DIGITAL NOTES ON
ADHOC AND SENSOR NETWORKS
(R20A6908)

B.TECH IV YEAR - I SEM
(2023-24)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY
(Autonomous Institution – UGC, Govt. of India)
(Affiliated to JNTUH, Hyderabad, Approved by AICTE - Accredited by NBA & NAAC – 'A' Grade - ISO 9001:2015 Certified)
Maisammaguda, Dhulapally (Post Via. Hakimpet), Secunderabad – 500100, Telangana State, INDIA.

MALLA REDDY COLLEGE OF ENGINEERING AND TECHNOLOGY

IV Year B.Tech. CSE- I Sem **L/T/P/C**

3 -/-/3

PROFESSIONAL ELECTIVE –V

(R20A6908)AD-HOC&SENSORNETWORKS

COURSE OBJECTIVES:

1. To understand the concepts of sensor networks
2. To understand the MAC and transport protocols for adhoc networks
3. To understand the security of sensor networks
4. To understand the applications of adhoc and sensor networks

COURSE OUTCOMES:

1. Ability to understand the state-of-the-art research in the emerging subject of Ad Hoc and Wireless Sensor Networks
2. Ability to solve the issues in real-time application development based on ASN.
3. Ability to conduct further research in the domain of ASN

UNIT-I

Introduction to AdHoc Networks-Characteristics of MANETs, Applications of MANETs and Challenges of MANETs.

Routing in MANETs - Criteria for classification, Taxonomy of MANET routing algorithms, Topology-based routing algorithms-**Proactive**: DSDV; **Reactive**: DSR, AODV; Hybrid: ZRP; Position-basedroutingalgorithms-**LocationServices**-DREAM,Quorum-based.

UNIT-II

Data Transmission- Broadcast Storm Problem, **Rebroadcasting Schemes**-Simple-flooding, Probability-based Methods, Area-based Methods, Neighbor Knowledge-based: SBA, Multipoint Relaying, AHBP. **Multicasting: Tree-based:** AMRIS, MAODV; **Mesh-based:** ODMRP, CAMP; **Hybrid:** AMRoute**,** MCEDAR.

UNIT-III

Geocasting: Data-transmission Oriented-LBM; Route Creation Oriented-GeoTORA, MGR**.**TCP over AdHoc TCP protocol Overview: TCP Basics, TCP Header Format, Congestion Control; TCP and Manets: Effects of Partition on TCP, Impact Of Lower Layers On TCP.

UNIT-IV

Basics of Wireless, Sensors and Lower Layer Issues: Applications, Classification of sensor networks, Architecture of sensor network, Physical layer, MAC layer, Link layer, Routing Layer.

UNIT-V

Upper Layer Issues of WSN: Transport layer, High-level application layer support, Adapting to the inherent dynamic nature of WSNs, Sensor Networks and mobile robots.

TEXTBOOKS:

1. AdHoc and Sensor Networks–Theory and Applications, Carlos Corderio Dharma P.Aggarwal, World Scientific Publications, March 2006,ISBN–981–256–681–3.
2. WirelessSensorNetworks:AnInformationProcessingApproach,FengZhao,Leonidas Guibas,ElsevierScience,ISBN –978-1-55860-914-3 (MorganKauffman).

INDEX

UNIT-I

Introduction to AdHoc Networks-Characteristics of MANETs, Applications of MANETs and Challenges of MANETs.

Routing in MANETs - Criteria for classification, Taxonomy of MANET routing algorithms, Topology-based routing algorithms-**Proactive**: DSDV; **Reactive**: DSR, AODV; Hybrid: ZRP; Position-basedroutingalgorithms-**LocationServices**-DREAM, Quorum-based; **ForwardingStrategies:GreedyPacket,Restricted** Directional Flooding-DREAM,LAR

Introduction

Simply stating, a Mobile Ad hoc NETwork (MANET) is one that comes together as needed, not necessarily with any support from the existing Internet infrastructure or any other kind of fixed stations. We can formalize this statement by defining an ad hoc network as an autonomous system of mobile hosts (also serving as routers) connected by wireless links, the union of which forms a communication network modeled in the form of an arbitrary graph. This is in contrast to the well-known single hop cellular network model that supports the needs of wireless communication by installing base stations as access points. In these cellular networks, communications between two mobile nodes completely rely on the wired backbone and the fixed base stations. In a MANET, no such infrastructure exists and the network topology may dynamically change in an unpredictable manner since nodes are free to move.

As for the mode of operation, ad hoc networks are basically peer-to-peer multi- hop mobile wireless networks where information packets are transmitted in a store-and- forward manner from a source to an arbitrary destination, via intermediate nodes as shown in Figure 1. As the nodes move, the resulting change in network topology must be made known to the other nodes so that outdated topology information can be updated or removed. For example, as MH2 in Figure 1 changes its point of attachment from MH3 to MH4 other nodes part of the network should use this new route to forward packets to MH2.

Note that in Figure 1, and throughout this text, we assume that it is not possible to have all nodes within range of each other. In case all nodes are close-by within radio range, there are no routing issues to be addressed. In real situations, the power needed toobtain complete connectivity may be, at least, infeasible, not to mention issues such as battery life. Therefore, we are interested in scenarios where only few nodes are within radio range of each other.

Figure 1 raises another issue of symmetric (bi-directional) and asymmetric (unidirectional) links. As we shall see later on, some of the protocols we discuss consider symmetric links with associative radio range, i.e., if (in Figure 1) MH1 is within radio range of MH3, then MH3 is also within radio range of MH1. This is to say that the communication links are symmetric. Although this assumption is not always valid, it is usually made because routing in asymmetric networks is a relatively hard task [Prakash 1999]. In certain cases, it is possible to find routes that could avoid asymmetriclinks, since it is quite likely that these links imminently fail. Unless stated otherwise, throughout this text we consider symmetric links, with all nodes having identical capabilities and responsibilities.

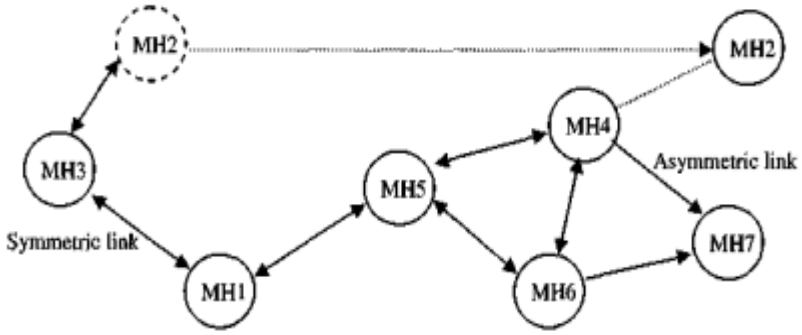


Figure 1.1 – A mobile ad hoc network (MANET)

Note that in Figure 1.1, and throughout this text, we assume that it is not possible to have all MHs within range of each other. In case all MHs are close-by within radio range, there are no routing issues to be addressed. In real situations, the power needed to obtain complete connectivity may be, at least, infeasible, not to mention issues such as battery life and spatial reusability. Therefore, we are interested in scenarios where only few MHs are within radio range of each other. Figure 1.1 raises another issue of symmetric (bi-directional) and asymmetric (unidirectional) links. As we shall see later on, some of the protocols we discuss consider symmetric links with associative radio range, i.e., if (in Figure 1.1) MH1 is within radio range of MH3, then MH3 is also within radio range of MH1. This is to say that the communication links are symmetric. Although this assumption is not always valid, it is usually made because routing in asymmetric networks is a relatively hard task [Prakashl999]. In certain cases, it is possible to find routes that could avoid asymmetric links, since it is quite likely that these links imminently fail. Unless stated otherwise, throughout this text we consider symmetric links, with all MHs having identical capabilities and responsibilities. The issue of symmetric and asymmetric links is one among the several challenges encountered in a MANET. Another important issue is that different nodes often have different mobility patterns. Some MHs are highly mobile, while others are primarily stationary. It is

difficult to predict a MH's movement and pattern of movement. Table 1.1 summarizes some of the main characteristics [Cordeiro2002] and challenges in a MANET. A comprehensive look at the current challenges in ad hoc and sensor networking.

Table 1.1 – Important characteristics of a MANET

Characteristic	Description
Dynamic Topologies	Nodes are free to move arbitrarily with different speeds; thus, the network topology may change randomly and at unpredictable times.
Energy-constrained Operation	Some or all of the nodes in an ad hoc network may rely on batteries or other exhaustible means for their energy. For these nodes, the most important system design optimization criteria may be energy conservation.
Limited Bandwidth	Wireless links continue to have significantly lower capacity than infrastructured networks. In addition, the realized throughput of wireless communications – after accounting for the effects of multiple access, fading, noise, and interference conditions, etc., is often much less than a radio's maximum transmission rate.
Security Threats	Mobile wireless networks are generally more prone to physical security threats than fixed-cable nets. The increased possibility of eavesdropping, spoofing, and minimization of denial-of-service type attacks should be carefully considered.

The issue of symmetric and asymmetric links is one among the several challenges encountered in a MANET. Another important issue is that different nodes often have different mobility patterns. Some nodes are highly mobile, while others are primarily stationary. It is difficult to predict a node's movement and pattern of movement. Table 1 summarizes some of the main characteristics [Duggirala 2000] and challenges faced in a MANET.

Wireless Sensor Networks [Estrin 1999, Kahn 1999] is an emerging application area for ad hoc networks which has been receiving a large attention. The idea is that a collection of cheap to manufacture, stationary, tiny sensors would be able to sense, coordinate activities and transmit some physical characteristics about the surrounding environment to an associated base station. Once placed in a given environment, these sensors remain stationary. Furthermore, it is expected that power will be a major drivingissue behind protocols tailored to these networks, since the lifetime of the battery usually defines the sensor's lifetime. One of the most cited examples is the battlefield urveillance of enemy's territory wherein a large number of sensors are dropped from an airplane so that activities on the ground could be detected and communicated. Other potential commercial fields include machinery prognosis, bio sensing and environmentalmonitoring.

This rest of this text is organized as follows. We initially provide necessary background on ad hoc networking by illustrating its diverse applications. Next, we cover the routing aspect in a MANET, considering both unicast and multicast communication. MAC issues related to a MANET are then illustrated. Following, sensor networks, its diverse applications, and associated routing protocols are discussed. Finally, we conclude this text by discussing the current standard activities at both IETF and the Bluetooth SIG, and also bringing up some open problems that have not received much attention so far and still need to be addressed.

Characteristics of MANET

Let's have a look at some of the characteristics of MANET:

1. **Dynamic topologies**:- As nodes are free to move in any direction, the network topology can alter at any time and is mainly composed of bidirectional links. A unidirectional link may sometimes exist when the transmission power of two nodes differs.

2. **Bandwidth-constrained and variable capacity links**:- Wireless links continue to have much lesser capacity than infrastructure networks.
3. **Energy-constrained operation**:- Batteries or other non-renewable sources may power some or all of the nodes in a MANET. Energy conservation may be the most essential system design optimization requirement for these nodes or devices.
4. **Limited physical security**:- MANETs are generally more vulnerable to physical security threats than wireline networks. The increased risk of eavesdropping, spoofing, and denial of service (DoS) attacks should be carefully examined. Many existing link security solutions are frequently used within wireless networks to reduce security concerns.
5. **Less Human Intervention**:- They require minimal human involvement to configure the network, hence they are dynamically autonomous

Advantages of MANETs

1. Each node can act as both a router and a host, demonstrating its autonomous nature.
2. Separation from the central network administration.
3. Highly expandable and suitable for the addition of new network hubs.
4. Nodes that self-configure and self-heal do not require human involvement.
5. In MANETs infrastructure is not required because it is a decentralized network.
6. Due to the multi-hop approach in which information is conveyed, decentralized networks are often more robust than centralized networks. In a cellular network, for example, if a base station fails, coverage is lost; however, the likelihood of a single point of failure in a MANET is greatly decreased because data can travel via several paths.
7. Other advantages of MANETs over fixed-topology networks include flexibility (mobile devices can be used to form an ad hoc network anywhere), scalability (you can quickly add more nodes to the network), and cheaper management expenses (no need to build infrastructure first).

Disadvantages of MANETs

- There are no authorization facilities.

- Resources are limited due to various constraints such as noise, interference situations, and so on.
- High latency means data is transferred between two sleeping nodes with a significant delay.
- Due to inadequate physical security, they are more vulnerable to attacks.

Applications of MANETs

There are many applications to ad hoc networks. As a matter of fact, any day- to-day application such as electronic email and file transfer can be considered to be easily deployable within an ad hoc network environment. Web services are also possible in case any node in the network can serve as a gateway to the outside world. In this discussion, we need not emphasize the wide range of military applications possible with ad hoc networks. Not to mention, the technology was initially developed keeping in mind the military applications, such as battlefield in an unknown territory where an infrastructured network is almost impossible to have or maintain. In such situations, the ad hoc networks having self-organizing capability can be effectively used where other technologies either fail or cannot be deployed effectively. Advanced features of wireless mobile systems, including data rates compatible with multimedia applications, global roaming capability, and coordination with other network structures, are enabling new applications. Some well-known ad hoc network

Applications are:

- **Collaborative Work** – For some business environments, the need for collaborativecomputing might be more important outside office environments than inside. After all, it is often the case where people do need to have outside meetings to cooperateand exchange information on a given project.
- **Crisis-management Applications** – These arise, for example, as a result of natural disasters where the entire communications infrastructure is in disarray. Restoring communications quickly is essential. By using ad hoc networks, an infrastructure could be set up in hours instead of days/weeks required for wire-line communications.
- **Personal Area Networking and Bluetooth** – A personal area network (PAN) is a short-range, localized network where nodes are usually associated with a given

Person. These nodes could be attached to someone's pulse watch, belt, and so on.

In these scenarios, mobility is only a major consideration when interaction among several PANs is necessary, illustrating the case where, for instance, people meet in real life. Bluetooth [Haarsten 1998], is a technology aimed at, among other things, supporting PANs by eliminating the need of wires between devices such as printers, PDAs, notebook computers, digital cameras, and so on, and is discussed later.

Education via the internet: - Educational opportunities are available on the internet or in remote places due to the practical impossibility of providing pricey last-mile wireline internet access to all users in these areas.

Virtual Navigation:- A big metropolis' physical properties, including its buildings, streets, and other physical features, are graphically represented in a remote database. Additionally, they might be able to "virtually" view a building's interior layout, including a strategy for an emergency evacuation, or locate potential places of interest.

Vehicular area network(VANETs):- This is a growing and precious ad-hoc network application for providing emergency services and other information. This works equally well in both urban and rural settings. The fundamental and necessary data interchange in a specific circumstance.

Challenges of MANET

- **Limited Bandwidth**

  The wireless networks have a limited bandwidth in comparison to the wired networks. Wireless link has lower capacity as compare to infrastructure networks. The effect of fading, multiple accesses, interference condition is very low in ADHOC networks in comparison to maximum radio transmission rate.

- **Dynamic topology**

  Due to dynamic topology the nodes has less truest between them. I some settlement are found between the nodes then it also make trust level questionable.

- **High Routing**

In ADHOC networks due to dynamic topology some nodes changes their position which affects the routing table.

- **Problem of Hidden terminal**

  The Collision of the packets are held due to the transmission of packets by those node which are not in the direct transmission range of sender side but are in range of receiver side.

- **Transmission error and packet loss**

  By increasing in collisions , hidden terminals, interference, uni-directional links and by the mobility of nodes frequent path breaks a higher packet loss has been faced by ADHOC networks.

- **Mobility**

  Due to the dynamic behavior and changes in the network topology by the movement of the nodes .ADHOC networks faces path breaks and it also changes in the route frequently.

- **Security threats**

  New security challenges bring by Adhoc networks due to its wireless nature. In Adhoc networks or wireless networks the trust management between the nodes leads to the numerous security attacks.

- **Some Other Major Challenges in MANET**

  Some other challenges of the MANET are describe briefly as below:

- **Dynamic topologies**

  In Dynamic Topology the nodes are free to move in any direction. Topology of network changes rapidly and unpredictably by the time. Due to this topology the bidirectional and unidirectional routing exists. Challenging task is to transferring the packets between the nodes because the topologies are changes continuously.

- **Multicast Routing**

  Another challenge of MANET is multicast. The multicast dynamic this networks because the nodes are randomly changes its position. The nodes have multiple hopes instead of single hop and they are complex. The new device adds in the network need to know all the other nodes. To facilitate automatic optimal route selection dynamic update is necessary due to existence of node

- **Variability in capacity links due to Bandwidth Constraint**

  Being a link of wireless they continue with low capacity in comparison to the hardwired.

- **Power-constrained and operation**

  This is also considered as a challenge for the MANET network. The MANET is a network where all nodes rely on the batteries or some exhaustible source of energy. Conversion in the energy is the optimized criteria and an important system design. Lean power consumption is also used for the light weight mobile terminals. Conservation of power and power-aware routing is another aspect which must be considered.

- **Security and Reliability**

  Nasty Neighbor relaying packets is also a security problem alongwith other vulnerabilities connected. Different schemes are used for authentication and key management in distributed operations. Reliability problem is also a wireless link characteristic due to limited wireless transmission. Due to the broadcast of wireless medium packets loss and errors in the data occur. In comparison to the wired network the wireless networks are more vulnerable to security threats.

- **Quality of Service**
- In MANET the environment will change constantly so that it provides different quality of service levels which are challengeable. The random nature in the quality communication of MANET it is difficult to server good guarantee of service of the device. To support multimedia services adaptive Quality of Services can be implement over traditional resources.
- **Inter-networking**
- Communication with fixed networks is also expected from MANET in many cases. The existence of routing protocols in both the networks is quite challengeable for pleasant mobility management.

ROUTING IN A MANET

It has become clear that routing in a MANET is intrinsically different from traditional routing found on infra structured networks. Routing in a MANET depends on many factors including topology, selection of routers, and initiation of request, and

specific underlying characteristic that could serve as a heuristic in finding the path quickly and efficiently. The low resource availability in these networks demands efficient utilization and hence the motivation for optimal routing in ad hoc networks. Also, the highly dynamic nature of these networks imposes severe restrictions on routing protocols specifically designed for them, thus motivating the study of protocols which aim at achieving routing stability.

One of the major challenges in designing a routing protocol [Jubin 1987] for ad hoc networks stems from the fact that, on one hand, a node needs to know at least the reachability information to its neighbors for determining a packet route and, on the other hand, the network topology can change quite often in an ad hoc network. Furthermore, as the number of network nodes can be large, finding route to the destinations also

requires large and frequent exchange of routing control information among the nodes. Thus, the amount of update traffic can be quite high, and it is even higher when high mobility nodes are present. High mobility nodes can impact route maintenance overhead of routing protocols in such a way that no bandwidth might remain leftover for the transmission of data packets [Corson 1996].

Proactive and Reactive Routing Protocols

Ad hoc routing protocols can be broadly classified as being Proactive (or table- driven) or Reactive (on-demand). Proactive protocols mandates that nodes in a MANET should keep track of routes to all possible destinations so that when a packet needs to beforwarded, the route is already known and can be immediately used. On the other hand, reactive protocols employ a lazy approach whereby nodes only discover routes to destinations on demand, i.e., a node does not need a route to a destination until that destination is to be the sink of data packets sent by the node.

Proactive protocols have the advantage that a node experiences minimal delay whenever a route is needed as a route is immediately selected from the routing table. However, proactive protocols may not always be appropriate as they continuously use a substantial fraction of the network capacity to maintain the routing information current. To cope up with this shortcoming, reactive protocols adopt the inverse approach by

finding a route to a destination only when needed. Reactive protocols often consume much less bandwidth than proactive protocols, but the delay to determine a route can be significantly high and they will typically experience a long delay for discovering a route to a destination prior to the actual communication. In brief, we can conclude that no protocol is suited for all possible environments, while some proposals using a hybrid approach have been suggested.

Unicast Routing Protocols

Proactive Routing Approach

In this section, we consider some of the important proactive routing protocols.

Destination-Sequenced Distance-Vector Protocol

The destination-sequenced distance-vector (DSDV) [Perkins 1994] is a proactive hop-by-hop distance vector routing protocol, requiring each node to periodically broadcast routing updates. Here, every mobile node in the network maintains a routing

table for all possible destinations within the network and the number of hops to each destination. Each entry is marked with a sequence number assigned by the destination node. The sequence numbers enable the mobile nodes to distinguish stale routes from new ones, thereby avoiding the formation of routing loops. Routing table updates are periodically transmitted throughout the network in order to maintain consistency in the table.

To alleviate the potentially large amount of network update traffic, route updates can employ two possible types of packets: full dumps or small increment packets. A full dump type of packet carries all available routing information and can require multiple network protocol data units (NPDUs). These packets are transmitted infrequently during periods of occasional movement. Smaller incremental packets are used to relay only the information that has changed since the last full dump. Each of these broadcasts should fit into a standard-size NPDU, thereby decreasing the amount of traffic generated. The mobile nodes maintain an additional table where they store the data sent in the

incremental routing information packets. New route broadcasts contain the address of the destination, the number of hops to reach the destination, the sequence number of the information received regarding the destination, as well as a new sequence number unique to the broadcast. The route labeled with the most recent sequence number is always used. In the event that two updates have the same sequence number, the route with the smaller metric is used in order to optimize (shorten) the path. Mobiles also keep track of settling time of the routes, or the weighted average time that routes to a destination could fluctuate before the route with the best metric is received. By delaying the broadcast of a routing update by the length of the settling time, mobiles can reduce network traffic and optimize routes by eliminating those broadcasts that would occur if a better route could be discovered in the very near future.

Note that each node in the network advertises a monotonically increasing sequence number for itself. The consequence of doing it so is that when a node B decides that its route to a destination D is broken, it advertises the route to D with an infinite metric and a sequence number one greater than its sequence number for the route that has broken (making an odd sequence number). This causes any node A routing packets through B to incorporate the infinite-metric route into its routing table until node A hears a route to D with a higher sequence number.

The Wireless Routing Protocol

The Wireless Routing Protocol (WRP) [Murthy 1996] described is a table-based protocol with the goal of maintaining routing information among all nodes in the network. Each node in the network is responsible for maintaining four tables: Distance table, Routing table, Link-cost table, and the Message Retransmission List (MRL) table. Each entry of the MRL contains the sequence number of the update message, a re-transmission counter, an acknowledgment-required flag vector with one entry per neighbor, and a list of updates sent in the update message. The MRL records which updates in an update message need to be retransmitted and neighbors should acknowledge the retransmission.

Mobiles inform each other of link changes through the use of update messages. An update message is sent only between neighboring nodes and contains a list of updates (the destination, the distance to the destination, and the predecessor of the destination), as well as a list of responses indicating which mobiles should acknowledge (ACK) the update. After processing updates from neighbors or detecting a change in a link, mobiles send update messages to a neighbor. In the event of the loss of a link between two nodes, the nodes send update messages to their neighbors. The neighbors then modify their distance table entries and check for new possible paths through other nodes. Any new paths are relayed back to the original nodes so that they can update their tables accordingly.

Nodes learn about the existence of their neighbors from the receipt of acknowledgments and other messages. If a node is not sending messages, it must send a hello message within a specified time period to ensure connectivity. Otherwise, the lack of messages from the node indicates the failure of that link; this may cause a false alarm. When a mobile receives a hello message from a new node, that new node is added to the mobile's routing table, and the mobile sends the new node a copy of its routing table information.Part of the novelty of WRP stems from the way in which it achieves freedom from loops. In WRP, routing nodes communicate the distance and second-to-last hop

information for each destination in the wireless networks. WRP belongs to the class of path-finding algorithms with an important exception. It avoids the "count-to-infinity" problem by forcing each node to perform consistency checks of predecessor information reported by all its neighbors. This ultimately (although not instantaneously) eliminates looping situations and provides faster route convergence when a link failure occurs.

Reactive Routing Approach

In this section, we describe some of the most cited reactive routing protocols.

Dynamic Source Routing

The Dynamic Source Routing (DSR) [Johnson 1996] algorithm is an innovative approach to routing in a MANET in which nodes communicate along paths stored in source routes carried by the data packets. It is referred as one of the purest examples of an on-demand protocol [Perkins 2001]. In DSR, mobile nodes are

required to maintain route caches that contain the source routes of which the mobile is aware. Entries in the route cache are continually updated as new routes are learned. The protocol consists of two major phases: route discovery and route maintenance. When a mobile node has a packet to send to some destination, it first consults its route cache to determine whether it already has a route to the destination. If it has an unexpired route to the destination, it will use this route to send the packet. On the other hand, if the node does not have such a route, it initiates route discovery by broadcasting a route request packet. This route request contains the address of the destination, along with the source node's address and a unique identification number. Each node receiving the packet checks whether it knows of a route to the destination. If it does not, it adds its own address to the route record of the packet and then forwards the packet along its outgoing links.

To limit the number of route requests propagated on the outgoing links of a node, a mobile node only forwards the route request if the request has not yet been seen by the mobile and if the mobile's address does not already appear in the route record.

A route reply is generated when the route request reaches either the destination itself, or an intermediate node that contains in its route cache an unexpired route to the destination. By the time the packet reaches either the destination or such an intermediate node, it contains a route record yielding the sequence of hops taken. Figure 2(a) illustrates the formation of the route record as the route request propagates through the network. If the node generating the route reply is the destination, it places the route

record contained in the route request into the route reply. If the responding node is an intermediate node, it appends its cached route to the route record and then generates the route reply. To return the route reply, the responding node must have a route to the initiator. If it has a route to the initiator in its route cache, it may use that route. Otherwise, if symmetric links are supported, the node may reverse the route in the route record. If symmetric links are not supported, the node may initiate its own route discovery and piggyback the route reply on the new route request. Figure 2(b) shows the transmission of route record back to the source node.

Route maintenance is accomplished through the use of route error packets and acknowledgments. Route error packets are generated at a node when the data link layer encounters a fatal transmission problem. When a route error packet is received, the hop in error is removed from the node's route cache and all routes containing the hop are truncated at that point. In addition to route error messages, acknowledgments are used to verify the correct operation of the route links. These include passive acknowledgments, where a mobile is able to hear the next hop forwarding the packet along the route.

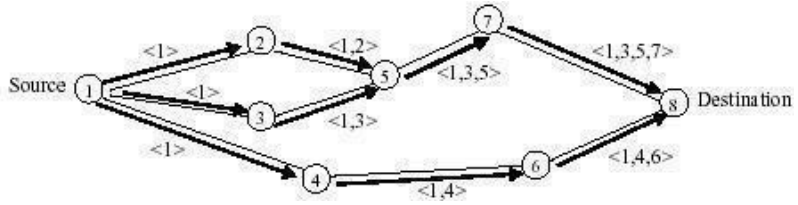


Figure 2(a) – Route discovery in DSR

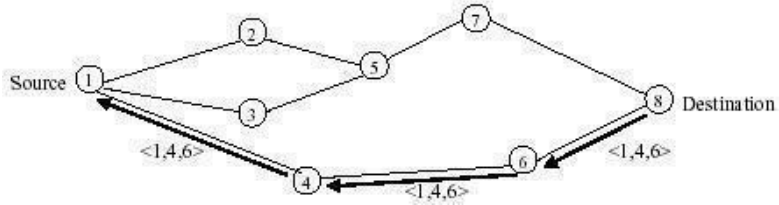


Figure 2(b) – Propagation of route reply in DSR

The Ad Hoc On-Demand Distance Vector Protocol

The Ad Hoc On-Demand Distance Vector (AODV) routing protocol [Perkins 1999] is basically a combination of DSDV and DSR. It borrows the basic on-demand mechanism of Route Discovery and Route Maintenance from DSR, plus the use of hop-by-hop routing, sequence numbers, and periodic beacons from DSDV. AODV minimizes the number of required broadcasts by creating routes on an on-demand basis, as opposed to maintaining a complete list of routes as in the DSDV algorithm. Authors of AODV classify it as a pure on-demand route acquisition system since nodes that are not on a selected path, do not maintain routing information or participate in routing table exchanges. It supports only symmetric links with two different phases:

- Route Discovery, Route Maintenance; and
- Data forwarding.

When a source node desires to send a message and does not already have a valid route to the destination, it initiates a path discovery process to locate the corresponding node. It broadcasts a route request (RREQ) packet to its neighbors, which then forwards the request to their neighbors, and so on, until either the destination or an intermediate node with a "fresh enough" route to the destination is located. Figure 3(a) illustrates the propagation of the broadcast RREQs across the network. AODV utilizes destination sequence numbers to ensure all routes are loop-free and contain the most recent route information. Each node maintains its own sequence number, as well as a broadcast ID. The broadcast ID is incremented for every RREQ the node initiates, and together with the node's IP address, uniquely identifies an RREQ. Along with the node's sequence number and the broadcast ID, the RREQ includes the most recent sequence number it has for the destination. Intermediate nodes can reply to the RREQ only if they have a route to the destination whose corresponding destination sequence number is greater than or equal to that contained in the RREQ.

During the process of forwarding the RREQ, intermediate nodes record in their route tables the address of the neighbor from which the first copy of the broadcast packet is received, thereby establishing a reverse path. If additional copies of the same RREQ are later received, these packets are discarded. Once the RREQ reaches the

destination or an intermediate node with a fresh enough route, the destination/intermediate node responds by unicasting a route reply (RREP) packet back to the neighbor from which it first received the RREQ (Figure 3(b)). As the RREP is routed back along the reverse path, nodes along this path set up forward route entries in their route tables that point to the node from which the RREP came. These forward route entries indicate the active forward route. Associated with each route entry is a route timer which causes the deletion of the entry if it is not used within the specified lifetime. Because the RREP is forwarded along the path established by the RREQ, AODV only supports the use of symmetric links.

Routes are maintained as follows. If a source node moves, it is able to reinitiate the route discovery protocol to find a new route to the destination. If a node along the route moves, its upstream neighbor notices the move and propagates a link failure notification message (an RREP with infinite metric) to each of its active upstream neighbors to inform them of the breakage of that part of the route. These nodes in turn propagate the link failure notification to their upstream neighbors, and so on until the source node is reached. The source node may then choose to re-initiate route discovery for that destination if a route is still desired. An additional aspect of the protocol is the use of hello messages, periodic local broadcasts by a node to inform each mobile node of other nodes in its neighborhood. Hello messages can be used to maintain the local connectivity of a node. However, the use of hello messages may not be required at all times. Nodes listen for re-transmission of data packets to ensure that the next hop is still within reach. If such a re-transmission is not heard, the node may use one of a number of techniques, including the use of hello messages themselves, to determine whether the next hop is within its communication range. The hello messages may also list other nodes from which a mobile node has recently heard, thereby yielding greater knowledge of network connectivity.

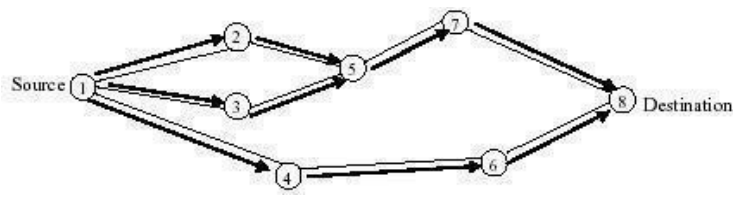
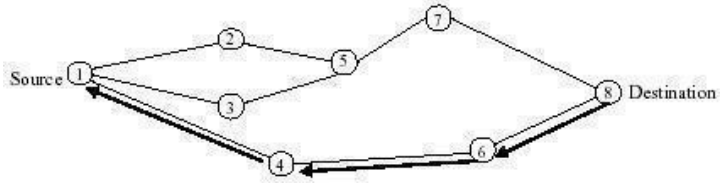


Figure 3(a) – Propagation of RREQ in AODV



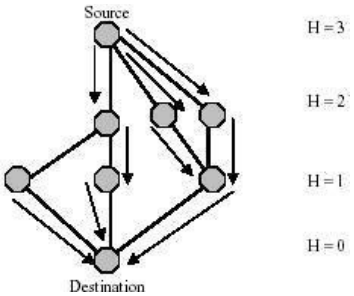


Figure 3(a) – Path taken by the RREP in AODV

Link Reversal Routing and TORA

The Temporally Ordered Routing Algorithm (TORA) [Park 1997] is a highly adaptive loop-free distributed routing algorithm based on the concept of link reversal. It is designed to minimize reaction to topological changes. A key design concept in TORA is that it decouples the generation of potentially far-reaching control messages from the rate of topological changes. Such messaging is typically localized to a very small set of nodes near the change without having to resort to a dynamic, hierarchical routing solution with its added complexity. Route optimality (shortest-path) is considered of secondary importance, and longer routes are often used if discovery of newer routes could be avoided. TORA is also characterized by a multipath routing capability.

The actions taken by TORA can be described in terms of water flowing downhill towards a destination node through a network of tubes that models the routing state of the real network. The tubes represent links between nodes in the network, the junctions of tubes represent the nodes, and the water in the tubes represents the packets flowing towards the destination. Each node has a height with respect to the destination that is computed by the routing protocol. If a tube between nodes A and B becomes blocked such that water can no longer flow through it, the height of A is set to a height greater than that of any of its remaining neighbors, such that water will now flow back out of A (and towards the other nodes that had been routing packets to the destination via A).

Figure 4 illustrates the use of the height metric. It is simply the distance from the destination node.

TORA is proposed to operate in a highly dynamic mobile networking environment. It is source initiated and provides multiple routes for any desired source/destination pair. To accomplish this, nodes need to maintain routing information about adjacent (one-hop) nodes. The protocol performs three basic functions:

- Route creation,
- Route maintenance, and
- Route erasure.

For each node in the network, a separate directed acyclic graph (DAG) is maintained for each destination. When a node needs a route to a particular destination, it broadcasts a QUERY packet containing the address of the destination for which it requires a route. This packet propagates through the network until it reaches either the destination, or an intermediate node having a route to the destination. The recipient of the QUERY then broadcasts an UPDATE packet listing its height with respect to the destination. As this packet propagates through the network, each node that receives the UPDATE sets its height to a value greater than the height of the neighbor from which the UPDATE has been received. This has the effect of creating a series of directed links from the original sender of the QUERY to the node that initially generated the UPDATE. When a node discovers that a route to a destination is no longer valid, it adjusts its height so that it is a local maximum with respect to its neighbors and transmits an UPDATE packet. If the node has no neighbors of finite height with respect to this destination, then the node instead attempts to discover a new route as described above. When a node detects a network partition, it generates a CLEAR packet that resets routing state and removes invalid routes from the network.

TORA is layered on top of IMEP, the Internet MANET Encapsulation Protocol [Corson 1997], which is required to provide reliable, in-order delivery of all routing control messages from a node to each of its neighbors, plus notification to the routing protocol whenever a link to one of its neighbors is created or broken. To reduce overhead, IMEP attempts to aggregate many TORA and IMEP control messages (which IMEP refers to as objects) together into a single packet (as an object block) before transmission. Each block carries a sequence number and a response list of other nodes

from which an ACK has not yet been received, and only those nodes acknowledge the block when receiving it; IMEP retransmits each block with some period, and continues to retransmit it if needed for some maximum total period, after which TORA is notified of each broken link to unacknowledged nodes. For link status sensing and maintaining a list of a node's neighbors, each IMEP node periodically transmits a BEACON (or "BEACON-equivalent") packet, which is answered by each node hearing it with a HELLO (or "HELLO-equivalent") packet.

As we mentioned earlier, during the route creation and maintenance phases, nodes use the "height" metric to establish a DAG rooted at the destination. Thereafter,

links are assigned a direction (upstream or downstream) based on the relative height metric of neighboring nodes as shown in Figure 5(a). In times of node mobility the DAG route is broken, and route maintenance is necessary to reestablish a DAG rooted at the same destination. As shown in Figure 5(b), upon failure of the last downstream link, a node generates a new reference level that effectively coordinates a structured reaction to the failure. Links are reversed to reflect the change in adapting to the new reference level. This has the same effect as reversing the direction of one or more links when a node has no downstream links.

Timing is an important factor for TORA because the "height" metric is dependent on the logical time of a link failure; TORA assumes that all nodes have synchronized clocks (accomplished via an external time source such as the Global Positioning System). TORA's metric is a quintuple comprising five elements, namely:

- Logical time of a link failure,
- The unique ID of the node that defined the new reference level,
- A reflection indicator bit,
- A propagation ordering parameter,
- The unique ID of the node.

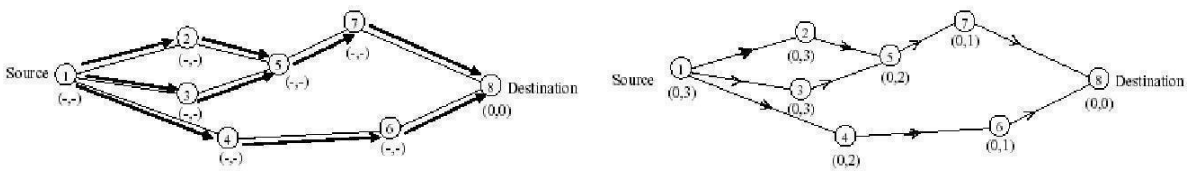


Figure 5(a) – Propagation of the query message

Figure 5(b) – Node's height updated as a result of the update message

The first three elements collectively represent the reference level. A new reference level is defined each time a node loses its last downstream link due to a link failure. TORA's route erasure phase essentially involves flooding a broadcast clear packet (CLR) throughout the network to erase invalid routes. In TORA, there is a potential for oscillations to occur, especially when multiple sets of coordinating nodes are concurrently detecting partitions, erasing routes, and building new routes based on each other (Figure 6). Because TORA uses inter-nodal coordination, its instability is similar to the "count-to-infinity" problem, except that such oscillations are temporary and route convergence ultimately occurs. Note that TORA is partially proactive and partially reactive. It is reactive in the sense that route creation is initiated on demand. However, route maintenance is done on a proactive basis such that multiple routing options are available in case of link failures.

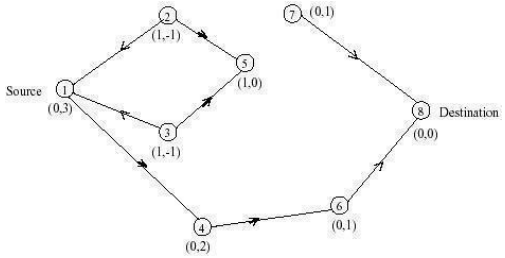


Figure 6 – Route maintenance in TORA

Hybrid Routing Protocols

Zone Routing Protocol

Zone Routing Protocol (ZRP) [Haas 1998] is a hybrid example of reactive and proactive schemes. It limits the scope of the proactive procedure only to the node's local neighborhood, while the search throughout the network, although it is global, can be performed efficiently by querying selected nodes in the network, as opposed to querying all the network nodes. In ZRP, a node proactively maintains routes to destinations within a local neighborhood, which is referred to as a routing zone and is defined as a collection of nodes whose minimum distance in hops from the node in question is no greater than a parameter referred to as zone radius. Each node maintains its zone radius and there is an overlap of neighboring zones.

The construction of a routing zone requires a node to first know who its neighbors are. A neighbor is defined as a node that can communicate directly with the node in question and is discovered through a MAC level Neighbor discovery protocol (NDP). The ZRP maintains routing zones through a proactive component called the Intrazone routing protocol (IARP) which is implemented as a modified distance vector scheme. On the other hand, the Interzone routing protocol (IERP) is responsible for acquiring routes to destinations that are located beyond the routing zone. The IERP uses a query-response mechanism to discover routes on demand.

The IERP is distinguished from the standard flooding algorithm by exploiting the structure of the routing zone, through a process known as bordercasting. The ZRP provides this service through a component called Border resolution protocol (BRP).

The network layer triggers an IERP route query when a data packet is to be sent to a destination that does not lie within its routing zone. The source generates a route query packet, which is uniquely identified by a combination of the source node's ID and request number. The query is then broadcast to all the source's peripheral nodes. Upon receipt of a route query packet, a node adds its ID to the query. The sequence of recorded node Ids specifies an accumulated route from the source to the current routing zone.If the destination does not appear in the node's routing zone, the node border casts the query to its peripheral nodes. If the destination is a member of the routing zone, a route reply is sent back to the source, along the path specified by reversing the accumulated route. A node will discard any route query packet for a query that it has previously encountered. An important feature of this route discovery process is that a single route query can return multiple route replies. The quality of these returned routes can be determined based on some metric. The best route can be selected based on the relative quality of the route.

Fisheye State Routing (FSR)

The Fisheye State Routing (FSR) protocol [Iwata 1999] introduces the notion of multi-level fisheye scope to reduce routing update overhead in large networks. Nodes exchange link state entries with their neighbors with a frequency which depends on distance to destination. From link state entries, nodes construct the topology map of the

entire network and compute optimal routes. FSR tries to improve the scalability of a routing protocol by putting most effort into gathering data on the topology information that is most likely to be needed soon. Assuming that nearby changes to the network topology are those most likely to matter, FSR tries to focus its view of the network so that nearby changes are seen with the highest resolution in time and changes at distant nodes are observed with a lower resolution and less frequently. It is possible to think the FSR as blurring the sharp boundary defined in the network model used by ZRP.

Landmark Routing (LANMAR) for MANET with Group Mobility

Landmark Ad Hoc Routing (LANMAR) [Pei 2000] combines the features of FSR and Landmark routing. The key novelty is the use of landmarks for each set of nodes which move as a group (viz., a group of soldiers in a battlefield) in order to reduce routing update overhead. Like in FSR, nodes exchange link state only with their neighbors. Routes within Fisheye scope are accurate, while routes to remote groups of nodes are "summarized" by the corresponding landmarks. A packet directed to a remote destination initially aims at the Landmark; as it gets closer to destination it eventually switches to the accurate route provided by Fisheye. In the original wired landmark scheme [Tsuchiya 1988], the predefined hierarchical address of each node reflects its position within the hierarchy and helps find a route to it. Each node knows the routes to all the nodes within it hierarchical partition. Moreover, each node knows the routes to various "landmarks" at different hierarchical levels. Packet forwarding is consistent with the landmark hierarchy and the path is gradually refined from top-level hierarchy to lower levels as a packet approaches the destination.

LANMAR borrows from [Tsuchiya 1988] the notion of landmarks to keep track of logical subnets. A subnet consists of members which have a commonality of interests and are likely to move as a "group" (viz., soldiers in the battlefield, or a group of students from the same class). A "landmark" node is elected in each subnet. The routing scheme itself is modified version of FSR. The main difference is that the FSR routing table contains "all" nodes in the network, while the LANMAR routing table includes only the nodes within the scope and the landmark nodes. This feature greatly improves scalability by reducing routing table size and update traffic overhead. When a node

needs to relay a packet, if the destination is within its neighbor scope, the address is found in the routing table and the packet is forwarded directly. Otherwise, the logical subnet field of the destination is searched and the packet is routed towards the landmark for that logical subnet. The packet however does not need to pass through the landmark. Rather, once the packet gets within the scope of the destination, it is routed to it directly.

The routing update exchange in LANMAR routing is similar to FSR. Each node periodically exchanges topology information with its immediate neighbors. In each update, the node sends entries within its fisheye scope. It also piggy-backs a distance vector with size equal to the number of logical subnets and thus landmark nodes.

Through this exchange process, the table entries with larger sequence numbers replace the ones with smaller sequence numbers.

FORWARDINGSTRATEGIES:

RFDR PROTOCOL

Restricted Flooding and Directional Routing (RFDR) is new proposed protocol that restricts the broadcast region will reduce routing packets, packet collisions and lowers the delay with more percentage of packet delivered.

2.1 Restricted Flooding (RF)

The main approach in restricted flooding is to restrict the flooding region. Restriction depends on distance, angle and distance covered by the next intermediate All the proposed protocols shown in figure 2 require quite complex mathematical computation of the distance, angle and coverage at all intermediate nodes to determine the nodes' participation. Information of the source and destination are required and must be inserted in the incoming packet. In MANET, route discovery is initiated by total flooding of route request (RREQ) messages that consume a large portion of the already limited bandwidth in MANET.

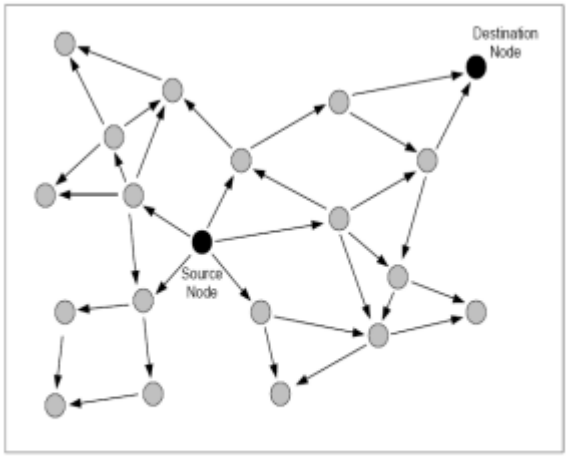


Fig. 3. Route Request (RREQs) broadcast based on Total Flooding.

As illustrated in Figure three, route request RREQ is broadcasted to all neighbours whereby frequent broadcast causes network congestion and degrades the performance of routing protocol. As we suggest utilizing restricted flooding mechanism to optimize the route establishment phase of AODVbis.

Restricted flooding is broadcasting messages to a selected number of nodes which is more than one that are located in an area in the vicinity of the destination. Location information of the destination can be obtained from any location service while location of the destination can be obtained with the aid of any other self-positioning system proposed for MANET. Then if this information is piggybacked in the reply or query packet, nodes will calculate its location with reference to the source and destination and will then decide to broadcast the query or not. Figure 4 illustrates that the same network topology .

but with restricted flooding. RREQ packets will be broadcast by nodes located in the request zone which is a quadrant drawn with respect to source node coordinates. Nodes participation is denoted by shaded circles with arrows indicating the direction of broadcast while lesser-toned circles indicate non-participating nodes. With this unique approach of using quadrant as the broadcast region, we proposed Quadrant-Based Directional Routing or RFDR.

Quadrant Based Directional Routing

QBDR is a restricted flooding routing that concentrates on a specified zone using location information provided by a location service. It restricts the broadcast region to all nodes in the same quadrant as the source and destination and does not require maintenance of a separate neighbours table at each node as in [3, 4, 5 & 6]. QBDR determines the quadrant of the current node based on the coordinates of source, destination and the current node that will direct the packet towards the destination. Even though [4] uses all these information to determine the distance or area covered, it requires trigonometric computations which will further incur delay if computed in kernel space. Decision to broadcast.

QBDR, the route request (RREQ) packet which contains the coordinates of the source and destination will be the only information the current node needs to decide to participate in the routing or not. The decision to participate at each node is made immediately as the node receives the RREQ packet and a neighbours table is not required to make the decision. RFDR will significantly reduce not only energy but also reduce the probability of packet collisions of messages rebroadcast by neighbours using the same transmission channel. This will result in

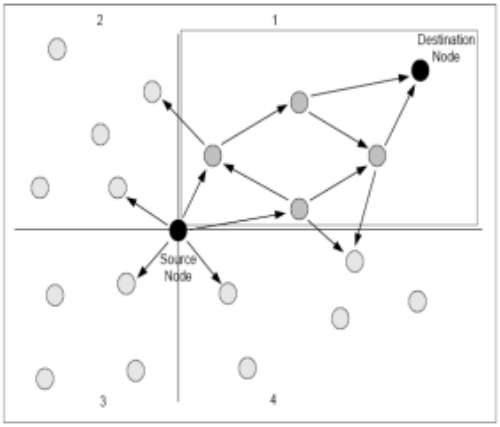


Fig.4. Route Request (RREQ) broadcast based on Restricted Flooding.

Distance Routing Effect Algorithm for Mobility

DREAM (Distance Routing Effect Algorithm for Mobility) [Basagni 1998] is a routing protocol for ad hoc networks built around two novel observations. One, called the distance effect, uses the fact that the greater the distance separating two nodes, the slower they appear to be moving with respect to each other. Accordingly, the location information in routing tables can be updated as a function of the distance separating nodes without compromising the routing accuracy. The second idea is that of triggering the sending of location updates by the moving nodes autonomously, based only on a node's mobility rate.Intuitively, it is clear that in a directional routing algorithm, routing information about the slower moving nodes needs to be updated less frequently than that about highly mobile nodes. In this way each node can optimize the frequency at which it sends updates to the networks and correspondingly reduce the bandwidth and energy used, leading to a fully distributed and self-optimizing system. Based on these routing tables, the proposed directional algorithm sends messages in the "recorded direction" of the destination node, guaranteeing delivery by following the direction with a given probability.

Routing Using Location Information

In this section we discuss some ad hoc routing protocols that take advantage of some sort of location information in the routing process.

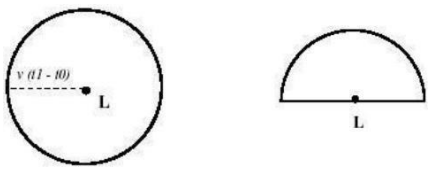
Location-Aided Routing

The Location-Aided Routing (LAR) [Ko 1998] protocol exploits location information to limit the scope of route request flood employed in protocols such as AODV and DSR. Such location information can be obtained through GPS (Global Positioning System). LAR limits the search for a route to the so-called request zone, determined based on the expected location of the destination node at the time of route discovery. Two concepts are important to understand how LAR works: Expected Zone and Request Zone.

Let us first discuss what is an Expected Zone. Consider a node S that needs to find a route to node D. Assume that node S knows that node D was at location L at time t0, and that the current time is t1. Then, the ―expected zone‖ of node D, from the

viewpoint of node S at time t1, is the region expected to contain node D. Node S can determine the expected zone based on the knowledge that node D was at location L at time t0. For instance, if node S knows that node D travels with average speed v, then S may assume that the expected zone is the circular region of radius v(t1 - t0), centered at location L (see Figure 7(a)). If actual speed happens to be larger than the average, then the destination may actually be outside the expected zone at time t1. Thus, expected zone is only an estimate made by node S to determine a region that potentially contains D at time t1.

If node S does not know a previous location of node D, then node S cannot reasonably determine the expected zone (the entire region that may potentially be occupied by the ad hoc network is assumed to be the expected zone). In this case, LAR reduces to the basic flooding algorithm. In general, having more information regarding mobility of a destination node can result in a smaller expected zone. For instance, if S knows that destination D is moving north, then the circular expected zone in Figure 7(a) can be reduced to the semi-circle of Figure 7(b).



(a) (b) Figure 7 – Examples of expected zone

Based on the expected zone, we can define the request zone. Again, consider node S that needs to determine a route to node D. The proposed LAR algorithms use flooding with one modification. Node S defines (implicitly or explicitly) a request zone for the route request. A node forwards a route request only if it belongs to the request zone (unlike the flooding algorithm in AODV and DSR). To increase the probability that the route request will reach node D, the request zone should include the expected zone (described above). Additionally, the request zone may also include other regions around

the request zone.

Based on this information, the source node S can thus determine the four corners of the expected zone. S includes their coordinates with the route request message transmitted when initiating route discovery. When a node receives a route request, it discards the request if the node is not within the rectangle specified by the four corners included in the route request. For instance, in Figure 8, if node I receives the route request from another node, node I forwards the request to its neighbors, because I determines that it is within the rectangular request zone. However, when node J receives the route request, node J discards the request, as node J is not within the request zone (see Figure 8).

The algorithm just described in the called LAR scheme 1. The LAR scheme 2 is a slight modification to include two pieces of information within the route request packet: assume that node S knows the location (Xd; Yd) of node D at some time t0 – the time at which route discovery is initiated by node S is t1, where t1 ≥ t0. Node S calculates its distance from location (Xd; Yd), denoted as DISTS, and includes this distance with the route request message. The coordinates (Xd; Yd) are also included in the route request packet. With this information, a given node J forwards a route request forwarded by I (originated by node S), if J is within an expected distance from (Xd; Yd) than node I.

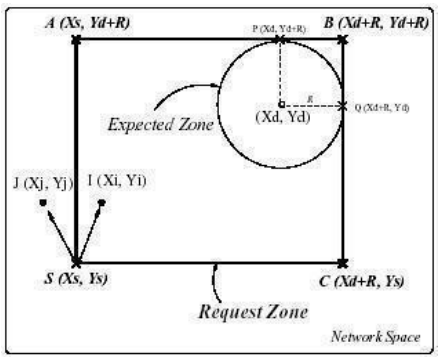


Figure 8 – LAR scheme .

Comparison Table

Table 2 summarizes the main characteristics of the most cited protocols discussed so far.

Table 2 – Protocol characteristics

Routing Protocol	Route Acquisition	Flood for Route Discovery	Delay for Route Discovery	Multipath Capability	Upon Route Failure
DSDV	Computed a priori	No	No	No	Floods route updates throughout the network
WRP	Computed a priori	No	No	No	Ultimately, updates the routing tables of all nodes by exchanging MRL between neighbors
DSR	On-demand, only when needed	Yes. Aggressive use of caching often reduces flood scope	Yes	Not explicitly. The technique of salvaging may quickly restore a Route	Route error propagated up to the source to erase invalid path
AODV	On-demand, only when needed	Yes. Conservative use of cache to reduce flood scope	Yes	No, although recent research indicate viability	Route error broadcasted to erase invalid path
TORA	On-demand, only when needed	Usually, only one flood for initial DAG construction	Yes. Once the DAG is constructed, multiple paths are found	Yes	Error is recovered locally, and only when alternative routes are not available
LAR	On-demand, only when needed	Localized flood by using location information	Yes	No	Route error propagated up to the source
ZRP	Hybrid	Only outside a source's zone	Only if the destination is outside the source's zone	No	Hybrid of updating nodes' tables within a zone and propagating route error to the source

UNIT –II
DATA TRANSMISSION IN MANETS

UNIT-II

Data Transmission- Broadcast Storm Problem, **Rebroadcasting Schemes**-Simple-flooding, Probability-based Methods, Area-based Methods, Neighbor Knowledge-based: SBA, Multipoint Relaying, AHBP. **Multicasting: Tree-based:** AMRIS, MAODV; **Mesh-based:** ODMRP, CAMP; **Hybrid:** AMRoute**,** MCEDAR.

1. DATA TRANSMISSION

In general, any one of the mobile hosts has to broadcast the information to all its neighbors during data transmission. For far-away devices, the message is rebroadcasted which could cause collision if multiple devices broadcast at the same time and are in the neighborhood. This is also known as the *broadcasting storm problem*; we will discuss ways to perform efficient broadcasting of messages.

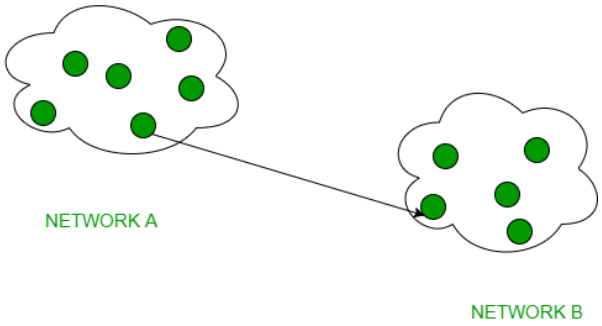
The MHs in the MANET share a single common channel with carrier sense multipleaccess (CSMA), but no collision detection (CD) capability (e.g., the IEEEstandard 802.11 [IEEE-802.111997]). Synchronization in such a network with mobility isunlikely, and global network topology information is unavailable to facilitate the scheduling of abroadcast. Thus, one straightforward and obvious solution is to achieve broadcasting by flooding(for example, as it is done by mostly all MANET routing algorithms). Unfortunately, it is observed that redundancy, contention, and collision could exist if flooding is doneblindly. Several problems arise in these situations including:

- As the radio propagation is omnidirectional (All directions) and a physical location may be covered by the transmission ranges of several hosts, many rebroadcasts are considered to be redundant.
- Heavy contention could exist because rebroadcasting hosts are close to each other; and
- As the RTS/CTS handshake is inapplicable for broadcast transmissions, collisions are more likely to occur as the timing of rebroadcasts is highly correlated.

a. UNICAST

This type of information transfer is useful when there is a participation of a single sender and a single recipient. So, in short, you can term it a one-to-one transmission. For example, if a device

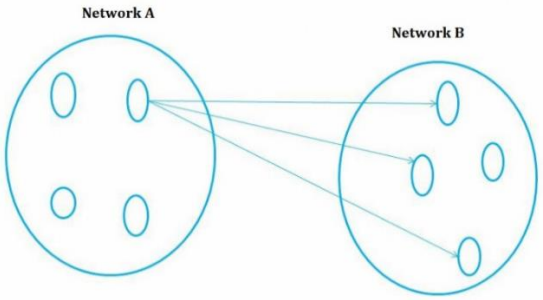
having IP address 10.1.2.0 in a network wants to send the traffic stream (data packets) to the device with IP address 20.12.4.2 in the other network, then unicast comes into the picture. This is the most usual form of data transfer over networks.



b. MULTICAST

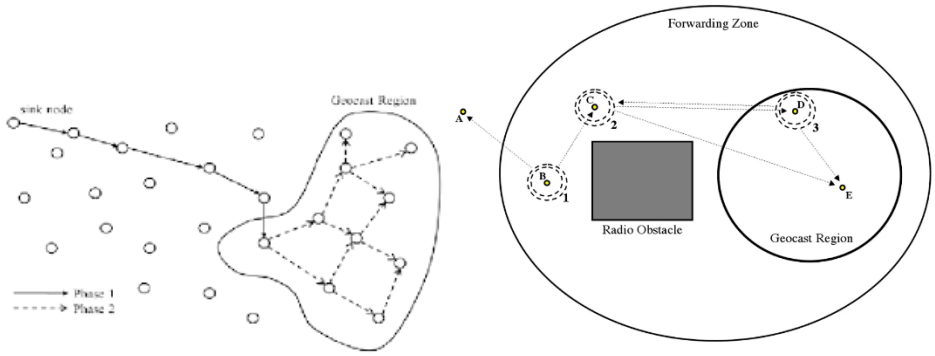
When the data is transmitted from a single source host to a specific group of hosts having the interest to receive the data, it is known as multicast transmission. Multicast can be more efficient than unicast when different groups of receivers need to see the same data.

Example − Multicast is the technique used in Internet streaming of video or audio teleconference, sending an email to a particular group of people, etc.



c. GEOCASTING

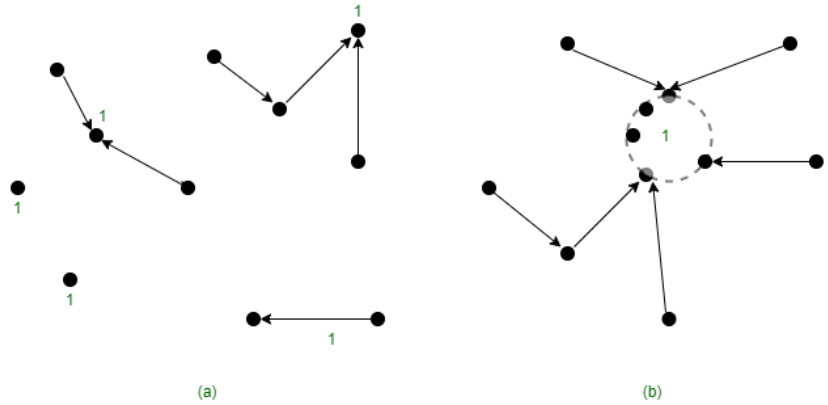
In a geocasting problem in ad hoc networks, a message is sent from one node to all the nodes located in a designated region. For example, monitoring center needs to contact all active sensors within a monitored area to either gather data from them periodically, or to provide its location to sensors covering certain area for event reporting. Intelligent flooding methods exist for this task when all active sensors belong to the monitored area. However, when a particular area containing only a small subset of active sensors needs to be monitored, the problem reduces to geocasting.



d. ANYCASTING

Anycast is a method for routing network traffic where the sender distributes packets to a destination that is adjacent to it in terms of network topology. The features of Anycastingis that the networking approach can allow for messages to be shared to a team of receivers that all have a similar destination address.

Suppose we want to anycast to the members of group 1. They will be given the address "1", instead of different addresses. Distance vector routing will distribute vectors as usual, and nodes will choose the shortest path to destination 1. This will result in nodes sending to the nearest instance of destination 1. That is, it believes that all the instances of node 1 are the same node, as in the topology.



e. BROADCATING

In Broadcast transmission, the data is transmitted from one or more senders to all the receivers within the same network or in other networks. This type of transmission is useful in network management packets such as ARP (Address Resolution Protocol) and RIP (Routing Information Protocol) where all the devices must see the data.

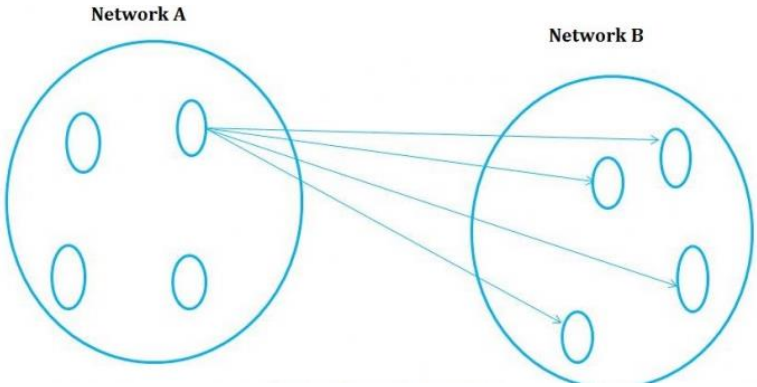
There are two types of broadcast transmission −

- Directed Broadcast, and
- Limited Broadcast

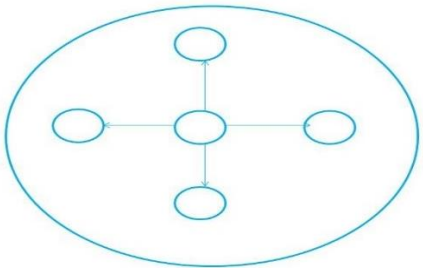
Directed Broadcast transmits data from one source host to all the other hosts that exist in some other

network. It is used in two scenarios −

- When the hosts are responsible for parsing data from broadcast packets.
- When all the hosts require the same data.



In Limited Broadcast, the data is transmitted from a single source host to all the other hosts residing in the same network. Suppose we have to send a stream of packets to all the devices over the network that you reside, this broadcasting comes in handy. For this to achieve, it will append 255.255.255.255 (all the 32 bits of IP address set to 1) called Limited Broadcast Address in the destination address of the datagram (packet) header which is reserved for information transfer to all the recipients from a single client (sender) over the network.



f. BROADCAST STORM PROBLEM

A broadcast storm is an abnormally high number of broadcast packets within a short period of time. They can overwhelm switches and endpoints as they struggle to keep up with processing the flood of packets. When this happens, network performance degrades, i.e. A broadcast storm occurs when a network system is overwhelmed by continuous multicast or broadcast traffic. When different nodes are sending/broadcasting data over a network link, and the other network devices are rebroadcasting the data back to the network link in response, this eventually causes the whole network to melt down and lead to the failure of network communication. There are many reasons a broadcast storm occurs, including poor technology, low port rate switches and improper network configurations. A broadcast storm is also known as a network storm.

For example, suppose there is a small LAN network consisting of three switches (Switch A, Switch

B and Switch C), and three network segments (Segment A, Segment B and Segment C). Two nodes are attached within this network. Node A is attached to Segment B, while Node B is directly attached to Switch A. Now, if Node B wants to transmit a data packet to Node A, then traffic is broadcast from Switch A over to Segment C; if this fails, then Switch A also broadcasts traffic over Segment A. Because Node A neither attaches to Segment C, nor Segment A, these switches would further create a flood to Segment B. If neither device/switch has learned the Node A address, then traffic is sent back to Switch A. Hence, all devices/switches keep sending and resending the traffic, eventually resulting in a flood loop or broadcast loop. The final result is that the network melts down, causing failure in all network links, which is referred to as a broadcast storm.

The following elements play an active role in the creation of a broadcast storm:

- Poor network management
- Poor monitoring of the network
- The use of cheap devices, including hubs, switches, routers, cables, connectors, etc.
- Improperly maintained network configuration and inexperienced network engineers
- The lack of a network diagram design, which is needed for proper management and to provide guidelines for all network traffic routes. This can be done on paper and with the help of application software that creates an automated network diagram.

How to reduce broadcast storms

- **Storm control and equivalent protocols** allow you to rate-limit broadcast packets. If your switch has such a mechanism, turn it on.
- **Ensure IP-directed broadcasts are disabled on your Layer 3 devices**. There is little to no reason why you would want broadcast packets coming in from the internet going to a private address space. If a storm originates from the WAN, disabling IP-directed broadcasts will shut it down.
- **Split up your broadcast domain**. Creating a new VLAN and migrating hosts into it will load and balance the broadcast traffic to a more acceptable level. Broadcast traffic is necessary and useful, but too much of it eventually leads to poor network experience.
- **Check how often ARP tables are emptied**. The more frequently they are emptied, the more often ARP broadcast requests occur.
- **Sometimes, when switches have a hardware failure**, their switch ports begin to spew out broadcast traffic onto the network. If you have a spare switch of the same or similar model, clone the config of the active switch onto the spare and swap the hardware and cables during

a maintenance window. Does the storm subside? If it does, it was a hardware issue. If not, then you've to keep digging.

- **Check for loops in switches**. Say there was an unmanaged Layer 2 switch connected upstream to an unmanaged switch, and someone is connected a cable between two ports on the same unmanaged switch (let us say ports 1 and 2). The unmanaged switch will respond to all broadcasts multiple times and flood the broadcast domain with packets, causing a denial-of-service attack on the network.
    - **BPDU** and **PortFast** or equivalent features should be implemented as a best practice to prevent loops.
    - Discourage users from connecting unmanaged switches to managed switch ports by enforcing a maximum number of MAC addresses per port. This may be up to two MAC addresses if users have a computer plugged into an IP phone, which in turn is plugged into the switch.

2. REBROADCASTING SCHEMES

Broadcasting is the simple and basic process in Mobile Ad Hoc Networks in which the same packet has been transmitted from the sender node to all the remaining nodes of the network. Due to the limited radio range of mobile nodes, the MANET is multihop in nature. Hence, the packet which is transmitted from the valid mobile source cannot reach the target in a single hop. So, here some other nodes of the network are required to forward the source transmitted packet to the destination. These nodes are often known as intermediate nodes. It can rebroadcast the packet to enable its delivery when the destination is in communication range though the intermediate nodes. This process is commonly known as "packet forwarding" or "flooding." In general, the broadcasting strategies can be grouped into four families: Simple flooding, Probability-based methods, Area-based methods, and Neighbor knowledge-based methods.

3. SIMPLE-FLOODING

In this method, a sender node initiates a message to all its neighbors. Each of these neighbors will check if they have seen this message before, if yes, the message will be dropped, if not the message will rebroadcasted at once to all their neighbors. The process goes on until all nodes have the message. This method is suitable for MANET with low density nodes and high mobility. It ensures no packet losses. But it may cause network congestion and quickly drain the battery power. Blind flooding ensures coverage; the broadcast packet is guaranteed to be received by every

node in the network. Redundant transmissions in blind flooding may cause the broadcast storm problem, in which redundant packets cause contention and collision.

a. PROBABILITY-BASED METHODS

Probability Based Flooding

When a node receives a broadcast message for the first time, the node rebroadcasts the message with a probability P. If the message received is already seen, then the node drops the message irrespective of whether the node retransmitted the message when received for the first time. Thus, randomly having some nodes not rebroadcast saves node and network resources without harming delivery effectiveness. Probabilistic broadcasting is one of the simplest and most efficient broadcast techniques. In this approach, each intermediate node rebroadcast received packets only with a predetermined forwarding probability. When the probability reaches 100% then it is identical to simple flooding.

Counter Based Methods

An inverse relationship between the number of times a packet is received at a node and the probability of that node being able to reach additional area on a rebroadcast. This result is the origin of their Counter-Based scheme. Upon reception of a broadcast packet, the node initiates a counter with a value of one and sets RDT (Random Delay Timer). During the RDT, the counter is incremented by one for each redundant packet received. If the counter is less than a threshold value when the RDT expires, the packet is rebroadcast. Otherwise, it is simply dropped.

b. AREA-BASED METHODS

Distance based approach

In Distance based approach, a node compares the distance between itself and each neighboring node that has previously forwarded a given packet. Upon reception of a previously unseen packet, a Random Assessment Delay (RAD) is initiated, and redundant packets are cached. When the RAD expires, all source node locations are examined to see if any node is closer than a threshold distance value. If true, the node does not rebroadcast. So, a node using the distance-based approach needs the information of the geographic locations of its neighbors‟ to make a rebroadcast decision. Measuring the distance of the source of the received packet may be accomplished by physical layer parameter i.e., signal strength at the node. Otherwise, if a GPS receiver is available, the location information can be included in each packet of the nodes that are transmitted.

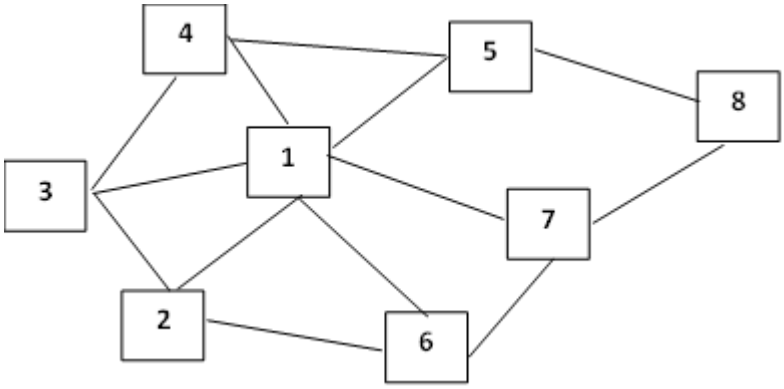
Location based Method

In this method each node has to identify its own location relative to the location of the sender using the geo location technique e.g., Global Positioning System. Each node in a MANET will add its own location to the header of each message it sends or rebroadcasts. When a neighboring node receives the packet, it notes the location of the sender and computes the additional coverage area obtainable if it were to rebroadcast. If the additional coverage area to rebroadcast is less than the given threshold, the node will not rebroadcast, and the same packets are ignored. Otherwise, the node assigns an RDT before delivery. During RAD, a redundant packet is received by a node then it is recomputed to the additional coverage area and compares that value to the threshold. The comparison of the area calculation and threshold occurs for all redundant broadcasts received until the packet reaches either the scheduled send time or is dropped.

c. NEIGHBOR KNOWLEDGE-BASED

Self-Pruning

Self-Pruning is the simplest method in reducing broadcast redundancy. In this method, each node requires knowledge of its 1-hop neighbors, which is acquired by sending periodic “Hello” packets. A node contains a list of known neighbors in the header of each broadcast packet. Broadcast packet received by neighbor nodes checks whether their list matches Additional nodes, it rebroadcasts else it avoids doing rebroadcasting. In the below figure, after receiving a message from node 6 node 2 will rebroadcast the message to node 3 and node 1 as its only additional nodes. Note that node 1 also will rebroadcast the same message to node 3 as its only additional node. In this situation still the message redundancy takes place.



Dominant Pruning

Dominant pruning also uses two-hop neighbor knowledge, obtained via hello packets, for routing decisions. Unlike SBA, however, dominant pruning requires rebroadcasting nodes to proactively choose some or all its one-hop neighbors as rebroadcasting nodes. Whenever a node receives a broadcast packet, it checks the header to see if its address is a part of the list. If so, it now must determine which of its neighbors should rebroadcast its packet to include them in the packet header. For that, it uses a Greedy Set Cover algorithm given the knowledge of which neighbors have already been covered by the sender's broadcast. One such algorithm recursively chooses one-hop neighbors that cover all two-hop neighbors.

Multi-point Relaying

Multipoint Relaying is like Dominant Pruning; upstream senders will choose the rebroadcasting nodes. For example, Node A is instigating a broadcast packet. It will select some or all one hop neighbor nodes to rebroadcast packet that they receive from Node A. The chosen nodes are called Multipoint relays (MPRs). Each MPR is necessary to decide a subset of its one hop neighbors to act as MPRs. Since a node knows the network topology within a 2-hop radius, it can select 1-hop neighbors as MPRs that most efficiently reach all nodes within the two-hop neighborhood. For a node to choose it MPRs:

- 1-hop neighbor must discover all 2-hop neighbors that can be reached by them.1-hop neighbors must be allotted as MPRs.
- Decide the resultant cover set (i.e., the collection of 2-hop neighbors that will receive the packet from the current MPR set).
- From the left over 1-hop neighbors not yet in the MPR set, determine the one that would cover the majority 2-hop neighbors not in the cover set.
- Till all 2-hop neighbors are covered repeat step 2.

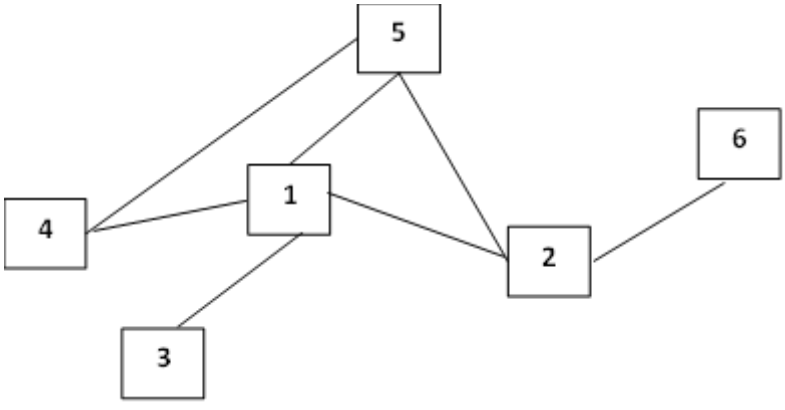
SBA

The Scalable Broadcast Algorithm (SBA) requires all the nodes to have knowledge about neighbors with two hop radius. With the neighbor knowledge, nodes determine whether it could reach additional nodes by rebroadcasting. By sending periodically "hello" packets, 2-hop neighbor knowledge is attainable; each "hello" packets have identifier of node and the list of known neighbors. Two hop radius node information are known by nodes after receiving "Hello" packets from neighbor nodes. Suppose Node A sends a broadcast packet to Node B. Node B knows all neighbors of Node A since it is a neighbor. Other than Node A"s broadcast if Node B has additional neighbors, it prepares the packet for delivery with a RAD. suppose redundant broadcast packet from another neighbor are received by Node B, it finds out whether can reach by rebroadcasting.

AD HOC BROADCASTING APPROACH (AHBP)

In this approach, only nodes selected as gateway nodes and a broadcast message header are allowed to rebroadcast the message. The approach is described as follows:

- Locate all two hop neighbors that can only be reached by one hop neighbor. Select these one hop neighbors as gateways.
- Calculate the cover set that will receive the message from the current gateway set for the neighbors not yet in the gateway set, find the one that would cover the most two hop neighbors not in the cover set. Set this one hop neighbor as a gateway.
- Repeat process 2 and 3 until all two hop neighbors are covered. When a node receives a message and is a gateway, this node determines which of its neighbors already received the message in the same transmission.



In Ad hoc broadcasting approach (figure), node 2 has 1, 5 and 6 nodes as one hop neighbors, 3 and 4 nodes as two hop neighbors. Node 3 can be reached through node 1 as a one hop neighbor of node 2. Node 4 can be reached through node 1 or node 5 as one hop neighbors of node 2. Node 3 selects node 1 as a gateway to rebroadcast the message to nodes 3 and 4. Upon receiving the

message node 5 will not rebroadcast the message as it is not a gateway.

4. MULTICASTING

Multicasting is the transmission of packets to a group of hosts identified by a single destination address. Multicasting is intended for group-oriented computing (broadcast a message to a subset of MANET MHs). The members of a host group are dynamic, that is hosts may join and leave groups at any time. There is no restriction on the location or number of members in a host group. A host may be a member of more than one group at a time. A host does not have to be a member of a group to send packets to it.

Protocols, designed for fixed networks, may fail to keep up with node movements and frequent topological changes as MHs become increasingly mobile, these protocols need to evolve to cope with the new environment. But the host mobility increases the protocol overheads. The broadcast protocols cannot be used either as multicasting requires a selected set of nodes to receive the message while all multicast algorithms depend on the topology of the network and do not consider whether a node belongs to a group or not. With this approach, data packets are sent throughout the ad hoc network and every node that receives this packet broadcasts it to all its immediate neighbors' nodes exactly once.

We can classify the protocols into four categories based on how route to the members of the group is created:

- Tree-Based Approaches
- Meshed-Based Approaches
- Stateless Multicast
- Hybrid Approaches

a. TREE-BASED Approaches

Most of the schemes for providing multicast in wired network are either source-based or shared tree-based. Due to simplicity of tree structures, many characteristics can be identified such as: a packet traverses each hop and node in a tree at most once, very simple routing decisions at each node, and the number of copies of a packet is minimized,

tree structure built representing shortest paths amongst nodes, and a loop-free data distribution structure.

On the other hand, there are many issues that must be addressed in tree-based approaches. Trees provide a unique path between any two nodes. Therefore, having even one link failure could mean reconfiguration of the entire tree structure and could be a major drawback. In addition, multiple packets generated by different sources will require some consideration when utilizing multicast trees such that efficient routing can be established and maintained. Thus, it is common to consider the use of either a shared tree or establish a separate tree per each source

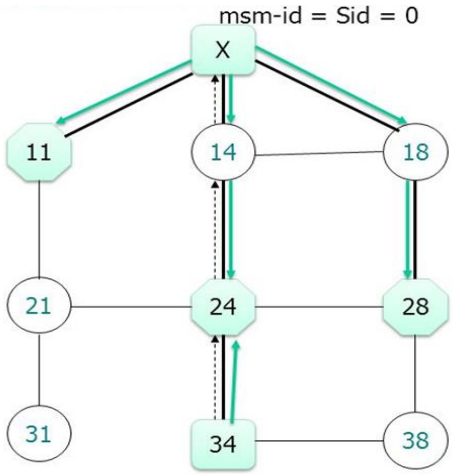
For separate source trees, each router or node involved in multiple router groups must maintain a list of pertinent information for each group in which it is involved. Such management per router is inefficient and not scalable. On the other hand, for shared trees, there is a potential that packets may not only traverse shorter paths, but in fact may be routed on paths with much longer distances than the shortest paths. While any scheme has positive and negative sides, the simple structure coupled with ease of approach has made multicast trees the primary method for realizing multicasting on the Internet.

i. AMRIS

Ad hoc Multicast Routing Protocol utilizing *Increasing id-numberS (AMRIS)* is an on-demand protocol, which constructs a shared multicast delivery tree to support multiple senders and receivers in a multicast session. AMRIS dynamically assigns an id- number to each node in each multicast session. Based on the id-number, a multicast delivery tree rooted at a special node with Sid (Smallest-ID) is created and the id-number increases as the tree expands from the Sid. Generally, Sid is the source or the node that initiates a multicast session.

The first step in AMRIS protocol operation is the selection of Sid. If there is only one sender for a group, the Sid is the source of the group. In the case of multiple senders, a Sid is selected from the given set of senders. Once a Sid is identified, it sends a *NEW-SESSION* message to its neighbors. The content of this message includes Sid's *msm-id* (multicast session member id) and the routing metrics. Nodes receiving the *NEW-SESSION* message generate their own *msm-ids*, which are larger than the *msm-id* of the sender. In case a node receives multiple *NEW-SESSION* messages from different nodes, it keeps the message with the best routing metrics and calculates its *msm-ids*. To join an ongoing session, a node checks the *NEW-SESSION* message, determines a parent with smallest msm-ids, and unicast

a *JOIN-REQ* to its potential parent node. If the parent node is already in the multicast delivery tree, it replies with a *JOIN-ACK*. Otherwise, the parent itself tries to join the multicast tree by sending a *JOIN-REQ* to its parent. If a node is unable to find any potential parent node, it executes a branch reconstruction (BR) process to rejoin the tree. BR consists of two sub-routines, namely, subroutines 1 (BR1) and 2 (BR2). BR1 is executed when a node has potential parent node for a group. In case it does not find any potential parent node, BR2 is executed. In BR2, instead of sending a unicast *JOIN_REQ* to a potential parent node, the node broadcasts a *JOIN-REQ* which consists of a range field R to specify the nodes till R hops. Upon link breakage, the node with larger *msm-id* tries to rejoin the tree by executing any of the BR mechanisms. It is to be noted that AMRIS detects the link disconnection by a beaconing mechanism. Hence, until the tree is reconstructed there is a possibility of packets being dropped.



X, 34 -> Receivers / Senders, 14, 18 --> I- Nodes, 11,24,28 --> Receivers, 21,31,38 --> U-Nodes

ii.MAODV

The Multicast Ad hoc On-Demand Distance Vector (MAODV) protocol enables dynamic, self-starting, multihop routing between participating mobile nodes wishing to join or participate in a

multicast group within an ad hoc network. The membership of these multicast groups is free to change during the lifetime of the network. MAODV enables mobile nodes to establish a tree connecting multicast group members. Mobile nodes can respond quickly to link breaks in multicast trees by repairing these breaks in a timely manner. In the event of a network

partition, multicast trees are established independently in each partition, and trees for the same multicast group are quickly connected if the network components merge.

One distinguishing feature of MAODV is its use of sequence numbers for multicast groups. Each multicast group has its own sequence number, which is initialized by the multicast group leader and incremented periodically. Using these sequence numbers ensures that routes found to multicast groups are always the most current ones available. Given the choice between two routes to a multicast tree, a requesting node always selects the one with the greatest sequence number.

The AODV multicast algorithm uses similar RREQ and RREP messages as in unicast operation. The nodes join the multicast group on-demand, and a multicast tree is created in the process. The tree consists of the group members and nodes connected to the group members. This enables a recipient host to join a multicast group even if it is more than one hop away from a multicast group member. The unicast operation of the protocol also benefits from the information that is gathered while discovering routes for multicast traffic. This cuts down the signaling traffic in the network.

A MH originates an RREQ message when it wishes to join a multicast group, or when it has data to send to a multicast group, but it does not have a route to that group. Only a member of the desired multicast group may respond to a join RREQ

If the RREQ is not a join request, any node with a fresh route (based on group sequence number) to the multicast group may respond. If an intermediate node receives a join RREQ for a multicast group of which it is not a member or if it receives a RREQ and it does not have a route to that group, it rebroadcasts the RREQ to its neighbors.

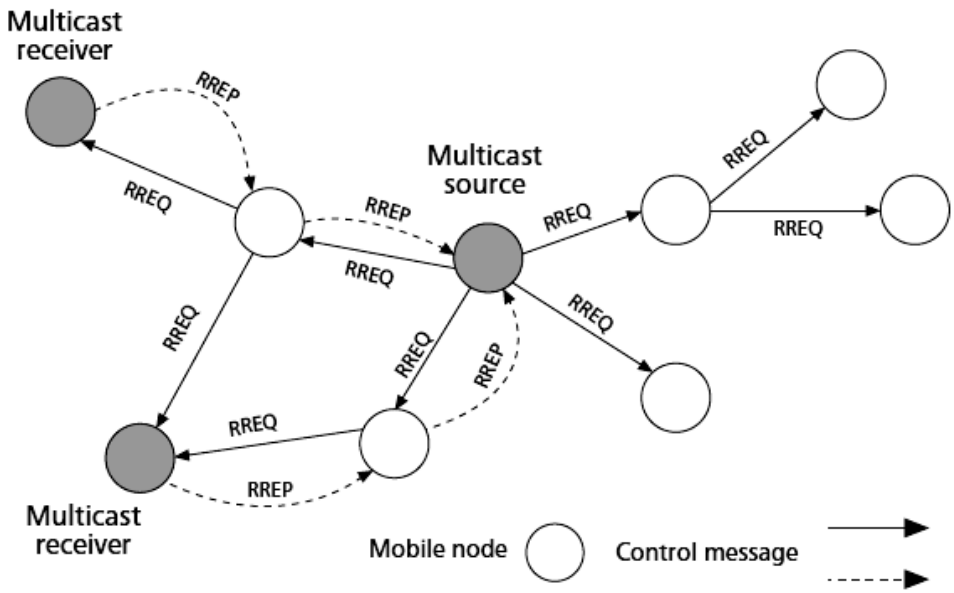
Route Request Message Format:

0 - 7	8	9	10	11 – 23	24 - 31
Type	J	R	G	Reserved	Hop Count
Other fields as in AODV					

J -- > Join flag; set when source node wants to join a multicast group **.**

R--> Repair flag, set when a node is responding to a repair request to connect two previously disconnected portions of the multicast tree.

G--> Group Leader flag **;**



Route Discovery:

When a node wishes to find a route for a multicast group, it sends an RREQ message. The destination address in the RREQ message is set to the address of the multicast group. If the node wants to join the group in question, the J-flag in the message is set. Any node may respond to a RREQ merely looking for a route, but only a router in the desired multicast tree may respond to a join RREQ. The corresponding RREP message may travel through nodes that are not members of the multicast group. This means that the eventual route may also include hops through non-member nodes

Route Reply:

The multicast RREP message is slightly different from the unicast RREP. The address of the multicast group leader is stored in a field called Group-Leader-Addr. In addition, there is a field called Mgroup-hop. This field is initialized to zero and it is incremented at each hop along the route. Mgroup-hop contains the distance in hops of the source node to the nearest member of the multicast tree.

Because the protocol relies on a group-wide destination sequence number (DSN) to ensure fresh routes, the group leader broadcasts periodical Group Hello messages. The Group Hello is an unsolicited RREP message that has a TTL (Time to Leave) greater than the diameter of the network. The message contains extensions that indicate the multicast group addresses and corresponding sequence numbers of all the groups for which the node is the group leader. The sequence number for each group is incremented each time the *Group Hello* is broadcast. The *Hop_Cnt* field in the message is initialized as zero and incremented by the intermediate nodes.

The nodes receiving the Group Hello use the information contained therein to update their request tables. If a node does not have an entry for the advertised multicast group, one is inserted. The hop counts are used to determine the current distance from the group leader.

0 - 7	8	9 - 18	19 – 23	24 - 31
Type	R	Reserved	Prefix Size	Hop Count
Other fields as in AODV				

R--> Repair flag, set when a node is responding to a repair request to connect two previously disconnected portions of the multicast tree.

b. MESH-BASED Approaches

Mesh-based multicast protocols may have multiple paths between any source and receiver pairs. The tree-based protocols are not necessarily the best suited for multicast in a MANET environment if the network topology changes frequently. In such an environment, mesh-based protocols outperform tree-based proposals due to availability of alternative paths, which allow multicast datagrams to be delivered to the receivers even if links fail.

The disadvantage of a mesh is the increase in data-forwarding overhead. The redundant forwarding consumes more bandwidth in the bandwidth constrained ad hoc networks. Moreover, the probability of collisions is higher when a larger number of packets are generated. Therefore, one common problem in mesh-based protocols must consider is how to minimize the data-forwarding overhead caused by flooding. As we shall see, different protocols attack this issue in different ways using forwarding groups, cores, and so on. There are multiple protocols of the mesh-based approaches that support multicast in MANETs.

i. ODMRP

ODMRP (On-Demand Multicast Routing Protocol) is a mesh-based, multicast protocol that provides richer connectivity among multicast members. By building a mesh and supplying multiple routes, multicast packets can be delivered to destinations in the face of node movements and topology changes. In addition, the drawbacks of multicast trees in mobile wireless networks (e.g., intermittent connectivity, frequent tree reconfiguration, traffic concentration, etc.) are avoided. To establish a mesh for each multicast group, ODMRP uses the concept of forwarding group. The forwarding group is a set of nodes responsible for forwarding multicast data on shortest paths between any member pairs. ODMRP also applies OnDemand routing techniques to avoid channel overhead and improve scalability. No explicit control message is required to leave a group.

When a multicast source has packets to send, but no route to the multicast group, it broadcasts a *Join-Query* control packet to the entire network. This *Join-Query* packet is periodically broadcasted to refresh the membership information and updates routes. When an intermediate node receives a *Join-Query* packet, it stores the source ID and the sequence number in its message cache to detect any potential duplicates. The routing table is updated with an appropriate node ID (i.e., backward learning) from which the message was received. If the message is not a duplicate and the TTL is greater than zero, it is rebroadcasted.

When a *Join-Query* packet reaches a multicast receiver, it creates and broadcasts a Join-Reply to its neighbors. When a node receives a Join-Reply, it checks if the next hop 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 a part of the forwarding group and sets the FG_FLAG (Forwarding Group Flag). It then broadcasts its own *Join-Reply* built upon matched entries. The next hop node ID field contains the information extracted from its routing table. In this way, each forward group member propagates the *Join-Reply* until it reaches the multicast source via the selected (shortest) path. This whole process constructs (or updates) the routes from sources to receivers and builds a mesh of nodes. After establishing a forwarding group and route construction process, a source can multicast packets to receivers via selected routes and forwarding groups. While a node has data to send, the source periodically sends Join-Query packets to refresh the forwarding group and the routes. When receiving the multicast data packet, a node forwards it only when it is not a duplicate and the setting of the FG_FLAG for the multicast group has not expired. This procedure minimizes the traffic overhead and prevents sending packets through stale routes.

In ODMRP, no explicit control packets need to be sent to join or leave the group. If a multicast source wants to leave the group, it simply stops sending Join-Query packets since it does not have any multicast data to send to the group. If a receiver no longer wants to receive from a particular multicast group, it does not send the Join-Reply for that group. Nodes in the forwarding group are demoted to non-forwarding nodes if not refreshed (no Join-Replies received) before they timeout.

ii. CAMP

The *Core-Assisted Mesh Protocol (CAMP)* supports multicasting by creating a shared mesh for each multicast group. Meshes thus created helps in maintaining the connectivity to the multicast users, even in case of node mobility. It borrows concepts from CBT (Core Based Tree Protocol), but the core nodes are used for control traffic needed to join multicast groups. The basic operation of the CAMP includes building and maintaining the multicast mesh for a multicast group. It assumes a mapping service, which provides routers with the addresses of groups identified by their names. Each router maintains a routing table (RT) built with the unicast routing protocol and is modified by CAMP when a multicast group needs to be inserted or removed. A router may update its MRT based on topological changes or messages received from its neighbors.

CAMP classifies the nodes in the network in three modes: simplex, duplex and non-member. A router joins a group in a simplex mode if it intends only to send traffic received from specific nodes or neighbors to the rest of the group and does not intend to forward packets from the group. A duplex member forwards any multicast packets for the group, whereas a non-member node needs not to be in the multicast delivery mesh. CAMP uses a receiver-initiated method for routers to join a multicast group. If a router wishing to join a group has multiple neighbors that are duplex members of the multicast group, then it simply changes its MRT and directly announces to its neighbors that it is a new member for the multicast group using multicast routing update. If it has no neighbors that are members of the multicast group, it either propagates a join request to one of the multicast group "cores" or attempts to reach a member through expanding ring search. Any router that is a regular member of the multicast group and has received the join request, is free to transmit a join acknowledgement (ACK) to the sending router. A router can leave a group if it has no hosts that are members of the group, and it has no neighbors for whom it is an anchor, i.e., if they are not needed to provide efficient paths for the dissemination of packets in the multicast meshes for the groups. Cores are also allowed to leave multicast group if there are no routers using them as anchors.

CAMP ensures that the mesh contains all reverse shortest paths between a source and the recipients. A receiver node periodically reviews its packet cache in order to determine whether it is receiving data packets from neighbors, which are on the reverse shortest path to the source. Otherwise, a HEARTBEAT message is sent to the successor in the reverse shortest

path to the source. This HEARTBEAT message triggers a PUSH JOIN (PJ) message. If the successor is not a mesh member, the PJ forces the specific successor and all the routers in the path to join the mesh.

CAMP has the advantage that it does not use flooding and the requests only propagate to mesh members. On the other hand, CAMP relies on an underlying unicast routing protocol to guarantee correct distances to all destinations within a finite time.

iii. HYBRID

The protocols to provide multicast in ad hoc networks discussed so far, either address efficiency or robustness but not both simultaneously. The tree-based approaches provide high data forwarding efficiency at the expense of low robustness, whereas mesh-based approaches lead to better robustness (link failure may not trigger a reconfiguration) at the expense of higher forwarding overhead and increased network load.

Thus, there is a possibility that a hybrid multicasting solution may achieve better performance by combining the advantages of both tree and meshed-based approaches. The different hybrid approaches which enable ad hoc multicasting are discussed below.

a) AMROUTE

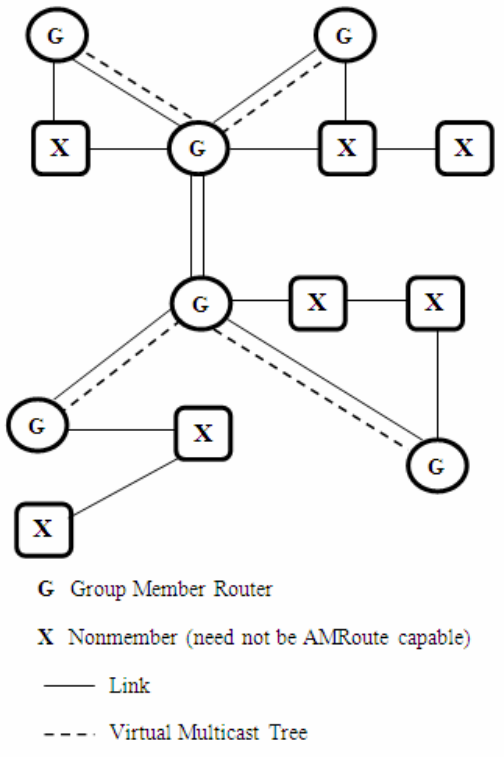
The Ad hoc Multicast Routing Protocol (AMRoute) creates a bi-directional, shared tree by using only group senders and receivers as tree nodes for data distribution. The protocol has two main components: mesh creation and tree setup.

The mesh creation identifies and designates certain nodes as logical cores, and these are responsible for initiating the signaling operation and maintaining the multicast tree to the rest of the group members. A non-core node only responds to messages from the core nodes and serves as a passive agent. The selection of logical core in AMRoute is dynamic and can migrate to any other member node, depending on the network dynamics and group membership. AMRoute does not address network dynamics and assumes the underlying unicast protocol to take care of it. To create a mesh, each member begins by identifying itself as a core and broadcasts JOIN_REQ packets with increasing TTL to discover other members. When a core receives JOIN_REQ from a core in a different mesh for the same group, it replies with a JOIN_ACK. A new bi-directional tunnel is created between the two cores and one of them is

selected as core after the mesh merger. Once the mesh has been established, the core initiates the tree creation process. The core sends out periodic TREE_CREATE messages along all links incident on its mesh. Using unicast tunnels, the TREE_CREATE messages are sent only to the group members. Group members receiving non-duplicate TREE_CREATE message forwards it to all mesh links except the incoming one and marks the incoming and outgoing links as tree links.

If a link is not going to be used as part of the tree, the TREE_CREATE is discarded and TREE_CREATE__NAK is sent back to incoming links. A member node, which wants to leave a group, can do so by sending a JOESLNAK message to its neighboring nodes.

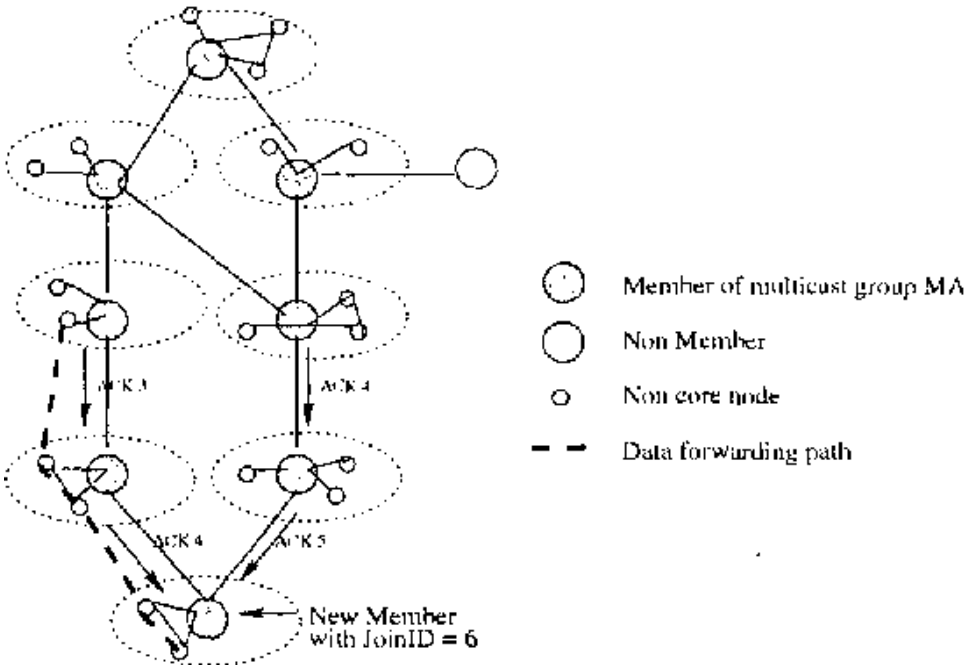
AMRoute employs virtual mesh links to establish the multicast tree, which helps in keeping the multicast delivery tree the same even with the change of network topology as long as routes between cores (nodes) and tree members exist via mesh links. The main disadvantage of this protocol is that it may have temporary loops and may create nonoptimal trees in case of mobility.



b) MCEDAR

The *Multicast Core-Extraction Distributed Ad hoc Routing* (MCEDAR) is a multicast extension to the CEDAR architecture. The main idea of MCEDAR is to provide the efficiency of the tree-based forwarding protocols and robustness of mesh-based protocols by combining these two approaches. It is worth pointing out that a source-based forwarding tree is created on a mesh. As such, this ensures that the infrastructure is robust and data forwarding occurs at minimum height trees. MCEDAR decouples the control infrastructure from the actual data forwarding to reduce the control overhead.

The underlying unicast protocol, CEDAR, provides the core broadcasting for multicasting. The core is used for routing management and link state inspection. Also, the cores make up the mesh infrastructure, which is referred to as an mgraph, and use joinlDs to perform the join operation. As MCEDAR uses a mesh as the underlying infrastructure, it can tolerate a few link breakages without reconfiguration. The efficiency is achieved by using a forwarding mechanism on the mesh that creates an implicit route-based forwarding tree. As mentioned earlier, this ensures that the packets need to travel only the minimum distance in the tree.



UNIT-III

GEOCASTING

Geocasting: Data-transmission Oriented-LBM; Route Creation Oriented-GeoTORA, MGR.TCP over AdHoc TCP protocol overview, TCP and MANETs, Solutions for TCP over Adhoc.

Geocasting

We now turn our attention to the problem of geocasting over MANETs. As we have mentioned earlier, geocasting is a variant of the conventional multicasting problem and distinguishes itself by specifying hosts as group members within a specified geographical region. In geocasting, the nodes eligible to receive packets are implicitly specified by a physical region and membership changes as mobile nodes move in or out of the geocast region.

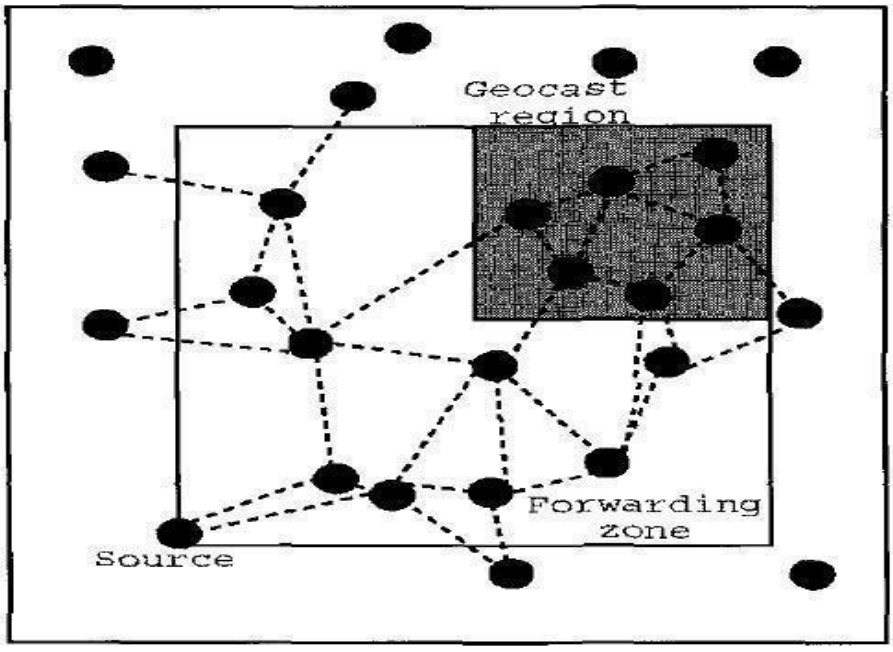
Protocol	Topology	Loop Free	Dependence on Unicast Protocol	Periodic Message	Control Packet Flooding Done/Required
Flooding	Mesh	Yes	No	No	No
AMRoute	Hybrid	No	Yes	Yes	Yes
AMRIS	Tree	Yes	No	Yes	Yes
MAODV	Tree	Yes	Yes	Yes	Yes
LAM	Tree	Yes	Yes	No	No
LGT-Based	Tree	Yes	No	Yes	No
ODMRP	Mesh	Yes	No	Yes	Yes
CAMP	Mesh	Yes	Yes	Yes	No
DDM	Stateless Tree	Yes	No	Yes	No
FGMP-RA	Mesh	Yes	Yes	Yes	Yes
FGMP-SA	Mesh	Yes	No	Yes	Yes
MCEDAR	Hybrid	Yes	Yes	Yes	Yes

from the source to the geocast region - and are described here.

3.4.1.1.1 Location-Based Multicast

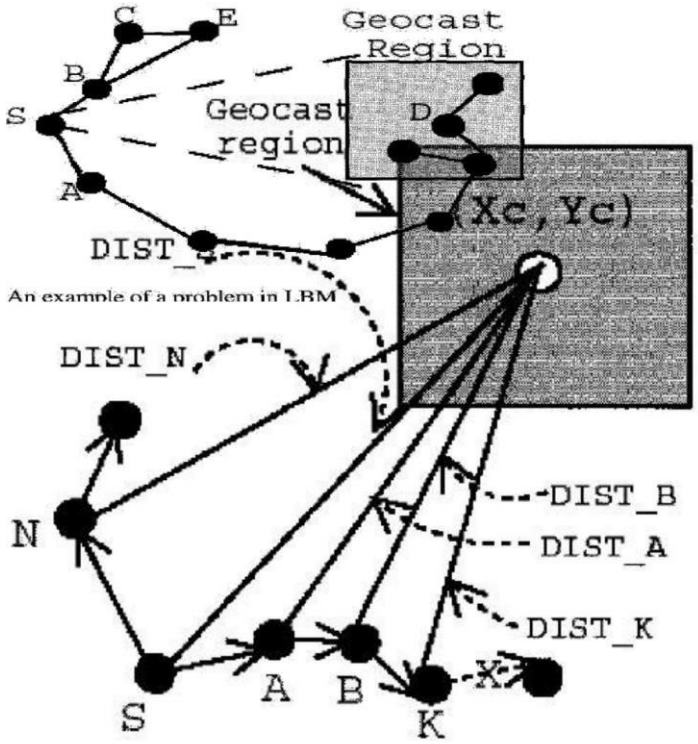
The Location-Based Multicast (LBM) protocol [Kol999] extends the LAR unicast routing algorithm for geocasting. As we have seen, LAR is an approach to utilize location information to improve the performance (i.e., higher data packet delivery ratio and lower overhead) of a unicast routing protocol in aMANET. Similarly, the goal of LBM is to decrease delivery overhead of geocast packets by reducing the forwarding space for geocast packets, while maintaining accuracy of data delivery. The LBM algorithm is based upon a flooding approach. LBM is essentially identical to flooding data packets, with the modification that a node determines whether to forward a geocast packet further via one of two schemes.

• **LBM Scheme 1:** When a node receives a geocast packet, it forwards the packet to its neighbors if it is within *& forwarding zone;* otherwise, it discards the packet. Thus, how to define the forwarding zone becomes the key point of this scheme. Figure 3.11 shows one example of a forwarding zone Boleng2001]. In Figure 3.11, the size of the forwarding zone is dependent on (i) the size of the geocast region and (ii) the location of the sender. In a BOX Forwarding Zone, the smallest rectangle that covers both the source node and the geocast region defines the forwarding zone. All the nodes in the forwarding zone forward data packets to their neighbors. Other kinds of forwarding zones are possible, such as the CONE Forwarding Zone [Boleng2001]. A parameter <5is discussed in [Kol999] to provide additional control on the size of the forwarding zone. When. *S* is positive, the forwarding zone is extended in both positive and negative X and Y directions by <5.(ie., each side increases by 2 *S*).



. BOX forwarding zone

LBM Scheme 2: Unlike scheme 1, in which a geocast packet is forwarded based on the forwarding zone, scheme 2 does not have a forwarding zone explicitly. Instead, whether a geocast packet should be forwarded isbased on the position of the sender node at the transmission of the packet and the position of the geocast region. That is, for some parameter *8*, node B forwards a geocast packet from node A (originated at node S), if node B is "at least. *S* closer" to the center of the geocast region *(Xc, Yc)* than node A. In other words, *DISTA* > *DISTB*+ *S*. We define *(Xc, Yc)* as the location of the geometrical center of the geocast region, and for any nodeZ, *DISTz* denotes the distance of node Z from *(Xc, Yc).* In Figure 3.12 [Kol999], node B will forward a geocast packet transmitted by node A since *DISTA* > *DISTB* and. *d*= 0. Node K will, however, discard a geocast packet transmitted by node B, since node K is not closer to *(Xc, Yc)* than node B. In brief, this protocol ensures that every packet transmission sends the packet closer to the destination. As for the performance, the accuracy (i.e., ratio of the number of geocast group members that actually receive the geocast packets to the number of group members that were supposed to receive the packets) of both LBM schemes is comparable with that of flooding geocast packets throughout the network. However, the number of geocast packets transmitted (a measure of the overhead) is consistently lower for LBM than simple flooding.

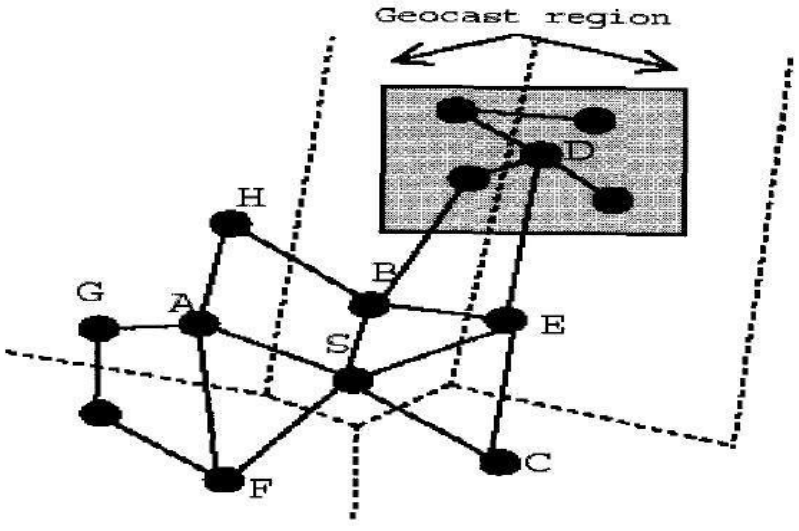


Forwarding zone in LBM scheme 2

Voronoi Diagram Based Geocasting

The goal of the Voronoi Diagram based Geocasting (VDG) protocol [Stojmenovicl999] is to enhance the success rate and decrease the hop count and flooding rate of LBM. It is observed that the forwarding zonedefined in LMB may be a partitioned network between the source node and the geocast region, although there exists a path between the source and the destination.

In VDG, the definition of the forwarding zone of LBM has been modified. The neighbors of node A thatare located within the forwarding zone in VDG are exactly those neighbors that are closest in the irection of the destination. This definition of a forwarding zone not only resolves the problem of having no nodes inside the forwarding zone, but also precisely determines the expansion of the forwarding zone. This forwarding zone can be implemented with a *Voronoi diagram* for a set of nodes in a given node's neighborhood of a MANET. A Voronoi diagram of *n* distinct points (i.e., *n* neighbors) in a plane is a partition of the plane into *n* Voronoi regions, which, when associated with node A, consists of all thepoints in the plane that are the closest to A. In other words, the *Voronoi diagram model* is a model where every point is assigned to a Voronoi region. The subdivision induced by this model is called the *Voronoidiagram* of the set of nodes [Berg]. For example, in Fig, five neighbors of source node S (A, B, C, E andF) carve up the plane into five Voronoi regions. The region associated with node A, consists of nodes G and H, since these two nodes are closer to node A than to any other node. The geocast region is



Example of a Voronoi diagram and the request zone

the rectangle with the center D. In Figure 3.14, the Voronoi regions of nodes B and E intersect the geocast region; thus, only nodes B and E will forward geocast packets from node S. Although there are not any simulations of the VDG algorithm, it is believed that VDG reduces the flooding rates of LBM Scheme 1,as fewer packets should be transmitted. On the other hand,

VDG may offer little improvement over LBM Scheme 2, as the end result of the two protocols appears to be similar.

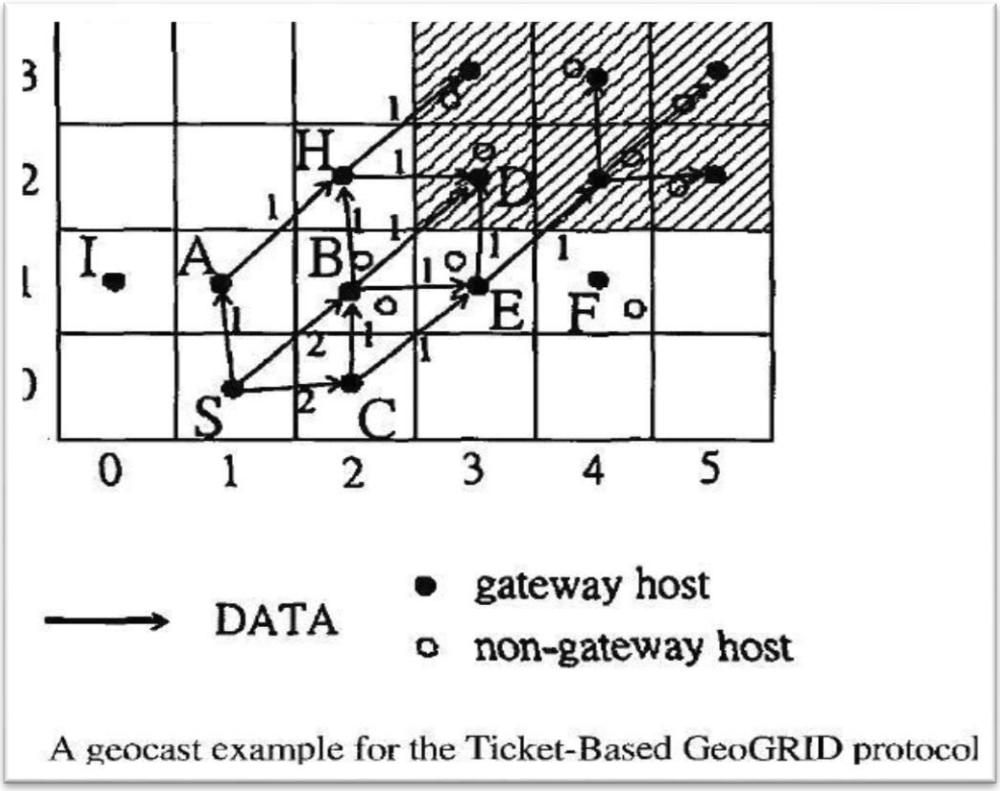
GeoGRID

Based on the unicast protocol GRID [Liao2001], the GeoGRID protocol [Liao2000] uses location information, which defines the forwarding zone and elects a special host (i.e, *gateway)* in each grid area responsible for forwarding the geocast packets. It is argued in [Liao2000] that the forwarding zone in LBM incurs unnecessary packet transmissions, and a tree-based solution is prohibitive in terms of control overhead. GeoGRID partitions the geographic area of the MANET into two-dimensional (2D) logical grids. Each grid is a square of size *d X d* (there are trade-offs in choosing a good value of *d*, as discussed in [Liao2000].) In GeoGRID, a gateway node is elected within each grid.

The forwarding zone is defined by the location of the source and the geocast region. The main difference between GeoGRID, LBM and VDG is the following: in GeoGRID, instead of every node in a forwarding zone transmitting data, only gateway nodes take this responsibility. There are two schemes on how to send geocast packets in GeoGRID:

Flooding-Based GeoGRID and *Ticket-Based GeoGRID*. In Flooding-Based GeoGRID, only gateways in every grid within the forwarding zone rebroadcast the received geocast packets. Thus, gateway election becomes the key point of this protocol. In Ticket-Based GeoGRID, the geocast packets are still forwarded by gateway nodes, but not all the gateways in the forwarding zone forward every geocast packet. A total of $m + n$ tickets are created by the source if the geocast region is a rectangle of $m X n$ grids. The source evenly distributes the $m + n$ tickets to the neighboring gateway nodes in the forwarding zone that are closer to the geocast region than the source. A gateway node that receives X tickets follows the same procedure as the one defined for the source. Consider the example in Figure 3.15 where node S begins with five (2+3) tickets. Node S may distribute two tickets to its neighboring nodes A and B, and one ticket to its neighbor node C, which are closer to the geocast region than node S. It is not mentioned in [Liao2000], however, why node C is given fewer tickets than nodes A and B. We believe the philosophy is that each ticket is responsible for carrying one copy of the geocast packet to the geocast region. Hence, if a node is sent a geocast packet that it has seen before, it does not discard it. For example, if node C decides to give its ticket to node B in Figure 3.15, (i.e., node B receives a geocast packet from node C), node B will rebroadcast the packet. In other words, node B will transmit the geocast packet (at least) two times. Both the Flooding-Based GeoGRID and the Ticket-Based GeoGRID protocols need an efficient solution for the gateway election. Once this node is elected, it remains the gateway until it moves out of the grid. One problem of this

selection process is when another potential gateway roams closer to the physical center of the grid than the currently assigned gateway and cannot be elected as the gateway until the current gateway leaves the grid. To eliminate this possibility, multiple gateways could be



A geocast example for the Ticket-Based GeoGRID protocol

allowed to reside in a grid temporally. In this situation, if a gateway hears a packet from another gateway at a location closer to the physical center of its grid, it silently turns itself into a non-gateway node and does not forward any further geocast packets. However, if the grid size is small, or the mobility of the node is low, this problem may not be severe. Another effective way of gateway election is via the concept Of Node Weigh. For example, we could assign the weight of a node as being inversely proportional to its speed. Flooding-Based GeoGRID and Ticket-Based GeoGRID have obvious advantages over LBM Scheme 1 and LBM Scheme 2, especially in dense networks. The two GeoGRID protocols should offer both higher accuracy and lower delivery cost than LBM and VDG due to the reduced number of transmitted packets.

GeoTORA

The goal of the GeoTORA protocol [Ko2000] is to reduce the overhead of transmitting geocast packets via flooding techniques, while maintaining high accuracy.

The unicast routing protocol TORA is used by GeoTORA to transmit geocast packets to a geocast region. As TORA is a distributed routing protocol based on a "link reversal" algorithm, it provides multiple routes to a destination. Despite dynamic link failures, TORA attempts to maintain a destination-oriented directed acyclic graph such that each node can reach the destination. In GeoTORA, a source node essentially performs an *anycast* to any geocast group member (i.e, any node in the geocast region) via TORA. When a node in the geocast region receives the geocast packet, it floods the packet such that the flooding is limited to the geocast region. The accuracy of GeoTORA is high, but not as high as pure flooding or LBM. One reason is only one node in the geocast region receives the geocast packet and if that node is partitioned from other nodes in the geocast region, the accuracy reduces.

Mesh-Based Geocast Routing Protocol

The Mesh-based Geocast Routing (MGR) protocol [Boleng2001] uses a mesh for geocasting inan ad hoc environment in order to provide redundant paths between the source and the group members. Since the group members in a geocast region are in close proximity to each other, it is less costly to provide redundant paths from a source to a geocast region than to provide the redundant paths from a source to a multicast group of nodes that may not be in close proximity of each other. Instead of flooding geocast packets, the MGR Protocol tries to create redundant routes via control packets. First, the protocol floods JOIN-DEMAND packets in a forwarding zone. A JOIN-DEMAND packet is forwarded in the network until it reaches a node in the geocast region. This node unicasts a JOIN-TABLE packet back to the source by following the reverse route of the JOIN-DEMAND packet. Thus, the nodes on the edge of the geocast region become a part of the mesh. Once the first JOINTABLE packet is received by the source, data packets can be sent to the nodes in the geocast region. Figure shows an example of geocast communication via a mesh.

TCP OVER ADHOC

INTRODUCTION:

Over the past few years, ad hoc networks have emerged as a promising approach for future mobile IP applications. This scheme can operate independently from existing underlying infrastructure and allows simple and fast implementation. Obviously, the more this technology evolves, higher will be the probability of being an integral part of the global Internet [Perkins2002]. Therefore, considerable research efforts have been put on the investigation of the Transmission Control Protocol (TCP) [Comer 1995] performance over ad hoc networks.

As we know, TCP is the prevalent transport layer in the IP world today and is employed by a vast majority of applications, especially those requiring reliability. More specifically, TCP is the most widely used transport protocol for data services like file transfer, email, web browsing, and so on. Therefore, it is essential to look at the TCP performance over ad hoc networks. TCP is now fine-tuned to work well in wired environments; however its performance is highly degraded due to the high error rates and longer delays of wireless mediums, as well as mobility.

In recent years, several improvements for TCP have been proposed for cellular [Balakrishnanl997, Zhang2001] wireless networks, but still much work has to be done for the ad hoc networks paradigm.

Unlike cellular networks, where only the last hop is based on a wireless medium, ad hoc networks are composed exclusively of wireless links, where multi-hop connections are often in place. Besides, in an ad hoc scenario, all nodes can move freely and unpredictably, which makes the clock based TCP congestion control quite hard. In particular, as the errors in wireless networks occur not only due to congestion but also due medium constraints and mobility, TCP needs to distinguish the nature of the error so that it can take the most appropriate action. Factors such as path asymmetry (that may also be caused by lower layers strategies, among other elements) and congestion window size also impact the performance of this protocol. These issues and others are addressed in detail in this chapter.

Although there are a number of differences between cellular and ad hoc networks, some of the ideas developed for the former can be used in the latter as well. As a matter of fact, many of the proposed solutions for TCP over ad hoc networks represent a mix of old concepts developed for cellular network properly adapted for this multi-hop scenario. Nevertheless, we cover some solutions specifically tailored for ad hoc networks, while many issues are still open.

TCP Protocol Overview

The TCP protocol is defined in the Request For Comment (RFC) standards document number 793 [Postell981] by the Internet Engineering Task Force (IETF) [IETFwww]. The original specification written in 1981 was based on earlier research and experimentation in the original ARPANET. It is important to remember that most applications on the Internet make use of TCP, relying upon its mechanisms that ensure safe delivery of data across an unreliable IP layer below. In this section we explore the fundamental concepts behind TCP and how it is used to transport data between two endpoints. More specifically, we focus on those aspects of TCP

that are of importance to understand its performance over ad hoc networks. TCP adds a great deal of functionality to the IP service it is layered over:

• **Streams:** TCP data is organized as a stream of bytes, much like a file. The datagram nature of the network is concealed. A mechanism (the Urgent Pointer) exists to let out-of-band data be specially flagged; • Reliable delivery: Sequence numbers are used to coordinate which data has been transmitted and received. TCP will arrange for retransmission if it determines that data has been lost; 356 AD HOC & SENSOR NETWORKS

• **Network adaptation:** TCP will dynamically learn the delay characteristics of a network and adjust its operation to maximize throughput without overloading the network;

• **Flow control**: TCP manages data buffers, and coordinates traffic so that its buffers will never overflow. Fast senders will be stopped periodically to keep up with slower receivers

Designed and Fine-Tuned to Wired Networks

The design of TCP has been heavily influenced by what is commonly known as the "end-to-end argument" [Clarkl988]. As it applies to the wired Internet, the system gets unnecessarily complicated by putting excessive intelligence in physical and link layers to handle error control, encryption or flow control. While these functions need to be done at the endpoints anyway, the result is the provision of minimal functionality on a hop-by-hop basis and maximal control between endto-end communicating systems.

TCP performance is often dependent on the flow control and the congestion control. Flow control determines the rate at which data is transmitted between a sender and a receiver. Congestion control defines the methods for implicitly interpreting signals from the network in order for a sender to adjust its rate of transmission. Ultimately, intermediate devices, such as IP routers, would only be able to control congestion. A recent study on congestion control examines the current state of activity.

Timeouts and retransmissions handle error control in TCP. The nature of TCP and the underlying packet switched network provide formidable challenges for managers, designers and researchers. In the design of TCP for wireless networks, it is important to incorporate link layer acknowledgements and error detection/correction functionality. Furthermore, when we consider MANETs, the mobility comes into picture. Therefore, high error rates, longer delays, and mobility makes MANET environments extremely challenging to the operation of TCP as it tears down most the assumptions over which TCP was designed.

TCP Basics

TCP is often described as a byte stream, connection-oriented, fullduplex, reliable delivery transport layer protocol. In this subsection, we discuss the meaning for each of these descriptive terms.

Byte Stream Delivery

TCP interfaces between the application layer above and the network layer below. When an application sends data to TCP, it does so in 8-bit byte streams. It is then up to the sending TCP to segment or delineate the byte stream in order to transmit data in manageable pieces to the receiver. It is this lack of "record boundaries" which gives it the name "byte stream delivery service".

Connection-Oriented

Before two communicating TCP entities (the sender and the receiver) can exchange data, they must first agree upon the willingness to communicate. Analogous to a telephone call, a connection must first be made before two parties exchange information.

Full-Duplex

No matter what a particular application may be, TCP almost always operates in full-duplex mode. It is sometimes useful to think of a TCP session as two independent byte streams, traveling in opposite directions. No TCP mechanism exists to associate data in the forward and reverse byte streams, and only during connection start and close sequences can TCP exhibit asymmetric behavior (i.e., data transfer in the forward direction but not in the reverse, or vice versa).

Reliability

A number of mechanisms help provide the reliability TCP guarantees. Each of these is described briefly below. • Checksums: All TCP segments carry a checksum, which is used by the receiver to detect errors with either the TCP header or data; • Duplicate data detection: It is possible for packets to be duplicated in packet switched network; therefore TCP keeps track of bytes.received in order to discard duplicate copies of data that has already been received;

• **Retransmissions:** In order to guarantee delivery of data, TCP must implement retransmission schemes for data that may be lost or damaged. The use of positive acknowledgements by the receiver to the sender confirms successful reception of data. The lack of positive acknowledgements, coupled with a timeout period (see timers below) calls for a retransmission;

• **Sequencing**: In packet switched networks, it is possible for packets to be delivered out of order. It is TCP's job to properly sequence segments it receives so that it can deliver the byte stream data to an application in order;

• **Timers**: TCP maintains various static and dynamic timers on data sent. The sending TCP waits for the receiver to reply with an acknowledgement within a bounded length of time. If the timer expires before receiving an acknowledgement, the sender can retransmit the segment.

TCP Header Format

As we know, the combination of TCP header and TCP in one packet is called a TCP segment. Figure 7.1 depicts the format of all valid TCP segments. The size of the header without options is 20 bytes. Below we briefly define each field of the TCP header.

Source Port: This is a 16-bit number identifying the application where the TCP segment originated from within the sending host. The port numbers are divided into three ranges: well-known ports (0 through 1023), registered ports (1024 through 49151) and private ports (49152 through 65535). Port assignments are used by TCP as an interface to the application layer. For example, the TELNET server is always assigned to the well-known port 23 by default on TCP hosts. A pair of IP addresses (source and destination) plus a complete pair of TCP ports (source and destination) defines a single TCP connection that is globally unique.

Destination Port: A 16-bit number identifying the application the TCP segment is destined for on a receiving host. Destination ports use the same port number assignments as those set aside for source ports.

Sequence Number: Within the entire byte stream of the TCP connection, a 32-bit number, identifying the current position of the first data byte in the segment. After reaching 232 -1, this number will wrap around to 0.

Acknowledgement Number: This is a 32-bit number identifying the next data byte the sender expects from the receiver. Therefore, this number will be one greater than the most recently received data byte. This field is used only when the ACK control bit is turned on.

Header Length: A 4-bit field that specifies the total TCP header length in 32-bit words (or in multiples of 4 bytes). Without options, a TCP header is always 20 bytes in length. On the other hand, the largest a TCP header is 60 bytes. Clearly, this field is required because the size of the options field(s) cannot be determined in advance. Note that this field is called "data offset" in the official TCP standard, but header length is more commonly used.

Reserved: A 6-bit field currently unused and reserved for future use. Control Bits:

• **Urgent Pointer (URG)** - If this bit field is set, the receiving TCP should interpret the urgent pointer field (see below);

• **Acknowledgement (ACK)** - If this bit is set, the acknowledgment field is valid;

• **Push Function (PSH)** - If this bit is set, the receiver should deliver this segment to the receiving application as soon as possible. An example of its use may be to send a Control-C request to an application, which can jump ahead of queued data;

• **Reset Connection (RST)** - If this bit is present, it signals the receiver that the sender is aborting the connection and all the associated queued data and allocated buffers can be freely relinquished;

• **Synchronize (SYN) -** When present, this bit field signifies that the sender is attempting to "synchronize" sequence numbers. This bit is used during the initial stages of connection establishment between a sender and a receiver;

• **No More Data from Sender (FIN) -** If set, this bit field tells the receiver that the sender has reached the end of its byte stream for the current TCP connection.

Window: This is a 16-bit integer used by TCP for flow control in the form of a data transmission window size. This number tells the sender how much data the receiver is willing to accept. The maximum value for this field would limit the window size to 65,535 bytes. However, a "window scale" option can be used to make use of even larger windows.

Checksum: A TCP sender computes the checksum value based on the contents of the TCP header and data fields. This 16-bit value will be compared with the value the receiver generates using the same computation. If the values match, the receiver can be very confident that the segment arrived intact.

Urgent Pointer: In certain circumstances, it may be necessary for a TCP sender to notify the receiver of urgent data that should be processed by the receiving application as soon as possible. This 16-bit field tells the receiver when the last byte of urgent data in the segment ends.

Options: In order to provide additional functionality, several optional parameters may be used between a TCP sender and a receiver. Depending on the option(s) used, the length of this field varies in size, but it cannot be larger than 40 bytes due to the maximum size of the header length field (4 bits). The most common option is the maximum segment size (MSS) option where a TCP receiver tells the TCP sender the maximum segment size it is willing to accept. Other options are often used for various flow control and congestion control techniques

Padding: Because options may vary in size, it may be necessary to "pad" the TCP header with zeroes so that the segment ends on a 32-bit word boundary as defined by the standard.

Data: Although not used in some circumstances (e.g., acknowledgement segments with no data in the reverse direction), this variable length field carries the application data from TCP sender to receiver. This field coupled with the TCP header fields constitutes a TCP segment.

Congestion Control

TCP congestion control and Internet traffic management issues in general is an active area of research and experimentation. Although this section is a very brief summary of the standard congestion control algorithms widely used in TCP implementations today, it covers the main points to understand behavior of the TCP over MANETs. These algorithms are defined in [Jacobsonl988] and [Jacobson 1990a], and the most update version of the TCP congestion control algorithm can be found in.

Slow Start Slow Start, a requirement for TCP software implementation, is a mechanism used by the sender to control the transmission rate, otherwise known as sender-based flow control. The rate of acknowledgements returned by the receiver determines the rate at which the sender can transmit data. Whenever a TCP connection starts, the Slow Start algorithm at the sender initializes a congestion window (CWND) to one segment. As the connection is carried out and acknowledgements are returned by the receiver, the congestion window increases by one segment for each acknowledgement returned. Thus, the sender can transmit the minimum of the congestion window and the advertised window (contained in the header of the acknowledgment packet) of the receiver, which is simply called the transmission window and is increased exponentially.

TCP and MANETs

MANETs pose some tough challenges to TCP primarily because TCP has not been designed to operate in such a highly dynamic environment [Oliveira2002]. In general, we can identify three main different types of challenges posed to TCP over ad hoc networks.

• First, as the topology changes, the path is interrupted and TCP goes into repeated, exponentially increasing time-outs with severe performance impact. Efficient retransmission strategies have been proposed to overcome such problems (e.g., [Chandran2001, Holland 1999a]) and are discussed later in this chapter; Chapter 7: TCP over Ad Hoc Networks 365

• The second problem has to do with the fact that the TCP performance in ad hoc multi-hop environment depends critically on the congestion window in use. If the window grows too large, there are too many packets (and ACKs) on the path, all competing for the same wireless shared medium. Congestion builds up and causes "wastage" of the broadcast medium with consequent throughput degradation [Fu2005, Oliveira2005];

• The third problem is significant TCP unfairness which has been recently revealed and reported through both simulations and testbed measurements.

In reality, even though TCP has evolved significantly over the years toward a robust and reliable service protocol, the focus has been primarily on wired networks. In this scenario, the additiveincrease/multiplicative-decrease strategy coupled with the fast recovery and fast retransmits mechanisms [Allmanl999], inherent in most of current TCP versions; provide an effective congestion control solution. The key idea of TCP is to probe the network in order to determine the availability of resources. It injects packets at an increasing rate into the network until a packet loss is detected, whereby it infers the network is facing congestion. Then, the TCP sender shrinks its CWND, retransmits the lost packet and resumes transmission at a lower increasing rate. If the losses persist (no timely ACK received), at every retransmission the sender doubles (up to 64s) its wait timer (i.e., the RTO) so that it can wait longer for the ACK of the current packet being transmitted.

Clearly, the aforementioned mechanisms work quite well in a wired network where the Bit Error Rate (BER) is typically very low allowing any lost packet to be treated as an indication of network congestion. In a wireless mobile ad hoc network, however, two more factors can induce packet losses in addition to network congestion: non-negligible wireless medium losses (high BER) and frequent connectivity disruptions caused by node mobility. As a result, lost packets no longer serve as an indication of congestion for TCP. In order for TCP to perform efficiently in wireless networks, it needs to be aware of what caused which loss.

Ad hoc networks have a high level of BER as all links involved are wireless and hence suffer from fading, multipath effects, and interference [Rappaportl999]. In addition, since nodes can move freely and unpredictably, there is a high probability of the peers in the ongoing connections to get abruptly disconnected by a temporary or permanent lack of a route between them, and this can last for a significant period of time. Such a discontinuity (whether temporary or permanent) is typically referred to as partition and is generally very degrading for TCP performance, as explained later. Another important characteristic of TCP is its dependence on reception of timely ACKs from the receiver to the sender so as to increase its data transfer rate. Thus, in case there is an asymmetrical path in place, with lower ACK rate as compared to the data rate, TCP can be prevented from sending at a higher rate. If we apply this fact to ad hoc networks where path asymmetry is a common place, we can clearly see that TCP algorithms may also have to deal with this issue. Finally, and not surprisingly, the lower layers (MAC and network) protocol strategies used in this scenario also play a key role on TCP performance,

which demands special handling tailored to the ad hoc environment. In the following, we address all the issues aforementioned in detail in order to clarify why and how they take place.

Effects of Partitions on TCP

A network partition occurs when a given mobile node moves away, or is interrupted by the medium, thereby splitting its adjacent nodes into two isolated parts of the network that are called partitions. Here, the term partition is defined within the context of a connection which is interrupted due to a link breakage, and not necessarily because there are no alternate paths available through some other nodes.

depict a scenario in which a partition takes place and interrupts an ongoing connection between nodes 3 and 9. there is a direct link between nodes 6 and 7, while this is not the case in Figure We consider these two cases to illustrate the different between a short-term (Figure 7.3) and a long-term partition. Both figures show the case when node 5 moves away from node 3 causing a link breakage and consequently disrupts the connection between nodes 3 and 9. In the case of the routing protocol in use will find an alternate route from node 3 to node 9 through node 6, possibly, in one attempt. The resulting connection topology is shown in Figure 7.5. Therefore, we refer to this type of partition as being short term.

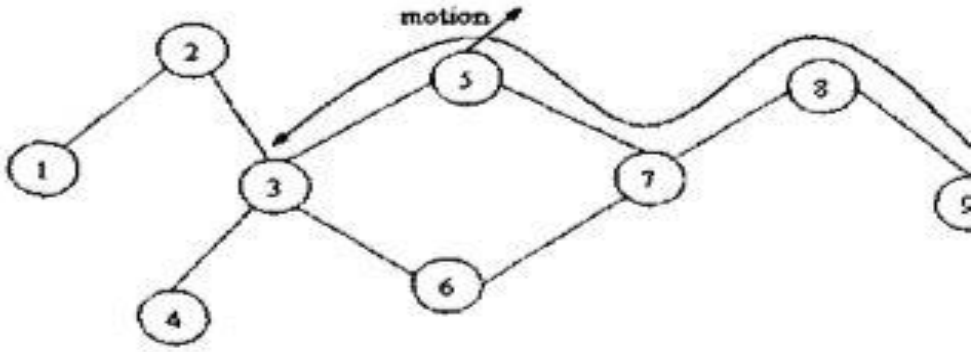


Figure 7.3 – Node 5 moves away from node 3 (short-t

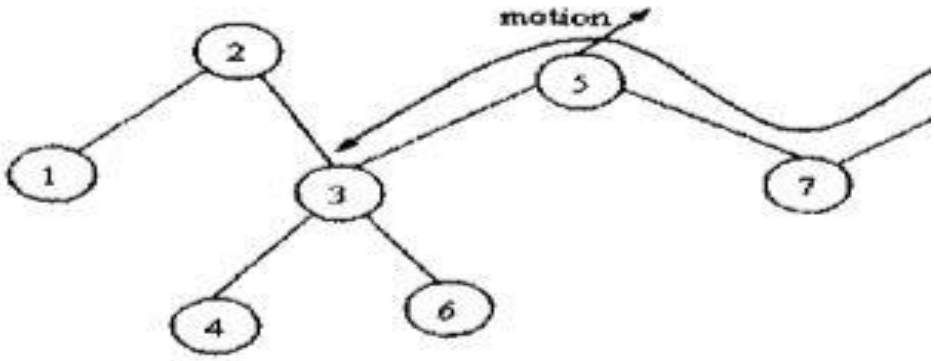


Figure 7.4 – Node 5 moves away from node 3 (long-t

Another possibility is as shown in Figure 7.4 where there is no link between nodes 6 and 7. In this case, upon movement of node 5 and consequent link breakage between nodes 3 and 5, not only is the connection between nodes 3 and 9 disrupted, but a long-term network partition takes place as can be seen from Figure 7.6. In this case, the connection can only be reestablished whenever the topology rejoins. For example, in Figure 7.6 node 6 moves towards node 7 rejoining the network and allowing the connection to be reestablished as presented in Figure 7.5.Figure

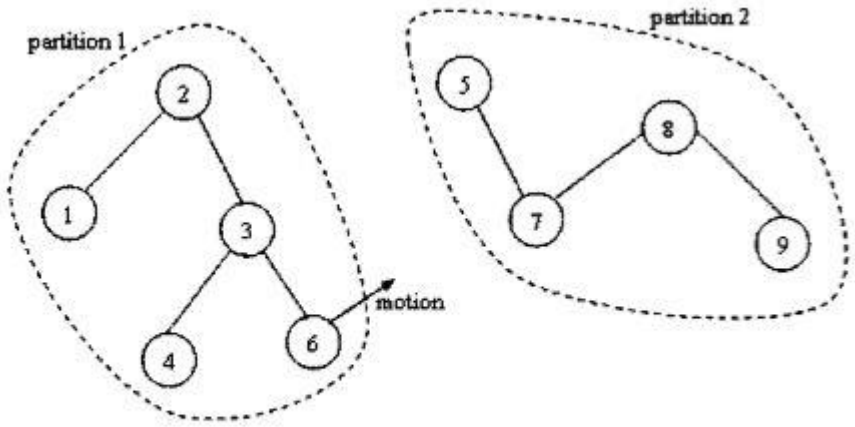


Figure 7.6 – No communication between the partitions

As a matter of fact, the real impact of a partition on TCP performance depends on its duration. As we explain below, a long partition will trigger the TCP backoff mechanism, and possibly end up increasing the delay till the connection restoration. Figure 7.2 depicts an example of a long partition triggering the TCP exponential backoff mechanism [Stevens 1994]. For the sake of this discussion, we use the term packet and segment interchangeably throughout this chapter. With that in mind, Figure 7.2 shows how the delayed answer of the exponential backoff mechanism can lead the TCP sender to a long idle period, which we call "dead time", subsequently to the link restoration. The example shows that both packet 3 (P3) and the acknowledgment of packet 2 (ACK3) are dropped due to a link failure (left vertical line). As the sender does not receive the confirmation of packet 2 (P2) receipt, it retransmits P2 by timeout after 6 sec (6 sec is the typical initial RTO which changes over time according to measured RTTs), and doubles its RTO value. Whenever the timeout period expires, the TCP sender retransmits P2 and doubles the RTO up to the limit of 64sec that refers to the maximum timeout allowed (note that after 12 unsuccessful attempts TCP would give up). The example shows that shortly after triggering its timer for 64 sec, the link is recovered. However, it is too late for TCP and it will stay over one minute frozen which is indeed a dead time for the assumed starving connection. In terms of percentage, we have roughly 61% (100 sec/163 sec) of interruption due to link failure and the remaining 39% (63

sec/163 sec) is completely caused by TCP, which is certainly too much to be acceptable. It is appropriate to mention that the infrequent (not necessary low) and transient packet losses caused by mechanisms such as fading or short-term partitions, can also lead TCP to take inappropriate actions. That is, even though such losses are improbably to cause a long period of disconnection, they do can undesirably lead TCP to invoke its exponential backoff algorithm. Consequently, TCP prevents the applications from using precious bandwidth by decreasing its transmission rate when, in fact, it should not. More details about this issue can be found in [Caceresl994] where the focus is on cellular networks which can face a similar problem. Therefore, it is clear that the standard TCP needs to be adapted to work satisfactorily in the newMANET paradigm. An effective TCP algorithm must be capable of distinguishing the origin of a packet loss so as to take the most appropriate action. In fact, its error-detection mechanism needs to detect the nature of the error so that its error-recovery mechanism can be tailored for each specific case.

Impact of Lower Layers on TCP

Given the fact that TCP is a reliable protocol, providing end-to-end guarantees over a variety of local and unreliable protocols running in the lower layers (notably, MAC and routing protocol layers), it is no surprise that its performance depends strongly on such protocols. What is not so clear, however, is how either TCP or lower protocols can be fine tuned to avoid as much as possible any undesired interferences between them. In this subsection we give an appropriate insight into the issues resulted from TCP interaction with lower layers.

MAC Layer Impact

Like the transport layer, the wireless MAC layers almost invariably also rely on error control mechanisms in order to improve the transmission efficiency. However, while the former deals with end-toend recovery, the latter concentrates on link (one hop) recovery. Hence, unless a well defined synchronism between these both protocols is put in place, negative interactions can substantially deteriorate end-to-end throughput provided by TCP. As we have studied in Chapter 4, the IEEE 802.11 DCF is nowadays the standard MAC layer protocol adopted for ad hoc networks (obviously, other WLAN/WPAN systems could also be used). This MAC protocol, which defines both physical and link layer mechanisms, is intended for providing an efficient shared broadcast channel through which the involved mobile nodes can communicate. The main novelty of this protocol refers to the inclusion of acknowledgment for data frames (link layer's ACKs) in addition to RTS/CTS control frames to make it possible for link layer retransmissions. This makes it possible to recover a packet loss at the link level instead of waiting for TCP to detect the loss only at the destination when it has already taken too long time. Another mechanism introduced by the

IEEE 802.11 MAC is the virtual carrier sense used to track medium activity. In IEEE 802.11, RTS/CTS handshake is only employed when the DATA packet size exceeds some predefined threshold. However, given the fact that most existing commercial implementations of IEEE 802.11 use RTS/CTS handshake before any DATA/ACK transmission (obviously, broadcast packets are not preceded by RTS/CTS), for the sake of our discussion here we assume RTS/CTS handshake is always employed before any DATA/ACK exchange. As we saw in Chapter 4, each of these frames carries the remaining duration of time for the transmission completion, so that other nodes in the vicinity can hear it and postpone their transmissions accordingly. In fact, every node maintains an information parameter called NAV which is updated according to other nodes' transmission schedules. In order to provide fair access to the medium at the end of every such sequence, the nodes must await an IFS interval and then contend for the medium again. The contention is carried out by means of a binary exponential backoff mechanism which imposes a further random interval, aiming to avoid collisions to allow all nodes the same probability of gaining access to the medium. At every unsuccessful attempt, this random interval tends to become higher (its range of randomness is doubled at every attempt) and after some number of attempts (typically, seven times) the MAC layer gives up and drops the data, which is reported as a route failure to the network layer. Therefore, the MAC protocol is certainly very robust in dealing with the possibility of collisions in the wireless shared medium. By using short RTS/CTS control frames to reserve the medium, bandwidth wastage is considerably minimized in case of collisions as these frames are much smaller than DATA frames. As for the virtual carrier sense mechanism, it prevents the so-called hidden node problem (discussed in Chapter 4).

In such cases, under IEEE 802.11, the first node to succeed reserves the medium and the other (or others) becomes aware and defers its transmission. Moreover, although this MAC protocol has been designed to be fair in the sense that all neighboring stations would have equal chances of accessing the wireless medium, in fact it may result in strong unfairness for TCP traffic as some stations will have larger backoff values while others have smaller ones. IEEE 802.11 relies on the assumption that every node can reach each other or at least sense any transmission into the medium, which is not always true in an ad hoc scenario. Consequently, the hidden node and exposed node problems can arise in some conditions inducing what is termed as the capture problem [Cordeiro2002], which impairs not only TCP performance but also results in unfairness amongst simultaneous TCP connections. We explain in the following, by means of examples, how these problems can take place.

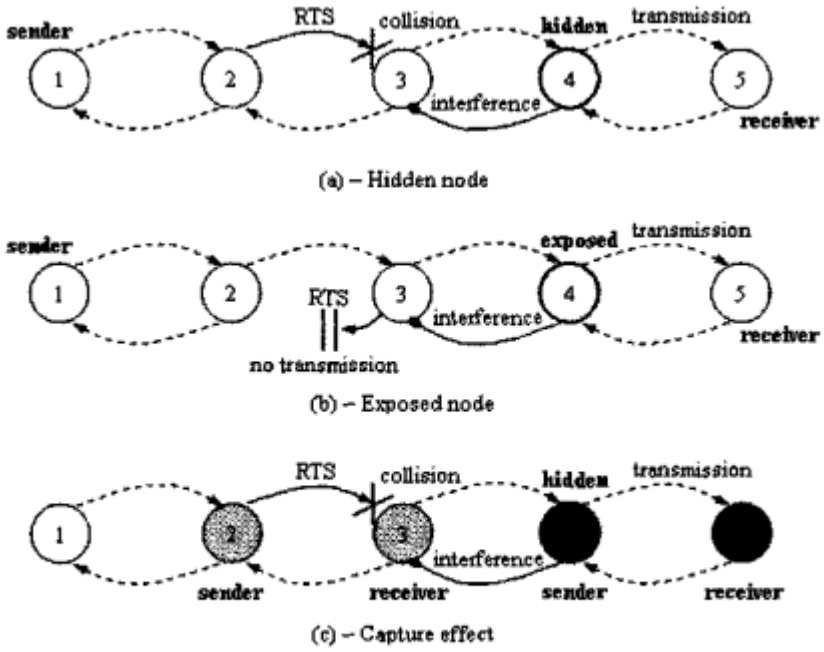


Figure 7.7 – Issues at the MAC layer

In Figure 7.7, we consider a linear topology in which each node can only communicate with its adjacent neighbors. In addition, consider that in Figures 7.7(a) and 7.7(b) there exist a single TCP connection running between nodes 1 and 5. Further, suppose node 1 starts transmitting to node 5 as illustrated in Figure 7.7(a). Once the first few packets reach the Chapter 5, there will be a condition in which node 2 wishes to communicate with node 3, while node 4 is transmitting to node 5. As node 2 cannot hear the transmission from node 4, it senses the medium idle (both physically and virtually using NAV) and so attempts its transmission by sending a RTS toward node 3. Even so, since node 3 is within node 4's interference range (that is, node 4's transmission to node 5 affects node 3's reception), it does not receive the data transmitted by node 2, which is dropped due to a collision.

This is a typical hidden node problem (see Chapter 4), where node 4 is the hidden node (in relation to node 2). Also, note that not one by many packets sent from node 2 to node 3 will be dropped due to collisions. Figure 7.7(b) depicts a particular condition for exposed node problem (see Chapter 4), where node 3 has a data frame (that is, related to a TCP ACK) to send to node 2. As node 4, which is within the sensing range of node 3 (that is, node 4's transmission affects node 3's transmission ability), is transmitting to node 5, node 3 must wait for the end of current transmission and then contend for the medium. Here, node 4 is the exposed node (in

regards to node 3). Note that as collisions only occur at the receiver, node 2 could receive the frame from node 3 correctly despite node 4's conversation with node 5, as node 4 is out of interfering range of node 2. These two problems can affect TCP throughput as follows. When a hidden node condition as illustrated in Figure 7.7(a) takes place, the MAC layer of node 2 invokes its backoff mechanism which attempts retransmission (locally) of the lost frame up to a maximum number of times (typically seven). In case it does not succeed (e.g., due to high traffic between nodes 4 and 5), node 2 will drop the packet and send a route error packet back to node 1. As a consequence, the routing protocol in node 1 will attempt to find a new route to the destination which will, by itself, delay the forwarding of TCP packets. Usually in these circumstances, the TCP sender will time out, reduce its transmission rate, and further delays the retransmission. Likewise, under the exposed node depicted in Figure 7.7(b), as long as the traffic between nodes 4 and 5 is high enough to delay the pending TCP ACK beyond the TCP timeout interval, the TCP sender at node 1 will also timeout.

If we look carefully at the example in Figure 7, we will notice that the same exposed and hidden terminal problems takes place in the reverse traffic (i.e., ACK packet flow) from the TCP receiver at node 5 to the TCP sender at node 1. In other words, the TCP backward ACKs need to compete for access to the wireless shared medium with its own TCP forward data packets. However, in terms of link layer frames, there are many more data frames in the forward direction (i.e., TCP data packets) than TCP ACK packets, as ACKs packet sizes are much smaller and also due to the possibility of the receiver sending cumulative ACKs through a single ACK packet. Therefore, the wireless medium will be highly loaded with by TCP data frames and, as a result, a significant amount of ACK packets will be lost. Due to the above facts, both hidden and exposed node problems can cause a considerable lack of ACKs at the TCP sender, which is characterized as an asymmetrical PATH problem for TCP as discussed later in this section. Such a problem will either trigger a TCP retransmission by timeout when the lost ACK regards a confirmation of a successfully received data packet, or impair the TCP fast retransmit mechanism for which three duplicate ACKs are required in order to trigger a fast retransmission without the need to wait for a timeout event.

Hence, in general, the larger the number of nodes a TCP connection needs to span, lower is the end-to-end throughput as there will be more medium contention taking place in several regions of the network. In this way, it has been shown [Holland 1999a] that the TCP throughput

over IEEE 802.11 decreases sharply and exponentially as a function of number of hops, and is shown in Figure 7.8.

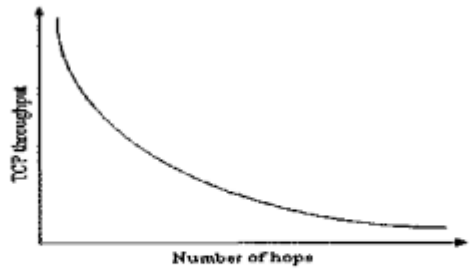


Figure 7.8 – TCP throughput is inversely proportional to the number of hops

Similar problems have been evaluated in [Xu2001a] where it has been shown that using smaller values for both the segment packet size and the maximum window size in TCP setup can mitigate such problems to a certain extent. The idea behind using smaller values for these parameters is to prevent TCP from sending too much data packets before receiving an ACK (i.e., to reduce the number of outstanding packets). As a result, the probability of collisions is decreased and the local MAC retransmission scheme has a better chance to succeed within its seven times retry limit. In addition, it has been found in [Xu2001a] that a maximum window size of four segments should be enough to provide maximum stable throughput. Nevertheless, further analysis is needed in order to validate such results for higher speed networks, such as IEEE 802.11a/g (see Chapter 4), where the limited size for that parameter could represent a throughput bottleneck. It is also shown in [Xu2001] that TCP throughput in such a scenario can be improved by using the delayed ACK option in which an ACK is sent for every two received data packets. In principle, this might be an interesting idea as it reduces the traffic load. IEEE 802.11 has also been raising serious unfairness concerns in ad hoc networks mainly due to its binary exponential backoff mechanism, which leads to what is known as capture conditions [Cordeiro2002], In fact, such a phenomenon is also related to the hidden node or exposed node conditions. For simplicity, here we explain only the situation in which the hidden node problem induces the capture effect (the capture effect caused by the exposed node is similar). Consider Figure 7.7(c) where there are two independent connections, one between nodes 2 and 3 (connection 2-3) and another between nodes 4 and 5 (connection 4-5). Assuming that connection 2-3 experiences collision due to the hidden node problem caused by the active connection 4-5 (as explained earlier), node 2 will back off and retransmit the lost frame. As we know, at every retransmission the binary exponential backoff mechanism imposes an increasingly (although random) backoff interval. Implicitly, this is actually decreasing the possibility of success for the connection 2-3 to send a packet as connection 4-5 will "dominate" the medium access once it has lower backoff value for most of

the time. Besides, if the MAC retransmission scheme fails, TCP will eventually time out and will also 376 AD HOC & SENSOR NETWORKS invoke its exponential backoff mechanism, further increasing the delay for the next attempt. In consequence, the connection 2-3 will hardly obtain access to the medium while connection 4-5 will capture it. Note that the MAC protocol is designed in such a way that if the connection between nodes 4 and 5 has a large data to transfer, it will fragment and transmit it in smaller data frames with higher priority over all the other nodes, which is done by using a short IFS between the transfer of each fragment. Clearly, this behavior also contributes to the unfairness. Finally, the burstiness in TCP is another component which makes a connection to capture the medium. Contrary to mobility related TCP issues, the capture problem is mostly present when network nodes are static or possess small mobility since nodes stay longer within radio range of each other [Cordeiro2002], while in high mobility networks nodes are often moving out of range of each other and hence rarely have the chance to capture or to be captured. The capture problem is severe enough that nodes cannot access the medium for some amount of time they generate route error packets, even though the network is completely static. For TCP traffic, this causes retransmission timers to go off and throughput to degrade drastically. Capture conditions are very likely in current generation routing protocols as the same route is used for forward and reverse traffic given a pair [Perkins2001, Johnson2001]. For TCP, this implies that data packets in the forwarding direction and ACK packets in the reverse direction compete to access the same shared medium, frequently causing ACK packets to be unable to reach the source and, thus, TCP executing its congestion control algorithms. It has been shown that TCP data packets often capture the medium preventing ACK packets from reaching their destination [Xu2001a, Xu2001b]. This problem is worsened by the presence of multiple TCP flows. Fairness problems due to capture conditions have been investigated in [Gerlal999], and it was found that it can be mitigated by properly adjusting some MAC layer timers. Specifically, it has been shown that, as far as IEEE 802.11 is concerned, better fairness can be achieved by increasing the IFS interval (therein called yield time). However, it comes at the cost of degrading the aggregate throughput, which is somewhat expected as it makes the medium idle for a longer time. Therefore, it is Chapter 7: TCP over Ad Hoc Networks 377 clear that alternative solutions for inherently unfair behavior detected in this environment need to be explored. Furthermore, no solution appears to be effective enough by simply configuring either TCP or MAC parameters. Rather, hidden and exposed node problems have to be addressed to have a deeper robust approach that would provide not only a fairer but also throughput effective MAC protocol. Moreover, the solutions presented so far do not address the scenario where

multiple TCP connections are simultaneously competing for the medium access. That is, solving the problem from the point of view of a single connection clearly is the exception to the rule.

As we discuss later, it is proposed in [Cordeiro2002] a general solution to the fairness problem regardless of the number of active TCP connections in the network. Apart from what has been explained here with regards to the interactions between the MAC and TCP layers, there are many other issues that are potential sources of complications. For example, if the nodes have different interfering (and sensing) and communication ranges, then the exposed node problem gets exacerbated. Likewise, either hidden or exposed node problems would be quite difficult to be controlled if the nodes had different battery power levels, which may be likely the case in an actual network [Poojary2001]. Additionally, the inherently node mobility can give rise to synchronization issues which would compromise the effectiveness of the reservation scheme provided by the MAC protocol.

Network Layer Impact As ad hoc networks consist of a highly dynamic environment where frequent route changes are expected, routing strategies play a key role on TCP performance as well. Unlike the MAC layer, for which the IEEE 802.11 protocol has been widely used as a testbed, the network layer has been a subject of most research efforts on mobile ad hoc networks area towards standardization. As we have seen in Chapter 2, there have been a lot of proposed routing schemes and, typically, each of them have different effects on the TCP performance. To understand the network layer effect on TCP, in this subsection we consider two of the major routing protocols proposed for MANETs (both of them covered in 378 AD HOC & SENSOR NETWORKS Chapter 2) and show how a network layer protocol can affect the performance of an upper layer protocol.

DSR

As we know, the DSR protocol operates on an on-demand basis in which a node wishing to find a new route broadcasts a RREQ packet. Then, the destination node, or any other node which knows a route to the destination, responds back with a RREP packet. This packet informs the sender node the exact path to be followed by the data packets, which are sent with a list of nodes through which they must go. In addition, each node keeps a cache of routes it has learned or overheard. As a result, intermediate nodes do not need to keep an up-to-date table of routes, thereby avoiding periodic route advertisements that cause considerable overhead. The problem with this approach concerns the high probability of stale routes in environments where high mobility as well as medium constraints may be normally present. That can happen, for instance,

when a RREP message is in its way back to the sender but the replied route is no longer valid due to either an involved node that has moved away or a link that has somehow been interrupted. The problem is exacerbated by the fact that other nodes can overhear the invalid route reply and populate their buffers with stale route information.

Therefore, unless stale routes can be detected and recovered in a fast way, TCP can be led to backoff state which considerably degrades its performance. This problem has been studied in [Holland 1999] and shown that it can be mitigated by either manipulating TCP to tolerate such a delay or by making the delay shorter so that the TCP can deal with them smoothly. In these studies, it has been observed that by disallowing route replies from caches can improve route accuracy at the expense of the routing performance in terms of overhead, since every new route discovery implies flooding the network. On the other hand, such an additional overhead is outweighed by the accuracy in the route determination, mainly for high mobility conditions, resulting in an enhanced TCP performance.

TORA

As we saw in Chapter 2, TORA is also an on-demand based protocol but has pro-active features as well. TORA has been designed to be highly dynamic by establishing routes quickly and concentrating control messages within a small set of nodes close to the place where the topological change has occurred. It is accomplished by maintaining multiple routes between any possible peers. In consequence, most topological changes should entail no reaction at all concerning route discovery, as it only reacts when all routes to a specific destination are lost. As we have seen before, TORA makes use of directed acyclic graphs, where every node has a path to a given destination. In other words, all neighbors of a given sender have an alternative path to a given destination, which define multiple potential paths for every peer. The directed acyclic graph is established initially by each node advertising a query packet and receiving update packets when it first tries to discover a route.

A new query is only necessary when no more routes are available for a given sender. This can happen as the invalid routes, caused by partition, are removed from the nodes by having the affected node send a clear packet. From the TCP viewpoint, this protocol can also suffer from stale route problem similar to the DSR protocol. Nevertheless, as route discovery procedures are confined to situations less probable (no available path), such a drawback can be considered not too harmful to TCP. On the other hand, multiple path routing can indeed cause significant performance degradation. The problem occurs mainly because TORA does not prioritize shorter

paths, which can yield considerable amount of out-of-sequence packets for the TCP receiver, triggering retransmission of packets.

A typical situation could be to send an earlier packet through a longer path and then, due an instantaneous route problem, a new packet being forced through a shorter enough path to arrive first at the destination. Therefore, it should be interesting to have a self-adaptive mechanism to avoid this possibility.

Later in this chapter, we discuss this further. In conclusion, the characteristics presented here in regards with DSR and TORA reveal that the design of a routing protocol should take into 380 AD HOC & SENSOR NETWORKS consideration its impacts on the upper layer, especially on the widely used TCP protocol. Once more, solutions can be placed in both layers and cooperation between them (cross-layer design) may be extremely advantageous [Cordeiro2002].

Path Asymmetry Impact

As we know, TCP relies on time sensitive feedback information to perform its flow control and asymmetrical paths can seriously compromise its performance. In other words, if TCP does not receive timely ACKs, it cannot expand its CWND to make full use of the available channel capacity, thereby wasting the bandwidth. Hence, in case the forward path characteristics are considerably different from the ones of the backward path, TCP will quite likely face performance problems. In ad hoc networks where the topology as well as the environment conditions can change quite frequently and unpredictably, asymmetry can occur by different reasons, including lower layer strategies. Based on the work presented in [Balakrishnan2001], asymmetry in a TCP-based wireless mobile ad hoc network can be categorized into the following classes:

• Loss rate asymmetry: This sort of asymmetry takes place when the backward path is significantly more error prone than the forward path. In ad hoc environment, this can be a serious factor as all links involved are wireless having high error rate which depends on local constraints that can vary from place to place and due to mobility patterns as well;

• Bandwidth asymmetry: This is the classical asymmetry found in satellite networks in which forward and backward data follow distinct paths with different speeds. In ad hoc networks this can happen as well, since all nodes need not have the same interface speed. So, even if a common path is used in both directions of a given flow, not necessarily they have the same bandwidth. Besides, as the routing protocols can assign different paths for forward and backward traffics [Cordeiro2002], asymmetry can definitely occur in ad hoc networks;

SOLUTIONS FOR TCP OVER AD HOC

We now present the most prominent schemes that have been specifically proposed to overcome TCP performance problems in ad hoc networks. Here, we classify the proposed solutions into Mobility-related and Fairness-related based on the key TCP issue they aim to overcome. The mobility-related approaches address the TCP problems resulting from node mobility which may mistakenly trigger TCP congestion control mechanisms. On the other hand, fairness-related solutions tackle the serious unfairness conditions raised when TCP is run over MANETs.

Mobility-Related

Notably, most of the solutions in this category have limitations which can compromise their widespread deployment. Nevertheless, it is of paramount importance to understand them as they may serve as the basis for future research. In the following we discuss the details of each one of them.

TCP-Feedback As the name suggests, TCP-Feedback (TCP-F) [CRVP97] is a feedback-based scheme in which the TCP sender can effectively distinguish between route failure and network congestion by receiving Route Failure Notification (RFN) messages from intermediate nodes. The idea is to push the TCP into a "snooze state" whenever such messages are received. In this state, TCP stops sending packets and freezes all its variables such as timers and CWND size, which makes sense once there is no available route to the destination. Upon receipt of a Route Re-establishment Notification (RRN) message from the routing protocol, indicating that there is again an available path to the destination, the sender leaves the frozen state and resumes transmission using the same variables values prior to the interruption. In addition, a route failure timer is employed to prevent infinite wait for RRN messages, and is started whenever a RFN is received. Upon expiration of this timer, the frozen timers of TCP are reset hence allowing the TCP congestion control to be invoked normally. Results from TCP-F shows gains over standard TCP in conditions where the route reestablishment delay are high, which are due to a fewer number of involved retransmission. Nevertheless, the simulation scenario employed to evaluate TCP-F has been quite simplified and so the results might not be a true representative. For example, the RFN and RRN messages employed in TCP-F are to be carried by the routing protocol, but no such protocol has been considered.

The ELFN Approach

The Explicit Link Failure Notification (ELFN) [Holland 1999] is a cross-layer proposal in which TCP also interacts with the routing protocol in order to detect route failure and take appropriate actions. Here, ELFN messages are sent back to the TCP sender from the node detecting the failure. Such messages are carried by the routing protocol that needs to be adapted for this purpose. In fact, the DSR's route error message has been modified to carry a payload similar to the "host unreachable" message of the Internet Control Message Protocol (ICMP) [Tanenbauml996]. Basically, the ELFN messages contain sender and receiver addresses and ports, as well as the TCP sequence number. This way, the modified TCP is able to distinguish losses caused by congestion from the ones due to mobility. In ELFN, whenever the TCP sender receives an ELFN message it enters a "stand-by" mode in which its timers are disabled and probe packets are sent regularly towards the destination in order to detect route restoration. Upon receiving an ACK packet, the sender leaves the "stand-by" mode and resumes transmission using its previous timer values and state variables.

This scheme was only evaluated for the DSR routing protocol where the stale route problem was found to be crucial for the performance of ELFN. Additionally, the interval between transmission of probe packets and the choice of which type of packet to be sent as a probe has also been evaluated. In essence, it has been suggested that a varying interval based on RTTs values could perform better than the fixed probe interval. In general, the ELFN approach provides meaningful enhancements over the standard TCP, but further evaluation may be needed. For instance, different routing protocols should be studied, and the performance of ELFN under congestion conditions should be considered. Last, but not the least, more appropriate values for the probe interval should be determined.

Fixed RTO

The fixed RTO scheme [Dyer2001] relies on the idea that routing error recovery should be accomplished in a fast fashion by the routing algorithm. As a result, any disconnection should be treated as a transitory period which does not justify the regular exponential backoff mechanism of TCP being invoked, as this can cause unnecessarily long recovery delays. Thus, it disables such a mechanism whenever two successive retransmissions due to timeout occur, assuming that it actually indicates route failure. By doing so, it allows the TCP sender to retransmit at regular intervals instead of at increasingly exponential ones. In fact, the TCP sender doubles the RTO once and if the missing packet does not arrive before the second RTO expires, the packet is retransmitted again and again but the RTO is no longer increased. It remains fixed until the route is recovered and the retransmitted packet is acknowledged.

The fixed RTO approach has been evaluated in [Dyer2001] considering different routing protocols along with TCP selective and delayed acknowledgements options. Sizeable enhancements have been accomplished with on-demand routing protocols, but only marginal improvements have been noticed when using the TCP options. Nevertheless, this proposal is limited to wireless networks only, which makes it somewhat discouraging as interoperation with wired networks is a mandatory requirement in the vast majority of applications.

The ATCP Protocol

Different from previously discussed approaches, the Ad hoc TCP (ATCP) protocol [Liu2001] does not impose changes to the standard TCP itself. Rather, it implements an intermediate layer between the network and the transport layers in order to provide an enhanced performance to TCP and still maintain interoperation with non-ATCP nodes. More specifically, ATCP relies on the ICMP protocol and on the Explicit Congestion Notification (ECN) [ECNwww] scheme to detect/distinguish network partition and congestion, respectively. This way, the intermediate layer keeps track of the packets to and from the transport layer so that the TCP congestion control is not invoked when it is not really needed, which is done as follows.

Whenever three duplicate ACKs are detected, indicating a lossy channel, ATCP puts TCP in "persistent mode" and quickly retransmits the lost packet from the TCP buffer; after receiving the next ACK, the normal state is resumed. In case an ICMP "Destination Unreachable" message arrives, pointing out a network partition, ATCP also puts the TCP in "persist modes" which only ends when the connection is reestablished. Finally, when network congestion is detected by the receipt of an ECN message, the ATCP does nothing but forwards the packet to TCP so that it can invoke its normal congestion control mechanism. ATCP was implemented in a testbed and evaluated under different constraints such as congestion, lossy scenario, partition, and packet reordering. In all cases, the transfer time of a given file by ATCP yielded better performance as compared to TCP. However, the scenario employed has been somewhat special as neither wireless links nor ad hoc routing protocols have been considered. In fact, such experiments relied on simple Ethernet networks connected in series, in which each node had two Ethernet cards. Moreover, some assumptions such as ECN-capable nodes as well as the sender node being always reachable might be somehow hard to be met in reality. In case the latter is not fulfilled, for example, the ICMP message might not even reach the sender which would retransmit continuously instead of entering the "persist mode". Also, the deployment of the ECN scheme is

known to raise many security concerns [ECNwww], and it might compromise the viability of ATCP.

TCP-DOOR

Due to its dynamic environment, mobility in MANETs is extremely frequent. Therefore, a natural effect of mobility is that the packet usually arrive out-of-order (OOO) at the destination. If the OOO delivery event is appropriately monitored, it might be just enough to detect link failure inside the network and, hence, be able to effectively distinguish between mobility and congestion. The TCP-DOOR (Detection of Out-of-Order and Response) [Wang2002] protocol focuses on the idea that OOO delivery of packets can happen frequently in MANETs as a result of nodes mobility. TCP-DOOR imposes changes to TCP code but does not require intermediate nodes to cooperate, which represents its main differentiation from all previously described proposals. In this way, TCPDOOR detects OOO events and responds accordingly as explained below. Based on the fact that not only data packets but also ACK packets can experience OOO deliveries, TCP-DOOR implements a detection of such deliveries at both entities: TCP sender and TCP receiver. For this, additional ordering information is used in both types of packets (data and ACK) which are conveyed as TCP options, where one extra byte is required for ACKs and two extra bytes are required for data. Thus, for every packet sent the sender increments its own stream sequence number inside the two-byte option regardless whether it is a retransmission or not (standard TCP does not increment sequence number of retransmitted packets).

This allows the receiver to precisely detect OOO delivery of data packets and notify the sender via a specific bit into the return packet. Additionally, because all ACKs associated with a given missing data packet have identical contents, the receiver increments its own ACK stream sequence number inside the one-byte option for every retransmitted ACK, so that the sender can distinguish the exact order of every (retransmitted or not) packet sent. Therefore, the explained mechanism provides the sender with reliable information about the order of the packet stream in both directions, allowing the TCP sender to act accordingly. After detecting OOO events, the TCP sender can respond with two mechanisms: temporarily disabling congestion control and instant recovery during congestion avoidance. In the former, the TCP sender keeps its state variables constant for a while after the OOO detection. The rationale behind this is that such condition might be short (route change) not justifying the invocation of the congestion avoidance mechanism. In the latter, whenever an OOO condition is detected the TCP sender checks if the congestion control mechanism has been invoked in the recent past. If so, the connection state

prior to the congestion control invocation is restored, as such an invocation may have been caused by temporary disruption instead of by congestion itself. Different scenarios combining all the aforementioned mechanisms have been simulated in [Wang2002].

The effects of the route cache property of the DSR routing protocol on TCP-DOOR performance have also been considered. The results indicate that only sender detection mechanism (ACK OOO detection) should suffice. Both responses mechanisms are important and instant recovery during congestion avoidance performs better than temporarily disabling congestion control. In addition, the DSR route cache impaired the performance improvement mainly due to stale caches. In general, TCP-DOOR improves TCP performance by an average of 50%, while other protocols such as ATCP report from 200% to 300% improvement (contrary to other solutions, however, TCP-DOOR confines the changes to the TCP protocol only). On the other hand, the assumption in TCP-DOOR that OOO packets are the exclusive result of route disturbance deserves much more careful analysis. Multipath routing algorithms (e.g., TORA) can induce OOO packets that are not necessarily related to route failures. Besides, as we have seen before, diverse factors can cause path asymmetry inducing events as well. Obviously, the independence from intermediate nodes makes TCP-DOOR quite attractive which calls for further developments towards a more general approach.

Discussions The main drawbacks of the proposed schemes are as follows. The approaches that rely on feedback information from inside the network (TCP-F, ELFN-based, ATCP) may fail in situations where TCP sender is unable to receive data from the next hop node (e.g., due to mobility). In such cases, the TCP sender would retransmit continuously instead of entering a frozen state. Furthermore, the usage of explicit notification by the intermediate nodes, such as ECN, raises many security concerns.

The fixed RTO scheme, on the other hand, does not seem to be appropriate for possible future interoperation with wired networks. Finally, the assumption in TCP-DOOR that OOO packets are exclusive results of route disturbance may not be true in a quite a few scenarios. In fact, the main concern addressed by the approaches presented so far is how to avoid the TCP exponential backoff mechanism when losses take place by factors other than congestion. However, as discussed in the previous section, other factors such as path asymmetry and fine tuning with lower layers, among others, should also be considered by an effective design. Moreover, challenging approaches should also address prominent issues such as power management, interoperation with wired networks (e.g., Internet), security, and so on. Thus, it is noticeable that

the proposed approaches are somewhat limited, highlighting the necessity for further investigation in this area.

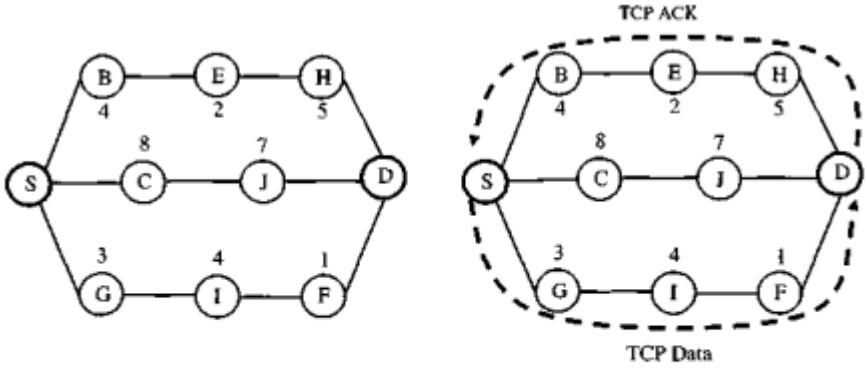
Fairness-Related Compared to the number of mobility-related studies, little has been done to address the serious unfairness conditions raised by TCP over ad hoc networks. In this section we present some important solutions in this category. Other related studies can be found in [Bottigliengo2004].

COPAS As we have seen before, the problem of capture is severe due to the interplay of the MAC layer and TCP backoff policies and results in a single node within its radio range being able to access the medium at all times, while others in its neighborhood starve. To the same extent as mobility related issues, the capture problem drastically affects TCP performance and is stressed in wireless MAC protocols that employ exponential backoff schemes such as IEEE 802.11 and FAMA (Floor Acquisition Multiple Access) [Fullmerl995] as their backoff mechanisms always favor the last successful station. A protocol called COPAS (COntention-based PAth Selection) has been proposed in [Cordeiro2002] to address TCP performance drop due to the capture problem and resulting unfairness. COPAS implements two novel routing techniques in order to contention-balance the network, namely, the use of disjoint forward (for TCP data) and reverse (for TCP ACK) paths to reduce the conflicts between TCP packets traveling in opposite directions, as well as a dynamic contention-balancing technique that continuously monitors network contention and selects routes with minimum MAC layer contention. COPAS works as follows. In on-demand protocols, a route discovery process is initiated when a route to a destination is needed and none is available. The source floods the network with a RREQ packet to discover a route to the destination. When the destination receives the RREQ, it responds with a unicast RREP packet back to the source. In COPAS, upon receipt of a non-duplicate RREQ packet, to this packet nodes append a weighted average of the number of times it has backed off in a "recent past" due to activity in the medium.

The RREQ packet is then rebroadcast. By keeping track of the recent average number of times a node has backed off, COPAS is actually determining how busy the wireless shared medium is in the neighborhood of a node. More times a node backs off, means that more busy is the medium around it. This provides precise information on the contention experienced along the paths traveled by a RREQ. After receiving the first RREQ packet, the destination waits for an appropriate amount of time to learn all possible routes. The destination node accepts duplicate RREQ received from different previous nodes. When the RREQ collection timer expires,

COPAS employs two selection criteria in order to choose exactly two routes: path disjointness, and least contented routes.

Disjoint path routing has been explored before in connection with both DSR [Nasipuri2001] and AODV [Marina2001] routing protocols. COPAS uses similar techniques to choose all possible node-disjoint routes (between source and destination) at the destination and selects the two least contented routes based on the information collected by the arriving RREQ packets. Least contented routes are computed by evaluating the sum of the contentions experienced by the RREQ packets on each node-disjoint route and then minimizing the sum over all disjoint routes available. Ties are resolved by favoring lower route lengths (in hops) and then by the arrival order of the RREQ packets. In the absence of disjoint paths, COPAS behaves similarly to existing routing protocols with the difference that it can take advantage of network contention information. The destination responds with at most two RREPs along the chosen paths. Along with the RREP, the destination also sets a direction flag in the packet header to indicate to the source node which path is to be used as forward (for TCP data packets) and reverse (for TCP ACK packets) traffic. This direction information is also kept in a node's routing table.



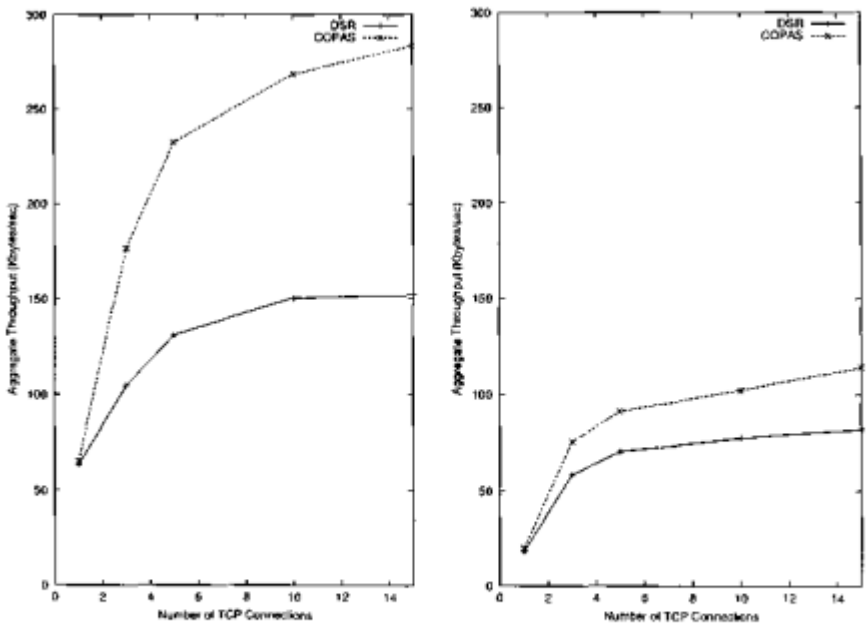
(a) – Network contention perceived at node D

(b) – Routes selected by node D

Figure 7.9 – Route establishment in COPAS [Taken from IEEE Publication Cordeiro2002]

To illustrate this, consider the scenario of Figure 7.9(a) wherein the source node S sends a RREQ packet towards the destination node D. In this case and with the contention values as depicted in the Figure 7.9(a), the destination first applies the disjointness path rule and finds out routes i = ,; = < S-C-J-D >, and k = < S-G-I-F-D> to be disjoint. Next, it applies the minimum contention

sum rule and end up selecting routes i and k to be used as reverse (for TCP ACK) and forward (for TCP data) paths respectively, as showed in Figure 7.9(b). Employing disjoint forward and reverse paths is also desirable for robustness reasons. Capture conditions can be so severe that links appear to be broken even when there is no mobility. Therefore, to guarantee continuous operation even in link breakage situations COPAS makes use of previously established forward and reverse routes. In a capture scenario, it is usually the MAC layer which reports to the network layer the link breakage since it is in this layer where the capture problem is rooted. When a route is disconnected, the immediate upstream node of the broken link sends a RERR message to the source of the route notifying the route invalidation. Nodes along the path to the source remove the route entry upon receiving this message and relay it to the source. In traditional on-demand routing protocols, the source reconstructs a new route by flooding a RREQ when informed of a route disconnection. In COPAS - in addition to flooding a RREQ to reconstruct the broken route -TCP packets are redirected using the second alternate path when available, hence providing uninterrupted communication. It is up to TCP to recover from potential lost packets due to link breakage, while COPAS attempts to minimize the route disruption by rerouting data packets. In this case, COPAS behaves similar to existing approaches. COPAS also includes provisions for dynamic contention-balancing. Traffic pattern across the network changes a lot with time and space. Therefore, routes that were optimal during the initial route construction process may no longer be good paths as contention might have increased with the new traffic pattern. Therefore, COPAS implements a scheme to dynamically monitor and change routes between any pair that have their contention increased noticeably. Recent research either evaluates a single TCP session [Holland 1999, Wang2002], or when multiple TCP sessions are considered the network is fully mobile [Dyer2001], or the connections mostly cover one hop employing unrealistic topologies such as ring and string However, in [Cordeiro2002] simulations are performed where it is considered multiple TCP connections under several scenarios, and where the network comprised of only static hosts. This is the worst case scenario where capture conditions are mostly severe since nodes remain within radio range of each other continuously, and where multiple TCP flows compete to have access to the shared medium. Nevertheless, COP AS could cooperate with any of the other proposed schemes in [Chandran2001, Dyer2001] as it tackles TCP degradation in static to



– 50-nodes scenario (b) – 100-nodes scenario

Figure 7.10 – Average aggregate throughput [Taken from IEEE Publication Cordeiro2002]

COPAS has been evaluated and compared with the DSR routing protocol. Figures 7.10(a) and 7.10(b) show some simulation results of COPAS applied to scenario of 50 and 100 nodes, respectively, where 394 AD HOC & SENSOR NETWORKS TCP connections range from 1 to 15. As we can see, for the 50-nodes scenario, COP AS is shown to drastically improve TCP throughput by up to 90%, whereas in the 100-nodes scenario, COPAS still achieves a considerable improvement but it is not as sizeable as in the 50-nodes scenario due to the large number of routes from any given source and destination which reduces conflicts among TCP connections. It has also been observed that nodes running COPAS experience much less medium contention due to the dynamic contention-balancing mechanism, while keeping the routing overhead low. However, there still issues that need to be addressed including how the protocol can handle unidirectional links, and the interrelationship between TCP and UDP traffic.

Neighborhood RED

a scheme called Neighborhood RED (NRED) is proposed, where it is claimed that two unique features of ad hoc wireless networks are the key to understand unfair TCP behaviors. One is the spatial reuse constraint, and the other is the location dependency. The former implies that space is also a kind of shared resource. TCP flows, which do not even traverse common nodes, may still compete for "shared space" and hence interfere with each other. The later, location dependency, triggers many of the problems we have mentioned discussed so far (channel capture, hidden and exposed terminals, and so on), which are often recognized as the primary

reasons for TCP unfairness. Clearly, TCP flows with different relative positions in the bottleneck may get different perception of the bottleneck situation in terms of packet delay and packet loss rate.

Since obtaining correct feedback information of the bottleneck is critical to the fairness of TCP congestion control, limited information of the bottleneck situation causes significant unfairness. If we view a node and its interfering neighbors to form a neighborhood (the neighborhood of a node X is formed by all nodes within communication range of X), the local queues at these nodes can be considered to form a distributed queue for this neighborhood (for instance, the neighborhood of node A and its distributed queue in Figure 7.11). Obviously, this distributed queue is not a FIFO queue. Flows sharing this queue have different and dynamic priorities determined by Chapter 7: TCP over Ad Hoc Networks 395 the topology and traffic patterns due to channel capture, hidden and exposed terminal situations, and so on. Therefore, they get different feedback in terms of packet loss rate and packet delay when congestion happens. The uneven feedback makes TCP congestion control diverge from the fair share. Similar situations may occur in wired networks when a buffer is full and drop tail queue management scheme is used. In these wired networks, the RED scheme has been shown to improve TCP fairness under such situations by keeping the queue size relatively small and dropping or marking packets roughly proportional to the bandwidth share of each flow through thegateway.

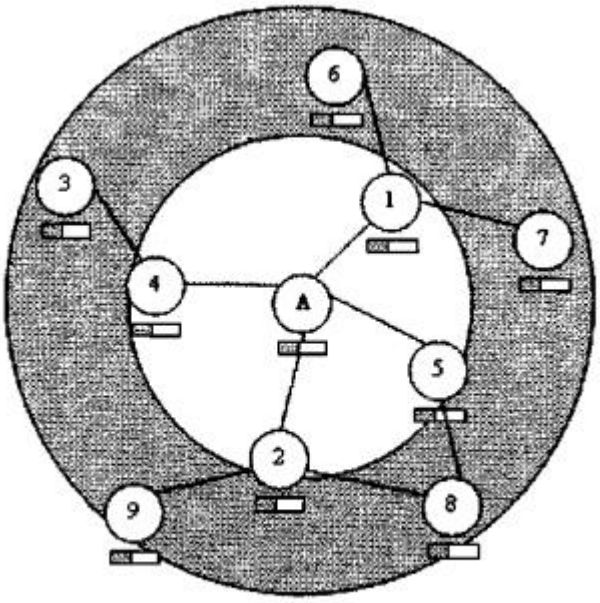


Figure 7.11 – Node A's neighborhood and its distributed queue

The idea of the NRED scheme is to extend the original RED mechanism to operate on the distributed neighborhood queue. Similar to RED, each node employing NRED keeps estimating the size of its neighborhood queue. Once the queue size exceeds a certain threshold, a drop probability is computed by using the algorithm from the original RED scheme.

Since a neighborhood queue is the aggregate of local queues at neighboring nodes, this drop probability is then propagated to neighboring nodes for cooperative packet drops. Each neighbor node computes its local drop probability based on its channel bandwidth usage and drops packets accordingly. The overall drop probability will realize the calculated drop probability on the whole neighborhood queue. Thus, the NRED scheme is basically a distributed RED suitable for ad hoc wireless networks. In [Xu2003], NRED is mostly evaluated under relatively long-lived TCP flows such as FTP connections transferring a medium or large size file. This is because TCP unfairness issues are more serious to such TCP traffic. If a TCP connection finishes its transfer in seconds, NRED may not have enough time to detect the network congestion and perform proper actions. Therefore, NRED may not be suitable for all types of TCP connections, even though short-lived TCP flows may not hurt other flows too much as they tend to quickly end. The main achievement of NRED is the ability to detect early congestion and drop packets proportionally to a flow's channel bandwidth utilization. By doing this, the NRED scheme is able to improve the TCP fairness. Finally, one major contribution of NRED is the design of a network layer solution that does not require any MAC modification.

UNIT-IV

Basics of Wireless, Sensors and Lower Layer Issues

Basics of Wireless, Sensors and Lower Layer Issues: Applications, Classification of sensor networks, Architecture of sensor network, Physical layer, MAC layer, Link layer, Routing Layer.

Introduction

Introduction In recent years, advances in miniaturization, yet simple low-power circuit design and improved low cost, small-size batteries have made a new technological vision possible: wireless sensor networks (WSN) .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. In other words, a sensor node is basically a device that converts a sensed attribute (such as temperature, vibrations) into a form understandable by the users. WSNs, which can be considered as a special case of ad hoc networks with reduced or no mobility, are expected to find increasing deployment in coming years, as they enable reliable monitoring and analysis of unknown and untested environments. These networks are "data centric", i.e., unlike traditional ad hoc networks where data is requested from a specific node, data is requested based on certain attributes such as, "which area has temperature over 35°C or 95°F". Therefore a large number of sensors need to be deployed to accurately reflect the physical attribute in a given area.

A sensor has many functional components as 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 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 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 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 Transceiver 10Kbps-1Mbps 50-125 m range Memory (3K-1Mb) Sensor Transducer AO Converter Embedded Processor (8bit4-8Mhz) Battery Functional Block Diagram of a typical Sensor Node transducers that could convert many physical quantities such as temperature, pressure, velocity, acceleration, stress and strain, fatigue, tilt, light intensity, sound, humidity, gas-sensors, biological, pollution, nuclear radiation, civil structural sensors, blood pressure, sugar level, white cell count, and many others. These basic generic transducers need to be interfaced and connected to other devices like and such custom made unit can be used for a given specific application. provides a comparison of existing sensor networks that are commercially available [Hill2004]. In the following sections we first consider the most popular sensor unit and then consider issues associated with sensor networks.

Design Issues

The advancement in technology has made it possible to have Chapter 8: Wireless Sensor Networks Sensi for SIM, Sensing area for Sensing and Communication range of SNs a network of 100s or even thousands of extremely small, low powered devices equipped with programmable computing, multiple parameter sensing and wireless communication capability, enhancing the reliability, accuracy of data and the coverage area. In short, some of the advantages of WSN over wired ones are as follows:

• 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 themselves 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.

The wireless medium does have a few inherent limitations such as low bandwidth, error prone transmissions, and potential collisions in channel access, etc. It is clear that the available bandwidth for sensor data is low and is of the order of 1-100 kb/s. Since the wireless nodes are not connected in any way to a constant power supply, they derive energy from batteries which limit the amount of energy available to the nodes. In addition, since these sensor nodes are deployed in places where it is difficult to either replace the nodes or their batteries, it is desirable to increase the longevity of the network and, preferably, all the nodes should die together so that new nodes could be replenished simultaneously in the whole area. Finding individual dead nodes and then replacing those nodes selectively would require dynamic deployment and eliminates major advantages of these networks. Thus, the protocols designed for these networks must strategically distribute the dissipation of energy, which also enhances the average life of the overall system. In addition, as we mentioned before, environments in which these nodes are expected to operate and respond are very dynamic in nature, with fast changing physical parameters.

Traditional routing protocols defined for MANETs (discussed in previous chapters) 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 Chapter 8: Wireless Sensor Networks 411 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).

An ideal sensor network should have the following additional features:

• 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 should 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.

In wireless sensor networks where efficient usage of energy is very critical, longer latency for non-critical data is preferable for longer node lifetime. However, queries for time critical data should not be delayed and should be handled immediately. Some protocols try to use the energy of the network very efficiently by reducing unnecessary data transmission for non-critical data but transmitting time-critical data immediately, even if we have to keep the sensors on at all times. Periodic data is transmitted at longer intervals so that historical queries can also be answered. All other data is retrieved from the system on-demand. As we can see, wireless sensor networks cover a very broad area and impacts the design of every layer in the network protocol stack. Many of the things we have discussed earlier on ad hoc networking may need to be revisited as the applications (and hence the requirements) have changed, which impact the appropriate solutions in support of these applications. Therefore, in this chapter we discuss wireless sensor networks with respect to every layer of the protocol stack. We start with by listing various applications of WSN and move on to discuss associated issues

Challenges

Despite their innumerable applications, WSN have several restrictions, e.g., limited energy supply, limited computing power, and limited bandwidth of the wireless links connecting sensor nodes. One of the main design goals of WSNs is to prolong the lifetime of the network and prevent connectivity degradation by employing aggressive energy management techniques. As we have seen before, existing routing protocols designed for other wireless networks and traditional networks cannot be used directly in WSNs for the following reasons:

• Sensor nodes should be self-organizing as the ad hoc deployment of these nodes requires the system to form connections and cope with the resultant nodal distribution. Coupled with the fact that the operation of the sensor networks is un-attended, the network organization and configuration should be performed

automatically and more often due to nodes failure; Chapter 8: Wireless Sensor Networks 413

• In most application scenarios, sensor nodes are stationary. Nodes in other traditional wireless networks are free to move, which results in unpredictable and frequent topological changes. However, in some applications, some sensor nodes may be allowed to move and change their location (although with very low mobility);

• Sensor networks are application specific, i.e., design requirements of a sensor network change with application. For example, the challenging problem of low-latency precision tactical surveillance is different from that required for a periodic weather-monitoring task;

• Data collected by many nearby sensors is based on common phenomena, thus there is a high probability that the data has redundancy. Therefore, data aggregation and in-network processing are desirable to yield energy-efficient data delivery before being sent to the destinations;

• In traditional networks, data is requested from a specific node. Sensor Networks are data centric i.e., data is requested based on certain attributes, i.e., attribute-based addressing. An attribute-based address is composed of a set of attribute-value pair query. For example, if the query is something like temperature > 35°C, then only those devices sensing temperature > 35°C need to respond and report their readings. Other sensors can remain in the sleep state. Once an event of interest is detected, the system should be able to configure itself so as to obtain very high quality results;

• WSNs have relatively large number of sensor nodes, which may be on the order of thousands of nodes. Therefore, sensor nodes need not have a unique ID as the overhead of ID maintenance is high. In datacentric WSNs, the data is more important than knowing the IDs of which nodes sent the data;

• Position awareness of sensor nodes is important since data collection is based on the location. Currently, it is not feasible to use GPS hardware for this purpose. Methods based on triangulation [Bulusu2000], for example, allow sensor nodes to approximate their position using radio strength from a few known points.

Algorithms based on triangulation can work quite well under conditions where only very few nodes know their positions a priori, e.g., using GPS hardware

Routing protocol design for WSNs is heavily influenced by many challenging factors, which must be overcome before efficient communication can be achieved. These challenges can be summarized as follows:

• **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 a 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. Chapter 8: Wireless Sensor Networks 415

• **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 timeconstrained 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 416 AD HOC & SENSOR NETWORKS 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.

Classifications of WSNs:

A WSN is deployed primarily to collect sensed data by different WSs and it is critical to see how requently 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 node not only send data periodically, but also respond to sudden changes in attributevalues.

Once the type of network is decided, protocols that efficiently route data from the SNs to the users have to be designed, perhaps using a suitable MAC protocol to avoid collisions and subsequent energy consumption. Attempts should be made to distribute energy dissipation evenly among all nodes in the network, as it is

usually not common to assume the presence of specialized high-energy nodes in the network. In this chapter we cover proactive, reactive and hybrid protocols, while highlighting the fact that the protocols ought to be directly related to application requirements.

Applications

Thousands of sensors over strategic locations are used in a structure such as an automobile or an airplane, so that conditions can be constantly monitored both from the inside and the outside and a real-time warning can be issued whenever a major problem is forthcoming in the monitored entity. These wired sensors are large (and expensive) to cover as much area is desirable. Each of these need a continuous power supply and communicates their data to the end- user using a wired network. The organization of such a network should be pre-planned to find strategic position to place these nodes and then should be installed appropriately. The failure of a single node might bring down the whole network or leave that region completely un-monitored. Unattendability and some degree of fault tolerance in these networks are especially desirable in those applications where the sensors may be embedded in the structure or places in an inhospitable terrain and could be inaccessible for any service. Undoubtedly, wireless sensor networks have been conceived with military applications in mind, including battlefield surveillance and tracking of enemy activities. However, civil applications considerably outnumber the military ones and are applicable to many practical situations

Architecture of Sensor Networks:

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 andon-chip memory are important

• Currently used radio transceivers include the RFM TR1001 or Infineon or Chip on devices; similar radio modems are available from various manufacturers. Typically, ASK or FSK is used, while the Berkeley Pico Nodes employ OOK modulation. Radio concepts like ultra-wideband are in an advanced stage (e.g., the projects undertaken by the IEEE 802.15 working group). 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 ultra-sound 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, self recharge 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 tuple spaces are an interesting starting point for the run-time environments for such SNs.

Network Architecture

As discussed in chapter 8, the WSN architecture need to cover a desired area both for sensing coverage and communication connectivity point of view. Therefore, density of the WSN network is critical for the effective use of the WSN. There is no well-defined measure of life-time of a WSN.

Some assume either the failure of a single sensor running out of battery power, is taken as life-time of the network. Perhaps a better Chapter 9: Data Retrieval in Sensor Networks 453 definition is if certain percentage of sensors stops working, may define the life-time as the network continues to operate. The percentage failure may depend on the nature of application and as long as the area is adequately covered by the operating sensors, a WSN may be considered operational. Here you could also have some quantitative measure such as the monitored area is 95% covered. The SNs are yet to become inexpensive to be deploying with some degree of redundancy.

For example, it is good to say that a region can be monitored by several sensors simultaneously. But this is still a theoretical concept as coverage of a region by a single sensor is currently adequate. In addition, the degree of data reduction by collaborative aggregation, plays a vital role in minimizing the energy consumption. A denser deployment of sensor and transmission of sensed data may cause more energy consumption and increased delay due to collisions. On the other hand, transmitting data between two far apart sensors, may cause increased energy consumption due to increased energy consumption in

A B ,. • • A « • * » » B D

(a) Direction Transmission between A and

(b) Transmission using intermediate B sensors

Transmission strategies between two sensors wireless transmission (Figure 9.3). Therefore, there is an optimal distance between two sensors that would maximize the sensor lifetime [Bhardwaj2002]. So, if the density of sensors is high, then some of the sensors can be put into sleep mode to have close to optimal distance between the sensors. 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 454 AD HOC & SENSOR NETWORKS structure" [Estrin2001]. 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 and their 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.

Physical Layer

Very little work has been done on protocols that suits well to the needs of WSNs. With respect to the radio transmission, the main question is how to transmit as energy efficiently as possible, taking into account all related costs (possible retransmissions, overhead, and so on). Some energy efficient modulation techniques have been discussed in the hardware aspect for CDMA in sensor nodes is considered and modulation issues are described. A discussion of communication protocol design based on the physical layer is found in Given the work being done at the IEEE level (e.g., the IEEE 802.15.4 standard) and also given the limited research in this area, we have chosen not to go into the details of the physical layer for sensor networks. We note, however, that this is a very important issue that needs careful consideration, by both the academia and the industry.

MAC Layer:

The MAC and the routing layers are the most active research areas in WSNs. Therefore, an exhaustive discussion of all schemes is impossible. However, most of the existing work addresses how to make SNs sleep as long as possible. Consequently, these proposals often tend to include at least some aspects of TDMA. The wireless channel is primarily a broadcast medium. All nodes within radio range of a node can hear its transmission. This can be used as a uni cast medium by specifically addressing a particular node and all other nodes can drop the packet they receive. 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 isdivided

In to N equal portions in frequency (FDMA), in time (TDMA), in code (CDMA), in space (SDMA) or In schemes such as OFDM or 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 burst 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 whileminimizing collisions with other SNs transmissions. When there is a collision, the SNs are forcedto retransmit data, thus leading to increased wastage of energy and unbounded delay.

As we will see shortly, in a hierarchical clustering model, once clusters have been formed, it is desirable to keep the number of nodes in the cluster fixed and due to hierarchical clustering; the number of nodes per cluster is not kept large. So, it may be better to use one of the static channel allocation schemes. In his scheme, each node transmits data in its own slot to the cluster head and at all other times, its radio can be switched off, thereby saving valuable energy.

When it is not feasible to use TDMA, the n des can use non persistent CSMA since the data packets are of fixed size.TDMA is suitable for either proactive or reactive type of networks. In proactive networks, as we have the nodes transmitting periodically, we can assign each node a slot and thus avoid collisions. In reactive Networks, since adjacent nodes have similar data, when a sudden change takes place in some attribute being sensed, all the nodes will respond immediately. This will lead to collisions and it is possible that the data may never reach the user on time. For this reason, TDMA is employed so that each node is given a slot and they transmit only in that slot. Even though this increases the delay and many slots might be empty, it is better than the energy consumption incurred due to dynamic channel allocation schemes.

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 andContention-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 IEEE 802.11 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.

LINK LAYER

Compared to the MAC and routing layers, very little work exists on the link layer for WSNs. The question of choosing suitable packet size for energy efficient operation is discussed in [Sankarasubramaniam2003a], while energy efficient issues at the link layer are also investigated.

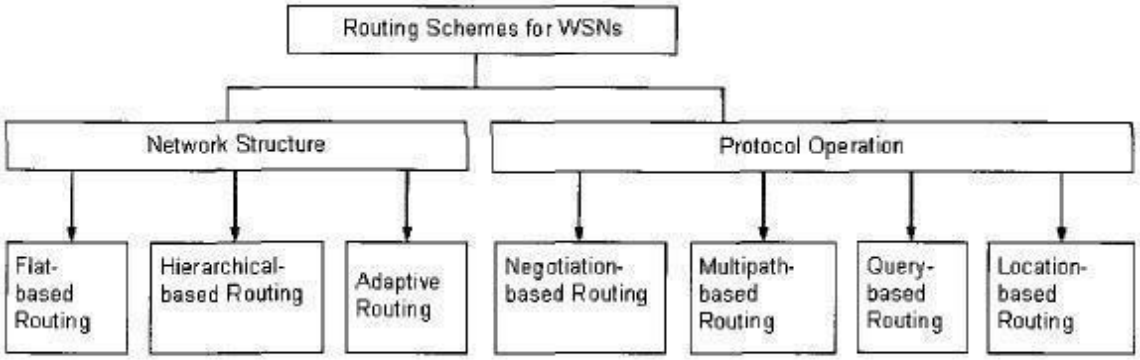
Routing Layer

By now, it must be clear that WSNs differ from traditional wireless networks. Conventional flooding-based protocols widely employed in MANETs suffer from data explosion problem, i.e., if a node is a common neighbor to nodes holding the same data item, then it will get multiple copies of the same data item. Therefore, the protocol wastes resources by sending and receiving duplicate data copies. In addition, flooding does not scale well in large networks and wastes resources.

The goal is to send the data from source node(s) to a known destination node, i.e., the BS. The destination node or the sink node is known and addressed by means of its location. A BS may be fixed or mobile, and is capable of connecting the sensor network to an existing

infrastructure (e.g., Internet) where the user can have access to the collected data. The task of finding and maintaining routes WSNs is nontrivial since energy restrictions and sudden changes in node status (e.g., failure) cause frequent unpredictable topological changes. Thus, the main objective of routing techniques is to minimize the energy consumption in order to prolong WSN lifetime. To achieve this objective, routing protocols proposed in the literature employ some well known routing techniques as well

as tactics special to WSNs. To preserve energy, strategies like data aggregation and in-network processing, clustering, different node role assignment, and data- centric methods are employed.



Classification of routing protocols for WSNs

In sensor networks, conservation of energy is considered relatively more important than quality of data sent. Therefore, energy-aware routing protocols need to satisfy this requirement. Routing protocols for WSNs have been extensively studied in the last few years. Routing protocols for WSNs can be broadly classified into flat-based, hierarchical-based, and adaptive, depending on the network structure. In Flat-based routing, all nodes are assigned equal role. In hierarchical based routing, however, nodes play different roles and certain nodes, called cluster heads (CHs), are given more responsibility. In adaptive routing, certain system parameters are controlled in order to adapt to the current network conditions and available energy levels. Furthermore, these protocols can be classified into multipath-based, query-based, negotiation-based, or location-based routing techniques. In this section we use a classification according to the network structure and protocol operation (i.e., routing criteria), and is shown in Figure 9.8. In majority of applications, sensor nodes are expected to be stationary. Thus, it may be preferable to have table driven routing protocols rather than employing reactive schemes where a significant amount of energy is used in

route discovery and setup. Another class of routing protocols is called the cooperative routing protocols. Where in SNs send data to a CH where data can be aggregated and may be subjected to further processing, hence reducing route cost in terms of energy use. Several other protocols, in turn, rely on timing and position information and are

also covered in this section.

Network Structure Based

In this class of routing protocols, the network structure is one of the determinant factors. In addition, the network structure can be further subdivided into flat, hierarchical and adaptive depending upon its organization.

Flat Routing

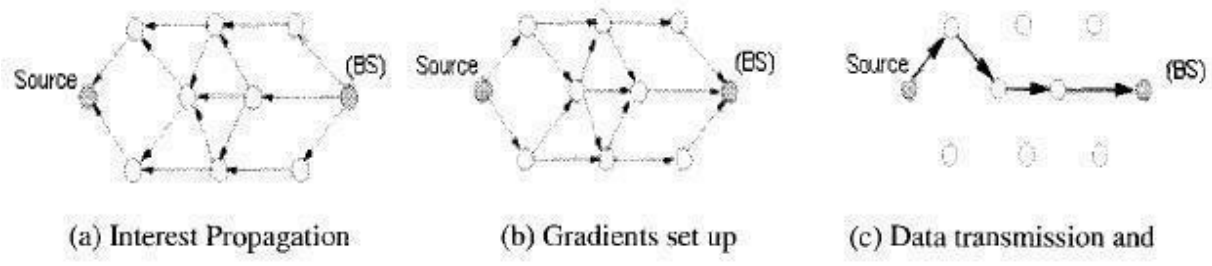
In flat routing based protocols, all nodes play the same role. Here, we present the mostprominent protocols falling in this category.

Directed Diffusion

Directed Diffusion [Intanagonwiwat2000] 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. Unlike traditional end-to-end routing, DC routing finds routes from multiple sources to a single destination (BS) that allows in-network consolidation of redundant data. In Directed Diffusion, sensors measure events and create gradients of information in their respective neighborhoods. 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. Figure 9.9 depicts an example of the operation of directed diffusion. Figure 9.9(a) presents the propagation of interests, Figure 9.9(b) shows the gradients construction, and Figure 9.9(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 ommunication 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.



Operation of directed diffusion

Sensor nodes in a directed diffusion-based network are application aware, which enables diffusion to achieve energy savings by choosing empirically good paths and by caching and processing data in the network. An application of directed diffusion is to spontaneously propagatean important event to regions of the sensor network. Such type of information retrieval is well suited for persistent queries where requesting nodes expect data that satisfy a query for a periodof time. However, it may be unsuitable for historical or one-time queries as it is not worth setting up gradients which employ the path only once. The performance of data aggregation methods employed by the directed diffusion paradigm is affected by the location of the source nodes in the network, the number of sources, and the network topology. In order to investigate these factors, two models of source placement shown in Figure 9.10 have been investigated. These models are called the event radius (ER) model, and the random sources (RS) model. In the ER model, a single point in the network area is defined as the location of an event. For example, this may correspond to a vehicle or some other phenomenon being tracked by the sensor nodes. All nodes within a distance S (called the sensing range) of this event that are not sinks, are considered to be data sources. In the RS model, K nodes that are not sinks are randomly selected to be sources. Unlike the ER model, in the RS model the sources are not necessarily closed to each other. In both of these source placement

models, however, for a given energy budget, a greater number of sources can be connected to the sink. Therefore, the energy savings with aggregation in directed diffusion can be transformed to provide a greater degree of robustness as per dynamics of the sensed activity.

Sequential Assignment Routing (SAR)

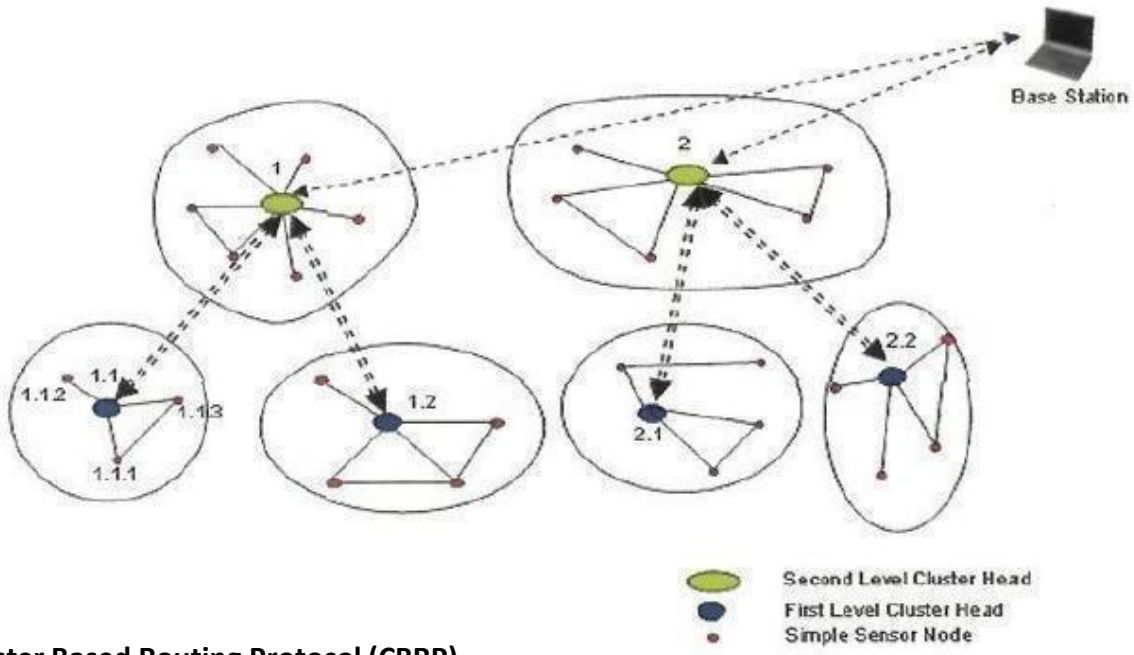
The routing scheme in SAR [Sohrabi2000] is dependent on three factors: energy resources, QoS on each path, and the priority level of each packet. To avoid single route failure, a multi-path approach coupled with a localized path restoration scheme is employed. To create multiple paths from a source node, 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. At the end of this process, each sensor node is part of multi-path tree. 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. Also, a path recomputation is carried out if the topology changes due to node failures. As a preventive measure, a periodic re-computation of paths is triggered by the BS to account for any changes in the topology. In addition, a handshake procedure based on a local path restoration scheme between neighboring nodes is used to recover from a failure.

Hierarchical Routing

Hierarchical, or cluster-based, routing has its roots in wired networks, where the main goals are to achieve scalable and efficient communication. As such, the concept of hierarchical routing has also been employed in WSN to perform energy-efficient routing. In a hierarchical architecture, higher energy nodes (usually called cluster heads) can be used to process and send the accumulated information while low energy nodes

can be used to sense in the neighborhood ofthe target and pass on to the CH. In these cluster-based architectures, creation of clusters and appropriate assignment of special tasks to CHs can contribute to overall system scalability, lifetime, and energy efficiency. An example of a general hierarchical clustering scheme is depicted in Figure 9.11. As we can see from this figure, each cluster has a CH which collects data from its cluster members, aggregates it and sends it to the BS or an upper level CH. For example in Figure 9.11, nodes 1.1.1, 1.1.2, 1.1.3 and 1.1 form a cluster with node 1.1 as the CH. Similarly, there exist other CHs such as 1.2, etc. These CHs, in turn, form a cluster with node 1 as their CH. So, node 1 becomes a second level CH as well. This pattern is repeated to form a hierarchy of clusters with the uppermost level cluster nodes reporting directly to the BS. The BS

forms the root of this hierarchy and supervises the entire network.



Hierarchical Clustering

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 adistributed 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 in [Heinzelman2000b] as a hierarchical clustering algorithm for sensor networks, called Low-Energy Adaptive Clustering Hierarchy (LEACH). LEACH is a good approximation of aproactive 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 dataimmediately. 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 asCHs.

The operation of LEACH is separated into two phases, the setup phase and the

steady state phase. In the setup phase, the clusters are organized and CHs are selected. In the steady state phase, actual data transfer to the BS takes place. Clearly, the duration of the steady state phase is longer than the duration of the setup phase in order to minimize overhead. During the setup phase, a predetermined fraction of nodes, say *p*, elect themselves as CHs as follows. A sensor node chooses a random number, say r, between 0 and 1. If this random number is less thana threshold value, say *T(n),* the node becomes a CH for the current round.

The threshold value, in turn, is calculated based on an equation that incorporates the desired percentage to become a CH, the current round, and the set of nodes that have not been selected as a CH in the last *{lip)*rounds, denoted by *G.* As a result, *T(n)* is given by:

T(n) =

-

ifneG

I-

p(rmo

d(l/

p))

where G is the set of nodes that are involved in the CH election. Each elected CH broadcast an advertisement message to the rest of the nodes in the network, informing that they are the new CHs. All the non- CH nodes, after receiving this

advertisement, decide on the cluster to which they want to attach to. In LEACH, this decision is based on the signal strength of the advertisement. The non-CH nodes then inform the corresponding CH of their decision to be a

member of its cluster. Based on the number of nodes in the cluster, the CH node creates a TDMA schedule and assigns each node a time slot within this schedule where it can transmit. This schedule is then broadcast to all cluster members. During the steady state phase, sensor nodes begin sensing and transmitting data to their respective CHs. Once the CH receives the data from all of its members, it aggregates before relaying data to the BS. After a period time, which is determined a priori, the network goes back into the setup phase and initiates another round for selecting new CHs. Although LEACH is able to increase the network lifetime, there are still a number of issues regarding many assumptions. For example, LEACH assumes that all nodes can transmit with enough power to reach the BS if needed, and that every node has enough computational power to support different MAC protocols. It also assumes that nodes always have data to send, and nodes located close to each other have correlated data. Also, it is not obvious how the number of the predetermined CHs (p) is going to be uniformly distributed through the network. Therefore, there is the possibility that the elected CHs be concentrated in one part of the network. Thus, some nodes will not at all find CHs in their proximity. Finally, the protocol assumes that all nodes begin with the same amount of energy capacity in each election round.

LEACH could be extended to account for non-uniform energy nodes, i.e., to use energy-based threshold. An extension to LEACH, also known as LEACH with negotiation, has been introduced in [Heinzelman2000b] with the goal of preceding data transfers with negotiations similar to meta-data descriptors used in the SPIN protocol discussed later. This ensures that only data that actually provides new information is transmitted to the CHs.

Multipath-Based Routing

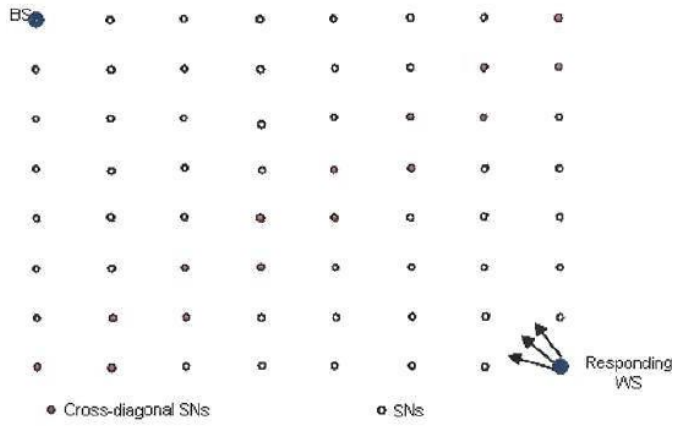
Network performance, and possibly lifetime, in WSNs can be significantly

improved if the routing rotocol is able to maintain multiple, instead of a single, paths to a destination, and protocols in this class are called multipath protocols. By employing multipath protocols, the fault tolerance (resilience) of the network is considerably increased. The fault tolerance of a protocol is measured by the likelihood that an alternate path exists between a source and a destination when the primary path fails. Clearly, this can be increased if we maintain multiple paths between the source and the destination at the expense of an increased energy consumption and traffic generation (i.e., overhead), as alternate paths are kept alive by sending periodic messages. We would also like to note here that multipath routes between a source and a destination can be or not node-disjoint. Multiple paths between a source and destination are said to be node-disjoint when there is no node overlap amongst them. For the purpose of our discussion here, we refer to

alternate routes as not being node disjoint, i.e., their routes are partially overlapped. In addition,unless otherwise noted, all multiple paths are of alternate types in this section.

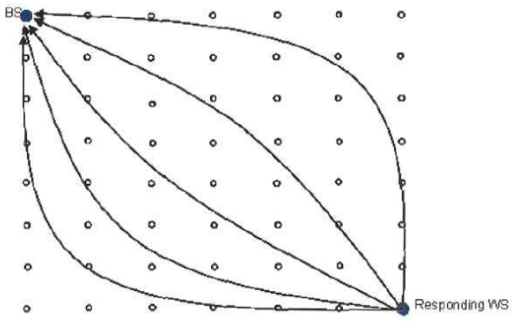
A set of suboptimal paths can be occasionally employed as to increases the network lifetime [Rahul2002]. These suboptimal paths are chosen by means of a probability which depends on how low the energy consumption of each path is. Packets are routed through a path with largest residual energy in the algorithm proposed in [Chang2000]. Here, the path is changed whenever a better path is discovered. The primary path is used until its energy falls below the energy of the backup path, at which time the backup path is used. By employing this mechanism, the nodes in the primary path do not deplete their energy resources through continual use of the same route, hence prolonging their lifetime. One issue with this scheme is the cost associated with switching paths, and how to deal with packets which are en-route. As we have seen before, there is a tradeoff between minimizing the total power consumed and the residual energy of a network. As a result, routing packets through paths with largest residual energy may turn out to be very energy-expensive.

To minimize this effect, it has been proposed in [Li2001b] a scheme in which the residual energy of the route is relaxed a bit in order to select a more energy efficient path. Multipath routing was employed in [Dulman2003] to enhance the reliability of WSNs. This scheme is useful for delivering data in unreliable environments. Here, reliability is enhanced by providing several paths from source to destination and by sending the same packet thorough each and every path. Obviously, by using this technique, traffic increases significantly. Therefore, there is a tradeoff between the amount of traffic and the network reliability. This tradeoff is investigated in [Dulman2003] using a redundancy function that is dependent on the multipath degree



(a) Routing Region and Cross-diagonal SNs

and on failing probabilities of the available paths.



(b) Paths Through cross-diagonal SNs

Determination of Multiple Paths

Directed Diffusion (discussed earlier). Based on the directed diffusion paradigm, a multipath routing scheme that finds several partially disjoint paths is

presented in [Ganesan2001j. The idea is that the use of multipath routing provides a viable alternative for energy efficient recovery from failures in WSN. The motivation of using these braided paths isto keep the cost of maintaining the multiple paths low. In this scheme, the costs of alternate paths are comparable to the primary path as they tend to be much closer to the primary path.

UNIT-V
Upper Layer Issues of WSN

Upper Layer Issues of WSN: Transport layer, High-level application layer support, Adapting to the inherent dynamic nature of WSNs, Sensor Networks and mobile robots.

High-Level Application Layer Support

The protocols we have presented so far are also found, albeit in some different form in traditional wired, cellular, or ad hoc networks. For specific applications, a higher level of abstraction specifically tailored to WSN appears to be useful. In this section, we outline someof the activities in this direction.

Distributed Query Processing:

The number of messages generated in distributed query processing is several magnitudes less than in centralized scheme. [Bonnet2000, Bonnet2001], discusses 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 [Madden2003] for query processing in homogeneous sensor networks.

Sensor Databases:

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. The problems encountered here are in finding energyefficiency ways of executing such queries and of defining proper query languages that can express the full richness of WSNs. The TinyDB project [TinyDBwww] carried out at the University of California at Berkeley is looking at these issues. A model for sensor database systems known as COUGAR [COUGARwww] defines appropriate user and internal representation of queries. The sensor queries are also considered so that it is easier to aggregate the data and to

combine two or morequeries. In COUGAR, routing of queries is not handled. COUGAR has three-tier architecture.

- The Query Proxy: A small database component running on the sensor nodes to interpretand execute queries
- A Front end Component: A query-proxy that allows the sensor network to connect tothe 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.

Distributed Algorithms:

WSNs are not only concerned with merely *sensing* the environment but also with interacting with the environment. Once actuators like valves are added to WSNs, the question of distributed algorithms becomes inevitable. One showcase is the question of distributed consensus, where several actuators have to reach a joint decision (a functionality which is also required for distributed software update, for example). This problem has been investigated to some degree for ad hoc networks [Malpani2000, Nakano2002, Srinivasan2003, Walter2001], but it has not been fully addressed in the context of WSNs where new scalability and reliability issues emerge and where the integration in the underlying, possibly data-centric routing architecture, has not yet been investigated.

Adapting to the Inherent Dynamic Nature of WSNs:

Some important goals that current research in this area is aiming to achieve are asfollows:

- Exploit spatial diversity and density of sensor/actuator nodes to build an adaptive node sleepschedule
- Spontaneously create and assemble network, dynamically adapt to device failure and degradation, manage mobility of sensor nodes and react to changes in task and sensor requirements
- Adaptability to drastic changes in the traffic
- Having finer control over the precision and coverage.

The Scalable Coordination Architectures for Deeply Distributed Systems (SCADDS)

project SCADDS www], also a part of DARPA Sens IT program [Sens IT www], focuses on adaptive fidelity, dynamically adjusting the overall fidelity of sensing in response to task dynamics (turn on more sensors when a threat is perceived). They use additional sensors (redundancy) to extend lifetime. Neighboring nodes are free to talk to each other irrespective of their listening schedules; there is no clustering and no inter-cluster communication and interference. Adaptive Self-Configuring Ensor Network Topologies (Ascent) [Cerpa2001b], which is part of SCADDS, focuses on how to decide which nodes should join the routing infrastructure to adapt to a wide variety of environmental dynamics and terrain conditions producing regions with non uniform communication density. A node signals and reduces its duty cycle when it detects high message loss, requesting additional nodes in the region to join the network in order to relay messages to it. It probes the local communication environment and does not join the multi-hop routing infrastructure until it is helpful to do so. In addition, it avoids transmitting dynamic state information repeatedly across the network

In-Network Processing

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 (although in-network processing may take place even in the absence of an associated query). 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 (also called spatio-temporal correlation).

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, 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 [Biswas2005, Jain2005c]: X= Value of the data sent to the higher level root of the tree (fr = Data reduction factor Figure 9.16 - A Heterogeneous WSN with resource rich wireless sensor nodes for data aggregation.

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 Data Retrieval in Sensor Networks 505 velocity". This type of queries could be possibly responded by RRWSs. So, the query processing in a WSN need to be correlated to access data at RRW nodes as query tree need to be mapped to the flow of data along the routing tree between the RRWS and to the BS].

The energy consumed in transmitting the query and receiving response from WSs and RRWSs, could represent the cost of the query and hence minimization of power consumption is fairly involved. In-network query processing for multi-target regions is addressed by an energy aware routing scheme for spatio-temporal queries Queries are then evaluated based on a computation plan that is provided to the sink in the form of a query tree. Query trees are, in turn, defined as a logical representation of operator hierarchy in a given query with target regions as leaf nodes.

A routing tree is defined as the set of routes constructed in the network to route data from target regions to the sink through the intermediate resource rich nodes, executing query operators in the order defined in the query tree. The problem of mapping a query tree to a routing tree is non-trivial. This query is usually specified in a declarative language like SQL containing operators such as selects, joins, projections and aggregations.

SENSOR NETWORKS AND MOBILE ROBOTS

Sensors for Mobile Robots

Why should a robotics engineer know about sensors?Is the key technology for perceiving the environmentUnderstanding the physical principle enables appropriate useUnderstanding the physical principle behind sensors enables us:To properly select the sensors for a given applicationTo properly model the sensor system, e.g. resolution, bandwidth, uncertainties.

Different sensors

differ in their precision and the kind of data that they provide, but none of them is able to completely solve the localization problem on its own.E.g. encoder measures position, but used in this function only on robotic arms.for nonholonomic robot: motions that return the encoder values to their initial position, do not necessarily drive the robot back to its starting point.Different data: e.g.accelerometer samples real-valued quantities that are digitized with some precisionodometer delivers discrete values that correspond to encoder increments.vision sensor delivers an array of digitized real values (colors).

- **Classifying sensors Type of information Physical Principle**
  Absolute vs. derivative
- Amount of information (Bandwidth)
- Low and high reading (Dynamic range)
- Accuracy and Precision