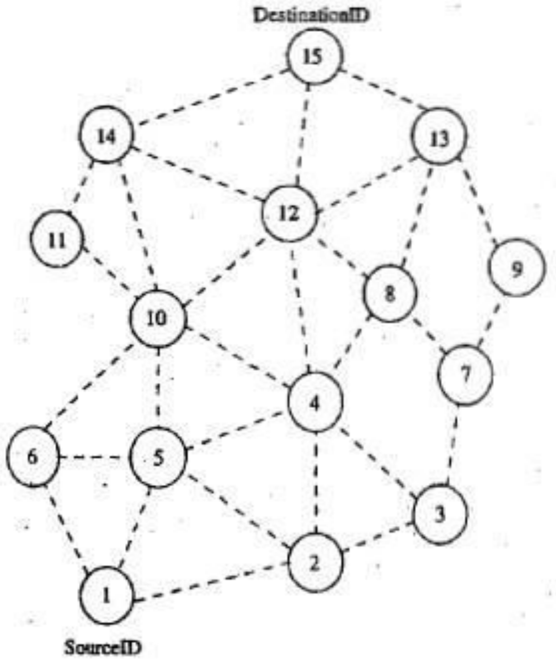
1. Takes a single network data packet unit (NDPU). These are used when a node does

not observe significant changes in the local topology.

2. Takes multiple NDPUs. It is done either when the local topology changes significantly or when an incremental update requires more than a single NDPU.

- Table updates are initiated by a destination with a new sequence number which is always greater than the previous one.

- Consider the example as shown in figure (a). Here node 1 is the source node and node 15 is the destination. As all the nodes maintain global topology information, the route is already available as shown in figure (b).
- Here the routing table node 1 indicates that the shortest route to the destination node is available through node 5 and the distance to it is 4 hops, as depicted in figure (b)
- The reconfiguration of a path used by an on-going data transfer session is handled by the protocol in the following way.
- The end node of the broken link initiates a table update message with the broken link's weight assigned to infinity (∞) and with a sequence number greater than the stored sequence number for that destination.
- Each node upon receiving an update with weight ∞, quickly disseminates it to its neighbors in order to propagate the broken-link information to the whole network.
- A node always assign an odd number to the link break update to differentiate it from the even sequence number generated by the destination.
- Figure 7.6 shows the case when node 11 moves from its current position.

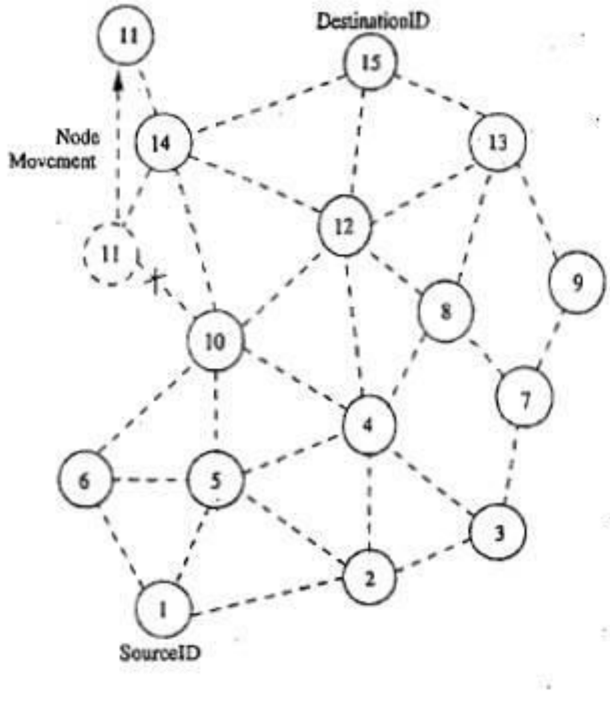


Dest	NextNode	Dist	SeqNo
2	2	1	22
3	2	2	26
4	5	2	32
5	5	1	134
6	6	1	144
7	2	3	162
8	5	3	170
9	2	4	186
10	6	2	142
11	6	3	176
12	5	3	190
13	5	4	198
14	6	3	214
15	5	4	256

(a) Topology graph of the network

(b) Routing table for Node 1

Figure 7.5. Route establishment in DSDV.



Routing Table for Node 1

Dest	NextNode	Dist	SeqNo
2	2	1	22
3	2	2	26
4	5	2	32
5	5.	1	134
6	6	1	144
7	2	3	162
8	5	3	170
9	2	4	186
10	6	2	142
11	5	4	180
12	5	3	190
13	5	4	198
14	6	3	214
15	5	4	256

Figure 7.6. Route maintenance in DSDV.

Advantages:

- Mechanism of incremental update with sequence number tags makes the existing wired network protocols adaptable to ad hoc wireless networks.
- The updates are propagated throughout the network in order to maintain an up-to-date view of the network topology at all nodes.
- The updates due to broken links lead to a heavy control overhead during high mobility.
- Even a small network with high mobility or a large network with low mobility can completely choke the available bandwidth**.**

Disadvantages:

- Suffers from excessive control overhead.
- In order to obtain information about a particular destination node, a node has to wait for a table update message initiated by the same destination node.
- This delay could result in state routing information at nodes.

2.Expalin about WRP?

Ans)Wireless Routing Protocol

- WRP is similar to DSDV, it inherits the properties of the distributed bellman-ford algorithm.
- To counter the count-to-infinity problem and to enable faster convergence, it employs a unique method of maintaining information regarding the shortest distance to every destination node in the network and penultimate hop node on the path to every destination node.
- 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 table that are maintained by a node are :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 the neighbor for a particular destination.): contains the up-to-date view of the network for all known destinations. It keeps the shortest distance, the predecessor/penultimate node, the successor node, and a flag indicating the status of the path. The path status may be a simplest (correct) path or a loop (error), or destination node not marked

(null).contains the cost of relaying messages through each link. The cost of broken link is ∞.it also contains the number of update periods passed since the last successful update was received from that link. contains an entry for every update message that is to be retransmitted and maintains a counter for each entry.

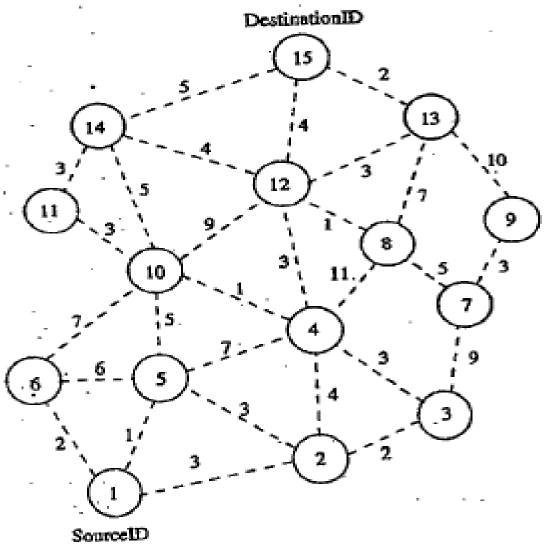
- After receiving the update message, a node not only updates the distance for transmitted neighbors but also checks the other neighbors' distance, hence convergence is much faster than DSDV.
- Consider the example shown in figure below, where the source of the route is node 1 and destination is node 15. As WRP proactively maintains the route to all destinations, the route to any destination node is readily available at the source node.
- From the routing table shown, the route from node 1 to node 15 has the next node as node 2. The predecessor node of 15 corresponding to this route is route 12.
- The predecessor information helps WRP to converge quickly during link breaks.
- When a node detects a link break, it sends an update message to its neighbors with the link cost of the broken link set to ∞. After receiving the update message; all affected nodes update their minimum distances to the corresponding nodes. The node that initiated the update message then finds an alternative route, if available
- from its DT. Figure 7.8 shows route maintenance in WRP.

Advantages:

- WRP has the same advantages as that of DSDV.
- It has faster convergence and involves fewer table updates.

Disadvantages:

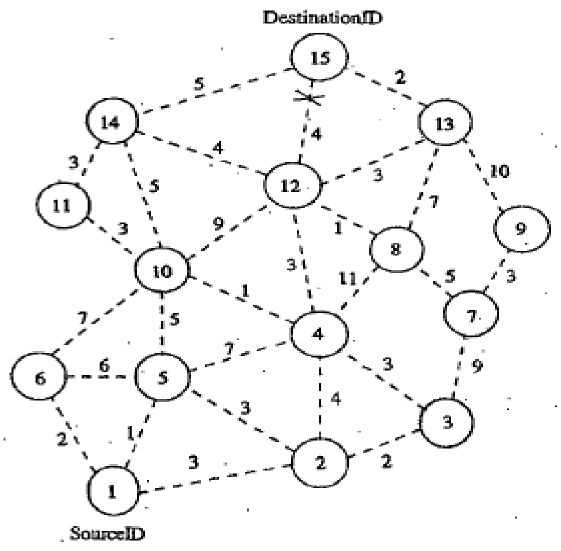
- The complexity of maintenance of multiple tables demands a larger memory and greater processing power from nodes in the adhoc wireless network.
- It is not suitable for highly dynamic and also for very large ad hoc wireless networks.



Routing Entry at Each Node
for DestinationID 15

Node	NextNode	Pred	Cost
15	15	15	0
14	15	14	5
13	15	13	2
12	15	12	4
11	14	14	8
10	4	12	8
9	13	13	12
8	12	12	5
7	8	12	10
6	10	12	15
5	10	12	13
4	12	12	10
3	4	12	7
2	4	12	11
1	2	12	14

Figure 7.7. Route establishment in WRP.



Routing Entry at Each Node
for DestinationID 15

Node	NextNode	Pred	Cost
15	15	15	0
14	15	14	5
13	15	13	2
12	15	13	5
11	14	14	8
10	4	13	9
9	13	13	12
8	12	13	6
7	8	13	11
6	10	13	16
5	10	13	14
4	12	13	8
3	4	13	11
2	4	13	12
1	2	13	15

Figure 7.8. Route maintenance in WRP.

3. **Explain about cluster head gateway switch routing protocol?**

- Uses a hierarchical network topology.
- CGSR organizes nodes into clusters, with coordination among the members of each cluster entrusted to a special node named *cluster-head.*
- This cluster-head is elected dynamically by employing a least cluster change (LCC) algorithm.
- According to this algorithm, a node ceases to be a cluster-head only if it comes under the range of another cluster-head, where the tie is broken either using the lowest ID or highest connectivity algorithm.
- Clustering provides a mechanism to allocate bandwidth, which is a limited resource, among different clusters, thereby improving reuse.
- A token-based scheduling is used within a cluster for sharing the bandwidth among the members of the cluster.
- CGRS assumes that all communication passes through the cluster-head. Communication between 2 clusters takes place through the common member nodes that are members of both the cluster are called *gateways.*
- A gateway is expected to be able to listen to multiple spreading codes that are currently in operation in the clusters in which the node exist as a member.
- A gateway conflict is said to occur when a cluster-head issues a token to a gateway over spreading code while the gateway is tuned to another code.
- Gateways that are capable of simultaneously communicating over two interfaces can avoid gateway conflicts.
- The performance of routing is influenced by token scheduling and code scheduling that is handled at cluster heads and gateways, respectively.
- Every member node maintains a routing table containing the destination cluster-head for every node in the network.
- In addition to the cluster member table, each node maintains a routing table which keeps the list of next-hop nodes for reaching every destination cluster.
- The cluster routing protocol is used here.
- Figure below shows the cluster head, cluster gateways, and normal cluster member nodes in an ad hoc wireless network.
- CGSR is a hierarchical routing scheme which enables partial coordination between nodes by electing cluster heads.

Advantages

- Better bandwidth utilization is possible.
- Easy to implement priority scheduling schemes with token scheduling and gateway code scheduling.

Disadvantages

- Increase in path length and instability in the system at high mobility when the rate of change of cluster-head is high.
- In order to avoid gateway conflicts, more resources are required.
- The power consumption at the cluster-head node is also a matter of concern.
- Lead to Frequent changes in the cluster-head, which may result in multiple path breaks.

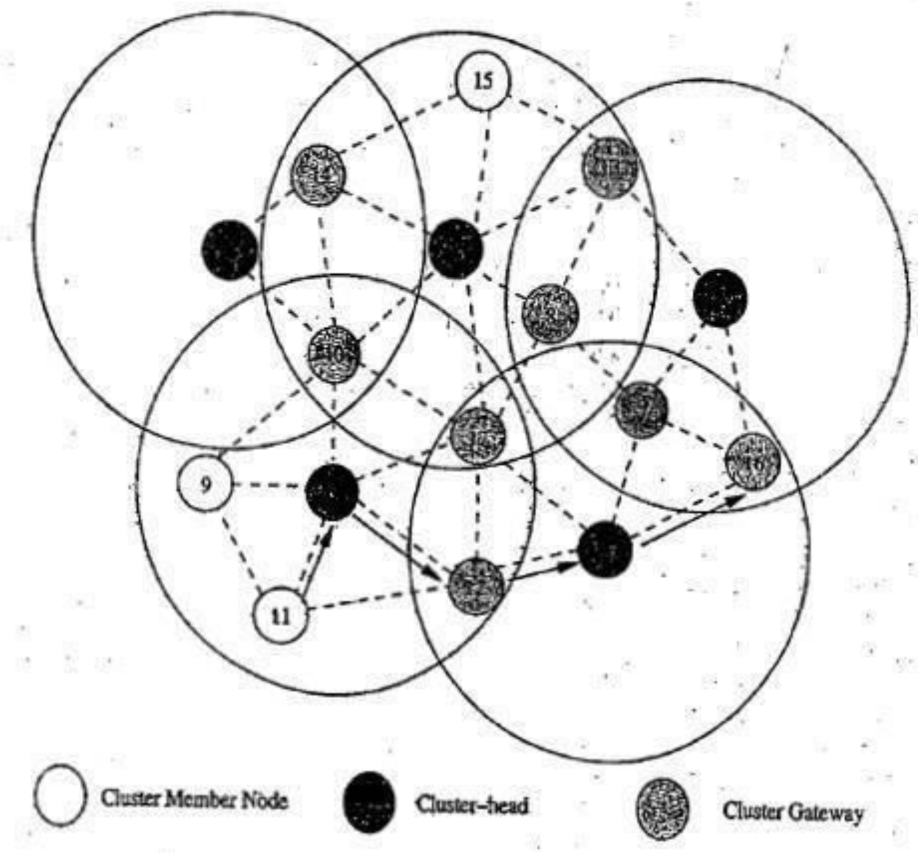


Figure 7.9. Route establishment in CGSR.

4. Expalin about STAR routing protocol?

- Key concept -least overhead routing approach (LORA)
- This protocol attempts to provide feasible paths that are not guaranteed to be optimal
- Involves much less control overhead

- In STAR protocol, every node broadcasts its source tree information
- The source tree of a node consists of the wireless links used by the node.
- Every node builds a partial graph of the topology.
- During initialization, a node sends an update message to its neighbors
- Each node will have a path to every destination node.
- The path would be sub-optimal.
- The data packet contains information about the path to be traversed in order to prevent the possibility of routing loop formation.
- In the presence of a reliable broadcast mechanism, STAR assumes implicit route maintenance.
- In addition to path breaks, the intermediate nodes are responsible for handling the routing loops
- The RouteRepair packet contains the complete source tree of node k and the traversed path of the packet.
- When an intermediate node receives a RouteRepair update message, it removes itself from the top of the route repair path and reliably sends it to the head of the route repair path.
- Very low communication overhead
- Reduces the average control overhead

5. Define the Reactive routing approach? Explain about Dynamic source routing protocol?

Reactive Routing Approach

They execute the path-finding process and exchange routing information only when a path is required by a node to communicate with a destination

Dynamic source routing protocol

Designed to restrict the bandwidth consumed by control packets in adhoc wireless networks by eliminating the periodic table update messages

- It is beacon-less and does not require periodic hello packet transmissions
- Basic approach to establish a route by flooding Route Request packets in the network.
- Destination node responds by sending a Route Reply packet back to the source
- Each Route Request carries a sequence number generated by the source node and the path it has traversed
- A node checks the sequence number on the packet before forwarding it.
- The packet is forwarded only if it is not a duplicate Route Request.
- The sequence number on the packet is used to prevent loop formations and to avoid multiple transmissions.

- Thus, all nodes except the destination forward a Route Request packet during the route construction phase.
- In figure 7.10, source node 1 initiates a Route Request packet to obtain a path for destination node 15
- This protocol uses a route cache that stores all possible information extracted from the source route contained in a data packet.
- During network partitions, the affected nodes initiate Route Request packets.
- DSR also allows piggy-backing of a data packet on the Route Request.
- As a part of optimizations, if the intermediate nodes are also allowed to originate Route Reply packets, then a source node may receive multiple replies from intermediate nodes.
- In fig 7.11, if the intermediate node 10 has a route to the destination via node 14, it also sends the Route Reply to the source node.
- The source node selects the latest and best route and uses that for sending data packets.
- Each data packet carries the complete path to its destination.
- If a link breaks, source node again initiates the route discovery process

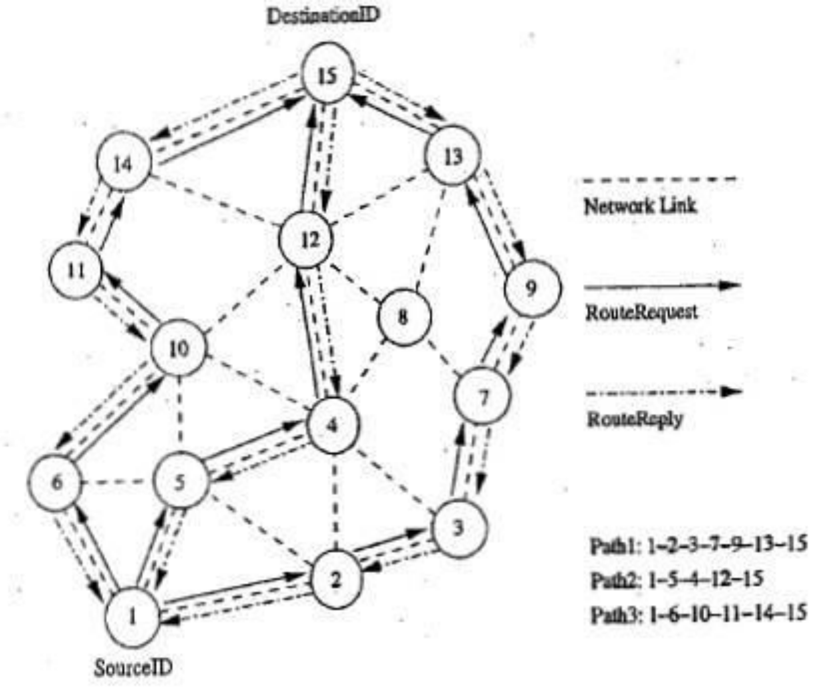


Figure 7.10. Route establishment in DSR.

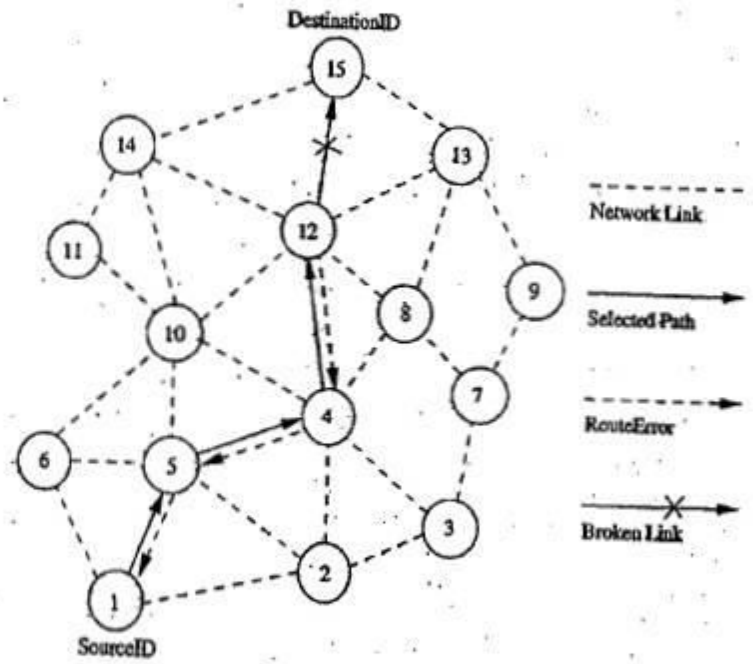


Figure 7.11. Route maintenance in DSR.

Advantages

- Uses a reactive approach which eliminates the need to periodically flood the network with table update messages.
- Route is established only when required.
- Reduce control overhead

Disadvantages

- Route maintenance mechanism does not locally repair a broken link
- Stale route cache information could result in inconsistencies during route construction phase Connection set up delay is higher
- Performance degrades rapidly with increasing mobility

- Routing overhead is more & directly proportional to path length

6. Explain about ad hoc on demand distance vector routing protocol?

- 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
- Source node and intermediate nodes store the next hop information corresponding to each flow for data packet transmission
- Uses DestSeqNum to determine an up-to-date path to the destination.
- A Route Request carries the source identifier, the destination identifier, the source sequence number, the destination sequence number, the broadcast identifier and the time to live field.
- DestSeqNum indicates the freshness of the route that is accepted by the source.
- When an intermediate node receives a Route Request, it either forwards it or prepares a Route Reply if it has a valid route to the destination.
- The validity of the intermediate node is determined by comparing the sequence numbers.
- If a Route Request is received multiple times, then duplicate copies are discarded.
- Every intermediate node enters the previous node address and its BcastID.
- A timer is used to delete this entry in case a RouteReply packet is not received.
- AODV does not repair a broken path locally
- When a link breaks, the end nodes are notified
- Source node re-establishes the route to the destination if required

Advantages

- Routes are established on demand and DestSeqNum are used to find latest route to the destination
- Connection setup delay is less

Disadvantages

- Intermediate nodes can lead to inconsistent routes if the source sequence number is very old.
- Multiple Route Reply packets to single Route Request packet can lead to heavy control overhead.
- Periodic beaconing leads to unnecessary bandwidth consumption.

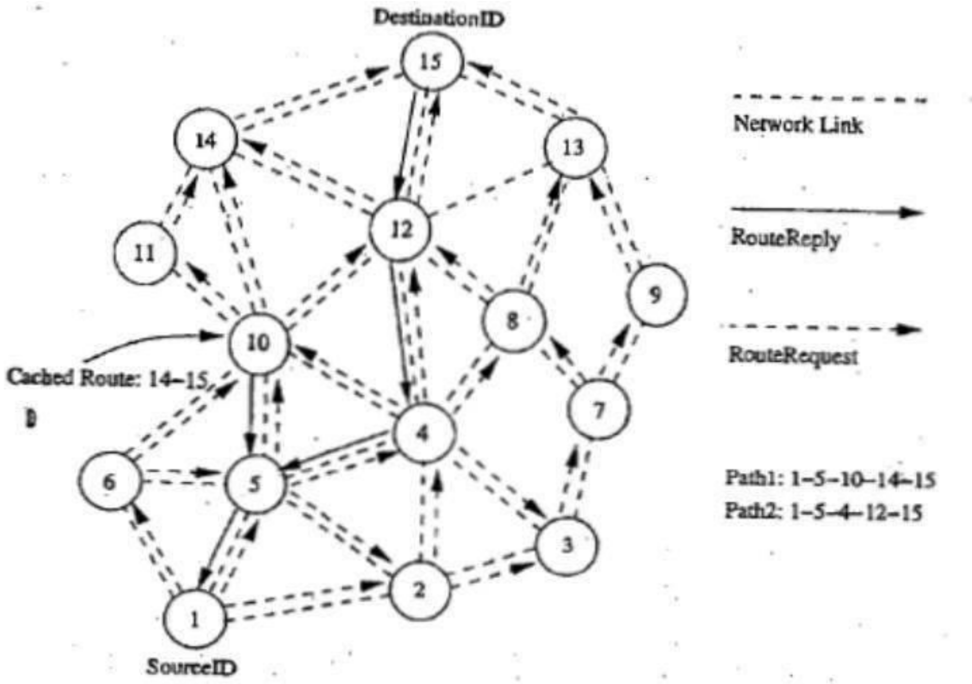


Figure 7.12. Route establishment in AODV.

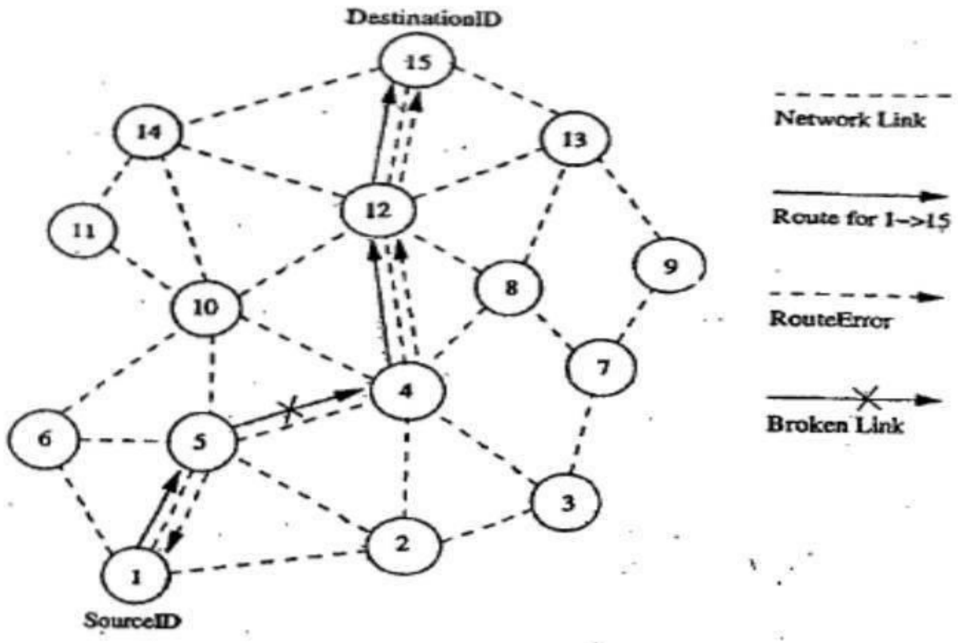


Figure 7.13. Route maintenance in AODV.

7. Explain about Link Reversal and TORA (Temporally ordered routing algorithm) reactive routing protocols?

- Source-initiated on-demand routing protocol.
- Uses a link reversal algorithm.
- Provides loop free multi path routes to the destination.
- Each node maintains its one-loop local topology information.
- Has capability to detect partitions.
- Unique property limiting the control packets to a small region during the reconfiguration process initiated by a path break
- TORA has 3 main functions: establishing, maintaining and erasing routes
- The route establishment function is performed only when a node requires a path to a destination but does not have any directed link.
- This process establishes a destination-oriented directed acyclic graph using a query/update mechanism.
- Once the path to the destination is obtained, it is considered to exist as long as the path is available,
- irrespective of the path length changes due to the re-configurations that may take place during the course of data transfer session
- If the node detects a partition, it originated a clear message, which erases the existing path information in that partition related to the destination.

Advantages

- Incur less control overhead
- Concurrent detection of partitions
- Subsequent deletion of routes

Disadvantages

- Temporary oscillations and transient loops
- Local reconfiguration of paths result in non-optimal routes

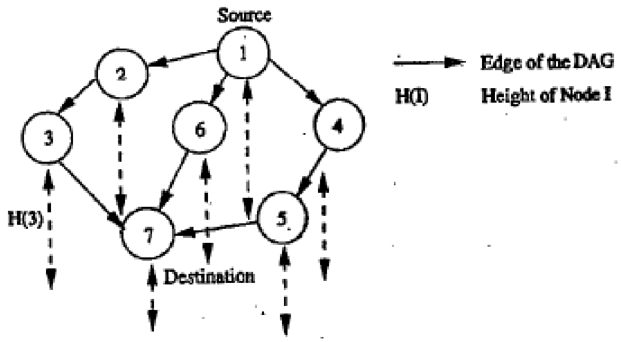
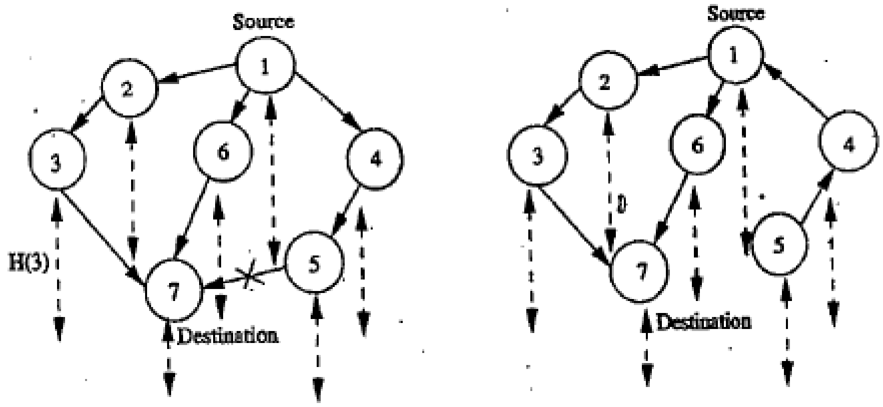


Figure 7.14. Illustration of temporal ordering in TORA.



Link break between Nodes 5 and 7

Nodes 4 and 5 reverse their links in order to update the path

Figure 7.15. Illustration of route maintenance in TORA.

8. Explain about LAR (Location aided routing)position based routing protocol?

- It utilizes the location information for improving the efficiency of routing by reducing the control overhead
- LAR assumes the availability of the global positioning system (GPS) for obtaining the geographical position information necessary for routing.
- LAR designates two geographical regions for selective forwarding of control packets, namely, Expected Zone and Request Zone.
- The Expected Zone is the region in which the destination node is expected to be present, given information regarding its location in the past and its mobility information.
- The is a geographical region within which the path-finding control packets are permitted to be propagated.
- This area is determined by the sender of a data transfer session.
- The control packets used for path-finding are forwarded by nodes which are present in the Request Zone and are discarded by nodes outside the zone.
- LAR uses flooding, but here flooding is restricted to a small geographical region.
- The nodes decide to forward or discard the control packets based on two algorithms, namely, LAR1 & LAR2

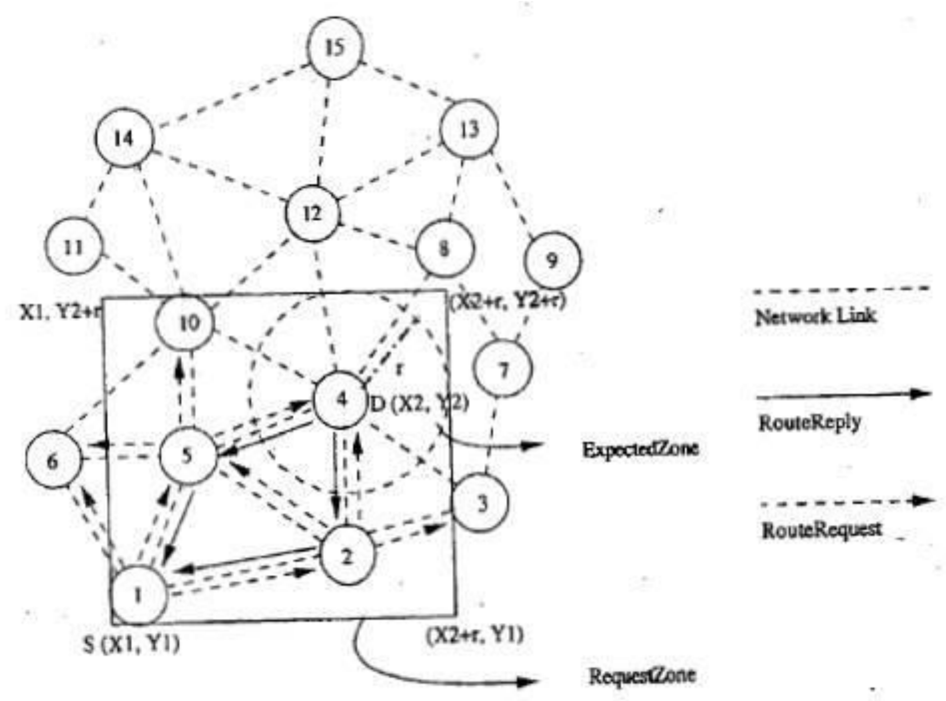


Figure 7.16. *RequestZone* and *ExpectedZone* in LAR1.

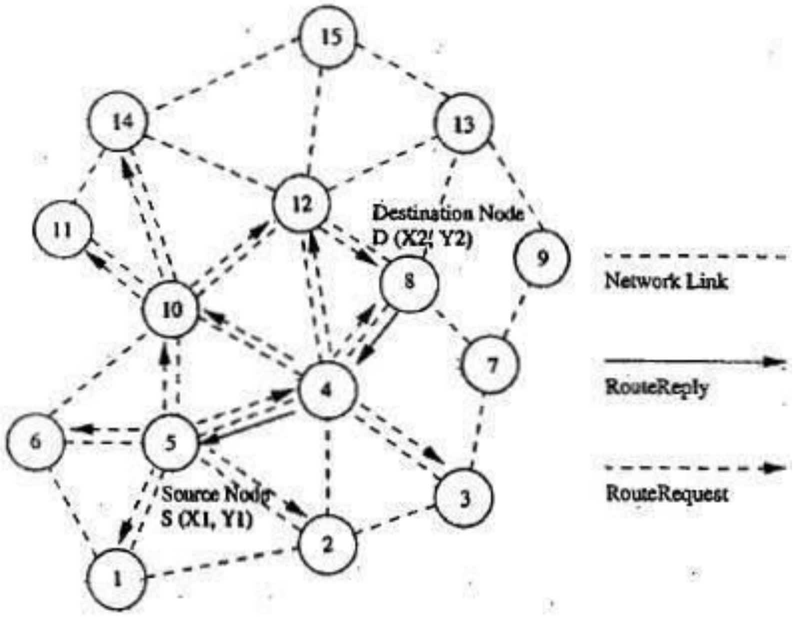


Figure 7.17. Route establishment in LAR2.

- In the LAR1 algorithm (fig 7.16), the source node explicitly specifies the Request Zone in the Route Request packet which is broadcast to its neighbors.
- These nodes verify their own geographical locations to check whether they belong to the Expected Zone.
- Finally, when the Route Request reaches the destination node, it originates a Route Reply that contains the current location and current time of the node.
- In LAR2 algorithm (fig 7.17), the source node includes the distance between itself and the destination node.
- When the intermediate node receives this Route Request packet, it computes the distance to the node D.
- A Route Request packet is forwarded only once and the distance between the forwarding node and D is updated in the Route Request packet for further relaying.
- In order to compensate for the location error, a larger Request Zone that can accommodate the amount of error that occurred is considered

Advantages:

- LAR reduces the control overhead by limiting the search area for finding a path.
- Efficient use of geographical position information.
- Reduced control overhead.
- Increased utilization of bandwidth

Disadvantages:

- Depends heavily on the availability of GPS infrastructure.
- Hence, cannot be used in situations where there is no access to such information

9. Explain about Associatively Based routing (ABR)?

- It is a distributed routing protocol that selects routes based on the stability of the wireless links.
- It is a beacon-based on-demand routing protocol.
- A link is classified as stable or unstable based on its temporal stability.
- The temporal stability is determined by counting the periodic beacons that a node receives from its neighbors.
- Each node maintains the count of its neighbor's beacons and classifies each link as stable or unstable.
- The link corresponding to a stable neighbor is termed as a stable link, while a link to an unstable neighbor is called an unstable link.
- A source node floods Route Request packets throughout the network if a route is not available in its router cache.
- All intermediate nodes forward the Route Request packet.
- Route Request packet carries the path it has traversed and the beacon count for the corresponding nodes in the path.
- When the first Route Request reaches the destination, the destination waits for a time period T to receive multiple Route Requests through different paths.
- If two paths have the same proportion of stable links, the shorter of them is selected.
- If more than one path is available, then a random path among them is selected as the path between source and destination.

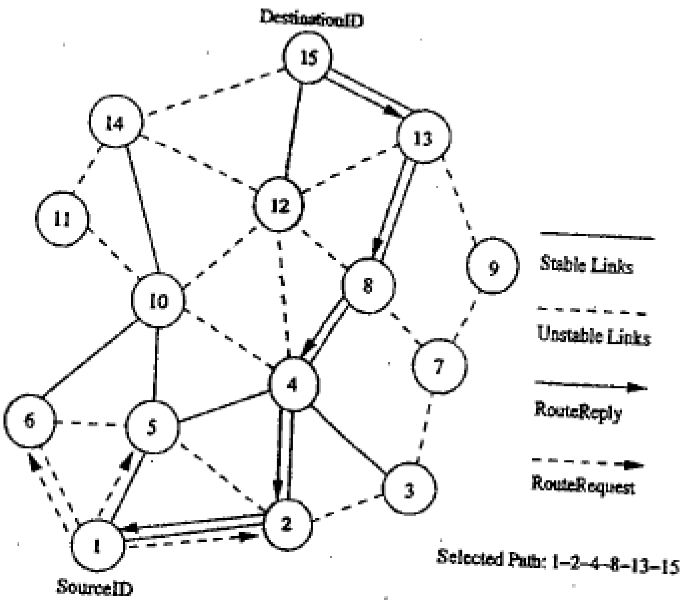


Figure 7.18. Route establishment in ABR.

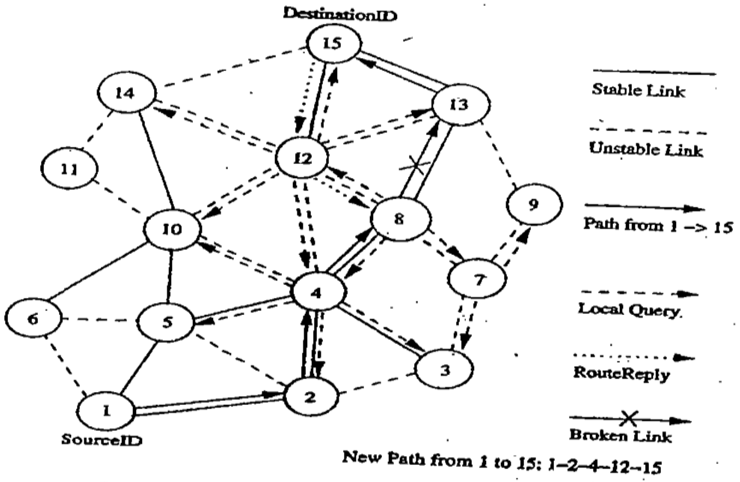


Figure 7.19. Route maintenance in ABR.

- In figure 7.18, source node initiates the Route Request to the flooded for finding a route to the destination node
- Solid lines represent stable links.
- Dotted lines represent unstable links.
- ABR uses stability information only during the route selection process at the destination node.
- If a link break occurs at an intermediate node, the node closer to the source, which detects the break, initiates a local route repair process.

- In this process, the node locally broadcasts a route repair packet, termed the local query (LQ) broadcast, with a limited time to live (TTL), as shown in figure 7.19.
- This way a broken link is bypassed locally without flooding a new Route Request packet in the whole network.

Advantages

- Stable routes have a higher preference compared to shorter routes
- They result in fewer path breaks which, in turn, reduces the extent of flooding due to reconfiguration of paths in the network

Disadvantages

- Chosen path may be longer than the shortest path between the source and destination because of the preference given to stable paths.
- Repetitive LQ broadcasts may result in high delays during route repairs

10 Explain about Signal stability routing protocol?

- Uses signal stability as the prime factor for finding stable routes.
- This protocol is beacon-based, in which signal strength of the beacon is measured for determining link stability.
- The signal strength is used to classify a link as stable or unstable.
- This protocol consists of two parts: forwarding protocol (FP) and dynamic routing protocol (DRP).
- These protocols use an extended radio interface that measures the signal strength from beacons.
- DRP maintains the routing table by interacting with the DRP processes on other hosts.
- FP performs the actual routing to forward a packet on its way to the destination.
- Every node maintains a table that contains the beacon count and the signal strength of each of its neighbors.
- If a node receives strong beacons, then link is classified as strong/stable link.
- The link is otherwise classified as weak/unstable link.
- Each node maintains a table called the signal stability table (SST) which is based on the signal strengths of its neighbors' beacons.
- This table is used by the nodes in the path to the destination to forward the incoming Route Request over strong links for finding the most stable end-to-end path.

- A source node which does not have a route to the destination floods the network with Route Request packets.
- SSA protocol process a Route Request only if it is received over a strong link.
- A Route Request received through a weak link is dropped without being processed.
- The destination selects the first Route Request packet received over strong links.
- The destination initiates a Route Reply packet to notify the selected route to the source.
- In figure 7.20, source node broadcasts a Route Request for finding the route to the destination node.
- Solid lines represent the stable links.
- Dotted lines represent the weak links.
- SSA restricts intermediate nodes from forwarding a Route Request packet if the packet has been received over a weak link.
- When a link breaks, the end nodes of the broken link notify the corresponding end nodes of the path.
- A source node, after receiving a route break notification packet, rebroadcasts the Route Request to find another stable path to the destination.
- Stale entries are removed only if data packets that use the stale route information fail to reach the next node.
- If no strong path is available when a link gets broken, then the new route is established by considering weak links also.
- This is done when multiple Route Request attempts fail to obtain a path to the destination using only the stable links.

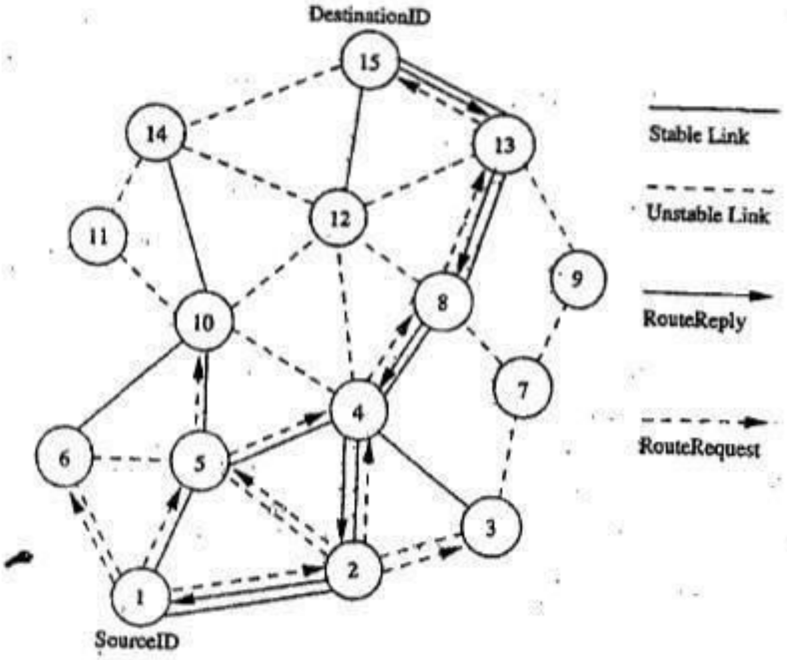


Figure 7.20. Route establishment in SSA.

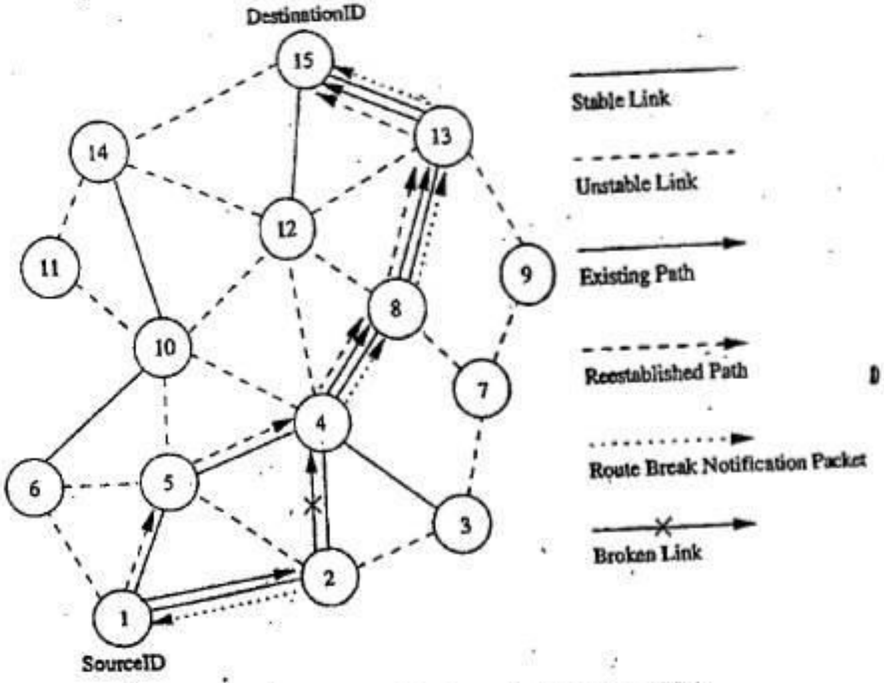


Figure 7.21. Route maintenance in SSA.

Advantages

- Finds more stable routes when compared to the shortest path route selection protocols.
- Accommodates temporal stability by using beacon counts to classify a link as stable or weak

Disadvantages

- It puts a strong RouteRequest forwarding condition which results in RouteRequest failures.
- Multiple flooding of RouteRequest packets consumes significant amount of bandwidth.
- Increases the path setup time.
- Strong link criterion increases the path length

11. Explain about CEDAR routing Protocol?

Core extraction Distributed Adhoc routing

- each node maintains the network topology information up to m nodes.
- It is based on extracting core nodes (also called as Dominator nodes) in the network.
- Core nodes together approximate the minimum Dominating Set (DS).
- A DS of a graph is defined as a set of nodes such that every node in the graph is either present in the DS or is a neighbor of some node present in the DS.
- There exists at least one core node within every three hops.
- The nodes that choose a core node as their dominating node are called core member nodes of the core node concerned.
- The path between two core nodes is termed as virtual link.
- CEDAR employs a distributed Algorithm to select core nodes.
- The selection of core nodes represents the core extraction phase.
- CEDAR uses the core broadcast mechanism to transmit any packet throughout the network in the unicast mode, involving as minimum number of nodes as possible.
- Route Establishment in CEDAR: It is carried out in two phase.
- The first phase finds a core path from source to destination. The core path is defined as the path from dominator of the source node (source core) to the dominator of the destination node (destination core).
- In the second phase, a QoS feasible path is found over the core path.
- A node initiates a RouteRequest if the destination is not in the local topology table of its core node; otherwise the path is immediately established.

- For establishing a route, the source core initiates a core broadcast in which the RouteRequest is sent to all neighboring core nodes which inturn forwards it.
- A core node which has the destination node as its core member replies to the source core.
- Once the core path is established, a path with the requested QoS support is then chose

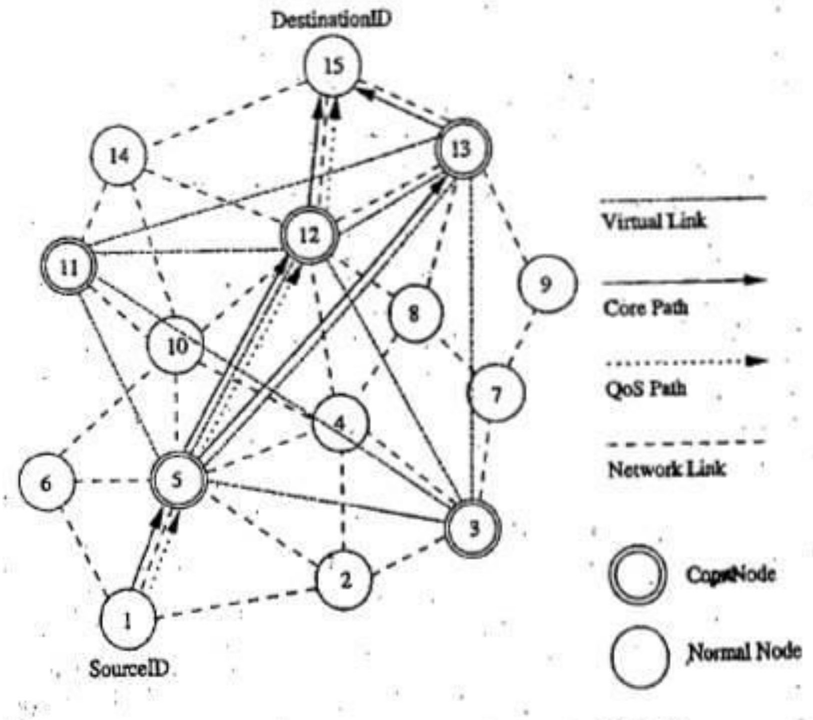


Figure 7.24. Route establishment in CEDAR.

- Route Maintenance in CEDAR: attempts to repair a broken route locally when a path break occurs.
- A node after which the break occurred:

1. Sends a notification of failure.
2. Begins to find a new path from it to the destination.
3. Rejects every received packet till the moment it finds the new path to the destination.

- Meanwhile, as the source receives the notification message:

1. It stops to transmit.
2. Tries to find a new route to the destination.
3. If the new route is found by either of these two nodes, a new path from the source to the
destination is established.

Advantages

- Performs both routing and QoS path computation very efficiently with the help of core nodes.
- Utilization of core nodes reduces traffic overhead.
- Core broadcasts provide a reliable mechanism for establishing paths with QoS support.

Disadvantages

- Since route establishment is carried out at core nodes, the movement of core nodes adversely affects the performance of the protocol.
- Core node update information causes control overhead.

12.Expalin about Zone Routing Protocol?

- Effectively combines the best features of both Proactive and Reactive routing protocols.
- It use a Proactive routing scheme within a limited zone in the r-hop neighborhood of every node.
- Use a Reactive routing scheme for nodes beyond this.
- An Intra-Zone Routing Protocol (IARP) is used in the zone where a particular node employs proactive routing.
- The Reactive routing protocol used beyond this zone is referred to as Inter-Zone Routing Protocol (IERP).
- The routing zone of a given node is a subset of the network, within which all nodes are reachable within less than or equal to.

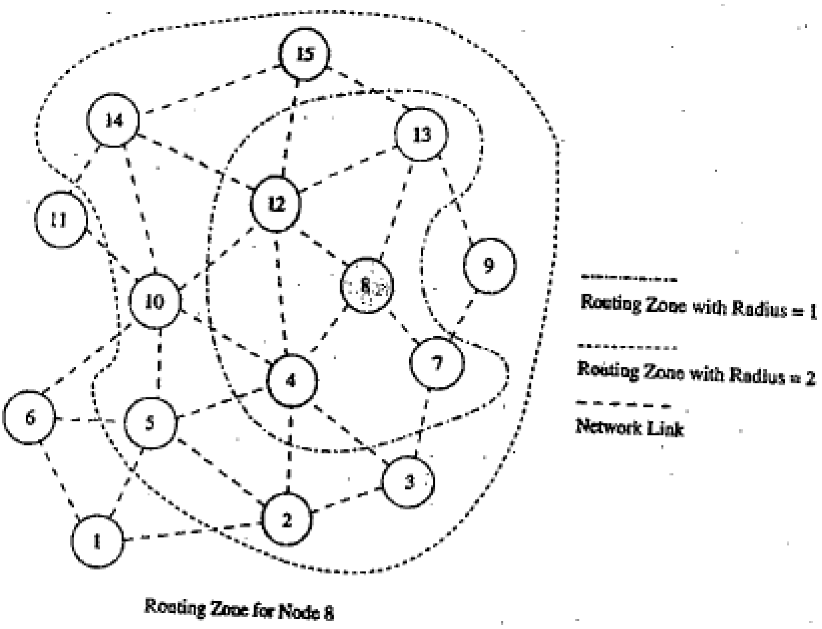


Figure 7.26. Routing zone for node 8 in ZRP.

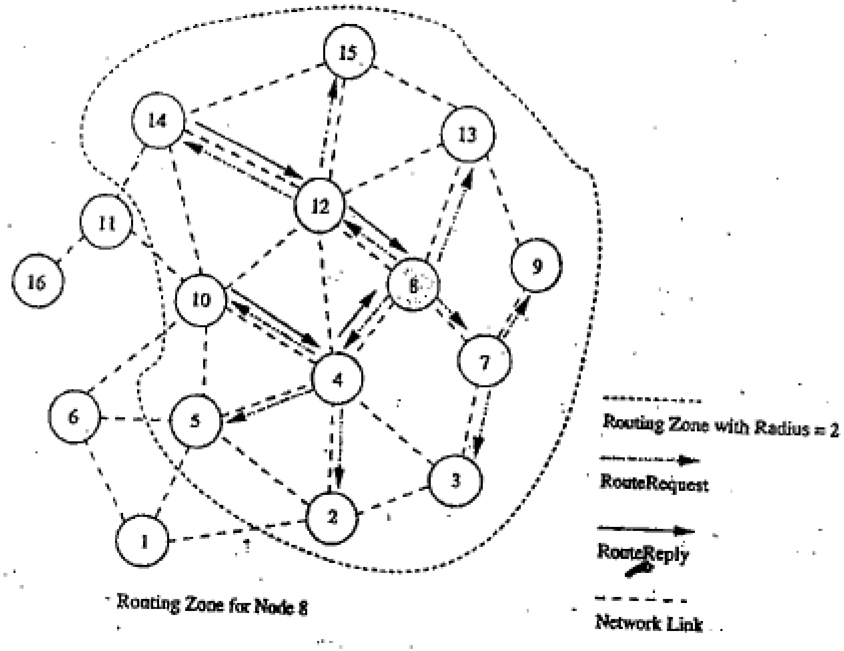
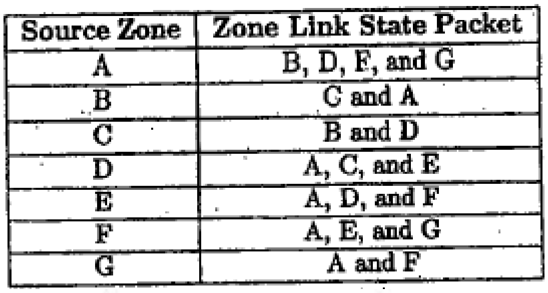


Figure 7.27. Path finding between node 8 and node 16.

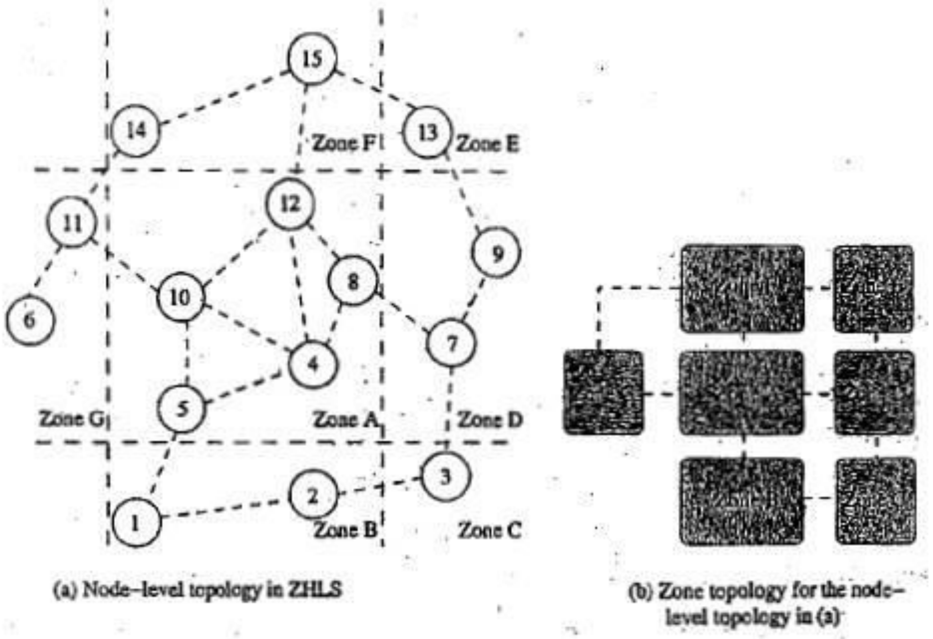
- When a node s (node 8 in the fig 7.27) has packets to be sent to a destination node d (node 15 in fig), it checks whether node d is within its zone.
- If the destination belongs to its own zone, then it delivers the packets directly.

- Otherwise, node s broadcasts the Route Request to its peripheral nodes (in fig, node 8 broadcasts Route Request to node 2, 3, 5, 7, 9, 10, 13, 14 and 15).
- If any peripheral node finds node d to be located within its routing zone, it sends a Route Reply back to node 8 indicating the path; otherwise, the node rebroadcasts the Route Request packet to the peripheral nodes.
- This process continues until node d is located.
- During Route Request propagation, every node that forwards the Route Request appends its address to it.
- This information is used for delivering the Route Reply packet back to the source.
- The criteria for selecting the best path may be the shortest path, least delay path etc.
- When an intermediate node in an active path detects a broken link in the path, it performs a local path reconfiguration in which the broken link is bypassed by means of a short alternate path connecting the ends of the broken link.
- A path update message is then sent to the sender node.
- This results in sub-optimal path between two end points.
- Reduce the control overhead by combining the best features of Proactive and Reactive protocols.
- Control overhead may increase due to the large overlapping of nodes routing zones.
- ZHLS uses the geographical location info of the nodes to form non-overlapping zones. A Hierarchical Addressing that consists of a zone ID and a node ID is employed.
- Similar to ZRP, ZHLS also employs a Proactive approach inside the geographical zone and a Reactive approach behind the zone.
- Every node requires GPS support for obtaining its own geographical location that is used to map itself into corresponding zone.
- The assignment of zone addresses to geographical areas is important and is done during a phase called the network design phase or network deployment phase.
- Each node maintains two link state packets: (LSP)
- Node level LSP: list of connected neighbors.
- Zone LSP: list of connected zones.
- Route Establishment-If a source node src wants to communicate with a destination node dest, src checks whether dest resides in its own zone.
- If dest belongs to same zone, then packets are delivered to the dest as per the Intra-Zone routing table.
- If dest does not belong to the same zone, then the src originates a location request packet containing the sender’s and destination’s information. This location info is forwarded to every other zone.
- The gateway node of a zone at which the location request packet is received verifies its routing table for the destination node.
- The gateway node that finds the destination node required by a location request packet originates a location response packet containing the zone information to the sender.

Table 7.1: Zone link state packets



Source Zone	Zone Link State Packet
A	B, D, F, and G
B	C and A
C	B and D
D	A, C, and E
E	A, D, and F
F	A, E, and G
G	A and F



(a) Node-level topology in ZHLS

(b) Zone topology for the node-level topology in (a)

Figure 7.28. Zone-based hierarchical link state routing protocol.

- If a given gateway node away causing a zone level connection failure, routing can still take place with the help of the other gateway nodes. This is due to the hierarchical addressing that makes use of zone ID and node ID.

Advantages

- Reduce storage requirements and common overhead.
- Robust and resilient to path breaks.
- Non overlapping zones.

Disadvantages

- Additional overhead incurred in creation of zone level topology.
- Path to Destination is suboptimal.
- Geographical info may not be available in all environments.

OBJECTIVE QUESTIONS

1. MANET Stands for **Mobile adhoc network**

2. The Important design criteria for MANET is **Energy Conservation**

3. **Bluetooth** is example for PAN

4. The parameters of Qos are **Jitter,Bandwidth and Random Delay Time**

5. In Manet the topology is **Dynamic**

6. All nodes in adhoc network **are Peer to Peer**

7. Security is the crucial issue in MANET due to **lack of Centralized Network**

8. DSDV stands for **Destination sequenced distance vector routing protocol**

9. Zone routing protocol is a **Hybrid type of protocol.**

10. Location aided routing is **Restricted directional flooding** type of forwarding stratergy.

11. Which type of network in which routers themselves are mobile?**[a]**

a.WAN

b.MANET

c.MNET

d.None

12.what is the routing algorithm used in MANET?**[d]**

a.Shortest path first

b.Routing information Protocol

c.Distance vector Protocl

d.Adhoc on demand distance vector protocol

13. Which of the following are challenges of MANETs?**[c]**

a.Dynamic topology

b.Less Bandwidth

c.Both

d.None

14. Adhoc multicast routing protocol with increasing sequence number is a type of **[a]**

a.on demand protocol

b. no demand protocol

c.full demand protocol

d.none

15. which of following is the applications of wireless networks**[d]**

a.Military

2.Smart Home

3.Traffic Management

d.All

16. Which of the following are common attributes of MANETS**[d]**

a.Dynamic topology

b.Enegy conservation with batteries

3.Limited Bandwidth

d.all

17. which of the following is not a location service**(b)**

a.some-to-some

b.some to all

c.All for some

d.None

18. which protocol is mostly used in dense and big area network due to optimized multi point relay;**[A]**

a.OLSR

b.AODV

c.TBRPF

19. The protocols acquire a rout befor sending packet**[a]**

a.DSDV&WRP

b.DSDV&ZRP

c.WRP&ZRP

d.STAR AND OSR

UNIT-2

Short Answers

1. Define the term Broadcasting and Broadcast storm problem?

Broadcasting is nothing but sending message to all the nodes. And for the nodes which are far away the message is rebroadcasted .In this process collisions will occur, because multiple sender broadcast at same time and are in neighborhood it is called broadcast storm problem.

2. What are the different categories of broadcasting protocols?

1. Simple flooding

2. probability based method

3. Area based method

4. Neighbour knowledge methods

3. Explain about simple flooding?

In this broadcasting starts off with a source node broadcasting a packet to all its neighbours.And the neighbors upon receiving this it will broadcast packet to only its neighboring nodes exactly once. This process is repeated until all reachable nodes have received and rebroadcast the packet at least once.

4. Explain about Probability Based Methods?

In the probability based scheme, this is a simple probabilistic approach of probability 1 or 0 for rebroadcasting. A node will broadcast either with probability 1 or with probability 0. That means with probability 1 it behaves like a flooding approach where as with 0 probability it is not broadcasting a single packet.

In case of dense network nodes are very closer so saves the resources because no need of rebroadcasting. But in case of sparse network all nodes are far away from each other so rebroadcasting is required.

5.Expalin about simple flooding?

In this method upon receiving the unseen packet ,a node initiates the counter with a value of one and sets RDT.During RDT the value of the counter will be incremented by one for each redundant packet. If the counter is less than the threshold value the packet has to be rebroadcasted otherwise it is simply dropped.

6. Define the term Multicasting?

It is transmission of data to a group of hosts identified by a single destination address and hence is intended for group oriented computing.

7. What are the different categories in Multicasting routing protocols?

.1.Tree based approach

2. Meshed Based approach

3. Stateless Multicast

4. Hybrid Approach

8. What are the advantages and disadvantages of Tree based approach?

1.A packet traverses each hop and node in a tree at most once.

2.Very simple routing decisions at each node

3. Number of copies of a packet is minimized.4.Provides loop free structure

Disadvantages:

In trees they will provide only unique path between any two nodes .Therefore failure of one link could mean reconfiguration of the entire tree structure.

9. What are the advantages and disadvantages of Meshed Based approach?

It provides Multiple paths between between sender and the receiver pairs.

Disadvantage:

The disadvantage of a mesh is increase in data forwarding overhead.

The probability of collisions is higher.

Redundant packets consume more bandwidth.

10. Explain about Stateless Multicast?

In this approach source itself maintains the list of destinations in the packet head. It focuses on small group of multicast. The routing protocol will take care of forwarding packet to the respective destinations based on the address present in the header.

11. Explain about Geocasting?

It is nothing but nodes are eligible to receive the packet are implicitly specified by a physical region. Membership in a geocast group changes whenever a mobile node moves in or out of the geocast region.

Long Answers

1. Explain about tree based approaches?

Tree based Approaches: As in fixed (non-mobile) multicast routing, tree-shared tree rather than evenly distributed throughout the network, which based protocols build a tree over which multicast data is forwarded.

Although tree based approaches are bandwidth-efficient, they do not always offer sufficient robustness and due to mobility susceptible for link failure.

1. Source-Tree-based approach: During this approach every source node produces a single multicast tree comprising all the members in a group. Generally, the path between the source and each member is not the one which is the shortest.

2. Shared-Tree based approach: A single multicast group encompassing every source node is produced is only produced for a multicast tree during this approach. After that this tree is rooted at a node treated as the core node. Every source utilizes this tree to start a multicast. Shared-Tree-based approach is not recommended for shortest path for routing, however it treats single point of failure, and so it often conserves more routing information which leads to overhead. Along with this, the traffic is aggregated on the gives it low throughput.

Multicast Ad Hoc On-Demand Distance Vector (MAODV):

The MAODV protocol is elongated from AODV . It conserves a shared tree for each multicast group, having only of receivers and relays (forwarding nodes). It figures out a multicast route on demand by utilizing a broadcast route discovery mechanism. The leader of that group is the one which is the very first member of that multicast group and this leader is liable for conserving the multicast group sequence number and disseminating this particular number to the multicast group. This is executed by a group HELLO message. Nodes utilize the group HELLO information to update their Request Table. MAODV uses an expanding ring search (ERS) to maintain the multicast tree. Whenever a broken link is traced between two nodes, the downstream node is liable for starting the repair link. The downstream node disseminates an RREQ packet by the way of utilizing an ERS. A node with a hop count to the multicast group leader less than or equal to the indicated value in the RREQ packet can only be able to give response. In the case of downstream node not getting a reply, it acknowledges that the multicast tree is divided into parts. Then the downstream node becomes the new multicast group leader for its involvement in the multicast tree split. Till the two parts of the network reconnects, the multicast tree remains as parts. Observation: The major disadvantages of MAODV are long delays and high overheads which are connected with fixing broken links in situations of high mobility and traffic load. Moreover, it has a low packet delivery ratio in situations with high mobility, large numbers of members, or in the case of a high traffic load. For this reason of its dependence on AODV, MAODV is not resilient. Eventually, it suffers from a single point of failure, which is the multicast group leader.

2. Explain about Multicast Ad Hoc On Distance vector protocol(MAODV)?

Both AODV and MAODV are routing protocols for adhoc networks, with AODV for unicast traffic and MAODV for multicast traffic. MAODV maintains a shared tree for each multicast group, consisting of only receivers and relays. Sources wishing to send to the group acquire routes to the group on demand in a way similar to the ad hoc on demand distance vector (AODV) protocol. Each multicast tree has a group leader, which is the first node to join the group in the connected component. The group leader in each connected component periodically transmits a group hello packet to become aware of reconnections. Receivers join the shared tree with a special route request as it is receiver-initiated approach MAODV allows each node in the network to send out multicast data packets, and the multicast data packets are broadcast when propagating along the multicast group tree. The core of the MAODV protocol is about how to form the tree, repair the tree when a link is broken, and how to merge two previously disconnected tree into a new tree.[5].There are four types of packets in MAODV: RREQ, RREP, MACT and GRPH. RREQ and RREP are also packets in AODV. A node broadcasts a RREQ when 1) it is a member node and wants to join the tree, or 2) it is a non-member node and has a data packet targeted to the group

Route Discovery and Maintenance for Reaching a Multicast Tree:

MAODV implementation supports that every node in the network can send out multicast traffic, we must consider how these data packets reach each multicast group member if the data source node is not a tree member. We choose a two-step approach: first step, there is a route
from that data source node to a tree member; then after the tree member receives the multicast data packets, it propagates the data through the whole tree, reaching every group member we borrow the mechanism used for route discovery and maintenance for reaching a specific node in AODV to accomplish the first step. The data source node initiates a RREQ to ask for a route to that multicast address. Usually, this RREQ is the same as the RREQ used in AODV, without any flags and broadcasted in the network.But with the Group Leader Table, the source node may already know a route to reach the group leader. By using the information recorded in the Group Leader Table, the RREQ can be sent unicastly towards the group leader if this is the first time the node sends RREQ. During RREQ propagation, the reverse route towards the source node is constructed as described in the AODV protocol. Any node not in the multicast tree but with fresh enough route to that multicast address, or any tree member with known group leader can respond to this RREQ with a RREP. While the RREP is sent back to the source node along the reverse route, every intermediate node and the source node updates the route to that tree member with the destination address set to the multicast group address, thus the forwarding route is established in their Unicast Route Tables. For this first step, the end node is a tree member. The second step is accomplished under multicast tree construction. Along with RREQ and RREP messages used in AODV in addition, MACT message is introduced to finish the tree construction.

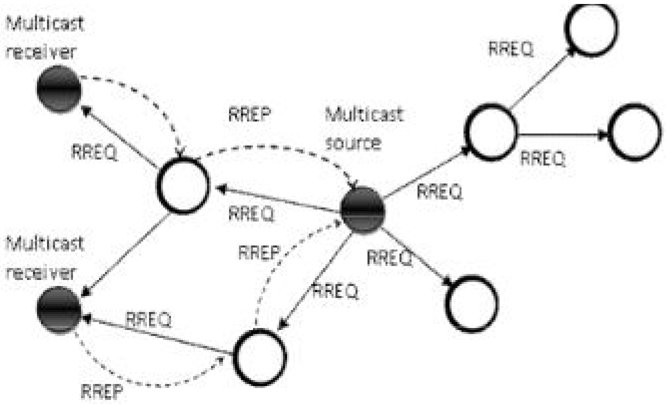


Figure-2: Route Discovery and Maintenance for Reaching a Multicast Tree.

3. Explain about Multicast zone routing protocol(MZRP) ?

The combination of reactive and proactive routing approaches of a source Commenced multicast protocol is known as MZRP [21].Routing sector is present in all nodes.Within the zone proactive approach is applied and outside the zone reactive approach is applied. A multicast tree is built within the routing zone and then across the zone the tree is expanded(the entire network). The category of a multicast advancing node transforms to multicast group member when the node desires to unite with multicast group. A multicast route request (MRREQ) messages launched by the other node. According to the data the source node has MRREQ is divided into two categories unicast or broadcast. A unicast MRREQ is sent down the route to the multicast tree and remains until a multicast route reply MRREQ, if the source node has a suitable route to all nodes on the tree and wishes to join that group Unicast MRREQ is promoted and invalidate ways are located in the multicast routing tables by the transitional nodes. MRREP is responded by the target when the MRREQ is obtained.A border cast MRREQ is commenced which is mailed through the border cast tree of the source if the uncast MRREQ is not successful or there is no convincing route to that groupby the source.On obtaining the border cast MRREQ, the secondary nodes will verify whether a suitable route to the multicast group or group leader is present or not.

In place of border cast MRREQs, unicast MRREQs are mailed and till the MRREPs are received they remain like that, or else through the border cast tree of the secondary nodes the MRREQs are mailed. Between the channel nodes reverse ways rebuilt. If the multicast group consists of a multicast tree and on obtaining an MRREQfor a multicast group, the end node mails an MRREP to the supply and to start the fresh division of the multicast tree, they stay till they obtain an MRACT letter from the supply node. Through the reverse way the MRREP is mailed tothe supply source.

Observation:

various group dimensions are ranged properly by MZRP. Since the MZRP runs entirely above the Zone Routing Protocol (ZRP) [32], both can transfer the data concluding that ODMRP has more power overhead than MZRP.A node external tothe supply routing zone must remain still for a substantial period to unite with the

group which is the major problem of this protocol.On evaluating, MZRP has more potential to produce many situations at nodes which are a part of many groups each having variety of nodes with the shared tree based apporoach.

4.Explain about Multicast Zone Routing Protocol?

Protocol Description

A ZRP network is partitioned into zones with a set of nodes.

MZR has two parts to it. A proactive protocol runs inside each zone, maintaining an up-to-date zone routing table at each table. A reactive multicast tree creation is initiated when a source needs to send multicast data to its group members. Each mobile node participating in an ad hoc network constructs a zone around itself with a pre-configured zone radius. A simplified distance vector protocol is implemented for creating zones and for maintaining a Zone Routing Table at each node. Every node in the mobile ad hoc network periodically broadcasts an ADVERTISEMENT packet, identifying itself. The propagation of the advertisement packets is restricted to a zone by setting the time-to-live(TTL) value of these packets to the zone radius. The nodes that are within the transmission range of a node A pick up the advertisement packet sent by A. Each node that receives an ADVERTISEMENT packet rebroadcasts it, if the TTL of the packet is still valid. When a node B receives the advertisement packet from A, a route entry for A is created and stored in B's zone routing table. The distance to A from B is set to the hop count in the advertisement packet and the next hop in the route entry is set to the node from which B received A's advertisement packet. A soft state approach is followed to remove stale routes from the zone routing table. Route entries expire if advertisement packets from the corresponding destination nodes do not periodically refresh them.

A zone routing table (containing unicast routes to each zone node) is kept up-to-date through this proactive protocol built on periodic advertisements. Routes to destinations that moved away are removed when they expire. Routes to new destinations are added to a node's zone routing table, when it receives advertisement packets from these destinations. Also, the protocol's reaction to the changes in topology is localized to a zone. Only the nodes within a zone are affected and only they need to update their zone routing tables. By looking at the zone routing table and the number of hops to each destination, a node can identify the interior zone nodes and the border nodes. Each node also maintains a Neighbor Table, which contains all those nodes from which a node received ADVERTISEMENT packets with a hop count of one.

A multicast source initiates the creation of a multicast data delivery tree. The tree creation is done in a two-stage process. The source initially forms the tree inside its zone and then tries to extend the tree to the entire network. The source sends a TREE-CREATE to each zone node. When a zone node, interested in the multicast group session, receives the TREE-CREATE packet, it creates a multicast route entry and replies to the source with a TREE-CREATE-ACK packet. The TREE-CREATE-ACK packet is sent back to the source through the reverse route created by the TREE-CREATE packet

Fig. 1.

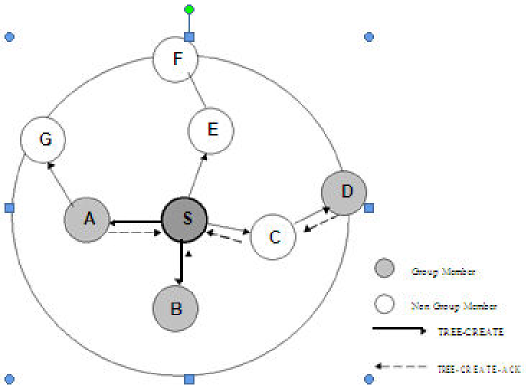
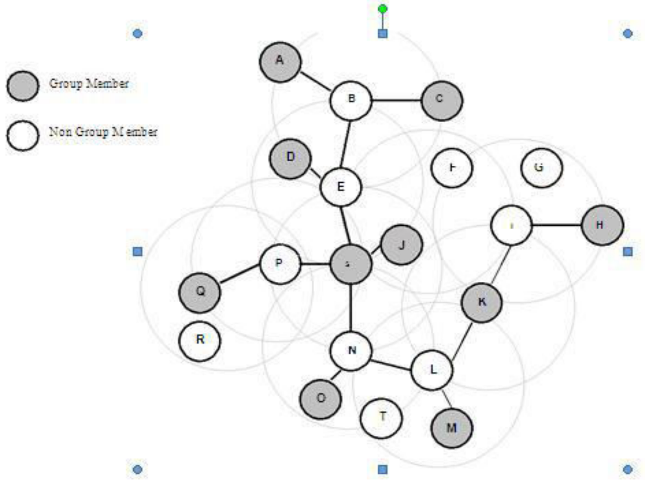


Figure 2 illustrates the multicast tree extension through the entire network. Source S initiates the multicast tree creation. Border nodes E, P and N in turn extend the multicast tree inside their zones and to the rest of the network. Nodes E, I, L, N and P, though not interested in the multicast group become multicast tree members because they provide connectivity to other member nodes.

Fig. 2. Multicast Tree Extension through the entire network



A node can be a member of multiple routing zones at the same time because the routing zones heavily overlap. It is possible that a node receives multiple TREE-CREATE and TREE-PROPAGATE messages, effectively flooding the network. This is prevented by early detection and termination of redundant TREE-PROPAGATE threads.

Routing Mechanism

The source starts transmitting data packets to the group members once the multicast delivery tree is created. When a node on the multicast tree receives a data packet from its upstream node, it replicates the data packet and sends a copy
to each node in the downstream list. A node stops transmitting data packets to a downstream node, if the downstream node migrates and moves out of its transmission range.

5.Explain about On Demand Multicast Routing Protocol?

On-Demand Multicast Routing Protocol (ODMRP) is a mesh-based multicast protocol. It can coexist with any unicast routing protocol.

ODMRP group membership and multicast routes are established and updated by the source on-demand. A request phase and a reply phase comprise the protocol . If a multicast source has packets to send, it floods a member advertising packet with data payload piggybacked. The so-called JOIN QUERY packet is periodically broadcasted to the entire network to refresh the membership information an update the routes as follows. When a node receives a non-duplicate JOIN QUERY, it stores the upstream node ID into the routing table and rebroadcasts the packet. When the JOIN QUERY packet reaches a multicast receiver, the receiver creates and broadcasts a JOIN REPLY to its neighbors. This is exactly the difference between AODV and ODMRP! In AODV the source broadcasts the RREQ and the destination unicasts the RREP, thus a tree is built up incrementally. While in ODMRP, the source broadcasts the JOIN QUERY and the destination also broadcasts the JOIN REPLY, thus a mesh is formed incrementally.

When a node receives a JOIN REPLY, it checks if the next node ID of one of the entries matches its own ID. If it does, the node realizes that it is on the path to the source and thus is part of the forwarding group. It then sets the FG_FLAG (Forwarding Group Flag) and broadcasts its own JOIN REPLY built upon matched entries. The JOIN REPLY is then propagated by each forwarding group member until it reaches the multicast source via the shortest path. This process constructs the routes from sources to receivers and builds a mesh of nodes, the forwarding group.

The forwarding group is a set of nodes which is in charge of forwarding multicast packets. It supports shortest paths between any members. Multicast group members and forwarding group nodes forward multicast data packets. A multicast receiver also can be a forwarding group node if it is on the path between a multicast source and another receiver.

Advantages

- Simplicity.
- Low channel and storage overhead.
- Usage of up-to-date shortest routes.
- Reliable construction of routes and forwarding group.
- Robustness to host mobility.
- Maintenance and utilization of multiple paths.
- Exploitation of the broadcast nature of the wireless environment.
- Unicast routing capability.

6. Explain about Core assisted Mesh Protocol?

CAMP : Core-Assisted Mesh Protocol
This approach, CAMP is the next generation core based trees CBT which were made known for Internet multicasting into multicasting meshes and further which possess higher connectivity than the conventional trees. In cases of repeated movement of the network routers, to facilitate better connectivity this approach defines a shared multicast group. CAMP establishes and maintains a multicast mesh, which is a subset of the network topology, which provides multiple paths between a source-receiver pair and ensures that the shortest paths from receivers to sources (called reverse shortest paths) are part of a group's mesh. One or multiple cores are defined per multicast group to assist in join operations; therefore, CAMP eliminates the need for flooding. CAMP uses a receiver-initiated approach for receivers to join a multicast group. A node sends a JREQ toward a core if none of its neighbors is a member of the group; otherwise, it simply announces its membership using either reliable or persistent updates. If cores are not reachable from a node that needs to join a group, the node broadcasts its JREQ using an ERS, which eventually reaches some group member. In addition, CAMP supports an alternate way for nodes to join a multicast group by employing simplex mode. Observation: CAMP needs an underlying proactive unicast routing protocol (the Bellman-Ford routing scheme) to maintain routing information about the cores, in which case considerable overhead may be incurred in a large network. Link failures have a

small effect in CAMP, so, when a link fails, breaking the reverse shortest path to a source, the node affected by the break may not have todo anything, because the new reverse shortest path may very well be part of the mesh already. Moreover, multicast data packets keep flowing along the mesh through the remaining paths to all destinations. However, if any branch of a multicast tree fails, the tree must reconnect all components of the tree for packet forwarding to continue to all destinations.

7. Explain main reasons why TCP is not suitable for Adhoc networks?

TCP OVER AD HOC WIRELESS NETWORKS:

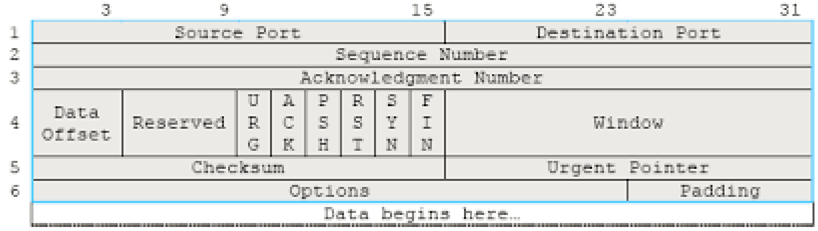
TCP is reliable, end to end, connection oriented TL protocol that provides a byte stream based service.
Major responsibilities of TCP include
Congestion control.
Flow control In order delivery of packets.
Reliable transportation of packets.

reasons why TCP does not perform well in Adhoc wireless network
The major reasons behind throughput degradation that TCP faces when used in ad hoc wireless
networks are the following.
1. Misinter pretation of packet loss:
In traditional TCP design, the packet loss is mainly attributed to network congestion. Ad hoc wireless network experience a much higher packets loss due to High bit rate Increased Collections etc.
2. Frequent path breaks:
If the route reestablishment time is greater than the RTO period of TCP sender, then the TCP sender assumes congestion in the networkretransmits lost packets and initiates congestion control algorithm. This leads to wastage of bandwidth and battery power.
3. Effect of path length:
As path length increases, the throughput decreases.
4. Misinterpretation of congestion window:
When there are frequent path breaks, the congestion window may not reflect the maximum transmission rate acceptable to the network and the receiver.
5. Asymmetric link behavior:
Radio channel used in ad hoc wireless network has different properties such as location dependent contention, directional properties etc leading to asymmetric links. This can lead to TCP invoking the congestion control algorithm and several retransmissions.
6. Unidirectional path:
TCP relies on end to end ACK for ensuring reliability. Path break on an entirely different reverse path can affect the performance of the network as much as a path breaks in the forward path.
7. Multipath Routing:

For TCP, multipath routing leads to significant amount of out of order packets, when intern generates a set of duplicate acknowledgement (DUPACKs),which cause additional power consumption and invocation of congestion control.

8.Explain about TCP header format

TCP/IP Header Format: TCP segments are sent as internet datagrams. The Internet Protocol header carries several information fields, including the source and destination host addresses. A TCP header follows the internet header, supplying information specific to the TCP protocol. This division allows for the existence of host level protocols other than TCP.



<table>
<tr><td></td><td>3</td><td>9</td><td></td><td></td><td></td><td></td><td></td><td>15</td><td>23 31</td></tr>
<tr><td>1</td><td colspan="8">Source Port</td><td>Destination Port</td></tr>
<tr><td>2</td><td colspan="9">Sequence Number</td></tr>
<tr><td>3</td><td colspan="9">Acknowledgment Number</td></tr>
<tr><td>4</td><td>Data Offset</td><td>Reserved</td><td>U
R
G</td><td>A
C
K</td><td>P
S
H</td><td>R
S
T</td><td>S
Y
N</td><td>F
I
N</td><td>Window</td></tr>
<tr><td>5</td><td colspan="8">Checksum</td><td>Urgent Pointer</td></tr>
<tr><td>6</td><td colspan="8">Options</td><td>Padding</td></tr>
<tr><td></td><td colspan="9">Data begins here…</td></tr>
</table>

Fig: TCP Header

- Source Port (16): The port number of the host sending the data
- Destination Port (16): The port number of the application requested on the destination host.
- Sequence Number (32): The sequence number of the first data octet in this segment (except when SYN is present). If SYN is present the sequence number is the initial sequence number (ISN) and the first data octet is ISN+1. It puts the data back in the correct order or retransmits missing or damages data, a process called sequencing.
- Acknowledgement Number (32): Define which TCP octet expected next. If the ACK control bit is set this field contains the value of the next sequence number the sender of the segment is expecting to receive. Once a connection is established this is always sent.
- Header Length (4): Stands for header length, which defines the number of 32 bit words in the header. This indicates where the data begins. The TCP header (even one including options) is an integral number of 32 bits long.
- Unused (6): Reserved for future use, it always set to 0.
- Flags (6): Control functions used to set up and terminate a session. Flags from left to right:

o URG: Urgent Pointer field significant
o ACK: Acknowledgment field significant
o PSH: Push Function

- RST: Reset the connection
- SYN: Synchronize sequence numbers
- FIN: No more data from sender

Windows (16): The window size of the sender is willing to accept, in octet. The number of data octets beginning with the one indicated in the acknowledgment field which the sender of this segment is willing to accept.

Checksum (16): An error detection code. The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header and text. If a segment contains an odd number of header and text octets to be check summed, the last octet is padded on the right with zeros to form a 16 bit word for checksum purposes. The pad is not transmitted as part of the segment. While computing the checksum, the checksum field itself is replaced with zeros.

The checksum also covers a 96 bit pseudo header conceptually prefixed to the TCP header. This pseudo header contains the Source Address, the Destination Address, the Protocol, and TCP length. This gives the TCP protection against misrouted segments.

Urgent Pointer (16): Indicates the ends of urgent data. This field communicates the current value of the urgent pointer as a positive offset from the sequence number in this segment. The urgent pointer points to the sequence number of the octet following the urgent data. This field is only be interpreted in segments with the URG control bit set.

Options (32): Options may occupy space at the end of the TCP header and are a multiple of 8 bits in length. All options are included in the checksum. An option may begin on any octet boundary. There are two cases for the format of an option:

Case 1: A single octet of option-kind.

Case 2: An octet of option-kind, an octet of option-length, a the actual option-data octets.

The option-length counts the two octets of option-kind and option-length as well as the option-data o

OBJECTIVE

1)TCP IS A protocol.
A.stream-oriented

B.message-oriented
C.block-oriented
D.packet-oriented

2) Which of the following is not the layer of TCP/IP protocol.
A.Physical
B.link
C.network
D.transport

3) TCP groups a number of bytes together into a packet called a ....
A. userdatagram
B.segment
C. datagram
D. packet

4) The .... of TCP/IP protocol is responsible for figuring out how to get data to its destination.
A. application layer
B. link layer
C. network layer
D.transport layer.

5) TCP is a(n) .... transport protocol.
A. protocol delivery
B. reliable
C. best-effort delivery
D. effortless delive

6) .... is the protocol that hides the underlying physical network by creating a virtual network view.
A. Internet Protocol(IP)
B. Internet Control Message Protocol(ICMP)
C. Address Resolution Protocol(ARP)
D. Bootstrap Protocol(BOOTP)

7) TCP assigns a sequence number to each segment that is being sent. The sequence number for each segment is number of the .... byte carried in that segment.
A. first
B. last
C. middle
D. zero

8) A port address in TCP/IP is ....bits long.
A. 32
B. 48
C. 16
D. 64

9)In Reverse Path Forwarding (RPF), router receives a packet and extracts the

1. Protocol Address
2. Source Address
3. IP Address
4. Standard Address

10)A network can receive a multicast packet from a particular source only through a

1. designated parent resolve
2. designated protocol router
3. designated parent rotator
4. designated parent router

11) To convert broadcasting to multicasting, protocol uses

1. Three Procedures
2. Two Procedures
3. One Procedure
4. Multi Procedures

12). In multicast communication, relationship is

1. one to one
2. one to many
3. many to one
4. one to all

13) A dynamic routing table is updated

1. manually
2. periodically
3. randomly
4. None

14) In Multicast Routing Protocol, flooding is used to broadcast packets but it creates

1. gaps
2. loops
3. holes
4. links

UNIT-4

Short Answers

1. Explain about distributed systems security threats?

Distributed Systems Security

The threats are divided into three categories: disclosure threats, integrity threats and denial of service threats. The disclosure threat involves the leakage of information from the system to a party that should not have seen the information and is a threat against the confidentiality of the information. The integrity threat involves an unauthorized modification of information. Finally, the denial of service threat involves inability to access a system resource that is being blocked by a malicious

2. Explain security issues in wireless sensor networks?

- Security requirements
- Availability
- Authorization and key management
- Confidentiality and Integrity
- Non repudation
- Security solutions constraints

3. Discuss the characteristics of security solutions for adhoc networks?

• **Lightweight:** Solutions should minimize the amount of computation and communication required to ensure the security services to accommodate the limited energy and computational resources of mobile, ad hoc-enabled devices;

• **Decentralized:** Like ad hoc networks themselves, attempts to secure them must be ad hoc: they must establish security without reference to centralized, persistent entities. Instead, security paradigms should levy the cooperation of all trustworthy nodes in the network;

• **Reactive:** Ad hoc networks are dynamic. Nodes - trustworthy and malicious - may enter and leave the network spontaneously and unannounced. Security paradigms must react to changes in network state; they must seek to detect compromises and vulnerabilities. Therefore, these solutions should be reactive.

• **Fault-Tolerant:** Wireless mediums are known to be unreliable; nodes are likely to leave or be compromised without warning. The communication requirements of security solutions should be Designed with such faults in mind; they should not rely on message delivery or ordering.

4. What are the Challenges of WSN?

- Key management
- Secure routing
- Intrusion Detection

- Authentication

  Trusted third parties

  Chain of trust

  Location limited authentication

5. Explain about Authentication in detail?

Authentication denotes the accurate, absolute identification of users who wish to participate in the network. Historically, authentication has been accomplished by a well-known central authentication server. The role of the server is to maintain a database of entities, or users, and their corresponding unique IDs. The ID may be a digital certificate, public key, or both. Unfortunately the ad hoc paradigm does not accommodate a centralized entity creating protocol deployment issues.

1. Trusted Third Parties

One of the most rudimentary approaches to authentication in ad hoc networks uses a Trusted Third Party (TTP). Every node that wishes to participate in an ad hoc network obtains a certificate from a universally trusted third party. When two nodes wish to communicate, they first check to see if the other node has a valid certificate. Although popular, the TTP approach is laden with flaws. Foremost, it probably is not reasonable to require all ad hoc network-enabled devices to have a certificate. Secondly, each node needs to have a unique name. Although this is reasonable in a large internet, it is a bit too restrictive in an ad hoc setting. Recent research has introduced many appropriate variations of TTPs, and these are discussed later.

2 .Chain of Trust

The TTP model essentially relies on a fixed entity to ensure the validity of all nodes' identities. In contrast, the chain of trust paradigm relies on any node in the network to perform authentication. That is, if a node wishes to enter a network session, it may request any of the existing nodes for authentication. This paradigm fails if there are malicious modes within the network or the incoming nodes cannot be authenticated at all.

3. Location-Limited Authentication

Location-limited authentication levies on the fact that two nodes are close to one another and most ad hoc networks exist in a small area. Bluetooth and infrared are two of the most widely used protocols for this form of authentication. Although it may not seem obvious, location limited authentication is potentially very secure. The security is obtained from physical assurance and tamper-detection. That is, the authenticating node can be reasonably certain that the node it thinks is being authenticated is the node it is actually authenticating (i.e., there is no man-in-the-middle) by physical indications - the transfer light on the requesting node is blinking, the person operating the device is physically present, etc. Although location-limited authentication is well-suited for most applications with a single end-point, it is not feasible for large, group-based settings.

6. Discuss and compare key management paradigms?

A) **Definition 1**: A group key is a secret that is used by two or more parties to communicate securely. Group keys are symmetric; that is, the same group key is used to encrypt and decrypt messages.

Definition 2: Key independence ensures that a passive adversary who knows a proper subset of group keys I c K cannot discover any other group key K e (K - K).

Definition 3: Forward secrecy ensures that a passive adversary (member or non-member) who knows a contiguous subset of old group keys cannot discover subsequent group keys.

Definition 4: Backward secrecy ensures that a passive adversary who knows a contiguous subset of group keys cannot discover preceding group keys.

Definition 5: Key establishment is the process, protocol, or algorithm by which a group key is created and distributed to the group.

Definition 6: Key agreement is a protocol by which two or more parties contribute to the creation of a shared group key.

Definition 7: Key distribution is the process by which each group member is apprised of the group key.

Definition 8: Key integrity ensures that the group key is a function of all authenticated group members and no one else.

One of the easiest ways for an adversary to sacrifice key integrity is by compromising prior keys or the individual contribution - the secret shared key - of a group member. This closely related form of attack is
known as a known key attack.

Definition 9: A protocol is vulnerable to a known key attack if compromise of past session keys allows a passive adversary to compromise future group keys, or an active adversary to impersonate one of the protocol parties.

7. Explain the different notations used in Diffie -Hellman Algorithm?

Notations

A ,B	protocol participants
P	large prime number
G	unique subgroup of Z»p of order *q* with *p, q* prime
a	exponentiation base - the generator in the group *G*
X	random secret chosen by *A* such that $1 < x < p - 2$
Y	random secret chosen by *B* such that $1 < y < p - 2$
K	random secret chosen by *B* such that $1 < y < p - 2$
Kt	the partial key created by member *i*

8. Discuss about the Problems Affecting Secure Ad Hoc Routing?

1 Infrastructure

An ad hoc network is an infrastructureless network. Unlike traditional networks, there is no pre-deployed infrastructure such as centrally administered routers or strict policy for supporting end-to-end routing.

2 Frequent Changes in Network Topology

Ad hoc network nodes may frequently change their locations. This results in frequently changing neighbors on whom a node has to rely for routing. As we shall see, this has a significant impact on the implementation of secure routing over MANETs.

3 Wireless Communication

As the communication is through a wireless medium, it is possible for any intruder to easily tap it. Wireless channels offer poor protection and routing related control messages can be tampered. The wireless medium is susceptible to signal interference, jamming, eavesdropping and distortion. An intruder may easily eavesdrop to find out sensitive routing information or jam the signals to prevent propagation of routing information. What is worse is that an intruder could even interrupt messages and distort them to manipulate routes. Secure routing protocols should be designed to handle such problems.

4 Problems with Existing Ad Hoc Routing Protocols

Existing ad hoc routing protocols possess many security vulnerabilities, routing security is very often peculiar to a specific protocol. It may happen that a given vulnerability is present in a given protocol, while it does not exist in another. Therefore, in the following discussions we consider security flaws in the context of a particular protocol.

9. What are the advantages and disadvantages of WatchDog?

Advantage

One advantage of the watchdog mechanism is that it can detect misbehaving nodes at forwarding level and not just the link level.

Weaknesses

Watchdog might not detect misbehaving nodes in presence of:

- Ambiguous collision:
- Receiver collision:
- False misbehavior
- Limited transmission power:
- Multiple colliding nodes
- Partial dropping

10. What are the different types of security solutions which are closely connected components in Token based procedure?

The security solution is composed of four closely connected components:

- **Neighbor verification:** describes how to verify whether each node in the network is a legitimate or malicious node
- **Neighbor monitoring:** describes how to monitor the behavior of each node in the network and detect occasional attacks from malicious nodes
- **Intrusion reaction**: describes how to alert the network and isolate the attackers
- **Security enhanced routing protocol:** explicitly incorporates the security information collected by other components into the ad hoc routing protocol.

11. What are the different types of Intrusion Detection Systems?

Based on the type of audit data, IDS can be classified into 2 types:

- **Network-based:** Network-based IDS sits on the network gateway and captures and examines network packets that go through the network hardware interface;
- **Host-based:** Host-based IDS relies on the operating system audit data to monitor and analyze the events generated by the users or programs on the host.

12. What are the different modules in IDS?

- **Local Data Collection:** The Local Data Collection module gathers streams of real time audit data from eclectic sources, which might include user and system activities within the mobile node, communication activities by this node as well as any communication activities within the radio range of this node and observable to this node.
- **Local Detection Engine:** The Local Detection Engine analyzes the local audit data for evidence of anomalies. This requires the IDS to maintain some expert rules for the node against which the audit data

collected would be checked. However, as more and more appliances are becoming wireless, the types of planned attacks against these appliances is going to increase and this may make the existing expert

rules insufficient to tackle these newer attacks. Moreover, updating these existing expert rules is not a simple job. Thus, any IDS meant for a wireless ad hoc network might have to resort to statistical

anomaly detection techniques. The normal behavior patterns, called "Normal Profiles", are determined using the trace data from a "training " process where all activities are normal. During the "testing" process, any deviations from the normal profiles are recorded if at all any occur. A detection module is computed from the deviation data to distinguish anomalies from normalcy.

- **Cooperative Detection:** If a node locally detects a known intrusion with strong evidence, it can very well on its own infer that the network is under attack and can initiate a response or a remedial action. However, if the evidence of an anomaly or intrusion is a weak one or is rather inconclusive, then the node decides it needs a broader investigation and can initiate a global intrusion detection procedure, which might consist of transmitting the intrusion detection state information among neighbors and further down the network if necessary.

13.Expalin the different steps involved in the IDS?

Intrusion Detection Algorithm can include the following steps:

- The node sends to its neighboring node an "intrusion state request".
- Each node, including the one which initiates this algorithm, then propagates the state information indicating the likelihood of an intrusion to its immediate neighbors;
- Each node then determines whether the majority of the received reports point towards an intrusion; if yes, then it concludes that the network is under attack;
- Any node that detects an intrusion to the network can then initiate the remedial/response procedure.

Long Answers

1.Explain about N-Party Diffie-Hellman Key Agreement

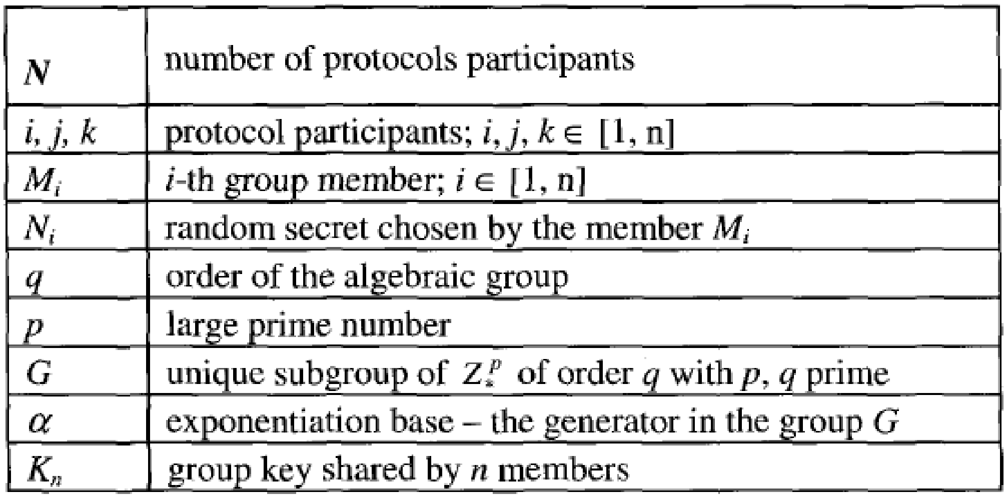
N-Party Diffie-Hellman Key Agreement

Overview

The Diffie-Hellman key agreement protocol was generalized to n participants.

The Protocol

Notation



N	number of protocols participants
i, j, k	protocol participants; $i, j, k \in [1, n]$
M_i	i-th group member; $i \in [1, n]$
N_i	random secret chosen by the member M_i
q	order of the algebraic group
p	large prime number
G	unique subgroup of Z_*^p of order q with p, q prime
α	exponentiation base – the generator in the group G
K_n	group key shared by n members

Key Agreement

The Generalized Diffie-Hellman(GDH) consists of two stages - upflow and downflow. Each member's contributions are collected during the upflow stage, and the resultant intermediate values are broadcast to the group in the downflow stage.

Setup

The setup for GDH is identical to that of two-party Diffie-Hellman: all participants, *Mh . . ., M,,* choose a cyclic group, *G,* of order *q,* and a generator, or in G; each member then chooses a secret share, TV, *e G.*

Upflow

During the upflow, each member *Mt* performs a single exponentiation, appends it to the flow, and forwards the flow to *Mi+1.*

$$M_i \xrightarrow{\alpha^{\prod (N_K | K\in[i,j])} | j\in[1,i]} M_{i+1}$$

The upflow stage terminates and the downflow commences when *Mn =Mi* - when the last member has received the upflow.

$$M_{n-1} \xrightarrow{(\alpha^{N_1}, \alpha^{N_1 N_2}, \ldots, \alpha^{N_1 N_2 \ldots N_{n-1}})} M_n$$

Upon receipt of the upflow, *Mn* calculates the new group key, *Kn,* by exponentiation of the last intermediate value in the flow:

$$K = K_n = (\alpha^{N_1 N_2 \ldots N_{n-1}})^{N_n}$$

Once *Kn* has been calculated, *Mn* commences the downflow.

Downflow

The downflow is initially comprised of *n-l* intermediate values, $(\alpha^{N_1 N_n}, \alpha^{N_1 N_2 N_n}, \ldots, \alpha^{N_1 N_2 \ldots N_{n-2} N_n})$, exponentiated to the n^{tn} group member's secret, , N_n. M_n sends the downflow to *M,,.j.*

$$M_{n-1} \xleftarrow{(\alpha^{N_1 N_n}, \alpha^{N_1 N_2 N_n}, \ldots, \alpha^{N_1 N_2 \ldots N_{n-2} N_{n-1}})} M_n$$

Upon receipt of the downflow, each member, M,, removes its own intermediate value, $(\alpha^{N_1 N_2, \ldots, N_{i-1}, N_{i+1} \ldots N_n})$ calculates the group key, $K_n = (\alpha^{N_1 N_2, \ldots, N_{i-1} N_{i+1} \ldots N_n})^{N_i}$, exponentiates the remaining *i-1* intermediate values in the flow, and forwards the flow to its predecessor, M_{i-1}.

$$\alpha^{N_1 N_n N_{n-1} \ldots N_{i+1} N_i},$$

$$\alpha^{N_1 N_2 N_n N_{n-1} \ldots N_{i+1} N_i},$$

$$M_{i-1} \xleftarrow{\alpha^{N_1 N_2 \ldots N_{i-2} N_{n-2} N_n \ldots N_{i+1} N_i}} Mi$$

2. Discuss about ARAN Protocol?

The ARAN Protocol

The Authenticated Routing for Ad-hoc Networks (ARAN) detects and protects against malicious actions by third parties and peers in an ad hoc environment. ARAN introduces authentication, message integrity and non-repudiation in an ad hoc environment. ARAN makes use of cryptographic certificates for the purposes of authentication and non-repudiation. Route discovery in ARAN is composed of two distinct stages. The first stage is simple and requires little extra work from peers beyond traditional ad hoc protocols. In the optional second stage, nodes increase the security of their routing, incuring additional cost to their peers who may decide not to comply (e.g., if they are low on battery resources). Below we describe the two stages employed in ARAN.

Stage 1

The stage one contains a preliminary certification stage and a mandatory end-to-end authentication stage. This lightweight stage does not demand too many resources.

a) Preliminary Certification

ARAN requires the use of a trusted certificate server, *T*. Before entering the ad hoc network, each node requests a certificate from *T*.

For example, the certificate is issued by *T* to a given node *A* would be:

T -» A: CertA = [IPA, KA + t, e]KTAs

we can see, the certificate contains the IP address of A (IPA), the public key of A (KA), a timestamp *t* of when the certificate has been created, and the time *e* at which the certificate expires. Finally, these

variables are concatenated and signed by *T*. All nodes in the network must maintain fresh certificates with the trusted server *T* and must know its public key (KT).

b) End-to-End Authentication

The goal of stage one is to enable the source node to verify if the intended destination has been reached. In this stage, the source trusts the destination to choose the return path. To better understand the behavior

of the various nodes in ARAN during the route discovery procedure, we now discuss it in two separate parts: route request and route reply.

Suppose a source node A wants to find a route to a destination node X. Node A initiates the route discovery procedure by broadcasting a route request (RREQ) packet (e.g., in on-demand protocols) to its neighbors in the following form:

A -> broadcast: [RREQ, IPX, CertA, NA, t]KA

The RREQ includes the packet type identifier ("RREQ"), the IP address of the destination (IPX), node A's certificate (CertA), a nonce NA, and the current time *t,* all signed with A's private key. Each time node A
initiates a route discovery, it monotonically increases its nonce (i.e., NA). Other nodes receiving the RREQ packet store the nonce they have last seen with its corresponding timestamp. Intermediate nodes receiving the RREQ record the neighbor from which the packet has been received. It then rebroadcasts the RREQ packet to each of its neighbors, signing the contents of the packet. This signature prevents spoofing attacks that may alter the route or form loops. Let B be a neighbor of node A. Node B
would rebroadcast the packet as:

B -» broadcast: [[RREQ, IPX, CertA, NA, t]KA-]K„-, CertB

Nodes do not rebroadcast duplicate packets for which they have already seen the (NA, IPA) tuple. Supposing node C is a neighbor of node B, it validates the packet signature with the given certificate upon receipt of node B's broadcast. Node C then rebroadcasts the RREQ to its neighbors after removing node B's signature, resulting in:

C -» broadcast: [[RREQ, IPX, CertA, NA, t]KA-]Kc-, Certc

Upon receipt of the first RREQ packet with the corresponding nonce at the destination node X, it replies back to the source with a route reply (RREP). Assume node D is the first hop in the reverse path from node X to node A. In this case, the RREP sent by node X takes the form:

X -> D: [RREP, IPA, Certx, NA, t]Kx-

Intermediate nodes receiving the RREP forward it to the predecessor node from which they received the corresponding RREQ. All RREP packets are signed by the sender. Let node D's next hop back to the
source be node C. The RREP sent by node D is:

D -» C: [[RREP, IPA, Certx, NA, t]Kx-]KD-, CertD and this process continues till the source is reached. Nodes along the path check the signature of the previous hop as the RREP is returned to the source. This procedure avoids attacks where malicious nodes instantiate routes by impersonation or replay of node X's packet. When the source node A receives the RREP from the destination node X, it first verifies that the correct nonce has been returned by the destination as well as the destination's signature. Only the destination can answer the RREQ packet and other nodes already having paths to the destination cannot reply on its behalf. While some routing protocols allow this networking optimization (e.g., DSR and AODV), we note that removing this feature also removes several possible security attacks and cuts down on the number of RREP packets received by the source. Since the destination is the only node that can originate a RREP, freedom from loops can be easily guaranteed.

Disadvantages

ARAN requires that nodes keep one routing table entry per source destination pair that is currently active. This is certainly more costly than per-destination entries in non-secure ad hoc routing protocols.

Stage 2

Stage two is performed only after discovery of shortest path in Stage one is over as the destination certificate is required in this phase. Data transfer can be pipelined with the shortest path discovery operation employed in Stage two. Using the same example, the source node A initiates the shortest path discovery operation by broadcasting a Shortest Path Confirmation (SPC) message to its neighbors as (the same arguments are used as in stage one):

A -> broadcast: SPC, JPX, Certx, [[IPx, CertA, NA, t]KA-]Kx+

The SPC message begins with the SPC packet identifier, followed by the destination node X's IP address and certificate. With this, the source concatenates a signed message containing the IP address of X, its own certificate, a nonce and a timestamp. This signed message is then encrypted with node X's public key so that other nodes cannot modify its contents. Intermediate nodes receiving this message rebroadcast the same after including its own cryptographic credentials. For example, a node, say B, would sign the encrypted portion of the received SPC, include its own certificate, and re-encrypt with the public key of X obtained in the certificate forwarded by node A. Therefore, the message rebroadcast by

node B would be:

B -> broadcast: SPC, IPX, Certx, [[[[IPX, CertA, NA, t]KA-]Kx+]KB-, CertB]Kx+

Similar to other non-secure routing algorithms, nodes receiving the SPC packet create entries in their routing table so as not to forward duplicate packets. In addition, this entry also serves to route the reply

packet from the destination to the source along the reverse path. Upon receipt of the packet, the destination node X checks that all the signatures are valid. Node X replies to the first SPC it receives and also to any SPC with a shorter recorded path. Then, it sends a Recorded Shortest Path

(RSP) packet to the source node A through its predecessor node, say D.

X -» D: [RSP, IPA, Certx, NA, route]Kx-

The source node A will eventually receive this packet and verify that the nonce corresponds to the SPC originally generated.

Advantages

The onion-ring like signing of messages prevents nodes in between source and destination from changing the path. First, to increase the path length of the SPC, malicious nodes require an additional valid certificate.

Second, malicious nodes cannot decrease the recorded path length or alter it because doing so would break the integrity of the encrypted data.

Route Maintenance

ARAN is an on-demand protocol where nodes keep track of whether routes are active or not. When an existing route is not used after some pre-specified lifetime, it is simply deactivated (i.e., expired) in the route table. Nodes also use Error (ERR) packets to report links in active routes that are broken due to node movement. For a given route between source node A and destination node X, an intermediate node B generates a signed ERR packet for its neighbor node C as follows:

B -» C: [ERR, IPA, IPX, Certc, NB, t]

Key Revocation

ARAN attempts a best effort key revocation that is backed up with limited time certificates. In the event that a certificate needs to be revoked, the trusted certificate server, node T, sends a broadcast message

to the ad hoc group announcing the revocation. Calling the revoked certificate *Cert R,* the transmission appears as:

T —> broadcast: [revoke, CertR]KT

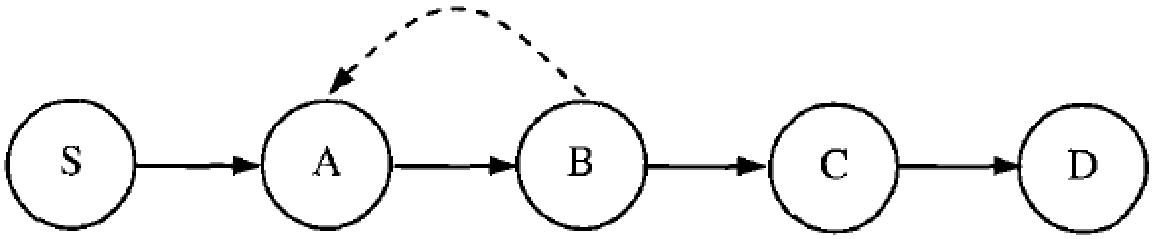
Any node receiving this message rebroadcasts it to its neighbors. Revocation notices need to be stored until the revoked certificate has expired normally. Neighbors of the node with the revoked certificate

may need to rebuild their routes so as to avoid paths passing through the now untrusted node. This method is, however, not failsafe. If an untrusted node whose certificate is being revoked is in between two other nodes in the ad hoc network, it may simply not propagate the revocation message to the other node, thus leading to a partitioned network.

3.Explain about the Watch Dog Method?

The watchdog method is used to detect misbehaving nodes. Figure illustrates how the watchdog works. In this figure, assume node S is transmitting packets to node D. Further assume that node A's

transmission cannot be heard by node C, but it can listen to node B'straffic. Thus, when node A transmits a packet to node B destined to node C, node A can often tell if node B retransmits the packet. If encryption is not performed separately for each link, then node A can also tell if node



Watch Dog Operation

B has tampered with the packet payload or the header. Watchdog can be implemented by maintaining a buffer of recently sent packets and comparing each overheard packet with the packet in the buffer to

determine if there is a match. If so, the packet in the buffer is removed and no longer monitored by the watchdog, as it has been forwarded on. If the packet has remained in the buffer for longer than a certain timeout, the watchdog increments a failure tally for the node responsible for not forwarding on the packet. If the tally exceeds a certain threshold bandwidth, it determines that the node is misbehaving and sends a

message to the source notifying it of the misbehaving station.

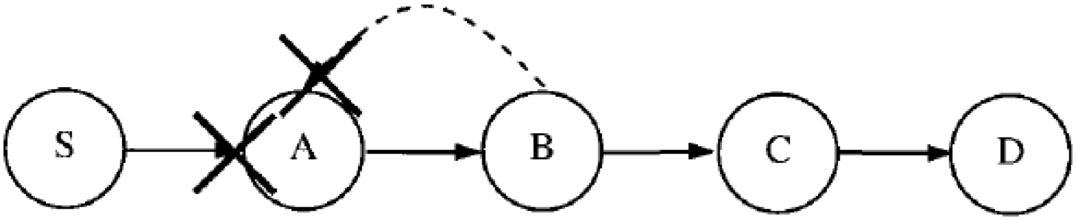
a) Advantage

One advantage of the watchdog mechanism is that it can detect misbehaving nodes at forwarding level and not just the link level.

b) Weaknesses

Watchdog might not detect misbehaving nodes in presence of:

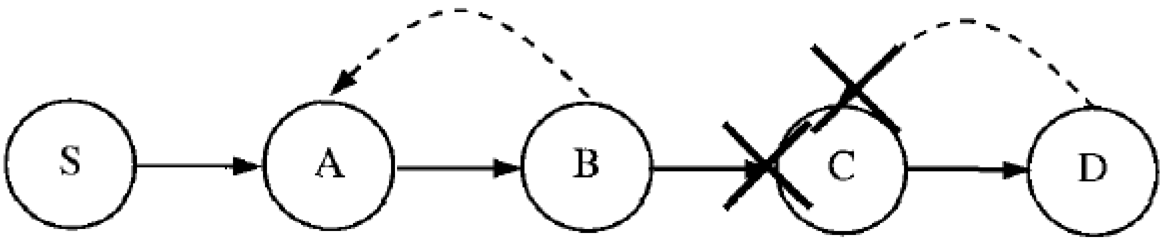
- **Ambiguous collision:** The ambiguous collision problem prevents a given node, say A, from overhearing transmissions from another node, say B. As Figure illustrates, a packet collision occurs at node A while it is waiting for node B to forward a packet. In this situation, node A is not able to figure out if the collision was caused by node B's transmission, or if node B never forwarded the packet and the collision was caused by other nodes in node A's neighborhood. Because of this uncertainty, node A should continue to watch node B over a longer period of time;



Ambiguous collission

Receiver collision: In the receiver collision problem, node A can only tell whether node B sends the packet to node C, but it cannot tell if node C receives it successfully. If a collision occurs at node C

when node B first forwards the packet, node A can only that node B has forwarded the packet and assumes that node C has successfully received it. In this scenario, node B could skip the packet retransmission and evade detection. This situation is shown in



Receiver collision

False misbehavior: False misbehavior can occur when nodes falsely report other nodes as misbehaving. A malicious node could attempt to partition the network by claiming that some nodes following it in

the path are misbehaving. For instance, in Figure 10.5 node A could report that node B is not forwarding packets when in fact it has done so. This will cause node S to mark node B as misbehaving when in

fact node A is the culprit. This behavior, however, can be detected in some cases. Since node A is passing messages onto node B (as verified by node S), then any acknowledgements from node D to

node S will go from node A to node S. In this case, node S will wonder why it receives replies from node D when node B is supposedly dropping packets in the forward direction. In addition, if node A drops acknowledgements to hide them from node S, node B will detect this misbehavior and report it to node D;

• **Limited transmission power:** A misbehaving node that can control its transmission power can circumvent the watchdog. A node could limit its transmission power such that the signal is strong enough to be overheard by the previous node but too weak to be received by the
true recipient;

• **Multiple colliding nodes:** Multiple nodes in collision can mount a more sophisticated attack. For example, nodes B and C in Figure could collide so as to cause a mischief. In this case, node B forwards a packet to node C but does not report to node A when node C drops the packet. Because of its limitation, it may be necessary to disallow two consecutive untrusted nodes in a routing
path.

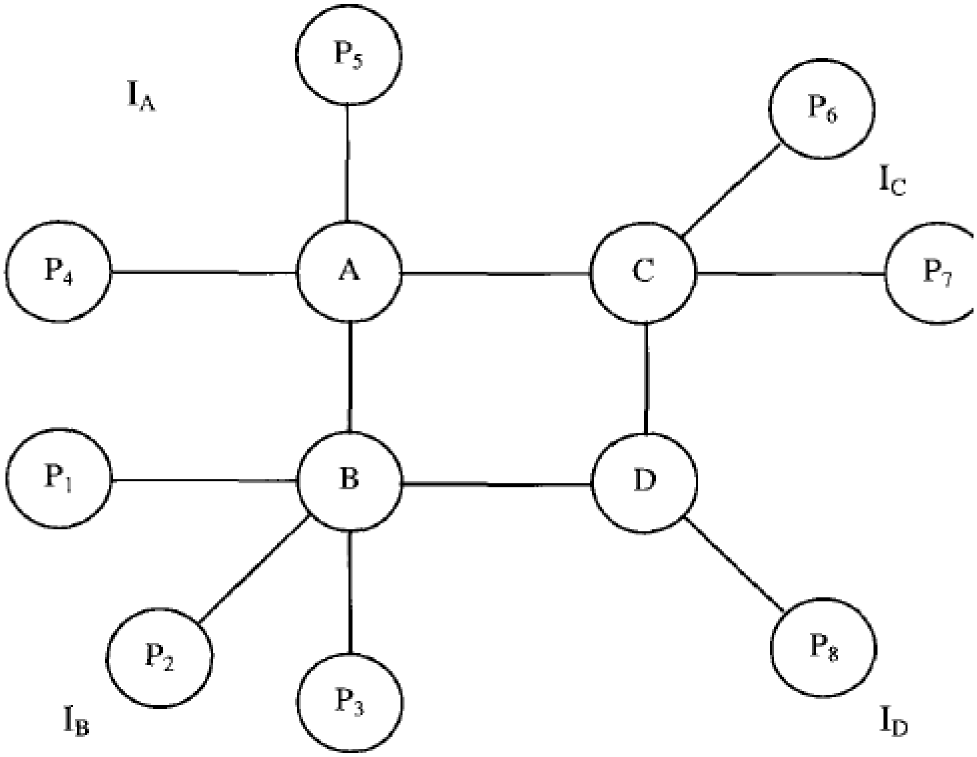
• **Partial dropping:** A node can also circumvent the watchdog by dropping packets at a lower rate than the watchdog's configured minimum misbehavior threshold. Although the watchdog will not detect this node as misbehaving, this node is forced to forward at the threshold bandwidth. This way, the watchdog serves to enforce this minimum bandwidth. For the watchdog to work properly, it must know where a packet should be in two hops.

4. Explain about the Octopus protocol?

The restrictive requirements of the Hypercube protocol – the necessity of *2d* group members - motivated Becker and Willie to develop a more flexible protocol. The resulting protocol, named Octopus allows an arbitrary number of group members to contribute to the group key. While minimizing the number of messages and rounds, the Octopus protocol has a desirable communication
complexity.

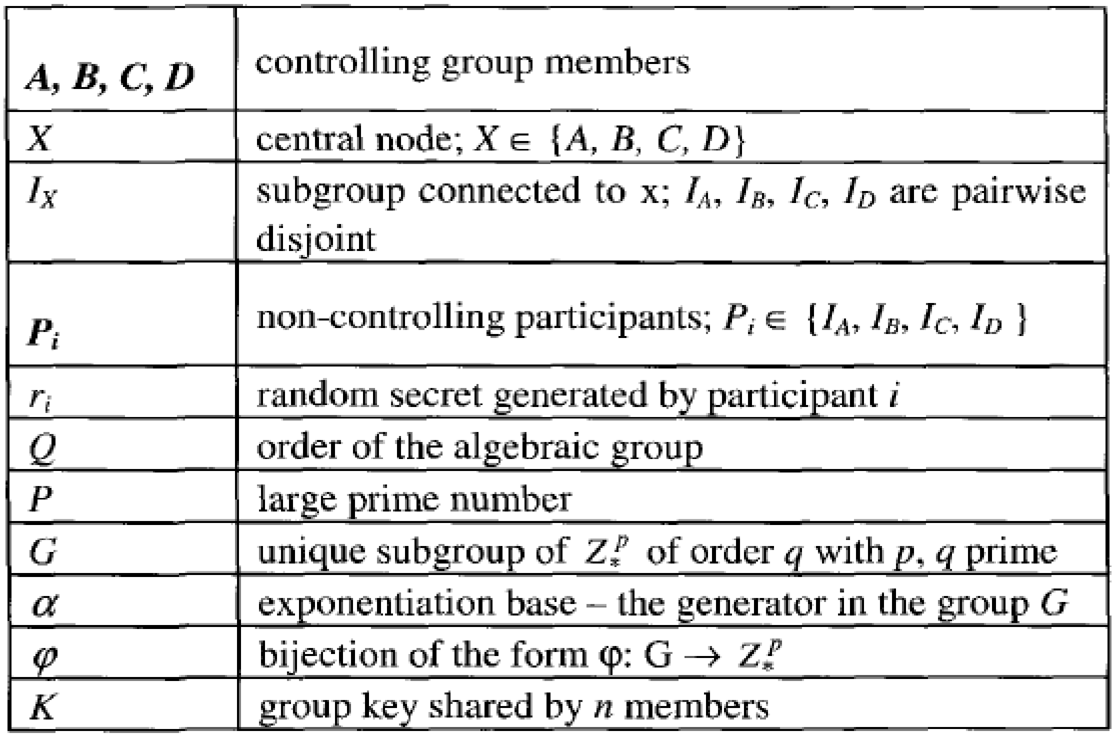
Design

The topology of the Octopus protocol loosely models an octopus: 4nodes comprise the core of the group and are positioned as a square; the remaining nodes are logically positioned as tentacles off the four nodes2.The four core nodes perform 2-Party Diffie-Hellman with each of the tentacles attached to it. Using the sub key generated from their tentacles,the four core nodes then perform the Hypercube protocol for *d - 2*. The key is then sent to all tentacles in the octopus. Figure 10.2 shows
an example of a topology in the Octopus protocol, with the
corresponding elements described next.



Topology of the octopus protocol

Notation



A, B, C, D	controlling group members
X	central node; $X \in \{A, B, C, D\}$
I_X	subgroup connected to x; I_A, I_B, I_C, I_D are pairwise disjoint
$\boldsymbol{P_i}$	non-controlling participants; $P_i \in \{I_A, I_B, I_C, I_D\}$
r_i	random secret generated by participant i
Q	order of the algebraic group
P	large prime number
G	unique subgroup of Z_*^p of order q with p, q prime
α	exponentiation base – the generator in the group G
φ	bijection of the form φ: G → Z_*^p
K	group key shared by n members

Setup

All participants elect controlling nodes, *A, B, C, D,* and the remaining nodes establish membership in some *Ix. A, B, C, D, Pt* then choose a cyclic group, *G,* of order *q,* and a generator, ***a*** in G; each member then chooses a secret share, r, e *G.*

Key Agreement

1. For all *X e A, B, C, D* and all ***i e*** *Ix, X* and P, perform 2-party Deffie-Hellman.

2. The controlling nodes, *A, B, C, D* perform the Hypercube protocol with *d - 2.* Each node uses the subkey generated during the previous steps as its secret for the Hypercube rounds of this step: *ra = K(IA), rb = K(IB), rc = K(IC), rd = K(ID).* Thus, after performing the Hypercube rounds, *A, B, C, D* hold the group key:

$$K = \alpha^{\varphi(\alpha^{K(I_A \cup I_B)})\cdot\varphi(\alpha^{K(I_C \cup I_D)})}$$

3. Each controlling node now sends the partial key to its subgroup:

A. For all $j \in I_A$:

i. A sends P_j: $\alpha^{\frac{K(I_A)\cup K(I_B)}{j}}$

ii. A sends P_j: $\alpha^{\varphi(\alpha^{K(I_C \cup I_D)})}$

iii. P_j calculates $(\alpha^{\frac{K(I_A \cup I_B)}{j}})^{\varphi^{(K_j)}} = \alpha^{K(I_A \cup I_B)}$

iv. P_j derives the group key:

$$K = (\alpha^{\varphi(\alpha^{K(I_C \cup I_D)})})^{\varphi(\alpha^{K(I_A \cup I_B)})}$$

B. For all $j \in I_D$:

i. D sends P_j: $\alpha^{\frac{K(I_C)\cup K(I_D)}{j}}$

ii. D sends P_j: $\alpha^{\varphi(\alpha^{K(I_A \cup I_B)})}$

iii. P_j calculates $(\alpha^{\frac{K(I_C \cup I_D)}{j}})^{\varphi^{(K_j)}} = \alpha^{K(I_C \cup I_D)}$

iv. P_j derives the group key:

$$K = (\alpha^{\varphi(\alpha^{K(I_A \cup I_B)})})^{\varphi(\alpha^{K(I_C \cup I_D)})}$$

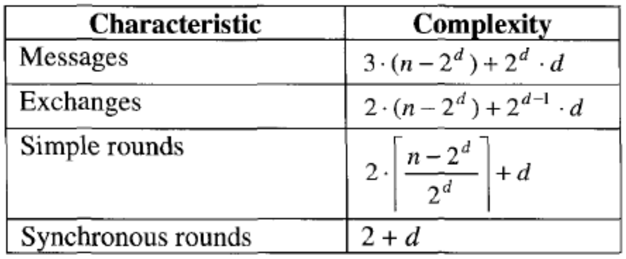
Complexity

The complexity of the Octopus protocol is shown in Table.

Analysis

The Octopus protocol inherits many of the performance benefits of the Hypercube protocol, while relaxing the restrictions on the size and the topology of the group.Although not explicitly mentioned, the logical structure of the Octopus protocol can be easily extended to allow group mutation: tentacles can simply be removed, and missing controlling nodes can be replaced by a tentacle. Nonetheless, the Octopus is quite sensitive to network failures and node movement. Thus, it is more suitable to static or low mobility ad hoc networks, and is not suitable for use in highly mobile environments.

Complexity of the Octopus protocol



Characteristic	Complexity
Messages	$3 \cdot (n - 2^d) + 2^d \cdot d$
Exchanges	$2 \cdot (n - 2^d) + 2^{d-1} \cdot d$
Simple rounds	$2 \cdot \left\lceil \frac{n - 2^d}{2^d} \right\rceil + d$
Synchronous rounds	$2 + d$

5.Explain about CLIQUES protocol?

The CLIQUES protocol remains one of the most effective and popular key management paradigms in use today. Unlike many of its predecessors, CLIQUES is more than simply a protocol: it is comprehensive key management paradigm, which includes well-defined group semantics, four key agreement protocols, an application programming interface (API), and an empirical analysis. In addition, CLIQUES has been specifically designed to accommodate dynamic and fault-prone group settings. That is, semantics for single and mass member join and leave, group merge and partition are included in the suite.The authors of CLIQUES were the first to formalize and prove the security of the GDH algorithm. Since its initial development, three additional variations of GDH have been developed: GDH.2, GDH.3, and STR. Each of these protocols is presented here.

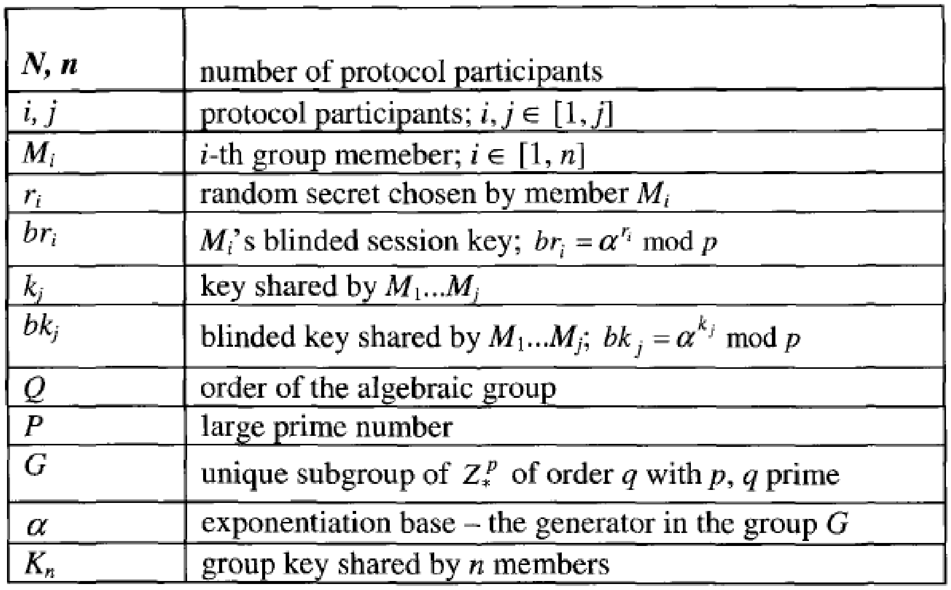
Design

Unlike the protocols discussed thus far, CLIQUES is largely event driven; that is, it uses membership events (e.g., join, leave, and merge) to trigger key regeneration. Because group events may occur concurrently, CLIQUES is designed above a synchronous networking layer. The well known Spread toolkit, which ensures total or causal ordering for broadcast messages, is employed at this layer.

CLIQUES use a group controller group mutation events - add, remove, merge, partition, etc. However, the group controller is not solely responsible for key generation. Although it is strongly discouraged, the controller may be a trusted third-party. An alternative that appears very attractive is to make the newest or oldest member of the group as the controller.

Notation

The following notation is common among all key agreement protocols in the CLIQUES suite - GDH.l (i.e., base GDH discussed earlier), GDH.2, GDH.3, STR.



N*, *n	number of protocol participants
i, j	protocol participants; $i, j \in [1, j]$
M_i	i-th group memeber; $i \in [1, n]$
r_i	random secret chosen by member M_i
br_i	M_i's blinded session key; $br_i = \alpha^{r_i} \bmod p$
k_j	key shared by $M_1 ... M_j$
bk_j	blinded key shared by $M_1 ... M_j$; $bk_j = \alpha^{k_j} \bmod p$
Q	order of the algebraic group
P	large prime number
G	unique subgroup of Z_*^p of order q with p, q prime
α	exponentiation base – the generator in the group G
K_n	group key shared by n members

7. Explain about *GDH.2 protocol?*

GDH.2 is a refinement of GDH.l, discussed earlier in this chapter.Like all CLIQUES protocols, GDH.2 consists of an upflow and downflow. *n* - 1 contributions are collected during the upflow stage; and

the resulting intermediate values are broadcast to the group during the downflow.

Setup

Members of the group select *p, q, G,* and *a.* In addition, each member is assigned a sequential identifier in *1..n. Mn* assumes the role of

the group controller.

Key Agreement

- Each *Mt* selects a random secret, r, **e** Z *. *i. Mi* receives the upflow, exponentiates each intermediate value, adds a new intermediate value that excludes its own contributions, updates the cardinal value, and forwards the upflow toMi+1:

$$M_i \xrightarrow{\alpha^{\frac{r_1 \cdots r_i}{r_j}} \mid j \in [1,i], \alpha^{r_1 \cdots r_i}} M_{i+1}$$

n. Mn receives the upflow from M,,_7, calculates the group key from the cardinal value, exponentiates all intermediate values, and broadcast the revised intermediate values to the group:

$$ALL \xleftarrow{\alpha^{\frac{r_1 \cdots r_n}{r_i}} \mid i \in [1,n], \alpha^{r_1 \cdots r_i}} M_n$$

Each member receives the broadcast message from *M,,,* extracts its intermediate value, and exponentiates it using its secret, r,, to calculate
the group key, *K:*

$$K = (\alpha^{r_1 \cdot r_2 \cdot \cdots \cdot r_{i-1} \cdot r_{i+1} \cdot \cdots \cdot r_{n-1} r_n})^{r_i}$$

Complexity

Table illustrates the complexity of GDH.2.

Complexity of GDH.2

Characteristic	Complexity
Messages	n
Rounds	n
Message size	$(n-1)(\frac{n}{2}+2)-1$
Exponentiations per member	$(i+1)-O(n)$
Total exponentiations	$\frac{(n+3)n}{2}-1$

8. Discuss about Wormhole Attack?

The wormhole attack is a severe threat against ad hoc routing protocols that is particularly challenging to detect and prevent. In a wormhole attack, a malicious node can record packets (or bits) at one location in the network and tunnel them to another location through a private network shared with a colluding malicious node. A dangerous threat can be perpetrated if a wormhole attacker tunnels all packets
through the wormhole honestly and reliably since no harm seems to be done: the attacker actually seems to provide a useful service in connecting the network more efficiently.

However, when an attacker forwards only routing control messages and not data packets, communication may be severely damaged. As an example, when used against an on-demand routing protocol such as DSR, a powerful application of the wormhole attack can be mounted by tunneling each RREQ message directly to the destination target node of the request. This attack prevents routes more than two hops long from being discovered, as RREP messages would arrive at the source faster than any other replies or, worse, RREQ messages arriving from other nodes next to the destination than the attacker would be discarded since already seen. *Temporal leashes* (using precise time synchronization) or *Geographical leashes* (using location information) are used to calculate delay bounds on a packet. These bounds when compared to actual values
can help in anomaly detection.

In some special cases, wormholes can also be detected through techniques that do not require precise time synchronization nor location information. As an example, it would be sufficient to modify the routing
protocol used to discover the path to a destination so that it could handle multiple routes: a verification mechanism would then detect anomalies when comparing the metric (e.g., number of hops) associated with each route. Any node advertising a path to a destination with a metric considerably lower than all the others could be branded as a suspect of a wormhole. Furthermore, if the wormhole attack is performed only on routing information while dropping data packets, other mechanisms can be used to detect this misbehavior. When a node does not correctly

participate in the network operation by not executing a particular function (e.g., packet forwarding), a collaborative monitoring technique can detect and
gradually isolate misbehaving nodes.

9. Discuss anomaly detection issues in IDS systems?

1 Detecting Abnormal Updates to Routing Tables

For ad hoc routing protocols, the primary concern is that false routing information generated and transmitted by a compromised node may be eventually used by other nodes in the network. Thus, a good
candidate for audit data would be the updates of routing information. A legitimate change in the routing table is caused by physical motion of the nodes or changes in the membership of the network. For a node, its own movement and the change in its own routing table are the only data it cantrust, and hence it is used as a basis of the trace data. The physical movement is measured by distance, direction and velocity. The routing table change is measured by Percentage of Changed Routes (PCR), and by the percentage changes in the sum of hops of all routes (PCH).Percentages are used as measurements as the number of nodes/routes is not fixed due to dynamic nature of the wireless ad hoc networks. During the "training" process, a wide variety of normal situations are simulated and the corresponding trace data is gathered for each node. The audit/trace data of all the nodes in the network are then merged together to obtain a set of all normal changes to the routing table for all nodes. The normal profile specifies the correlation of the physical movement of the node and the changes in the routing table. The classification algorithm classifies available trace data into ranges. For a particular trace data, if the PCR and/or PCR values are beyond the valid range for a particular movement (velocity, direction and distance), then it is
considered to be an anomalous situation and the necessary procedures are initiated.

2 Detecting Anomalous Activities in Other Layers

For medium access protocols, trace data could be in the form of total number of channel requests, the total number of nodes making those requests, etc., for last *s* seconds. The class can be the range of the current requests by a node. The classifier of the trace data describes the normal profile of a request. Anomaly detection model can then be computed on the basis of the deviation of the trace data from the normal profile. Similarly, the wireless application layer can use the service as the class and can contain the following features: for the past *s* seconds, the total number of requests to the same service, total number of services requested, the average duration of service, the number of nodes that requested service, the total number of service errors, etc. A classifier for each service then characterizes, for each service, a normal behavior for all requests.

10.Briefly discuss about the Hypercube protocol?

Overview

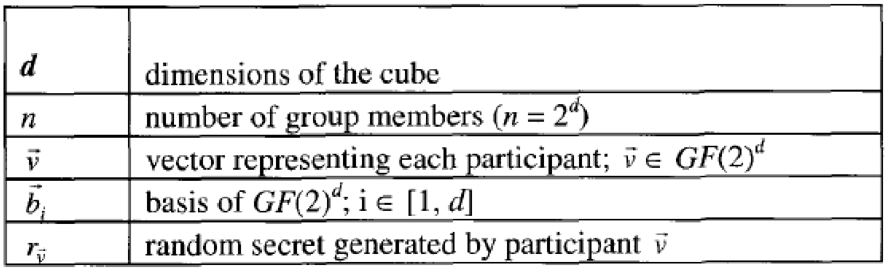
The high number of messages required by the Ingemarsson and Burmester protocols motivated Becker and Willie to define lower bounds on the communication complexity of the Diffie-Hellman based key

agreement protocols. To minimize communication overhead, they developed the Hypercube and Octopus protocols . The twoprotocols differ only in their logical arrangement of group members.

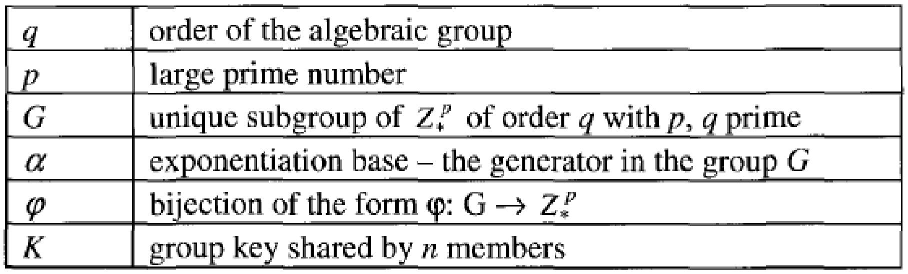
Design

The Hypercube Protocol requires *2d* participants, where *d* represents the dimensions of the cube. Each participant is logically positioned as a point in the cube; each edge of the cube depicts a key exchange. Because parallel edges represent exchanges that can be executed in parallel, a total of *d* rounds are required.

Notation



d	dimensions of the cube
n	number of group members ($n = 2^d$)
$\vec{v}$	vector representing each participant; $\vec{v} \in GF(2)^d$
$\vec{b}_i$	basis of $GF(2)^d$; i ∈ [1, d]
$r_{\vec{v}}$	random secret generated by participant $\vec{v}$



q	order of the algebraic group
p	large prime number
G	unique subgroup of Z_*^p of order q with p, q prime
α	exponentiation base – the generator in the group G
φ	bijection of the form φ: G → Z_*^p
K	group key shared by n members

Setup

Each participant in the J-dimensional vector space *GF{2)d* chooses a basis of *GF(2)d.* Each participant, v , then generates its secret, r;.

Key Agreement

1. Each participant, *v,* performs a 2-Party Diffie-Hellman key exchange with *bx.*

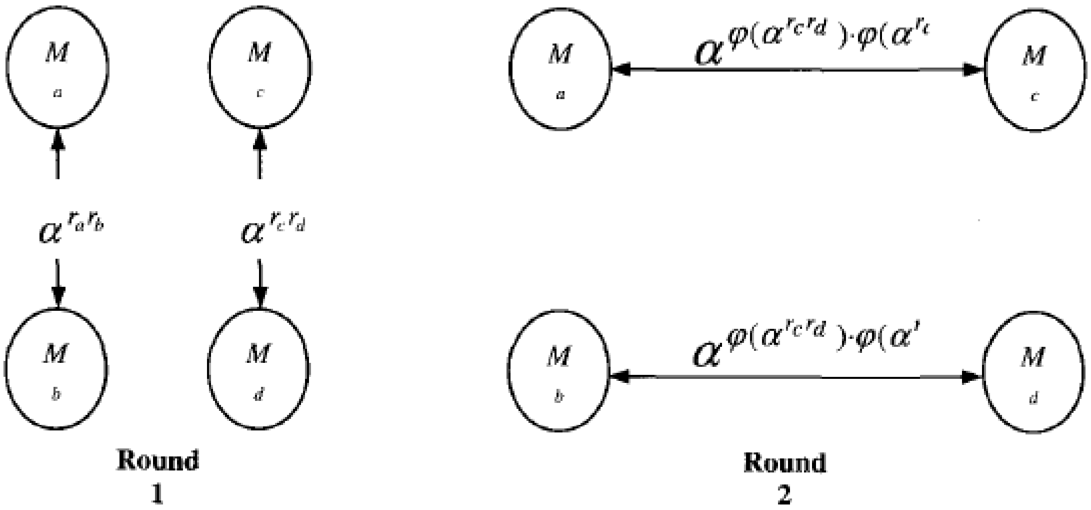
i. During the *f* round, each participant v performs a key exchange with v + *bt*,. The participant uses the result of the *i*-\ round as the secret.

The protocol terminates when *i* = *d,* since *bx,* ..., *bd* forms the basis of *GF(2)d.* Thus, every participant can generate the shared key:

$$K = \alpha^{\varphi(\alpha^{r_{\bar{v}_1} \cdot r_{\bar{v}_2}}) \cdot \varphi(\alpha^{r_{\bar{v}_3} \cdot r_{\bar{v}_4}}) \cdot \ldots \cdot \varphi(\alpha^{r_{\bar{v}_{d-1}} \cdot r_{\bar{v}_d}})}$$

Example

To clarify functioning of the protocol, examine the example shown in Figure .Note that there are 4 = 22 members, and, therefore, *d* = *2* rounds of the protocol.



M a
M c
$\alpha^{r_a r_b}$
$\alpha^{r_c r_d}$
M b
M d
Round 1
M a
$\alpha^{\varphi(\alpha^{r_c r_d})\cdot\varphi(\alpha^{r_c}}$
M c
M b
$\alpha^{\varphi(\alpha^{r_c r_d})\cdot\varphi(\alpha^{'}}$
M d
Round 2

Example of the Hypercube protocol *(d* = 2)

Complexity

Given the aforementioned characteristics, the complexity of the Hypercube protocol can be determined and is given in Table.

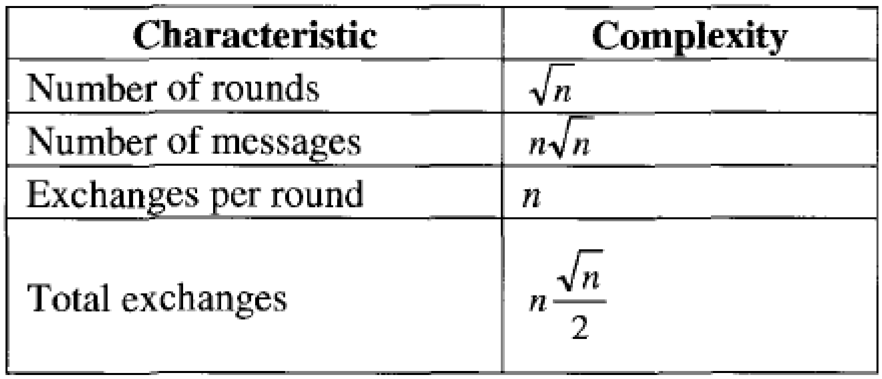
Analysis

The Hypercube protocol provides a reduction in the number of simple rounds if group members can be logically positioned as a cube.

Unlike many of the other protocols discussed in this chapter, the Hypercube protocol imposes the fact that group members can perform parallel key exchanges synchronously, which are difficult to achieve in

ad hoc networks. In addition, the Hypercube is extremely sensitive to network failures and member departure. For example, if a member moves during a key exchange, the entire cube must be reformed and the protocol restarted. Furthermore, if group semantics necessitate dynamic key management, whereby the key is regenerated whenever the state of the group changes, members must join and leave the group in powers of two so the cube structure can be maintained. Nonetheless, the Hypercube protocol provides an efficient, quite simple approach to key management if the group membership is static and the underlying transfer medium is reliable.

Complexity of the Hypercube protocol



Characteristic	Complexity
Number of rounds	$\sqrt{n}$
Number of messages	$n\sqrt{n}$
Exchanges per round	n
Total exchanges	$n\frac{\sqrt{n}}{2}$

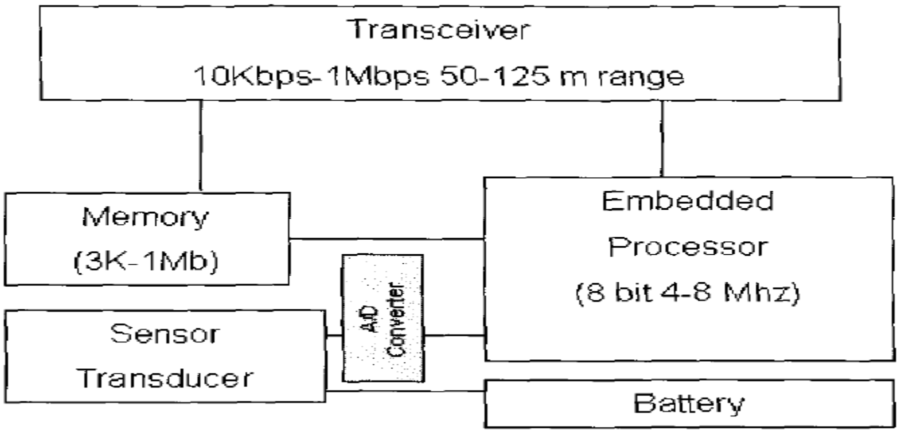
UNIT-III

1. Draw the block diagram of Sensor and explain

These networks combine wireless communication and minimal computation facilities with sensing of physical phenomenon which can be easily embedded in our physical environment. It is expected that the size of a sensor will be a few cubic millimeters, the target price range less than one US dollar, including radio front end, microcontroller, power supply and the actual sensor. All these components together in a single device form a so-called sensor node. A sensor has many functional components as shown in below Figure.

Due to lack of a better word, a typical sensor consists of a transducer to sense a given physical quantity with a predefined precision, an embedded processor for local processing, small memory unit for storage of data and a wireless transceiver to transmit or receive data and all these devices run on the power supplied by an attached battery. It is interesting to note that precise specifications of various components illustrated in Figure 1, may depend on the type of application in hand, but the basic characteristics are essentially present to fulfill desired application functionalities. There are few integrated sensors commercially available [Hill2004] and can be used directly as plug-

and-play unit to monitor and control some specific physical parameters as decided by the user. But, there are many basic sensors



Block diagram of typical Sensor Node

2. Discuss in brief about mica mote sensor node

MICA mote is a commercially available product that has been used widely by researchers and developers. It has all of the typical features of a mote and therefore can help you understand what this technology makes possible today. MICA motes are available to the general public through a company called Crossbow. These motes come in two form factors:

- Rectangular, measuring 2.25 x 1.25 by 0.25 inches (5.7 x 3.18 x.64 centimeters), it is sized to fit on top of two AA batteries that provide it with power.
- Circular, measuring 1.0 by 0.25 inches (2.5 x .64 centimeters), it is sized to fit on top of a 3 volt button cell battery.

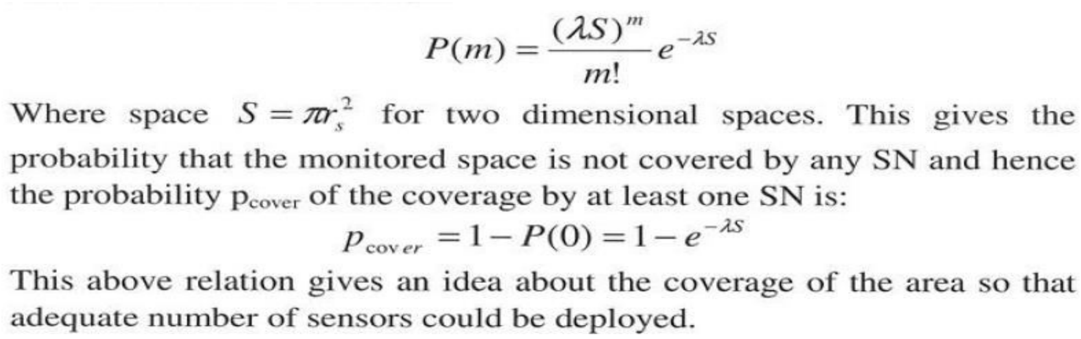
The MICA mote uses an Atmel ATmega 128L processor running at 4 megahertz. The 128L is an 8-bit microcontroller that has 128 kilobytes of onboard flash memory to store the mote's program. This CPU is about as powerful as the 8088 CPU found in the original IBM PC (circa 1982). The big difference is that the ATmega consumes only 8 milliamps when it is running, and only 15 microamps in sleep mode

3. How sensing and communication range is computed for sensor nodes

A wireless sensor network (WSN) consists of a large number of sensor nodes (SNs) and exploring their best possible use is a challenging problem. As the main objective of a SN is to monitor some physical quantity in a given area, the sensors need to be deployed with adequate density so that sensing of the complete area can be done, without leaving any void or unsensed area.

In other words, given the sensing range of each SN and the area to be covered, adequate number of SNs should be needed to be placed throughout the area so that no corner is left out. The SNs can be placed deterministically at pre-specified locations or could be distributed randomly. So, if N SNs are put in an area A=LxL, then the SNs density can be given by ʎ — N/A.

The sensing range rs of each sensor can be shown in Figure. As illustrated in Figure below, each SN has its own sensing range and to cover the whole space, adjacent SNs need to be located close to each other and at the most at a distance of 2r s from each other as illustrated by, three positions of SN2, SN2 and SN2 with different overlapping areas between SN1 and SN2. If the SNs are uniformly distributed with the node density of ʎ, the probability that there are m SNs within the space of S is Poisson distributed as

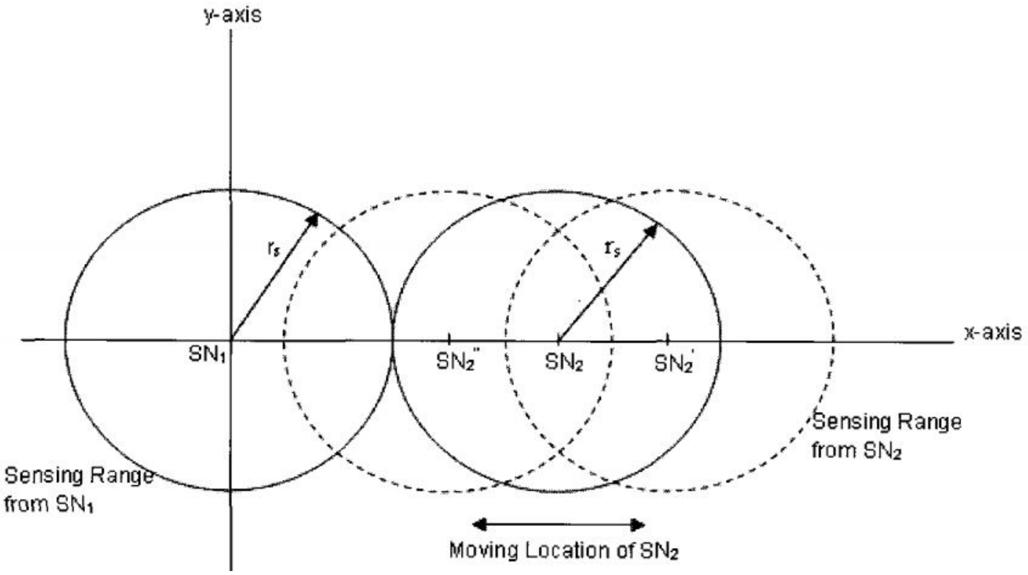


$$P(m) = \frac{(\lambda S)^m}{m!} e^{-\lambda S}$$

Where space $S = \pi r_s^2$ for two dimensional spaces. This gives the probability that the monitored space is not covered by any SN and hence the probability p_{cover} of the coverage by at least one SN is:

$$p_{cover} = 1 - P(0) = 1 - e^{-\lambda S}$$

This above relation gives an idea about the coverage of the area so that adequate number of sensors could be deployed.


y-axis
x-axis
r_s
r_s
SN_1
SN_2''
SN_2
SN_2'
Sensing Range from SN_1
Sensing Range from SN_2
Moving Location of SN_2

Transmission between adjacent SNs using the wireless transceiver and is feasible only when there is at least one SN within the communication range of each SN. Therefore, not just the sensing coverage, but the communication connectivity is at least equally important so that sensor data could be received by other SNs. It may be noted that the data from a single SN is not adequate to make any useful decision [Dasgupta2003] and data need to be collected from a set of SNs in arriving at an intelligent interpretation.

So, the real question is how far away the SNs can be located. As illustrated in Figure 8.4, two SNs can be separated by 2rs distance from sensing coverage point of view. Considering this to be an allowed maximum distance between two adjacent sensors SN] and SN2. To

enable a data transfer between them, the minimum allowed communication distance should be 2rs. This implies that the wireless communication coverage of a sensor must be at least twice the sensing

distance [Zhang2005] and is the minimum distance to ensure connectivity between sensing devices to communicate with each other.

4. Why Routing protocol design for WSNs is heavily influenced by many challenging factors

Ad hoc deployment - Sensor nodes are randomly deployed which requires that the system be able to cope up with the resultant distribution and form connections between the nodes. In addition, the system should be adaptive to changes in network connectivity as result of node failure.

- **Computational capabilities** - Sensor nodes have limited computing power and therefore may not be able to run sophisticated network protocols leading to light weighted and simple versions of routing protocols.

- **Energy consumption without losing accuracy** - Sensor nodes can use up their limited energy supply carrying out computations and transmitting information in a wireless environment. As such, energyconserving forms of communication and computation are crucial as the node lifetime shows a strong dependence on the battery lifetime. In a multi-hop WSN, nodes play a dual role as data sender and data router. Therefore, malfunctioning of some sensor nodes due to power failure can cause significant topological changes and might require rerouting of packets and reorganization of the network.

- **Scalability** - The number of sensor nodes deployed in the sensing area may be in the order of hundreds, thousands, or more. Any routing scheme must be scalable enough to respond to events and capable of operating with such large number of sensor nodes. Most of the sensors can remain in the sleep state until an event occurs, with data from only a few remaining sensors providing a coarse quality.

- **Communication range** - The bandwidth of the wireless links connecting sensor nodes is often limited, hence constraining intersensor communication. Moreover, limitations on energy forces sensor nodes to have short transmission ranges. Therefore, it is likely that a path from a source to a destination consists of multiple wireless hops.

- **Fault tolerance** - Some sensor nodes may fail or be blocked due to lack of power, physical damage, or environmental interference. If many nodes fail, MAC and routing protocols must accommodate formation of new links and routes to the data collection BSs. This may require actively adjusting transmit powers and signaling rates on the existing links to reduce energy consumption, or rerouting packets through regions of the network where more energy is available. Therefore, multiple levels of redundancy may be needed in a faulttolerant WSN.

- **Connectivity** - High node density in sensor networks precludes them from being completely isolated from each other. Therefore, sensor nodes are expected to be highly connected. This, however, may not prevent the network topology from varying and the network size from shrinking due to sensor nodes failures. In addition, connectivity depends on the, possibly random, distribution of nodes.

- **Transmission media** - In a multi-hop sensor network, communicating nodes are linked by a wireless medium. Therefore, the traditional problems associated with a wireless channel (e.g., fading, high error rate) also affect the operation of the sensor network. In general, bandwidth requirements of sensor applications will be low, in the order of 1-100 kb/s. As we have seen in Chapters 4 and 5 and in the previous section, the design of the MAC protocol is also critical in terms of conserving energy in WSNs.

- **QoS** - In some applications (e.g., some military applications), the data should be delivered within a certain period of time from the moment it is sensed, otherwise the data will be useless. Therefore, bounded latency for data delivery is another condition for time constrained applications

- **Control Overhead** - When the number of retransmissions in wireless medium increases due to collisions, the latency and energy consumption also increases. Hence, control packet overhead increases linearly with the node density. As a result, tradeoffs between energy conservation, self-configuration, and latency may exist.

• **Security** - Security is an important issue which does not mean physical security, but it implies that both authentication and encryption should be feasible. But, with limited resources,

implementation of any complex algorithm needs to be avoided. Thus, a tradeoff exists between the security level and energy consumption in a WSN.

5. "An ideal sensor network should have some additional features" Discuss

- **Attribute-based addressing**. This is typically employed in sensor networks where addresses are composed of a group of attribute-value pairs which specify certain physical parameters to be sensed. For example, an attribute address may be (temperature > 35°C, location = "Recife"). So, all sensor nodes located in "Recife" which sense a temperature greater than 35°C should respond;
- **Location awareness** is another important issue. Since most data collection is based on location, it is desirable that the nodes know their position whenever needed;Another important requirement in some cases is that the sensors should **react immediately** to drastic changes in their environment, for example, in time-critical applications. The end user hould be made aware of any drastic deviation in the situation with minimum delay, while making efficient use of the limited wireless channel bandwidth and sensor energy;
- **Query Handling** is another important feature. Users should be able to request data from the network through some base station (also known as sink) or through any of the nodes, whichever is closer. So, there should be a reliable mechanism to transmit the query to appropriate nodes which can respond to the query. The answer should then be re-routed back to the user as quickly as possible.

6. Explain the advantages of WSN over wired ones

• **Ease of deployment** - These wireless sensors can be deployed (dropped from a plane or placed in a factory) at the site of interest without any prior organization, thus reducing the installation cost and time, and also increasing the flexibility of deployment;

• **Extended range** - One huge wired sensor (macro-sensor) can be replaced by many smaller wireless sensors for the same cost. Such a macro-sensor can sense only a limited region whereas a network of smaller sensors can be distributed over a wider range;

• **Fault tolerant** -With macro-sensors, the failure of one node makes that area completely unmonitored till it is replaced. With wireless sensors, failure of one node does not affect the network operation substantially as there are other adjacent nodes collecting similar data. At most, the accuracy of data collected may be somewhat reduced

• **Mobility** - Since these wireless sensors are equipped with battery, they can possess limited mobility (e.g., if placed on robots). Thus, if a region becomes unmonitored we can have the nodes rearrange them selves to distribute evenly, i.e., these nodes can be made to move towards area of interest but having lower mobility as compared to ad hoc networks.

7. What are the applications of Wireless Sensor Networks and explain briefly

There are lots of applications of WSN:

1. Process Management: Area monitoring is a very common using WSNs. In area monitoring, the WSN is deployed spanning a region where some phenomenon is usually to be monitored. A military example may be the use of sensors detect enemy intrusion; a civilian example would be the geo-fencing of gas or oil pipelines. Area monitoring is most important part.

2. Healthcare monitoring: The medical applications might be of two sorts: wearable and implanted. Wearable devices are applied to the body surface of the human or maybe at close proximity from the user. The implantable medical devices are the ones that are inserted inside your body. There are numerous other applications too e.g. body position measurement and of the person, overall monitoring of ill patients in hospitals and also at homes.

3. Environmental/Earth sensing: There are numerous applications in monitoring environmental parameters samples of which are given below. They share any additional challenges of harsh environments and reduced power supply.

4. Polluting of the environment monitoring: Wireless sensor networks have been deployed in lots of cities (Stockholm, London and Brisbane) to monitor the power of dangerous gases for citizens. These can leverage the random wireless links instead of wired installations that also make them more mobile for testing readings in several areas.

5. Forest fire detection: A network of Sensor Nodes is usually positioned in a forest to detect every time a fire has begun. The nodes is usually with sensors to measure temperature, humidity and gases which are produced by fire within the trees or vegetation. The first detection is necessary to get a successful action of the fire fighters; As a result of Wireless as Sensor Networks, the fire brigade are able to know when a fire begins you bet it can be spreading.

6. Landslide detection: A landslide detection system uses a wireless sensor network to detect the slight movements of soil and modifications to various parameters that will occur before or throughout a landslide. With the data gathered it may be possible to know the appearance of landslides before it genuinely happens.

8. Why Traditional routing protocols defined for MANETs are not well suited for wireless sensor networks due to the following reasons

As we mentioned earlier, wireless sensor networks are "data centric",where data is requested based on particular criteria such as "which area has temperature 35°C";

• In traditional wired and wireless networks, each node is given a unique identification (e.g., an IP address) used for routing. This cannot be effectively used in sensor networks because, being data centric, routing to and from specific nodes in these networks is not required;

• Adjacent nodes may have similar data. So, rather than sending data separately from each sensor node to the requesting node, it is desirable to aggregate similar data before sending it;

• The requirements of the network change with the application and hence, it is application-specific. For example, in some applications, the sensor nodes are fixed and not mobile while others may need data based only on some selected attributes (viz., attribute is fixed in this network).

9. How Energy Consumption is done in Sensor Nodes

Minimizing the energy consumption of WSs is critical yet a challenge for the design of WSNs. The energy consumption in WSN involves three different components: Sensing

Unit (Sensing transducer and A/D Converter), Communication Unit (transmission and receiver radio), and Computing/Processing Unit. In order to conserve energy, we may make some SNs go to sleep mode and need to consider energy consumed in that state.

Sensing transducer is responsible for capturing the physical parameters of the environment. Its basic function is to do physical signal sampling and convert into electrical signals. The energy consumption of this part depends on the hardware and the application and sensing energy is only a small part of the total energy consumption.

A/D Converter: Based on paper [www.moteiv.com], an AD Converter for sensor consumes only 3.1µW , in 31 pJ/8-bit sample at 1 Volt supply. The standby power consumption at 1 V supply is 41pW. Assuming the D/C is not noise limited, the lower bound on energy per sample for the successive approximation architecture is roughly computed as: E_{min}-$C_{total}V_{ref}$, where C_{total} is the total capacitance of the array including the bottom plate parasites, and V_{ref} is a common mode input voltage the comparator works under.

Transmission Energy: Based on the transmission energy transmits a k-bit message to distance d can be computed as:

E_{Tx}(k,d)=$E_{Tx\text{-}elec}$(k)+ETx.amp(k,d)=Eetec*k+ £ *k*d2, where $E_{Tx\text{-}elec}$ is the transmission electronics energy consumption, Erx.amp is the transmit amplifier energy consumption.

10. Discuss briefly the different types of WSNs

A WSN is deployed primarily to collect sensed data by different WSs and it is critical to see how frequently the sensed values are collected. Looking at various ways in which one can employ the network resources, WSNs can be classified on the basis of their mode of operation or functionality, and the type of target applications.

Accordingly, we hereby classify WSNs into three types:

- Proactive Networks - The nodes in this network periodically switch on their sensors and transmitters, sense the environment and transmit the data of interest. Thus, they provide a snapshot of the relevant parameters at regular intervals and are well suited for applications requiring periodic data monitoring.

- Reactive Networks - In this scheme, the nodes react immediately to sudden and drastic changes in the value of a sensed attribute. As such, these are well suited for time critical applications.

- Hybrid Networks - This is a combination of both proactive and reactive networks where sensor nodes not only send data periodically, but also respond to sudden changes in attribute values

11. What are the devices present in the typical hardware platform of a wireless sensor nodes?

Due to the principle differences in application scenarios and underlying communication technology, the architecture of WSNs will be drastically different both with respect to a single WS and the network as a whole. The typical hardware platform of a wireless sensor node will consist of:

- **Simple embedded microcontrollers**, such as the Atmel or the Texas Instruments MSP 430. A decisive characteristic here is, apart from the critical power consumption, an answer to the important question whether and how these microcontrollers can be put into various operational and sleep modes, how many of these sleep modes exist, how long it takes and how much energy it costs to switch between these modes. Also, the required chip size and computational power and on-chip memory are important;

- Currently used **radio transceivers** include the RFM TR1001 or Infineon or Chipcon devices; similar radio modems are available from various manufacturers. Typically, ASK or FSK is used, while the Berkeley PicoNodes employ OOK modulation. Radio concepts like ultra-wideband are in an advanced stage. A crucial step forward would be the introduction of a reasonably working wake-up radio concept, which could either wake up all SNs in the vicinity of a sender or even only some directly addressed nodes. A wake-up radio allows a SN to sleep and to be wakened up by suitable transmissions from other nodes, using only a low-power detection circuit. Transmission media other than radio communication are also considered, e.g., optical communication or ultrasound for underwater-applications. However, this largely depends on the application;

- **Batteries** provide the required energy. An important concern is battery management and whether and how energy scavenging can be done to recharge batteries in the field. Also, self-discharge rates, selfrecharge rates and lifetime of batteries can be an issue, depending on the application;

- **The operating system and the run-time environment** is a hotly debated issue in the literature. On one hand, minimal memory footprint and execution overhead are required while on the other, flexible means of combining protocol building blocks are necessary, as meta information has to be used in many places in a protocol stack (e.g., information about location, received signal strength, etc., has an influence on many different protocol functions). Consequently, we believe that structures like blackboards, publish/subscribe or tuplespaces are an interesting starting point for the run-time environments for such SNs.

12. Explain briefly the different aspects are to be taken into account while developing a network architecture

The network architecture as a whole has to take various aspects into account including:

• The protocol architecture has to take both application- and energydriven point of view;

• QoS, dependability, redundancy and imprecision in sensor readings have to be considered;

• The addressing structures in WSNs are likely to be quite different:scalability and energy requirements can demand an "address-free structure". Distributed assignments of addresses can be a key technique, even if these addresses are only unique in a two-hop neighborhood. Also, geographic and data-centric addressing structures are required;

• A crucial and defining property of WSNs will be the need for andtheir capability to perform in-network processing. This pertains to aggregation of data when multiple sensor readings are convergecasted to a single or multiple sinks, distributed signal processing, and the exploitation of correlation structures in the sensor readings in both time and space. In addition, aggregating data reduces the number of transmitted packets;

• Based on such in-network processing, the service that a WSN offers at the level of an entire network is still an ill-defined concept. It is certainly not the transportation of bits from one place to another, but any simple definition of a WSN service ("provides readings of environmental values upon request", etc.) is incapable of capturing all possible application scenarios;

• As these services are, partially and eventually, invoked by agents outside the system, a gateway concept is required: How to structure the integration of WSNs into larger networks, where to bridge the different communication protocols (starting from physical layer upwards) are open issues;

- More specifically, integration of such ill-defined services in middleware architectures like CORBA [CORBAwww] or into web services is also not clear: how to describe a WSN service such that it can be accessed via a Web Service Description Language (WSDL) [WSDLwww] and Universal Description, Discovery and Integration (UDDI) [UDDIwww] description?;

- Other options could be working with non-standard networking architectures, e.g., the user of agents that "wander" around a given network and explore the tomography or the "topology" of the sensed values; and

- From time to time, it might be necessary to reassign tasks to the WSN, i.e., to provide all its SNs with new tasks and new operating software.

13. Explain briefly the types of schemes available to allocate a single broadcast channel among competing nodes

There are two types of schemes available to allocate a single broadcast channel among competing nodes:

Static Channel Allocation and Dynamic Channel Allocation.

- **Static Channel Allocation**: In this category of protocols, if there are N SNs, the bandwidth is divided into N equal portions in frequency (FDMA), in time (TDMA), in code (CDMA), in space (SDMA) or in schemes such as OFDM or ultra-wideband. Since each SN is assigned a private portion, there is no or minimal interference amongst multiple SNs. These protocols work very well when there are only a small and fixed number of SNs, each of which has buffered (heavy) load of data;

- **Dynamic Channel Allocation**: In this category of protocols, there is no fixed assignment of bandwidth. When the number of active SNs changes dynamically and data becomes bursty at arbitrary SNs, it is most advisable to use dynamic channel allocation scheme. These are contention-based schemes, where SNs contend for the channel when they have data while minimizing collisions with other SNs transmissions. When there is a collision, the SNs are forced to retransmit data, thus leading to increased wastage of energy and unbounded delay. Example protocols are: CSMA (persistent and non-persistent), MACAW

14. Explain in detail the important issues involved in the design of MAC protocols for WSNs

Design Issues

As with MAC protocols for traditional MANETs, WSNs have their own inherent characteristics that need to be addressed. Below we discuss some of the most important ones involved in the design of MAC protocols for WSNs.

Coping up with Node Failure When many SNs have failed, the MAC and routing protocols must accommodate formation of new links and routes to other SNs and the BS. This may require dynamically adjusting transmit powers and signaling rates on the existing links, or rerouting packets through regions of the network with higher energy level.

Sources of Resource Consumption at the MAC Layer

There are several aspects of a traditional MAC protocol that have negative impact on wireless sensor networks including:

- Collisions - When a transmitted packet is corrupted due to a collision, it has to be discarded. The follow-on retransmission increases the energy consumption and hence increases the latency;

- Overhearing - SNs listen to transmissions that are destined to other SNs;

- Control packets overhead - Sending and receiving control packets consume energy and reduce the payload. This overhead increases linearly with node density. Moreover, as more SNs fail in the network, more control messages are required to self configure the system, resulting in more energy consumption;

- Idle Listening -Waiting to receive anticipated traffic that is never sent. This is especially true in many sensor network applications. If nothing is sensed, SNs are in the idle mode for most of the time.

Measures to Reduce Energy Consumption

One of the most cited methods to conserve energy in sensor networks is to avoid listening to idle channels, that is, neighboring nodes periodically sleep (radio off) and auto synchronize as per sleep schedule. It is important to note that fairness, latency, throughput and bandwidth utilization are secondary in the WSNs.

Comparison of Scheduling & Reservation-based and Contention-based MAC Design

One approach of MAC design for WSNs is based on reservation and scheduling, for example TDMA-based protocols that conserve more energy as compared to contention-based protocols like the DCF. This is because the duty cycle of the radio is increased and there is no contention-introduced overhead and collisions. However, formation of cluster, management of inter-cluster communication, and dynamic adaptation of the TDMA protocol to variation in the number of nodes in the cluster in terms of its frame length and time slot assignment are still the key challenges.

15. Explain in detail the sensor MAC protocol

The Sensor-MAC (S-MAC) protocol explores design trade-offs for energy-conservation in the MAC layer. It reduces the radio energy consumption from the following sources: collision, control overhead, overhearing unnecessary traffic, and idle listening.

The basic scheme of S-MAC is to put all SNs into a low-duty-cycle mode -listen and sleep periodically. When SNs are listening, they follow a contention rule to access the medium, which is similar to DCR. In S-MAC, SNs exchange and coordinate on their sleep schedules rather than randomly sleep on their own. Before each SN starts the periodic sleep, it needs to choose a schedule and broadcast it to its neighbors.

To prevent long-term clock drift, each SN periodically broadcasts its schedule as the SYNC packet. To reduce control overhead and simplify broadcasting, S-MAC encourages neighboring SNs to choose the same schedule, but it is not a requirement. A SN first listens for a fixed amount of time, which is at least the period for sending a SYNC packet.

If it receives a SYNC packet from any neighbor, it will follow that schedule by setting its own schedule to be the same. Otherwise, the SN will choose an independent schedule after the initial listening period. It is possible that two neighboring SNs have two different schedules. If they are aware of each other's schedules, they have two options:

- Following two schedules by listening at both scheduled listen time;
- Only following its own schedule, but transmitting twice as per both schedules when broadcasting a packet.

In some cases the two SNs may not be aware of the existence of each other, if their listen intervals do not overlap at all. To solve the problem, S-MAC let each SN periodically perform neighbor discovery, i.e., listening for the entire SYNC period, to find unknown neighbors on a different schedule.

Below Figure depicts the low-duty-cycle operation of each SN. The listen interval is divided into two parts for both SYNC and data packets. There is a contention window for randomized carrier sense time before sending each SYNC or data (RTS or broadcast) packet. For example, it SN A wants to send a unicast packet to SN B, it first perform carrier sense during B's listen time for data. If carrier sense indicates an idle channel, node A will send RTS to node B, and B will reply with CTS if it is ready to receive data. After that, they will use the normal sleep time to transmit and receive actual data packets.

Broadcast does not use RTS/CTS due to the potential collisions on multiple CTS replies. Low-duty-cycle operation reduces energy consumption at the cost of increased latency, since a node can only start sending when the intended receiver is listening. S-MAC developed an adaptive listen scheme to reduce the latency in a multi-hop transmission.

The basic idea is to let the node who overhears its neighbor's transmissions (ideally only RTS or CTS) wake up for a short period of time at the end of the transmission. In this way, if the SN is the next-hop node, its neighbor is able to immediately pass the data to it, instead of waiting for its scheduled listen time. If the SN does not receive anything during the adaptive listening, it can go back to sleep mode.

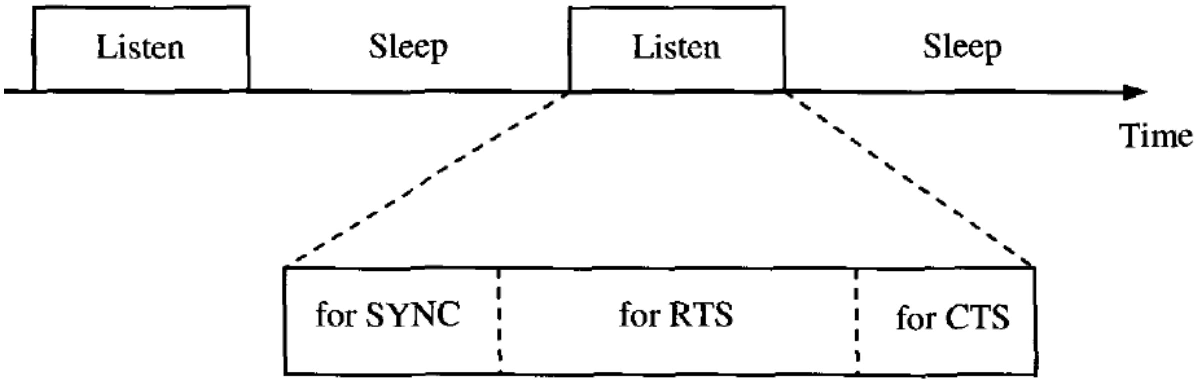


Figure :Low-duty-cycle operation in S-MAC

16. Describe in detail about self organizing MAC for sensor networks.

The SMACS is an infrastructure-building protocol that forms a flat topology (as opposed to a cluster hierarchy) for sensor networks. SMACS is a distributed protocol which enables a collection of SNs to discover their neighbors and establish transmission/reception schedules
for communicating with them without the need for any local or global master nodes. In order to achieve this ease of formation, SMACS combines the neighbor discovery and channel assignment phases. Unlike methods such as the Linked Clustering Algorithm (LCA) in
which a first pass is performed on the entire network to discover neighbors and then another pass done to assign channels, or TDMA slots, to links between neighboring nodes, SMACS assigns a channel to a link immediately after the link's existence is discovered. This way, links begin to form concurrently throughout the network. By the time all nodes hear all their neighbors, they would have formed a connected network. In a connected network, there exists at least one multi-hop path between any two distinct nodes. Since only partial information about radio connectivity in the vicinity of a SN is used to assign time intervals to links, there is a potential for time collisions with slots

assigned to adjacent links whose existence is not known at the time of channel assignment. To reduce the likelihood of collisions, each link is required to operate on a different frequency. This nfrequency band is chosen at random from a large pool of possible choices when the links are formed. The topology of Figure: Here, nodes A and D wake up at times *Ta* and *Td.* After they find each other, they agree to transmit and receive during a pair of fixed time slots.

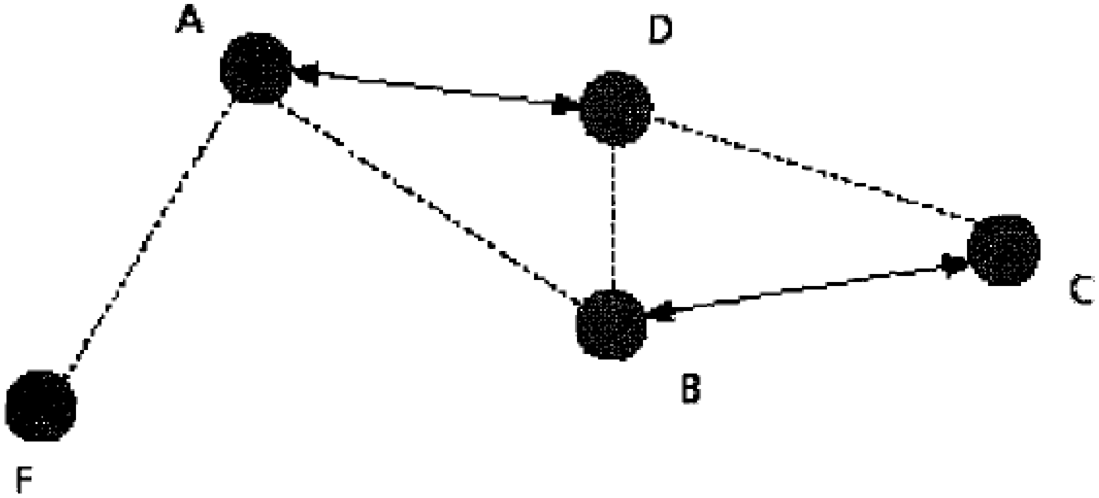


Figure: Network Topoloy

17. Explain in detail about eaves-drop and register protocol

Mobility can be introduced into a WSN as extensions to the stationary sensor network. Mobile connections are very useful to a WSN and can arise in many scenarios where either energy or bandwidth is a major concern. Where there is the constraint of limited power consumption, small, low bit rate data packets can be exchanged to relay data to and from the network whenever necessary.

At the same time, it cannot be assumed that each mobile node is aware of the global network state and/or node positions. Similarly, a mobile node may not be able to complete its task (data collection, network instruction, information extraction) while remaining motionless.

Thus, the EAR protocol attempts to offer continuous service to these mobile nodes under both mobile and stationary constraints. EAR is a low-power protocol that allows for operations to continue within the stationary network while intervening at desired moments for information exchange.

As battery power is the primary concern of stationary nodes, the communication channels between the mobile and stationary sensors must be established with as few messages as possible.

This can be accomplished by allowing the mobile node to determine when to invite/drop the stationary node for a connection. The network is primarily assumed to consist of stationary nodes, with few mobile randomly distributed nodes.

Such an assumption leads to the notion that only a few selected stationary sensors will be within the vicinity of a mobile sensor at any given time. To avoid unnecessary consumption of energy associated with lost messages, the mobile nodes assume full control of the connection process.

Furthermore, the overhead associated with acknowledgments can be eliminated as the proximity between SNs is adequate to ensure message reception. To avoid connection handoff, the mobile sensor keeps a registry of the surrounding SNs, selecting a new connection only when absolutely necessary.

Since there will be few stationary nodes aware of the presence of the mobile nodes, the EAR protocol could be transparent to the existing stationary protocol. Also, by placing the mobile MAC protocol in the background, very few specialized messages are needed to establish or drop connections.

EAR assumes that stationary nodes use a TDMA-like frame structure, within which slots are designated for inviting neighboring nodes into the network. This message need not occur in every epoch of the TDMA structure; it is only needed at some semi regular interval, and serves as the "pilot signal" for the mobile SNs.

Since the stationary node does not require a response to this message (although it waits for a predetermined time for a response), the mobile node is simply "eavesdropping" on the control signals of the stationary MAC protocol. It then decides the best course of action regarding the transmitting stationary sensor; hence, this invitation message acts as the trigger for the EAR algorithm.

In order to keep a constant record of neighboring activity, the mobile node forms a registry of neighbors. This registry holds only the required information for forming, maintaining, and tearing down connections. From the transmitted invitation message, the mobile can extract the received signal-to-noise ratio (SNR), node ID, transmitted power, and so on.

Making or breaking a connection is based on the status of connections, as well as the location and mobility information inferred from the entries in the registry. The stationary node will simply register mobile sensors that have formed connections and remove them when the link is broken, effectively limiting participation in the connection procedures. To design a system in which the mobile assumes full responsibility for making and breaking connections, an appropriate signaling method must be defined.

The EAR algorithm employs the following four primary messages:

• **Broadcast Invite (BI)** - This is used by the stationary node to invite other nodes to join. If multiple Bis are received, the mobile node continues to register every stationary node encountered, until its registry becomes full;

• **Mobile Invite (MI)** - This message is sent by the mobile node in response to the BI message from the stationary node for establishing connection.

- **Mobile Response (MR)** - This is sent by the stationary node in response to a MI message, and indicates the acceptance of the MI request. This causes the stationary node to select slots along the TDMA frame for communication. In addition, the stationary node will enter the mobile node in its own registry;

- **Mobile Disconnect (MD)** - With this message, the mobile node informs the stationary node of a disconnection. For energy saving purposes, no response is needed from the stationary node. The decision to send a MD message is usually based on the SNR value.

18. Describe in detail about STEM Protocol

The Sparse Topology and Energy Management (STEM) protocol is based on the assumption that most of the time the sensor network is only sensing the environment, waiting for an event to happen. In other words, STEM may be seen as better suitable for reactive sensor networks where the network is in the monitoring state for vast majority of time.

One example of such application is a sensor network designed to detect fires in a forest. These networks have to remain operational for months or years, but sensing only on the occurrence of a forest fire. Clearly, although it is desirable that the transfer state be energy-efficient, it may be more important that the monitoring state be ultra-low-power as the network resides in this state for most of the time.

This observation holds true for many other applications as well. The idea behind STEM is to turn on only a node's sensors and some preprocessing circuitry during monitoring states. Whenever a possible event is detected, the main processor is woken up to analyze the sensed data in detail and forward it to the data sink.

However, the radio of the next hop in the path to the data sink is still turned off, if it did not detect the same event. STEM solves this problem by having each node to periodically turn on its radio for a short time to listen if someone else wants to communicate with it.

The node that wants to communicate, i.e., the *initiator SN,* sends out a beacon with the ID of the node it is trying to wake up, i.e., the *target SN.* This can be viewed as a procedure by which the initiator SN attempts to activate the link between itself and the target SN. As soon as the target SN receives this beacon, it responds back to the initiator node and both keep their radio on at this point. If the packet needs to be relayed further, the target SN will become the initiator node for the next hop and the process is repeated.

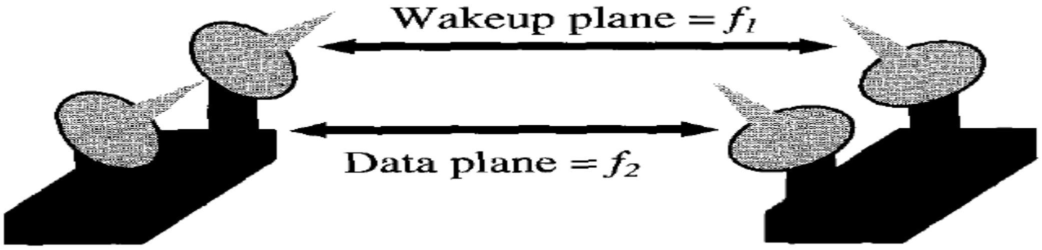


Figure: Sensor Node configuration in STEM

Once both the nodes that make up a link have their radio on, the link is active and can be used for subsequent packets. However, the actual data transmissions may still interfere with the wakeup protocol. To overcome this problem, STEM proposes the wakeup protocol and the data transfer to employ different frequency bands as depicted in above Figure.

In addition, separate radios would be needed in each of these bands. It is to be noted that there exists several commercially available sensors with this feature. In Figure we see that the wakeup messages are transmitted by the radio operating in frequency band $f_{1.}$ STEM refers to these communications as occurring in the *wakeup plane.* Once the initiator SN has successfully notified the target SN, both SNs turn on their radio that operates in frequency band *f2.* The actual data packets are transmitted in this band, called the *data plane.*

19. Explain in detail about the different flat routing protocols

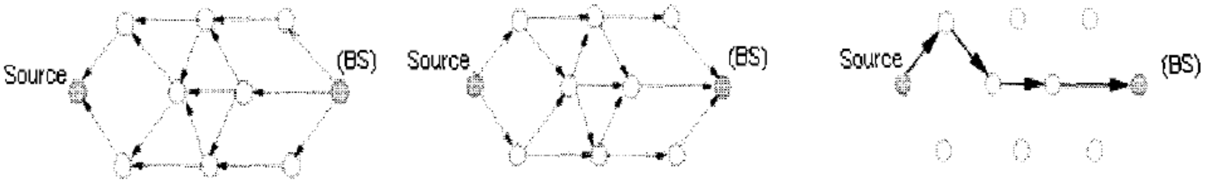
In flat routing based protocols, all nodes play the same role. Here, we present the most prominent protocols falling in this category.

Directed Diffusion

Directed Diffusion is a data aggregation and dissemination paradigm for sensor networks. It is a data-centric (DC) and application-aware approach in the sense that all data generated by sensor nodes is named by attribute-value pairs. Directed Diffusion is very useful for applications requiring dissemination and processing of queries. The main idea of the DC paradigm is to combine the data coming from different sources en-route (in-network aggregation) by eliminating redundancy, minimizing the number of transmissions; thus saving network energy and prolonging its lifetime.

The BS requests data by broadcasting *interests,* which describes a task to be done by the network. Interest diffuses through the network hop-by-hop, and is broadcast by each node to its neighbors. As the interest is propagated throughout the network, gradients are setup to draw data satisfying the query towards the requesting node. Each SN that receives the interest setup a gradient toward the SNs from which it receives the interest. This process continues until gradients are setup from the sources back to the BS.

The strength of the gradient may be different towards different neighbors, resulting in variable amounts of information flow. At this point, loops are not checked, but are removed at a later stage. Below Figure depicts an example of the operation of directed diffusion. Figure (a) presents the propagation of interests, Figure (b) shows the gradients construction, and Figure (c) depicts the data dissemination. When interests fit gradients, paths of information flow are formed from multiple paths, and the best paths are reinforced so as to prevent further flooding according to a local rule. In order to reduce communication costs, data is aggregated on the way. The BS periodically refreshes and re-sends the interest when it starts to receive data from the source(s). This retransmission of interests is needed because the medium is inherently unreliable.



(a) Interest Propagation (b) Gradients set up (c) Data transmission and path reinforcement

Figure : Operation of directed diffusion

Sequential Assignment Routing (SAR)

The routing scheme in SAR is dependent on three factors: energy resources, QoS on each path, and the priority level of each packet. To avoid single route failure, a multi-path approach coupled with a localized path restoration scheme is employed.

To create multiple paths from a source node, a tree rooted at the source node to the destination nodes (i.e., the set of BSs) is constructed. The paths of the tree are defined by avoiding nodes with low energy or QoS guarantees.

For each SN, two metrics are associated with each path: **delay** (which is an additive QoS metric); and **energy** usage for routing on that path.

The energy is measured with respect to how many packets will traverse that path. SAR calculates a weighted QoS metric as the product of the additive QoS metric and a weight coefficient associated with the priority level of the packet. The goal of SAR is to minimize the average weighted QoS metric throughout the lifetime of the network.

In addition, a handshake procedure based on a local path restoration scheme between neighboring nodes is used to recover from a failure.

Minimum Cost Forwarding Algorithm

The minimum cost forwarding algorithm (MCFA) exploits the fact that the direction of routing is always known, that is, towards fixed and predetermined external BS. Therefore, a SN need not have a unique ID nor maintain a routing table. Instead, each node maintains the least cost estimate from itself to the BS. Each message forwarded by the SN is broadcast to its neighbors.

When a node receives the message, it checks if it is on the least cost path between the source SN and the BS. If so, it re-broadcasts the message to its neighbors. This process repeats until the BS is reached. In MCFA, each node should know the least cost path estimate from itself to the BS, and this is obtained as follows. The BS broadcasts a message with the cost set to zero while every node initially set its least cost to the BS to infinity. Each node, upon receiving the broadcast message originated at the BS, checks to see if the estimate in the message plus the cost of the link on which the message was received is less than the current estimate.

Coherent and Non-Coherent Processing

Data processing is a major component in the operation of any WSN. Thus, different routing schemes usually employ different data processing techniques. In general, sensor nodes cooperate with each other in processing different data flooded throughout the network. Two examples of data processing techniques are coherent and non-coherent data processing-based routing [Sohrabi2000]. In non-coherent data processing routing, nodes locally process the raw data before being sent to other nodes for further processing. The nodes that perform further processing are called the aggregators. In coherent routing, the data is forwarded to aggregators after minimum processing of time stamping and duplicate suppression. To perform energy-efficient routing, normally coherent processing is selected. Non-coherent functions generate fairly low load.

Energy Aware Routing

A destination initiated reactive protocol is proposed in order to prolong the network lifetime. This protocol is similar to directed diffusion (discussed earlier) with the difference that it maintains a set of paths instead of maintaining or enforcing one optimal path. These paths are maintained and chosen by means of a certain probability, which depends on how low the energy consumption of each path can be achieved. By selecting different routes at different times, the energy of any single route will not deplete so quickly. With this scheme, the network degrades gracefully as energy is dissipated more equally amongst all nodes.

Cluster Based Routing Protocol (CBRP)

A simple cluster based routing protocol (CBRP) has been proposed in [Jiang 1998]. It divides the network nodes into a number of overlapping or disjoint two-hop-diameter clusters in a distributed manner. Here, the cluster members just send the data to the

CH, and the CH is responsible for routing the data to the destination. The major drawback with CBRP is that it requires a lot of hello messages to form and maintain the clusters, and thus may not be suitable for WSN. Given that sensor nodes are stationary in most of the applications this is a considerable and unnecessary overhead.

Scalable Coordination

In [Estrinl999], a hierarchical clustering method is discussed, with emphasis on localized behavior and the need for asymmetric communication and energy conservation in a sensor network. In this method the cluster formation appears to require considerable amount of energy (no experimental results are available) as periodic advertisements are needed to form the hierarchy. Also, any changes in the network conditions or sensor energy level result in re-clustering which may be not quite acceptable as some parameters tend to change dynamically.

Low-Energy Adaptive Clustering Hierarchy (LEACH)

LEACH is introduced as a hierarchical clustering algorithm for sensor networks, called Low-Energy Adaptive Clustering Hierarchy (LEACH). LEACH is a good approximation of a proactive network protocol, with some minor differences which includes a distributed cluster formation algorithm. LEACH randomly selects a few sensor nodes as CHs and rotates this role amongst the cluster members so as to evenly distribute the energy dissipation across the cluster. In LEACH, the CH nodes compress data arriving from nodes that belong to the respective cluster, and send an aggregated packet to the BS in order to reduce the amount of information that must be transmitted. LEACH uses a TDMA and CDMA MAC to reduce intra-cluster and inter-cluster collisions, respectively. However, data collection is centralized and is performed periodically. Therefore, this protocol is better appropriate when there is a need for constant monitoring by the sensor network. On the other hand, a user may not need all the data immediately. Hence, periodic data transmissions may become unnecessary as they may drain the limited energy of the sensor nodes. After a given interval of time, a randomized rotation of the role of the CH is conducted so that uniform energy dissipation in the sensor network is obtained. Based on simulation, it has been found that only 5% of the nodes actually need to act as CHs.

Power-Efficient Gathering in Sensor Information Systems (PEGASIS)

The Power-Efficient Gathering in Sensor Information Systems (PEGASIS) is a near optimal chain-based protocol which is an enhancement over LEACH. In order to prolong network lifetime, nodes employing PEGASIS communicate with their closest neighbors only and they take turns in communicating with the BS. Whenever a round of nodes communicating with the BS ends, a new round starts. This decreases the power required to transmit data per round, as energy dissipation is spread uniformly over all nodes. As a result, PEGASIS has two main goals. First, it aims at increasing the lifetime of each node

by using collaborative techniques. As a result, the overall network lifetime is also increased. Second, it only allows coordination between nodes that are close together, thus reducing the bandwidth consumed for communication. To locate the closest neighbor SN, SNs use the signal strength to measure the distance to all of its neighboring nodes and then adjust the signal strength so that only one node can be heard. The chain in PEGASIS consists of those nodes that are closest to each other and form a path to the BS. The aggregated form of the data is sent to the BS by any node in the chain, and nodes in the chain take turns in transmitting to the BS. By simulation, it has been shown [Lindsey2002a] that PEGASIS increases the lifetime of the network twice as much as compared to when LEACH is used. On the other hand, PEGASIS makes some assumptions which may not always be true. First, PEGASIS assumes that each sensor node is able to communicate with the BS directly. In practical cases, however, sensor nodes are expected to use multihop communication to reach the BS. In addition, it assumes that each node maintains a complete database about
the location of all other nodes in the network. However, the method by which node locations are obtained is not indicated. Finally, PEGASIS assumes all sensors to be immobile at all times, while this may not be true for certain applications.

Small Minimum Energy Communication Network (MECN)

The minimum energy communication network (MECN) protocol has been designed to compute an energy efficient Sub network for a given sensor network. On top of MECN, a new algorithm called Small MECN (SMECN) has been proposed to construct such a sub network. The sub network (i.e., sub graph G1) constructed by SMECN is smaller than the one constructed by MECN if the broadcast region around the broadcasting node is circular for a given power assignment. The sub graph *G'* of graph G, which represents the sensor network, minimizes the energy consumption satisfying the following conditions:

- The number of edges in G' is less than in G, while containing all nodes in G;
- The energy required to transmit data from a node to all its neighbors in sub graph *G'* is less than the energy required to transmit to all its neighbors in graph *G.*

Threshold-Sensitive Energy Efficient (TEEN) a Threshold-sensitive Energy Efficient sensor Network (TEEN) protocol has been described its time line as depicted in this scheme, at every cluster change time, the CH broadcasts the following to its members in addition to the attributes:

- Hard Threshold (HT): This is a threshold value for the sensed attribute. It is the absolute value of the attribute beyond which, the node sensing this value must switch on its transmitter and report to its CH;
- Soft Threshold (ST): This is a small change in the value of the sensed attribute which triggers the node to switch on its transmitter and transmit once the HT has been crossed.

20. Distinguish between Flat and Hierarchical routing protocols

Flat versus Hierarchical

As we can see, hierarchical and flat approaches have their own advantages and disadvantages as an underlying routing organization for WSN. To illustrate their differences and suitability for different Applications

Hierarchical	Flat
Reservation-based scheduling	Contention-based scheduling
Collisions avoided	Collision overhead present
Reduced duty cycle due to periodic	Variable duty cycle by controlling
sleeping	sleep time of nodes
Data aggregation by cluster head	Node on multi-hop path aggregates
Simple but non-optimal routing	incoming data from neighbors
Requires global and local	Routing is complex but optimal
synchronization	Links formed in the fly, without
Overhead of cluster formation	synchronization
throughout the network	Routes formed only in regions that
Lower latency as multi-hop	have data for transmission

21. Briefly discuss about Query-based Routing

Query-Based Routing

In query-based routing, the destination nodes propagate a query for data (sensing task) from a node throughout the network. A node having the data matching the query sends it back to the node which requested it. Usually, these queries are described in natural language or in high-level query languages. In query based routing, all the nodes have tables consisting of the sensing tasks queries that they received, and send back data matching these tasks whenever they receive it.

Directed diffusion (discussed earlier) is an example of this type of routing. Here, the sink node sends out messages of interest to SNs. As the interest is propagated throughout the WSN, the gradients from the source back to the sink (BS) are set up. Whenever the source has data matching the interest, it sends the data along the interests' gradient path. To reduce energy consumption, data aggregation is performed en-route. The rumor routing protocol uses a set of long-lived agents to create paths that are directed towards the events they encounter. Here, each node may generate an agent in a probabilistic fashion, and each agent contains an events table that is synchronized with every node that it visits. Agents have a lifetime of a certain number of hops after which they die. Whenever an agent comes across a path leading to an event that it has not encountered so far, it creates a path state leading to this new event. If agents come across shorter or more efficient paths, they optimize them in their routing tables

22. Write short notes on location based routing

In location-based routing, SNs are addressed by means of their locations. Here, the distance between neighboring SNs can be estimated on the basis of incoming signal strengths, and relative coordinates of neighboring SNs can be obtained by exchanging such information. Alternatively, the location of nodes may be available directly through GPS if we consider nodes are equipped with a small low power GPS receiver.

In order to conserve energy, some location-based schemes demand that SNs should go to sleep if there is no activity. Clearly, the more sleeping SNs in the network the more energy can be saved. However, the active SNs should be connected, should cover the entire sensing region,

and should provide basic routing and broadcasting functionalities. The problem of designing sleep period schedules for each node in a localized manner has been addressed. The sensor field is divided into small squares in such a way as to ensure that two SNs in two neighboring squares are connected.

23. Explain Distributed Query Processing

The number of messages generated in distributed query processing is several magnitudes less than in centralized scheme. The application of distributed query execution techniques to improve communication efficiency in sensor and device networks.

They discuss two approaches for processing sensor queries:

warehousing and distributed.

In the **warehousing approach**, data is extracted in a pre-defined manner and stored in a central database (e.g., the BS). Subsequently, query processing takes place on the BS.

In the **distributed approach**, only relevant data is extracted from the sensor network, when and where it is needed. A language similar to the Structured Query Language (SQL) has been proposed in for query processing in homogeneous sensor networks.

24. Explain Sensor Databases and COUGAR Architecture

One can view the wireless sensor network as a comprehensive distributed database and interact with it via database queries. This approach solves, en passant, the entire problem of service definition and interfaces to WSNs by mandating, for example, SQL queries as the interface.

A model for sensor database systems known as COUGAR defines appropriate user and internal representation of queries. The sensor queries is also considered so that it is easier to aggregate the data and to combine two or more queries. In COUGAR, routing of queries is not handled. COUGAR has a three-tier architecture:

- The **Query Proxy**: A small database component running on the sensor nodes to interpret and execute queries;
- A **Front end Component**: A query-proxy that allows the sensor network to connect to the outside world. Each front-end includes a full-fledged database server;
- A **Graphical User Interface** (GUI): Through the GUI, users can pose ad hoc and long running queries on the WSN. A map that allows the user to query by region and visualize the topology of sensors in the network.

25. What is In-Network Processing? Explain.

In-network processing, requires data to be modified as it flows through the network. It has become one of the primary enabling technologies for WSNs as it has the potential to considerably increase the energy efficiency of the network. In-network processing is often very closely related to distributed query processing (discussed earlier), as the former takes place in the execution of the latter. The rationale behind in-network processing is that sensors close to the event being monitored sense similar data. Obviously, the number of nodes that sense attributes related to an event in a geographical region depends on the footprint of the event, also referred to as the *target region.* Therefore, it is possible to exploit correlation in the observed data both in time and in space.

Possibilities for in-network processing include compression or aggregation, which is one of the most active research areas in WSNs. An important motivation for aggregation and in-network processing is that, typically, computation is much cheaper in terms of energy consumption than communication. Monitoring civil structures, machines, road traffic and environment are just a few applications that require spatio-temporal querying that could benefit from an in-network query processing architecture.

For aggregating data some of the sensors need to have enhanced capabilities than the majority of the simple sensors and such resource rich wireless sensors (RRWS) make the WSN heterogeneous in nature.

As the RRWSs nodes act as CHs, they also maintain partial network data. So, the next question is, how many RRWS nodes need to be deployed and what the ratio with respect to simple WS nodes is. This would depend on the application and the type of desired query as response could also be provided by RRWSs, rather than getting information from individual WS nodes. So, the queries can be broadly classified as

1. **Simple Queries**: this may require answer from a subset of WSs and could be provided by RRWS. An example could be, "What is the temperature in a given region?"
2. **Aggregate Queries**: This requires aggregation of currently sensed values by WSs in a given region.
3. **Approximate Queries**: This implies aggregation of data in the data form of a histogram, contour maps, or tables and the response could come from the RRWS nodes.
4. **Complex Queries**: This type of query would consist of several condition-based nested queries and one such example is, "Report the average temperature in a region has the highest wind velocity". This type of queries could be possibly responded by RRWSs

Bits

1. The MICA is powered by

    a. One volt button cell battery

    b. Two volt button cell battery

    c. Two AA batteries

    d. Four AA batteries

Ans: c

2. The probability of the number of sensor nodes (n) with in the space when the nodes are uniformly distributed with the node density of X, can be calculated by the Poisson's distribution as
    a. $P (n) = (\lambda s)^{n} / n! * e^{-} \lambda s$
    b. $P (n) = (\lambda s)^{n+1} /(n+1)! * e^{-} \lambda s$
    c. $P (n) = (\lambda s)n-1 /(n-1)! * e- \lambda s$
    d. $P (n) =\lambda n sn-1/n! * e- \lambda s$

    Ans:A

3. The wireless sensor network are
    a. Node centric
    b. Data centric
    c. Network centric
    d. Sensors centric

    Ans:B

4. If N number of sensor nodes are placed in an area A = L x L, the density (X) of sensor nodes is given as during active period and--------
    a. NA=ʎ
    b. 2NA=ʎ
    c. 2N/A=ʎ
    d. N/A=ʎ
    Ans:D

5. SSIM stands for
    a. Symmetric sensing information mobility
    b. Smart sensing information methodology
    c. Smart sensing information methodology
    d. Sensor's service for integrated machine
    Ans:B

6. The Berkley note hardware has been observed to draw during sleep period, respectively
    a. 5-20 milliamps, 5 microamps
    b. 5-20 microamps, 5 milliamps
    c. 5 millamps, 5-20 microamps
    d. 5-20 microamps, 5-20 milliamps
    Ans:A

7. .__________In Hexagonal placement, the sensing area to be covered by each sensor is
    a. $2\ \ /3\, r^2$
    b. $3\ \ /4\, r^2$
    c. $3\, /4\, r$
    d. $/4\, r^2$
    Ans:B

8. In rectangular placement, the total sensing area covered by N-sensors is
    a. N/r^2
    b. Nr
    c. Nr^2
    d. 2Nr
    Ans:C

9. _______________ based sensors is used to monitor crack
    a. Fiber optic
    b. Mica mote
    c. Berkley mote
    d. Bluetooth

Ans:A

10. Sens IT stands for

    a. Sensing integrated tele devices
    b. Sensor information technology
    c. Sensor integrated topology
    d. Sensing information topology

    Ans:B

11. The micro controller Atmel Atmega 1032 operates at_

    a. 4MHz
    b. 8MHz
    c. 12MHz
    d. 14MHz

    Ans:A

12. .__is used to capture the physical parameters of the environment

    a. Sensing Conductor
    b. Sensing transducer
    c. Both A & B
    d. None of the above

    Ans:B

13. The energy consumed for receiving a K bit message is $E_{Rx(k)}$ =

    a. $E_{Rx\text{-}elec}(k) = E_{elec} *k$
    b. 3 /4 r^2
    c. 3 /4 r
    d. /4 r^2

    Ans:A

14. The are application specific

    a. Wireless sensor networks
    b. Wire sensor networks
    c. Both A & B
    d. None of the above

    Ans:A

15. Sensor are put into ____________cycles to consume energy

    a. Sleep
    b. Wakeup
    c. Sleep-Wakeup
    d. None of the above

    Ans:C

16. If A is the total area covered by N number of sensors, r is the side of a triangle. Then the area covered by a triangular tile is given by
    a. $r^2/4$
    b. $r^2/8$
    c. $r^2/12$
    d. $r^2/14$
    Ans:A

17. A Bluetooth based have been proposed for monitoring humidity, temperature, vibration, stress etc.
    a. Scatternet
    b. Internet
    c. IntraNet
    d. None of the above
    Ans:A

18. The proposed architecture of 2-tier network comprise Of__
    a. Micro nodes and macro nodes
    b. Micro nodes
    c. macro nodes
    d. None of the above
    Ans:A

19. ALERT stands for_______
    a. Automated Local Evolution in Real-Time
    b. Automated Global Evolution in Real-Time
    c. Both A & B
    d. None of the above
    Ans:A

20. The networked micro sensors technologies have been identified as the future key application by
    a. Fixed RTO
    b. DARDA
    c. DSR-MB
    d. ATCP
    Ans:B

21. ___are used by wireless sensor node, to transmit and receive the data across the network.
    a. Radio Transceivers
    b. Transmitter
    c. Transceiver
    d. None of the above

Ans:A

22. The primary strategy of____ protocol is to keep all the sensor nodes of wireless network in a low duty cycle mode

    a. Cluster based routing protocol
    b. Chain-based protocol
    c. MECN protocol
    d. Sensor MAC protocol

    Ans:D

23. .__________protocol enables continuity of various operations in sensor network

    a. APTEEN protocol
    b. Eaves-drop-and-register protocol
    c. TEEN protocol
    d. STEM protocol

    Ans:B

24. .__________are the messages used by EAR algorithm for inviting its neighbours

    a. Response, invite and disconnect
    b. Mobile response and mobile disconnect
    c. Broadcast and mobile invite
    d. both (b) and (c)

    Ans:D

25. The main purpose of ________ protocol is to preserve both time and energy that are used by sensor nodes in sensing the environment.

    a. APTEEN protocol
    b. TEEN protocol
    c. STEM protocol
    d. MECN protocol

Ans:C

UNIT-V

1.Discuss in detail about TinyOS?

Tiny OS aims at supporting sensor network applications on resource-constrained hardware platforms, such as the Berkeley motes. To ensure that an application code has an extremely small foot-print, TinyOS chooses to have no file system, supports only static memory allocation, implements a simple task model, and provides minimal device and networking abstractions. Furthermore, it takes a language-based application development approach so that only the necessary parts of the operating system are compiled with the application. To a certain extent, each TinyOS application is built into the operating system

Like many operating systems, TinyOS organizes components into layers. Intuitively, the lower a layer is, the 'closer' it is to the hardware; the higher a layer is, the closer it is to the application. In addition to the layers, TinyOS has a unique component architecture and provides as a library a set of system software components. A components specification is independent of the components implementation. Although most components encapsulate software functionalities, some are just thin wrappers around hardware. An application, typically developed in the nesC language, wires these components together with other application-specific components.

A program executed in TinyOS has two contexts, tasks and events, which provide two sources of concurrency. Tasks are created (also called posted) by components to a task scheduler. The default implementation of the TinyOS scheduler maintains a task queue and invokes tasks according to the order in which they were posted. Thus tasks are deferred computation mechanisms. Tasks always run to completion without preempting or being preempted by other tasks. Thus tasks are non-preemptive. The scheduler invokes a new task from the task queue only when the current task has completed. When no tasks are available in the task queue, the scheduler puts the CPU into the sleep mode to save energy.

The ultimate sources of triggered execution are events from hardware: clock, digital inputs, or other kinds of interrupts. The execution of an interrupt handler is called an event context. The processing of events also runs to completion, but it preempts tasks and can be preempted by other events. Because there is no preemption mechanism among tasks and because events always preempt tasks, programmers are required to chop their code, especially the code in the event contexts, into small execution pieces, so that it will not block other tasks for too long.

Another trade-off between non-preemptive task execution and program reactiveness is the design of split-phase operations in TinyOS. Similar to the notion of asynchronous method calls in distributed computing, a split-phase operation separates the initiation of a method call from the return of the call. A call to split-phase operation returns immediately, without actually performing the body of the operation. The true execution of the operation is scheduled later; when the execution of the body finishes, the operation notifies the original caller through a separate method call.

In TinyOS, resource contention is typically handled through explicit rejection of concurrent requests. All split-phase operations return Boolean values indicating whether a request to perform the operation is accepted.

In summary, many design decisions in TinyOS are made to ensure that it is extremely lightweight. Using a component architecture that contains all variables inside the components and disallowing dynamic memory allocation reduces the memory management overhead and makes the data memory usage statically analyzable. The simple concurrency model allows high concurrency with low thread maintenance overhead. However, the advantage of being lightweight is not without cost. Many hardware idiosyncrasies and complexities of concurrency management are left for the application programmers to handle. Several tools have been developed to give programmers language-level support for improving programming productivity and code robustness.

2.Discuss about nesC?

nesC is an extension of C to support and reflect the design of TinyOS. It provides a set of language constructs and restrictions to implement TinyOS components and applications.

A component in nesC has an interface specification and an implementation. To reflect the layered structure of TinyOS, interfaces of a nesC component are classified as *provides* or *uses* interfaces. A provides interface is a set of method calls exposed to the upper layers, while a uses interface is a set of method calls hiding the lower layer components. Methods in the interfaces can be grouped and named.

Although they have the same method call semantics, nesC distinguishes the directions of the interface calls between layers as *event* calls and *command* calls. An event call is a method call from a lower layer component to a higher layer component, while a command is the opposite.

The separation of interface type definitions from how they are used in the components promotes the reusability of standard interfaces. A component can provide and use the same interface type, so that it can act as a filter interposed between a client and a service. A component may even use or provide the same interface multiple times.

3. **Explain about Component Implementation in Sensor network?**

There are two types of components in nesC, depending on how they are implemented: modules and configurations. Modules are implemented by application code (written in a C-like syntax).

Configurations are implemented by connecting interfaces of existing components. nesC also supports the creation of several instances of a component by declaring *abstract components* with optional parameters. Abstract components are created at compile time in configuration. As TinyOS does not support dynamic memory allocation, all components are statically constructed at compile time. A complete application is always a configuration rather than a module. An application must contain the main module, which links the code to the scheduler at run time. The main has single *StdControl* interface, which is the ultimate source of initialization of all components.

5. Explain about Dataflow-Style Language: TinyGALS?

Dataflow languages are intuitive for expressing computation on interrelated data units by specifying data dependencies among them. A dataflow diagram has a set of processing units called actors. Actors have ports to receive and produce data, and the directional connections among ports are FIFO queues that mediate the flow of data. Actors in dataflow languages intrinsically capture concurrency in a system, and the FIFO queues give a structured way of decoupling their executions. The execution of an actor is triggered when there are enough input data at the input ports.

Asynchronous event-driven execution can be viewed as a special case of dataflow models, where each actor is triggered by every incoming event. The globally asynchronous and locally synchronous (GALS) mechanism is a way of building event-triggered concurrent execution from thread-unsafe components. TinyGALS is such as language for TinyOS.

One of the key factors that affect component reusability in embedded software is the component composability, especially concurrent composability. In general, when developing a component, a programmer may not anticipate all possible scenarios in which the component may be used. Implementing all access to variables as atomic blocks, incurs too much overhead. At the other extreme, making all variable access unprotected is easy for coding but certainly introduces bugs in concurrent composition. TinyGALS addresses concurrency concerns at the system level, rather than at component level as in nesC. Reactions to concurrent events are managed by a dataflow-style FIFO queue communication.

6.How do we implement TinyGALS Programming Model

TinyGALS supports all TinyOS components, including its interfaces and module implementations. All method calls in a component interface are synchronous method calls- that is, the thread of control enters immediately into the callee component from the caller component. An application in TinyGALS is built in two steps: (1) constructing asynchronous actors from synchronous components, and (2) constructing an application by connecting the asynchronous components through FIFO queues.

An actor in TinyGALS has a set of input ports, a set of output ports, and a set of connected TinyOS components. An actor is constructed by connecting synchronous method calls among TinyOS components.

At the application level, the asynchronous communication of actors is mediated using FIFO queues. Each connection can be parameterized by a queue size. In the current implementation of TinyGALS, events are discarded when the queue is full. However, other mechanisms such as discarding the oldest event can be used.

7. **Explain about Node-Level Simulators?**

Node-level design methodologies are usually associated with simulators that simulate the behavior of a sensor network on a per-node basis. Using simulation, designers can quickly study the performance (in terms of timing, power, bandwidth, and scalability) of potential algorithms without implementing them on actual hardware and dealing with the vagaries of actual physical phenomena. A node-level simulator typically has the following components:

- *Sensor node model:* A node in a simulator acts as a software execution platform, a sensor host, as well as a communication terminal. In order for designers to focus on the application-level code, a node model typically provides or simulates a communication protocol stack, sensor behaviors (e.g., sensing noise), and operating system services. If the nodes are mobile, then the positions and motion properties of the nodes need to be modeled. If energy characteristics are part of the design considerations, then the power consumption of the nodes needs to be modeled.

- *Communication model:* Depending on the details of modeling, communication may be captured at different layers. The most elaborate simulators model the communication media at the physical layer, simulating the RF propagation delay and collision of simultaneous transmissions. Alternately, the communication may be simulated at the MAC layer or network layer, using, for example, stochastic processes to represent low-level behaviors.

- *Physical environment model:* A key element of the environment within a sensor network operates is the physical phenomenon of interest. The environment can also be simulated at various levels of details. For example, a moving object in the physical world may be abstracted into a point signal source. The motion of the point signal source may be modeled by differential equations or interpolated from a trajectory profile. If the sensor network is passive-that is, it does not impact the behavior of the environment-then the environment can be simulated separately or can even be stored in data files for sensor nodes to read in. If, in addition to sensing, the network also performs actions that influence the behavior of the environment, then a more tightly integrated simulation mechanism is required.

- *Statistics and visualization:* The simulation results need to be collected for analysis. Since the goal of a simulation is typically to derive global properties from the execution of individual nodes, visualizing global behaviors is extremely important. An ideal visualization tool should allow users to easily observe on demand the spatial distribution and mobility of the nodes, the connectivity among nodes, link qualities, end-to-end communication routes and delays, phenomena and their spatio-temporal dynamics, sensor readings on each node, sensor nodes states, and node lifetime parameters (e.g., battery power).

A sensor network simulator simulates the behavior of a subset of the sensor nodes with respect to time. Depending on how the time is advanced in the simulation, there are two types of execution models: *cycle-driven simulation* and *discrete-event simulation*. A cycle-driven (CD) simulation discretizes the continuous notion of real time into (typically regularly spaced) ticks and simulates the system behavior at these ticks. At each tick, the physical phenomena are first simulated, and then all nodes are checked to see if they have anything to sense, process, or communicate. Sensing and computation are assumed to be finished before the next tick. Sending a packet is also assumed to be completed by then. However, the packet will not be available for the destination node until next tick. This split-phase communication is a key mechanism to reduce cyclic dependencies that may occur in cycle-driven simulations. Most CD simulators do not allow interdependencies within a single tick.

Unlike cycle-driven simulators, a discrete-vent (DE) simulator assumes that the time is continuous and an event may occur at any time. As event is 2-tuple with a value and a time stamp indicating when the event is supposed to be handled. Components in a DE simulation react to input events and produce output events. In node-level simulators, a component can be a sensor node, and the events can be communication packets; or a component can be software module within and the events can be message passings among these nodes. Typically, components are causal, in the sense that if an output event is computed from an input event, then the time stamp of the output should not be earlier than that of the input event. Non-causal components require the simulators to be able to roll back in time, and worse, they may not define a deterministic behavior of a system . A DE simulator typically requires a global event queue. All events passing between nodes or modules are put in the event queue and sorted according to their chronological order. At each iteration of the simulation, the simulator removes the first event (the one with earliest time stamp) from the queue and triggers the component that reacts to that event.

In terms of timing behavior, a DE simulator is more accurate than a CD simulator, and as a consequence, DE simulators run slower. The overhead of ordering all events and computation, in addition to the values and time stamps of events, usually dominates the computation time. At an early stage of a design when only the asymptotic behaviors rather than timing properties are of concern, CD simulations usually require less complex components and give faster simulations. This is partly because of the approximate timing behaviors, which make simulation results less comparable from application to application, there is no general CD simulator that fits all sensor network simulation tasks. Many of the simulators are developed for particular applications and exploit application-specific assumptions to gain efficiency.

DE simulations are sometimes considered as good as actual implementations, because of their continuous notion of time and discrete notion of events. There are several open-source or commercial simulators available. One class of these simulators comprises extensions of classical network simulators, such as ns-2, J-Sim (previously known as JavaSim), and GloMoSim/ Qualnet. The focus of these simulators is on network modeling, protocol stacks, and simulation performance. Another class of simulators, sometimes called *software-in-the-loop simulators,* incorporate the actual node software into the simulation. For this reason, they are typically attached to particular hardware platforms and are less portable. Example include TOSSIM [12] for Berkeley motes and Em* for Linux-based nodes such as Sensoria WINS NG platforms.

8. **Explain about ns-2 Simulator and its Sensor Network Extensions?**

The simulator ns-2 is an open-source network simulator that was originally designed for wired, IP networks. Extensions have been made to simulate wireless/mobile networks (e.g. 802.11 MAC and TDMA MAC) and more recently sensor networks. While the original ns-2 only supports logical addresses for each node, the wireless/mobile extension of it (e.g. introduces the notion of node locations and a simple wireless channel model. This is not a trivial extension, since once the nodes move, the simulator needs to check for each physical layer event whether the destination node is within the communication range. For a large network, this significantly slows down the simulation speed.

There are two widely known efforts to extend ns-2 for simulating sensor networks: SensorSim form UCLA and the NRL sensor network extension from the Navy Research Laboratory . SensorSim also supports hybrid simulation, where some real sensor nodes, running real applications, can be executed together with a simulation. The NRL sensor network extension provides a flexible way of modeling physical phenomena in a discrete event simulator. Physical phenomena are modeled as network nodes which communicate with real nodes through physical layers.

The main functionality of ns-2 is implemented in C++, while the dynamics of the simulation (e.g., time-dependent application characteristics) is controlled by Tcl scripts. Basic components in ns-2 are the layers in the protocol stack. They implement the handlers interface, indicating that they handle events. Events are communication packets that are passed between consecutive layers within one node, or between the same layers across nodes.

The key advantage of ns-2 is its rich libraries of protocols for nearly all network layers and for many routing mechanisms. These protocols are modeled in fair detail, so that they closely resemble the actual protocol implementations. Examples include the following:

- TCP: reno, tahoe, vegas, and SACK implementations.
- MAC: 802.3, 802.11, and TDMA.
- Ad hoc routing: Destination sequenced distance vector (DSDV) routing, dynamic source routing (DSR), ad hoc on-demand distance vector (AOPDV) routing, and temporarily ordered routing algorithm (TORA).

Sensor network routing: Directed diffusion, geographical routing (GEAR) and geographical adaptivefidelity (GAF) routing

9.**Explain about The Simulator TOSSIM**

TOSSIM is a dedicated simulator for TinyOS applications running on one or more Berkeley motes. The key design decisions on building TOSSIM were to make it scalable to a network of potentially thousands of nodes, and to be able to use the actual software code in the simulation. To achieve these goals, TOSSIM takes a cross-compilation approach that compiles the nesC source code into components in the simulation. The event-driven execution model of TinyOS greatly simplifies the design of TOSSIM. By replacing a few low-level components such as the A/D conversion (ADC), the system clock, and the radio front end, TOSSIM translates hardware interrupts into discrete-event simulator events. The simulator event queue delivers the interrupts that drive the execution of a node. The upper-layer TinyOS code runs unchanged.

TOSSIM uses a simple but powerful abstraction to model a wireless network. A network is a directed graph, where each vertex is a sensor node and each directed edge has a bit-error rate. Each node has a private piece of state representing what it hears on the radio channel. By setting connections among the vertices in the graph and a bit-error rate on each connection, wireless channel characteristics, such as imperfect channels, hidden terminal problems, and asymmetric links can be easily modeled. Wireless transmissions are simulated at the bit level. If a bit error occurs, the simulator flips the bit.

TOSSIM has a visualization package called TinyViz, which is a Java application that can connect to TOSSIM simulations. TinyViz also provides mechanisms to control a running simulation by, for example, modifying ADC readings, changing channel properties, and injecting packets. TinyViz is designed as a communication service that interacts with the TOSSIM event queue. The exact visual interface takes the form of plug-ins that can interpret TOSSIM events. Beside the default visual interfaces, users can add application-specific ones easily.

10.Explain about State-centric Programming

Many sensor network applications, such as target tracking, are not simply generic distributed programs over an ad hoc network of energy-constrained nodes. Deeply rooted in these applications is the notion of states of physical phenomena and models of their evolution over space and time. Some of these states may be represented on a small number of nodes and evolve over time, as in the target tracking problem, while others may be represented over a large and spatially distributed number of nodes, as in tracking a temperature contour.

A distinctive property of physical states, such as location, shape, and motion of objects, is their continuity in space and time. Their sensing and control is typically done through sequential state updates. System theories, the basis for most signal and information processing algorithms, provide abstractions for state updates, such as:

$$\mathbf{x}_{\mathrm{k+1}} = f(\mathbf{x}_{\mathrm{k}}, u_k)$$

$$\mathbf{y}_{\mathrm{k}} = g(\mathbf{x}_{\mathrm{k}}, u_k)$$

where **x** is the state of a system, *u* is the system input, **y** is the output and *k* is an integer update index over space and/or time, f is the state update function, and g is the output or observation function. This formulation is broad enough to capture a wide variety of algorithms in sensor fusion, signal processing, and control (e.g., Kalman filtering, Bayesian estimation, system identification, feedback control laws, and finite-state automata).

However, in distributed real-time embedded systems such as sensor networks, the formulation is not as clean as represented in the above equations. The relationships among subsystems can be highly complex and dynamic over space and time. The following issues (which are not explicitly tackled in the above equations) must be properly addressed during the design to ensure the correctness and efficiency of the system.

- Where are the state variables stored?
- Where do the inputs come from?
- Where do the outputs go?
- Where are the functions *f* and *g* evaluated?

- How long does the acquisition of input take?

- Are the inputs in u_k collected synchronously?

- Do the inputs arrive in the correct order through communication?

- What is the time duration between indices k and $k+1$? Is it a constant?

These issues, addressing where and when, rather than how, to perform sensing, computation, and communication, play a central role in the overall system performance. However, these 'non-functional" aspects of computation, related to concurrency, responsiveness, networking, and resource management, are not well supported by traditional programming models and languages. State-centric programming aims at providing design methodologies and frameworks that give meaningful abstractions for these issues, so that system designers can continue to write algorithms on top of an intuitive understanding of where and when the operations are performed.

A collaborative group is such an abstraction. A collaborative group is a set of entities that contribute to a state update. These entities can be physical sensor nodes, or they can be more abstract system components such as virtual sensors or mobile agents hopping among sensors. These are all referred to as agents.

Intuitively, a collaboration groups provides two abstractions: its *scope* to encapsulate network topologies and its *structure* to encapsulate communication protocols. The scope

of a group defines the membership of the nodes with respect to the group. A software agent that hops among the sensor nodes to track a target is a virtual node, while a real node is physical sensor. Limiting the scope of a group to a subset of the entire space of all agents improves scalability. Grouping nodes according to some physical attributes rather than node addresses is an important and distinguishing characteristic of sensor networks.

The *structure* of a group defines the “roles” each member plays in the group, and thus the flow of data. Are all members in the group equal peers? Is there a “leader” member in the group that consumes data? Do members in the group form a tree with parent and children relations? For example, a group may have a leader node that collects certain sensor readings from all followers. By mapping the leader and the followers onto concrete sensor nodes, one can effectively define the flow of data from the hosts of followers to the host of the leader. The notion of roles also shields programmers from addressing individual nodes either by name or address. Furthermore, having multiple members with the same role provides some degree of redundancy and improves robustness of the application in the presence of node and link failures.