# Mobile Ad hoc Networking

**Carlos de Morais Cordeiro and Dharma P. Agrawal**
OBR Research Center for Distributed and Mobile Computing, ECECS
University of Cincinnati, Cincinnati, OH 45221-0030 – USA
{cordeicm, dpa}@ececs.uc.edu

*Abstract – Recent advances in portable computing and wireless technologies are opening up exciting possibilities for the future of wireless mobile networking. A Mobile Ad hoc NETwork (MANET) consists of mobile platforms which are free to move arbitrarily. This is in contrast with the topology of the existing Internet, where the router topology is essentially static (barring network configuration or router failures). In a MANET, the nodes are mobile and inter-node connectivity may change frequently during normal operation. In this course we will focus our attention on current protocols which provide connectivity in mobile ad hoc networks, such as routing and MAC protocols. Moreover, we will also cover an emerging promising area within ad hoc networks called Sensor Networks and demonstrate its wide applicability. We will conclude this course by discussing current challenges to mobile networking that have not received as much attention from the research community, and then highlighting some of the current wireless protocol standardization efforts within the IETF and the Bluetooth SIG (Special Interest Group).*

## 1. 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 to obtain 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 asymmetric links, 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.

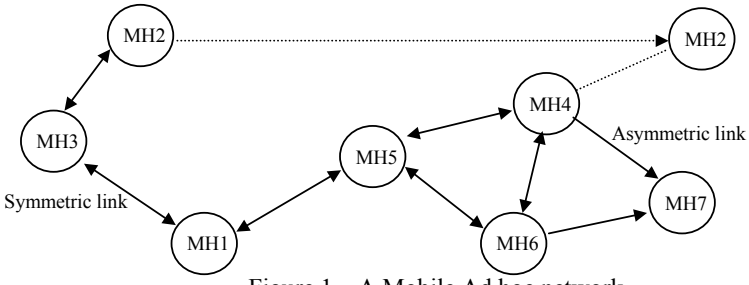


Figure 1 – A Mobile Ad hoc network

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 driving issue 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 surveillance 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 environmental monitoring.

Table 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. |

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.

### *1.1 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 collaborative computing 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 cooperate and 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.

## 2. Routing in a MANET

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

### *2.1 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 be forwarded, 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.

## 3. Unicast Routing Protocols

### *3.1 Proactive Routing Approach*

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

#### *3.1.1 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.

### *3.1.2 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.

### *3.2 Reactive Routing Approach*

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

#### *3.2.1 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.

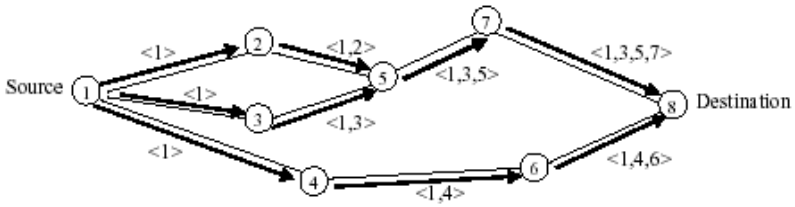



Figure 2(a) – Route discovery in DSR

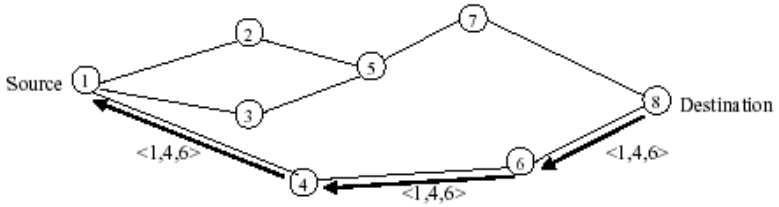



Figure 2(b) – Propagation of route reply in DSR

*3.2.2 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.

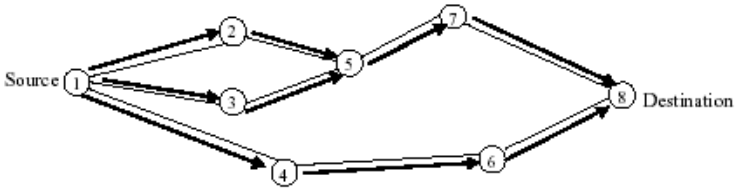



Figure 3(a) – Propagation of RREQ in AODV

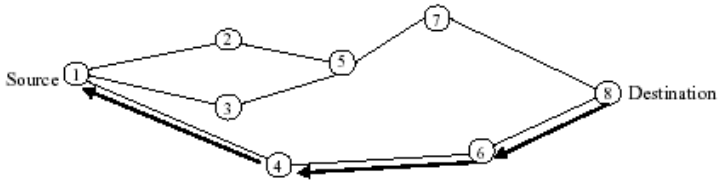



Figure 3(a) – Path taken by the RREP in AODV

*3.2.3 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.

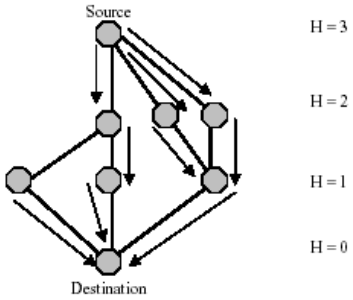


Figure 4 – TORA height metric

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.

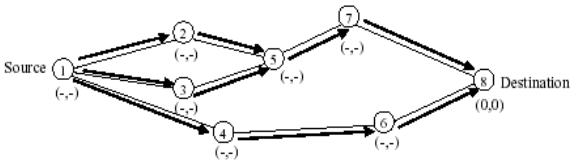



Figure 5(a) – Propagation of the query message

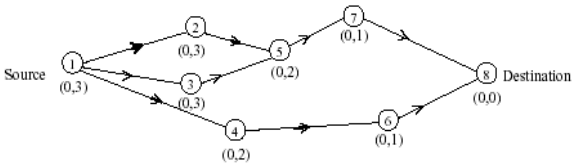



Figure 5(b) – Node's height updated as a result of the update message

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.

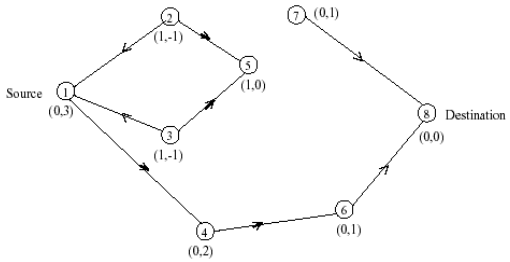


Figure 6 – Route maintenance in TORA

*3.2.4 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.

*3.2.4.1 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 $t_0$, and that the current time is $t_1$. Then, the *"expected zone"* of node D, from the viewpoint of node S at time $t_1$, 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 $t_0$. 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 $v(t_1 - t_0)$, 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 $t_1$. Thus, expected zone is only an estimate made by node S to determine a region that potentially contains D at time $t_1$.

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 ($X_d$; $Y_d$) of node D at some time $t_0$ – the time at which route discovery is initiated by node S is $t_1$, where $t_1 \geq t_0$. Node S calculates its distance from location ($X_d$; $Y_d$), denoted as $DIST_S$, and includes this distance with the route request message. The coordinates ($X_d$; $Y_d$) 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 ($X_d$; $Y_d$) than node I.

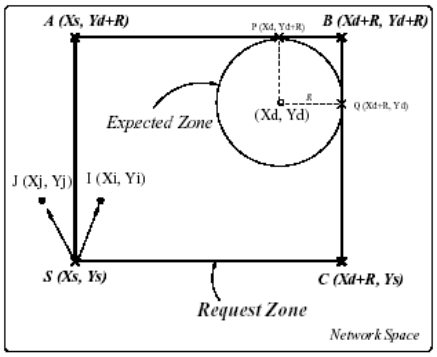


Figure 8 – LAR scheme

*3.2.4.2 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.

*3.2.4.3 Relative Distance Micro-Discovery Ad Hoc Routing*

The RDMAR (Relative Distance Micro-discovery Ad Hoc Routing) routing protocol [Aggelou 1999] is a highly adaptive, efficient and scaleable routing protocol. It is well-suited in large mobile networks whose rate of topological changes is moderate. A key concept in its design is that protocol reaction to link failures is typically localized to a very small region of the network near the change. This desirable behavior is achieved through the use of a novel mechanism for route discovery, called Relative Distance Micro-discovery (RDM). The concept behind *RDM* is that a query flood can be localized by knowing the relative distance (RD) between two terminals. To accomplish this, every time a route search between the two terminals is triggered, an iterative algorithm calculates an estimate of their RD, given an average nodal mobility and information about the elapsed time since they last communicated and their previous RD. Based on the newly calculated RD, the query flood is then localized to a limited region of the network centered at the source node of the route discovery and with maximum propagation radius that equals to the estimated relative distance. This ability to localize query flooding into a limited area of the network serves to minimize routing overhead and overall network congestion.

In RDMAR, calls are routed between the stations of the network by using routing tables which are stored at each station of the network; each node is treated as a host as well as a store-and-forward node. Each routing table lists all reachable destinations, wherein for each destination *i*, additional routing information is also maintained. This includes: the "D*efault Router*" field that indicates the next hop node through which the current node can reach *i*, the "*RD*" field which shows an estimate of the relative distance (in hops) between the node and *i*, the *"Time_Last_Update"* (TLU) field that indicates the time since the node last received routing information for *i*, a "*RT_Timeout*" field which records the remaining amount of time before the route is considered invalid, and a "R*oute Flag*" field which declares whether the route to *i* is active.

RDMAR comprises of two main algorithms:

- Route Discovery – When an incoming call arrives at node *i* for destination node *j* and there is no route available, *i* initiates a route discovery phase. Here, *i* has two options; either to flood the network with a route query in which case the route query packets are broadcast into the whole network, or instead, to limit the discovery in a smaller region of the network, if some kind of location prediction model for *j* can be established. The former case is straightforward. In the latter case, the source of the

route discovery, *i*, refers to its routing table in order to retrieve information on its previous relative distance with *j* and the time elapsed since *i* last received routing information for *j*. Let us designate this time as $t_{motion}$. Based on this information and assuming a moderate velocity, *Micro_Velocity*, and a moderate transmission range, *Micro_Range*, node *i* is then able to estimate its new relative distance to destination node *j* in terms of actual number of hops. To accomplish this, node *i* calculates the distance offset of *DST* (*DST_Offset*) during $t_{motion}$, and "adjusts" the result onto their previous relative distance (*RDM_Radius*).

- Route Maintenance – An intermediate node *i*, upon reception of a data packet, first processes the routing header and then forwards the packet to the next hop. In addition, node *i* sends an explicit message to examine whether a bi-directional link can be established with the previous node. RDMAR, therefore, does not assume bi-directional links but in contrast nodes *exercise* the possibility of having bi-directional links. In this way, nodes that forward a data packet will always have routing information to send the future acknowledgement back to the source. If node *i* is unable to forward the packet because there is no route available or a forwarding error occurs along the data path as a result of a link or node failure, *i* may attempt a number of additional re-transmissions of the same data packet, up to a maximum number of retries. However, if the failure persists, node *i* initiates a Route Discovery procedure.

### *3.3 Hybrid Routing Protocols*

#### *3.3.1 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.

### *3.3.2 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.

### *3.3.3 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.

### *3.4 Other Routing Protocols*

There is plenty of routing protocol proposals for mobile ad hoc networks. Our discussion here is far from being exhaustive. Below we will describe some other routing protocols which employ different optimization criteria as the ones we have previously described.

#### *3.4.1 Signal Stability Routing*

Another on-demand protocol is the Signal Stability-Based Adaptive Routing protocol (SSR) [Dube 1997]. Unlike the algorithms described so far, SSR selects routes based on the signal strength (weak or strong) between nodes and a node's location stability. The signal strengths of neighboring nodes are obtained by periodic beacons from the link layer of each neighboring node. This route selection criterion of SSR has the effect of choosing routes that have "stronger" connectivity [Chlamtac 1986].

#### *3.4.2 Power Aware Routing*

In this protocol, power-aware metrics [Singh 1998, Jin 2000] are used for determining routes in wireless ad hoc networks. It has been shown that using these metrics in a shortest-cost routing algorithm reduces the cost/packet of routing packets by 5 - 30 percent over shortest-hop routing (this cost reduction is on top of a 40-70 percent reduction in energy consumption over the MAC layer protocol used). Furthermore, using these new metrics ensures that mean time to node failure is increased significantly, but packet delays do not increase. A recent work [Lee 2000a] concentrates on selecting a route based the traffic and congestion characteristics in the network.

#### *3.4.3 Associativity-Based Routing*

This is a totally different approach in mobile routing. The Associativity-Based Routing (ABR) [Toh 1997] protocol is free from loops, deadlock, and packet duplicates, and defines a new routing metric for ad hoc mobile networks. In ABR, a route is selected based on a metric that is known as the degree of association stability. Each node periodically generates a beacon to signify its existence. When received by neighboring nodes, this beacon causes their associativity tables to be updated. For each beacon received, the associativity tick of the

current node with respect to the beaconing node is incremented. Association stability is defined by connection stability of one node with respect to another node over time and space. A high (low) degree of association stability may indicate a low (high) state of node mobility. Associativity ticks are reset when the neighbors of a node or the node itself move out of proximity. A fundamental objective of ABR is to derive longer-lived routes for ad hoc networks. The three phases of ABR are:

- Route discovery,
- Route reconstruction (RRC),
- Route deletion.

The route discovery phase is accomplished by a broadcast query and await-reply (BQ-REPLY) cycle. A node desiring a route broadcasts a BQ message in search of mobiles that have a route to the destination. All nodes receiving the query (that are not the destination) append their addresses and their associativity ticks with their neighbors along with QoS information to the query packet. A successor node erases its upstream node neighbors' associativity tick entries and retains only the entry concerned with itself and its upstream node. In this way, each resultant packet arriving at the destination contains the associativity ticks of the nodes along the route to the destination. The destination is then able to select the best route by examining the associativity ticks along each of the paths. When multiple paths have the same overall degree of association stability, the route with the minimum number of hops is selected. The destination then sends a REPLY packet back to the source along this path. Nodes propagating the REPLY mark their routes as valid. All other routes remain inactive, and the possibility of duplicate packets arriving at the destination is avoided.

RRC may consist of partial route discovery, invalid route erasure, valid route updates, and new route discovery, depending on which node(s) along the route move. Movement by the source results in a new BQ-REPLY process. The RN message is a route notification used to erase the route entries associated with downstream nodes. When the destination moves, the immediate upstream node erases its route and determines if the node is still reachable by a localized query (LQ[H]) process, where H refers to the hop count from the upstream node to the destination. If the destination receives the LQ packet, it REPLYs with the best partial route; otherwise, the initiating node times out and the process backtracks to the next upstream node. Here, a RN [0] message is sent to the next upstream node to erase the invalid route and inform this node that it should invoke the LQ[H] process. If this process results in

backtracking more than halfway to the source, the LQ process is discontinued and a new BQ process is initiated at the source.

### *3.5 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 |

## 4. Multicast Routing Protocols

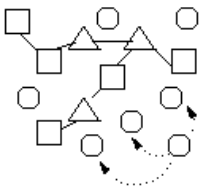
Multicasting is the process of sending packets from a transmitter to multiple destinations identified by a single address. As with their wired counterparts, multicasting in MANET is also a hard task to accomplish, and it is even harder in the MANET case since the physical topology changes quite frequently. Therefore, multicast protocols designed for a MANET have to take topological changes into consideration. In this section, we discuss two multicast routing protocols, namely AODV Multicasting and ODMRP, proposed within the MANET [MANET] working group at the IETF [IETF].

### *4.1 AODV Multicasting (MAODV)*

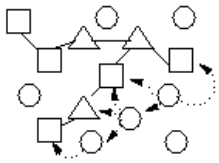
The AODV multicast algorithm [Royer 1999] uses similar RREQ and RREP messages as in unicast operation. The nodes join the multicast group on-demand, and a multicast tree is created in the process. The tree consists of the group members and nodes connected to the group members. This enables a recipient 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.

### *4.1.1 Route Discovery*

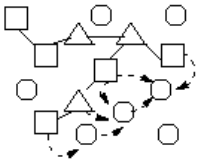
When a node wishes to find a route for a multicast group, it sends an RREQ message. The destination address in the RREQ message is set to the address of the multicast group. If the node wants to join the group in question, the *J_flag* in the message is set. Any node may respond to a RREQ merely looking 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 (see Figure 9).



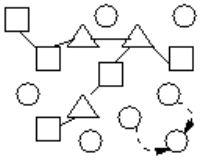
(a) A node sends RREQ to join multicast group



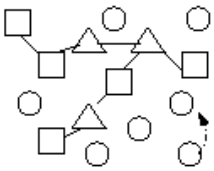
(b) Neighboring nodes rebroadcast the request



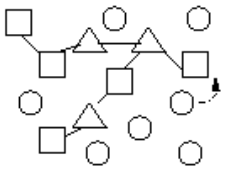
(c) Tree members and Group members send RREPs



(d) Duplicate RREPs are dropped



(e) MACT is sent to the node that has the shortest path



(f) MACT is forwarded to a group member


(g) The intermediate node becomes a tree member in the process

Figure 9 – AODV multicast tree branch addition

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 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.

### *4.1.2 Multicast Tree Maintenance*

In a network consisting of mobile nodes, link breakages are bound to happen. The breakages should be repaired promptly to ensure maximal connectivity of the multicast group. Multicast tree maintenance has three different scenarios: activating a link when a new node joins the group, pruning the tree when a node leaves the group, and repairing a broken link. Repairing consists of re-establishing the branches when a link goes down and reconnecting the tree after a possible partition in the network.

**Route activation.** If a node receives more than one RREP to a RREQ it has sent, it must pick only one RREP as the next hop. This avoids adding any extra branches to the tree and loops are eliminated. The source node waits for a specified Route Discovery timeout after it has sent an RREQ, and then selected the received route that has the greatest DSN and the

fewest hops to the nearest member of the multicast tree. The node then *unicasts* a multicast-specific message called Multicast Activation (MACT) to the selected next hop. The MACT message carries the source address, SSN, destination address and flags *P_flag* and *GL_flag*. P_flag is used in case of pruning and GL_flag is set when selecting the multicast group leader, as explained later.

When a group member receives a MACT message, it enables the entry for the source node in its own multicast routing table. If a node that is not a group member receives a MACT message, it acts in a similar way as the source node. A new MACT is sent to the best next hop towards the multicast group. Those nodes that have generated or forwarded RREP messages delete the corresponding route entries from their routing tables if they do not receive a MACT within a specified Multicast Tree Build timeout period.

The multicast tree can never have multiple paths to any tree node because the MACT messages are only propagated through one path. Hence, the tree is indeed a tree. Because the nodes forward data only along the activated routes in their routing tables, the data can never be forwarded to the source node by multiple intermediate nodes.

**Pruning.** Multicast group members sometimes remove themselves from the group. If a non-leaf node decides to leave the group it must continue as a router for the multicast tree. Leaf nodes may prune themselves by sending a MACT message with the *P_flag (prune)* set. The *Dest_addr* of the message points to the multicast group in question.

Because a leaf node can only have one next hop node for the multicast group, the MACT can be unicasted to that node. After the MACT has been sent, information about the group can be removed from the route table. When the recipient of the MACT notices the *P_flag*, it deletes the entry for the sender of the MACT from its own route table. If the recipient is not a member of the multicast group and it would become a leaf node after this operation, it can prune itself from the multicast tree using the same method. The pruning process terminates when it reaches either a multicast group member or a non-leaf node.

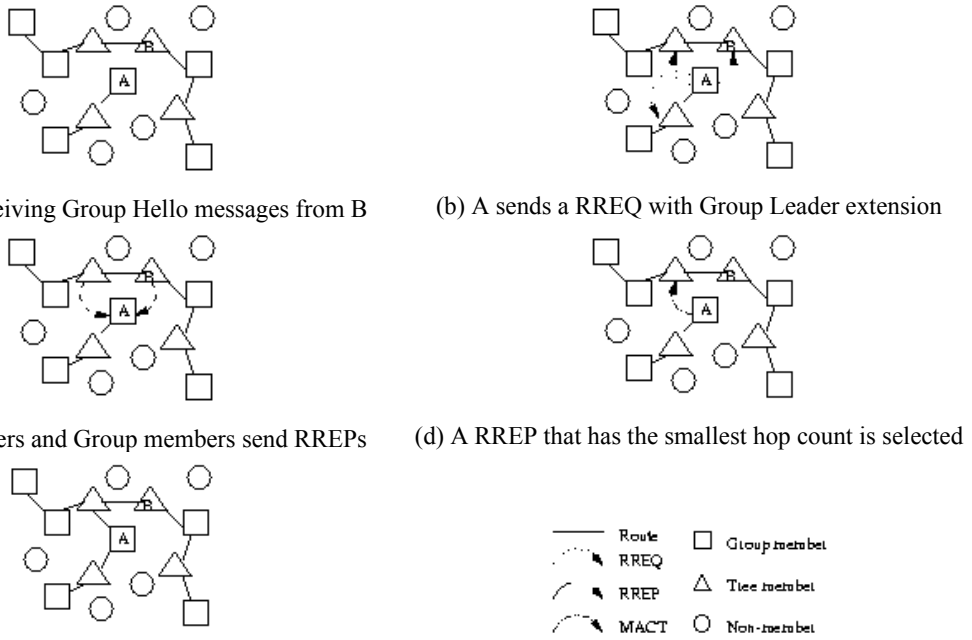
### *4.1.3 Triggered Repair of Broken Links*

Timeouts and node mobility cause links to break within the multicast tree. In multicast operation, routes must be reconstructed when a link goes down since current group members must stay connected.

If a node does not receive Group Hello messages from a neighbor within an acceptable timeout period, the link between the two nodes is considered broken (see Figure 10). The

node that is further away from the multicast group leader tries to rejoin the group as explained earlier. The RREQ carries a *Multicast Group Leader Extension* with a Hop Count that equals the distance to the group leader. However, the RREQ message is first broadcast with a restricted TTL value (two more than the recorded Multicast Group Hop Count). Because only one node starts the repair process, loops and duplicate work can be avoided. The use of a small TTL is an effort to keep the effects of the link breakage local. Only if no RREP is received within a timeout period, the RREQ is rebroadcast over the whole network.

A node that is nearer to the group leader than the RREQ hop count indicates is allowed to respond to the request. Duplicate RREPs are discarded. When a RREP has been received, the route must be activated. If the node was not a leaf node, it must also inform the downstream nodes of possible changes in distance to the group leader. In this case, a MACT message with an *U_flag* (Update) is broadcast. Those nodes that are not downstream neighbors of the sender ignore the update messages.




(a) A stops receiving Group Hello messages from B

(b) A sends a RREQ with Group Leader extension

(c) Tree members and Group members send RREPs

(d) A RREP that has the smallest hop count is selected

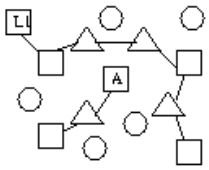
(e) MACT is sent to the sender of the chosen RREP

Figure 10 – AODV multicast tree repair

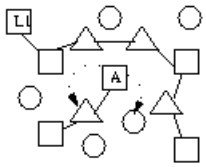
If the node that started the repair process never gets a RREP after a set number of retries, a partition is assumed to be present in the network (see Figure 11). In this case a new group leader must be elected. If the said node was a part of the multicast group, it becomes the new group leader. Otherwise it sends a pruning MACT message to its next hop, if there is only one of them. When the next hop node receives the MACT with *P_flag* set, it has the same options as the originating node. This is repeated until a new group leader is found.

If there is more than one next hop downstream, a node cannot prune itself. Instead it sends a MACT with *GL_flag* (Group Leader) set to the first of its next hops. This process is repeated until a multicast group member receives the MACT. This node then becomes the new group leader and it broadcasts a Group Hello message with *U_flag* set. The nodes that receive the Hello then update their route tables accordingly. Meanwhile, if the node upstream of the breakage became a non-member leaf node, it waits for a MACT from the downstream node. If it is not received within a set timeout period, the node prunes itself from the tree.

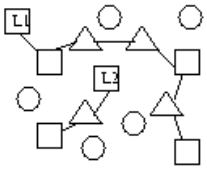
**Reconnecting two partitioned trees.** After a network partition, there are two group leaders (see Figure 12). A node can detect reconnection of the partitions if it receives a Group Hello with conflicting information about the group leader and hop count. The group leader that has the lower IP address (L2) initiates the repair process by sending a RREQ with *R_flag* (Repair) and *J_flag* set to the other group leader with the higher IP address (L1). The RREQ is sent via the next hop from which the Hello message was received. The current DSN for L2 is included in the message. The RREQ is only propagated upstream towards L1 so that no loops can appear. When L1 receives the RREQ, it creates a new DSN that is higher than its own or the one carried by the RREQ. It sends back a RREP message with the *R_flag* set and becomes the new leader.




(a) A stops receiving Group Hello messages from L




(b) A broadcasts a RREQ with Group Leader extension




(c) After a set number of retries No RREP is received.
A becomes a new Group Leader (L2)

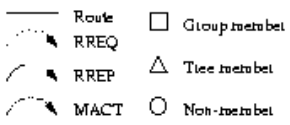
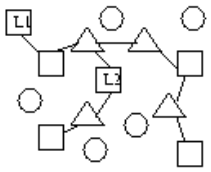



Figure 11 – Detecting a network partition

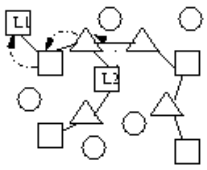
The nodes that used to belong to the group lead by L2 propagate the RREP towards L2. The node where the RREP was received from is marked as the next hop upstream and the next hop towards L2 is marked as being downstream. When L2 finally receives the message, it acknowledges L1 as the new leader, thus completing the reconnection of the tree. The next time that L1 sends a Hello message it sets the *U_flag* so that all the nodes will update their routing information.

*4.2 On-Demand Multicast Routing Protocol (ODMRP)*

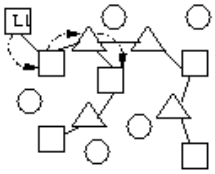
ODMRP (On-Demand Multicast Routing Protocol) [Bae 2000] is a *mesh*-based, instead of tree-based, multicast protocol that provides richer connectivity among multicast members. By building a mesh and supplying multiple routes, multicast packets can be delivered to destinations in the face of node movements 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 [Chiang 1998]. The forwarding group is a set of nodes responsible for forwarding multicast data on shortest paths between any member pairs. ODMRP also applies on-demand routing techniques to avoid channel overhead and improve scalability. No explicit control message is required to leave a group.




(a) L2 starts receiving Group Hello messages that point to L1 as the leader




(b) L2 unicasts a RREQ to L1 with *J_Flag* and *R_Flag* set




(c) L1 becomes the leader of the reconnected tree

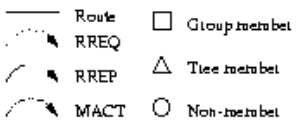



Figure 12 – Reconnecting partitioned subtrees

*4.2.1 Multicast Route and Mesh Creation*

In ODMRP, group membership and multicast routes are established and updated by the source on demand. Similar to on-demand unicast routing protocols, a request phase and a reply phase constitute the protocol (see Figure 13). While a multicast source has packets to send, it periodically broadcasts to the entire network a member advertising packet, called a JOIN QUERY, which refreshes the membership information and updates the route as follows. When a node receives a non-duplicate JOIN QUERY, it stores the upstream node ID (i.e., backward learning) and rebroadcasts the packet. When the JOIN QUERY packet reaches a multicast receiver, the receiver creates or updates the source entry in its *Member Table*. While valid entries exist in the *Member Table*, JOIN REPLIES are broadcasted periodically to the

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

Figure 14 visualizes the forwarding group concept. The forwarding group is a set of nodes in charge of forwarding multicast packets. It supports shortest paths between any member pairs. All nodes inside the *bubble* (multicast members and forwarding group nodes) forward multicast data packets. Note that a multicast receiver can also be a forwarding group node if it is on the path between a multicast source and another receiver. The mesh provides richer connectivity among multicast members compared to trees. Flooding redundancy among forwarding group helps overcome node displacements and channel fading. Hence, unlike trees, frequent reconfigurations are not required.

Figure 15 is an example to show the robustness of a mesh configuration. Three sources ($S_1$, $S_2$, and $S_3$) send multicast data packets to three receivers ($R_1$, $R_2$, and $R_3$) via three forwarding group nodes (A, B, and C). Suppose the route from $S_1$ to $R_2$ is < $S_1$-A-B- $R_2$>. In a tree configuration, if the link between nodes A and B breaks or fails, $R_2$ cannot receive any packets from $S_1$ until the tree is reconfigured. ODMRP, on the other hand, already has a redundant route <$S_1$-A-C-B-$R_2$> to deliver packets without going through the broken link between nodes A and B.

After group establishment and route construction process, a multicast source can transmit packets to receivers via selected routes and forwarding groups. Periodic control packets are sent only when outgoing data packets are still present. When receiving a multicast data packet, a node forwards it only if it is not a duplicate and the setting of the FG_Flag for the multicast group has not expired. This procedure minimizes traffic overhead and prevents sending packets through stale routes.

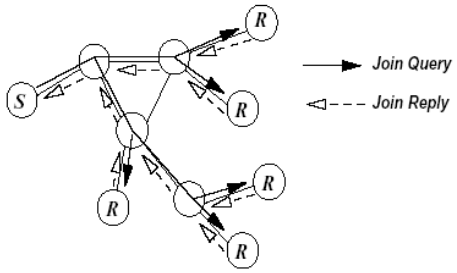



Figure 13 – On-demand procedure for membership setup and maintenance

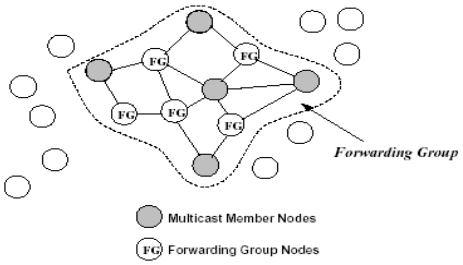



Figure 14 – The forwarding group concept

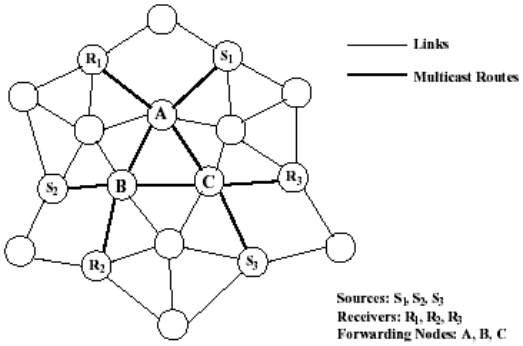



Figure 15 – The robustness of a mesh organization

*4.3 Other Protocols*

There are many more protocols with different characteristics for multicast in a MANET. They include: AMRoute [Bommaiah 1998], AMRIS [Wu 1998], CAMP [Garcia-Luna-Aceves 1999a], and flooding. A comparison of these protocols is performed in [Lee 2000b].

## 5. Medium Access Control (MAC) Protocols Issues

Maybe not as much as ad hoc routing protocols, but MAC protocols have also been receiving attention from the research community. There are still many issues that need to be addressed in order to design an efficient MAC protocol to be used in a wireless ad hoc network environment [Royer 2000]. There are several MAC protocols which can be employed for multi-hop ad hoc networking including IEEE 802.11 [Crow 1997], Bluetooth [Bluetooth] and HiperLAN [HiperLAN 1995]. Usually, the IEEE 802.11 standard is the platform employed to experiment multi-hop networking. However, it does not support multi-hop as is. In this section, we discuss some fundamental issues MAC protocols for wireless multi-hop ad hoc networks have to cope up with, along with their proposed solutions.

*5.1 Hidden Terminal Problem*

In CSMA, every station senses the carrier before transmitting, and if it detects carrier then the transmission is deferred. Carrier sense attempts to avoid collisions by testing the signal strength in the vicinity of the transmitter. However, collisions occur at the receiver, not the transmitter; i.e., it is the presence of two or more interfering signals at the receiver that constitutes a collision. Since the receiver and the sender are typically not co-located, carrier sense does not provide the appropriate information for collision avoidance. Two examples illustrate this point in more detail. Consider the configuration depicted in Figure 16. Station A can hear B but not C, and station C can hear station B but not A (and, by symmetry, we know that station B can hear both A and C). First, assume A is sending to B. When C is ready to transmit (perhaps to B or perhaps to some other station), it does not detect carrier and thus commences transmission; this produces a collision at B. Station C's carrier sense did not provide the necessary information since station A was "hidden" from it. This is the classic "hidden terminal" scenario.

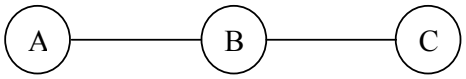


Figure 16 – Station B can hear both A and C, but A and C cannot hear each other.

An "exposed" terminal scenario results if now we assume that B is sending to A rather than A sending to B. Then, when C is ready to transmit, it does detect carrier and therefore defers transmission. However, there is no reason to defer transmission to a station other than B since station A is out of C's range. Station C's carrier sense did not provide the necessary information since it was exposed to station B even though it would not collide or interfere with B's transmission. The point to note here is that carrier sense provides information about potential collisions at the sender, but not at the receiver. This information can be misleading when the configuration is distributed so that not all stations are within range of each other.

The solution to the hidden terminal problem was proposed in [Karn 1990]. It consists of transmitting RTS (Request-to-Send) and CTS (Clear-to-Send) packets between nodes that wish to communicate. Among other things, these RTS and CTS packets carry the duration of the data transfer of the communicating parties. Stations in the neighborhood that do not participate in the communication but overhear either the RTS or CTS keep quiet for the duration of the transfer. Returning to our example of Figure 16, when node A wants to send a packet to node B, node A first sends a RTS packet to B. On receiving the RTS packet, node B responds by sending a CTS packet (provided node A is able to receive the packet). As a result of that, when node C overhears the CTS sent by B it keeps quiet for the duration of the

transfer contained in the CTS packet. As for the exposed terminal problem, while in IEEE 802.11 MAC layer there is almost no scheme to deal with it, MACAW [Bharghavan 1994] solves this problem by having the source transmit a data sending (DS) control packet to alert exposed nodes of the impending arrival of an ACK packet.

### *5.2 Reliability*

Wireless links are prone to errors. Actually, packet error rates of wireless mediums are much higher than that of their wired counterparts. As a result, some protocols – which were originally designed to work in wired world – suffer performance degradation when operating in a wireless environment. A classic example of this problem is TCP (which has been designed and fine-tuned for wired networks) that assumes transmission timer expiration as an indication of network congestion. This event triggers the execution of TCP congestion control mechanisms which ultimately decreases the transmission rate aiming at reducing the network congestion. As a matter of fact, this is often true in wired environments since wired media are usually very reliable. However, in wireless environment this is often not the case. Due to effects such as multipath fading, interference, shadowing, distance between transmitter and receiver, etc., packet losses occur every now and then. As a result, when a packet loss takes place in a communication using TCP, it erroneously assumes that the loss was due to congestion and enters its congestion control mechanisms. There have been some proposals to cope up with this TCP behavior in a MANET [Holland 1999, Liu 2001, Chandran 2001].

As for the MAC protocol issues, a common approach to reduce packet loss rates experienced by upper layers is to introduce acknowledgment (ACK) packets. Returning to our earlier example of Figure 16, whenever node B received a packet from node A, node B sends an ACK packet to A. In case node A fails to receive the ACK from B, it will retransmit the packet. This approached is adopted in many protocols [Bharghavan 1994]. As an example, the IEEE 802.11 DCF (Distributed Coordination Function) [Crow 1997] uses RTS-CTS to avoid the hidden terminal problem and ACK to achieve reliability.

### *5.3 Collision Avoidance*

The radios used in the wireless mobile nodes employed in wireless communications are half-duplex. This is to say that these radios are not able to transmit and receive at the same and, thus, collision detection is not possible. To minimize collisions, wireless MAC protocols, such as CSMA/CA, often use collision avoidance techniques in conjunction with a carrier

sense (be it physical or virtual) mechanism. Carrier sense is the mechanism whereby nodes wishing to transmit data first have to check whether the medium is idle or busy. The idea is that a station cannot transmit until the channel is idle. Collision avoidance is implemented by mandating that, when the channel is sensed idle, the node has to wait for a randomly chosen duration before attempting to transmit. This mechanism drastically decreases the probability that more that one node attempts to transmit at the same time, hence, avoiding collision. Of course there will be cases where more than one node initiate their transmission at the same time. In these cases, transmissions are corrupted and the corresponding nodes retry later on.

*5.4 Congestion Avoidance*

Congestion avoidance is another aspect of fundamental importance in wireless MAC protocols. In IEEE 802.11 DCF, when a node detects the medium to be idle, it chooses a *backoff interval* between [0, CW], where *CW* is called contention window which usually has a minimum (CW_min) and maximum value (CW_max). The idea is that the node will count down the backoff interval and when it reaches zero the node can transmit the RTS. In case the medium becomes busy while the node is still counting down the backoff interval, the countdown process is suspended.

To illustrate how DCF works, let us consider the example in Figure 17. In this figure, $BO_1$ and $BO_2$ are the backoff intervals of nodes 1 and 2, and we assume for this example that CW = 31. As we can see from Figure 17, node 1 and node 2 have chosen a backoff interval of 25 and 20, respectively. Obviously, node 2 will reach zero before five units of time earlier than node 1. When this happens, node 1 will notice that the medium became busy and freezes its backoff interval currently at 5. As soon as the medium becomes idle again, node 1 resumes its backoff countdown and transmits its data once the backoff interval reaches zero. Similarly, upon node's 1 transmission, node 2 will freeze its backoff countdown process and resume it as soon as node 1 finishes its transmission. To a certain extent, collision can be avoided by carrying out this procedure provided we choose a suitable value for the CW parameter. Choosing a large CW leads to large backoff intervals and can result in larger overhead, since nodes have to carry out the countdown procedure. On the other hand, choosing a small CW leads to a larger number of collisions, that is, it is more likely that two nodes will count to zero simultaneously.

*5.5 Congestion Control*

As we have mentioned earlier, the number of nodes attempting to transmit simultaneously may change with time. Therefore, some mechanism to manage congestion is needed. In IEEE 802.11 DCF, congestion control is achieved by dynamically choosing the contention window CW. When a node fails to receive a CTS in response to its RTS, it assumes that congestion has built up and, as a consequence, doubles its contention window up to CW_max. When a node successfully completes its transmission, it restores its contention window to CW_min. This mechanism of dynamically controlling the contention window is called Binary Exponential Backoff, since the contention window increases exponentially with a failed CTS.

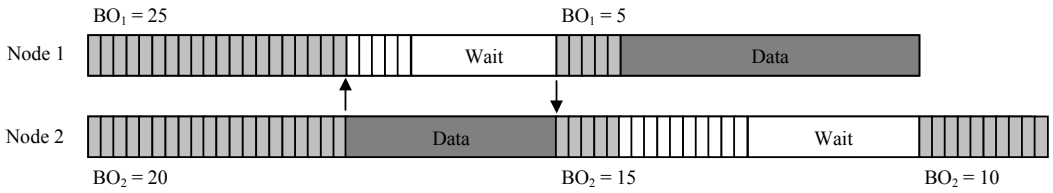


Figure 17 – Example of the backoff mechanism in DCF

*5.6 Energy Efficiency*

Since many mobile hosts are operated by batteries, there is an increasing interest for MAC protocols that conserve energy. The current proposals in this area usually suggest turning the radio off when it is not needed. IEEE 802.11 has a Power Saving (PS) mode whereby the Access Point (AP) periodically transmits a beacon indicating which nodes have packets waiting for them. Each PS node wakes up periodically to receive the beacon transmitted by the AP. In case a node has a packet waiting for it, it sends a PS-POLL packet to the AP after waiting for a backoff interval in [0, CW_min]. Upon receipt of the PS-POLL packet, the AP transmits the data to the requesting node. Using this procedure, it is possible to extend the battery life of mobile nodes for a longer period of time.

*5.7 Other MAC Issues*

Similarly to ad hoc routing protocols discussed earlier, our discussion on MAC protocol issues is also far from being exhaustive. There are many other issues to be considered such as *fairness*. Fairness has many meanings and one of them might say that stations should receive equal bandwidth. With the basic approach of IEEE 802.11, this fairness is not easy to accomplish since unfairness will eventually occur when one node backs off much more than some other node. MACAW's solution to this problem [Bharghavan 1994] is to append the contention window value (CW) to packets a node transmits, so that all nodes hearing that CW

use it for their future transmissions. Since CW is an indication of the level of congestion in the vicinity of a specific receiver node, MACAW proposes maintaining a CW independently for each receiver. There are also other proposals such as Distributed Fair Scheduling [Vaidya 2000] and Balanced MAC [Ozugur 1998].

A final comment must be made on receiver-related issues in wireless MAC protocols. All protocols discussed so far are sender-initiated protocols. In other words, a sender always initiates a packet transfer to a receiver. The receiver might take a more active role in the process by assisting the transmitter in certain issues such as collision avoidance [Garcia-Luna-Aceves 1999b], and some sort of adaptive rate control [Holland 2001].

## 6. Wireless Sensor Networks

Wireless Sensor Networks (WSN) [Estrin 1999, Kahn 1999] are a recent application of ad hoc networks that is 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 networks where data is requested from a specific node, data is requested based on certain attributes such as, "which area has temperature 100ºF". The routing protocols proposed for all the traditional networks are point-to-point and so these protocols are not well suited for wireless sensor networks.

Sensor networks consist of thousands of tiny disposable, low power devices equipped with programmable computing, multiple sensing and communication capability. They operate and respond in a very dynamic environment and their design must be application specific. Judging by the interest shown by the military, academia, and the media, innumerable applications exist for sensor networks. Examples include weather monitoring, security and tactical surveillance, distributed computing, fault detection and diagnosis in machinery, detecting ambient conditions such as temperature, movement, sound, light, or the presence of certain objects, etc. One example of a wireless sensor network is illustrated in Figure 18, with sensor nodes being deployed from a low-flying airplane. The example in this figure is the most quoted one, whereby a large number of sensors are dropped on the enemy's territory from an airplane so that activities on the ground can be detected and communicated.

A sensor node is basically a device that converts a sensed attribute (such as temperature, vibrations) into a form understandable by the users. Each of such devices may include a sensing module, a communicating module (display or a media to transmit data to the user), memory (to hold data till it can be used) and a power supply for the sensor.

Wired sensor networks have been used for years for a number of applications. Some examples include distribution of thousands of sensors and wires over strategic locations 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 structure. These wired sensors are made large (and expensive) to cover as much area as desirable. Each of these has a continuous power supply and communicate 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. These networks are usually unattended and some degree of fault-tolerance needs to be incorporated so that maintenance is minimized. This is especially desirable in those applications where the sensors may be embedded in the structure or places in an inhospitable terrain and are inaccessible for any service.

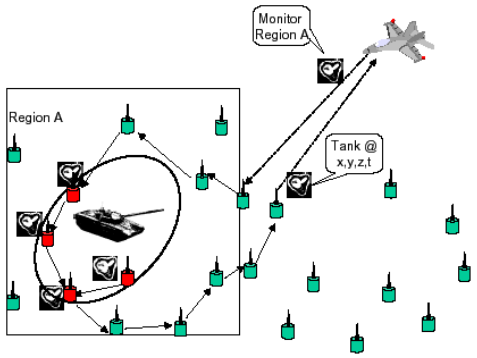



Figure 18 – A wireless sensor network

The advancement in technology has made it possible to have extremely small, low powered devices equipped with programmable computing, multiple parameter sensing and wireless communication capability. Also, the low cost of sensors makes it possible to have a network of hundreds or thousands of these wireless sensors, thereby enhancing the reliability, accuracy of data and the area coverage. Moreover, it is necessary that the sensors be easy to deploy (i.e., require very low or no installation cost etc). In short, the advantages of wireless sensor networks over wired ones are as follows:

- Ease of deployment – These wireless sensors can be deployed (dropped from plane or placed in factory) at the site of interest without any pre-organization, thus reducing the installation cost and increasing the flexibility of arrangement.

- Extended range – One huge wired sensor (Macro-sensor) can be replaced by many smaller wireless sensors for the same cost. One macro-sensor can sense only a limited region whereas a network of smaller sensors can be distributed over a wider region.
- Fault tolerant – Since sensor networks are mostly unattended, they should possess fault tolerant capability. With macro-sensors, the failure of one node makes that area completely unmonitored till it is replaced. Whereas with wireless sensors, failure of one node does not affect the network operation considerably as there are other nodes collecting similar data. At most, the accuracy of data collected may be reduced.
- Mobility – Since these wireless sensors are equipped with battery, they can possess limited mobility (e.g., 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. Note, however, that these nodes have lower mobility as compared to ad hoc networks.

There are a few inherent limitations of wireless medium such as low bandwidth, error prone transmissions, need for collision free channel access, etc. It is clear due to the nature of observed phenomenon, the required bandwidth of sensor data will be low, on the order of 1-100 kb/s. Since the wireless nodes are mostly mobile and are not connected in any way to a constant power supply, they derive energy from a personal battery. This limits 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 we can replace all the nodes simultaneously or put new nodes on the whole area. Finding individual dead nodes and then replacing those nodes selectively would require pre-planned deployment and eliminates some advantages of these networks. Thus, the protocols designed for these networks must strategically distribute the dissipation of energy, which also increases the average life of the overall system. In addition, environments in which these nodes operate and respond are very dynamic, with fast changing physical parameters.

Traditional routing protocols defined for wireless ad hoc networks are not well suited for wireless sensor networks due to the following reasons:

- Sensor networks are “data centric”, i.e., unlike traditional networks where data is requested from a specific node, data is requested based on certain attributes such as “which area has temperature 50ºF”.

- In traditional wired and wireless networks, each node is given a unique id, used for routing. This cannot be effectively used in sensor networks because these networks, being data centric, routing to and from specific nodes is not required.
- Adjacent nodes may have similar data. So, rather than sending data separately from each 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 one selected attribute (viz., attribute is fixed in this network).

Thus, sensor networks need protocols which are application specific, data centric, capable of aggregating data and minimizing energy consumption. An ideal sensor network should have the following additional features:

- *Attribute based addressing* is typically employed in sensor networks. The attribute-based addresses are composed of a series of attribute-value pairs which specify certain physical parameters to be sensed. For example, an attribute address may be (temperature > 40ºF, location = "Rio de Janeiro"). So, all nodes located in "Rio de Janeiro" which sense a temperature greater than 40ºF 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 additional feature. Users, using hand held wireless devices, should be able to request data from the network. Since these hand-held devices are also energy constrained, the user should be able to query through the base station 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. Since efficient query handling is a highly desirable feature, we explore it in the next section.

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 be answered. All other data is retrieved from the system on-demand.

### *6.1 DARPA Efforts Towards Wireless Sensor Networks*

The Defense Advanced Research Projects Agency (DARPA) has identified networked micro sensors technology as a key application for the future. There are many interesting projects and experiments going under the DARPA SensIT (Sensor Information Technology) program [SensIT]. The SensIT program aims to develop the software for distributed micro-sensors. On the battlefield of the future, a networked system of smart, inexpensive and plentiful microsensors, combining multiple sensor types, embedded processors, positioning ability and wireless communication, will pervade the environment and provide commanders and soldiers alike with heightened situation awareness. Therefore, software is needed to enable a variety of sensor nets, on the ground and in the air as well as on buildings and bodies, all functioning autonomously, operating with high reliability, and processing signals and information collaboratively in the network to provide useful information to the warfighter in a timely manner. In this text, we will focus on routing and MAC protocols targeted at supporting these wireless sensor networks.

### *6.2 Classification of Sensor Networks*

Looking at the various ways in which one can employ the network resources, sensor networks can be classified on the basis of their mode of operation or functionality, and the type of target applications. Accordingly, sensor networks are classified into two 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.

Once the type of network is decided, protocols that efficiently route data from the nodes to the users have to be designed, preferably using a suitable MAC sub-layer protocol to avoid collisions. Attempts should be made to distribute energy dissipation evenly among all nodes in the network as we do not have specialized high energy nodes in the network.

There are some basic functionalities and characteristics expected from a protocol for proactive networks. To illustrate this fact, let us take as an example a hierarchical clustering scheme whereby a group of nodes, called cluster members, synchronize and elect one of its members as the cluster-head (see Figure 19). At each cluster change time, once the cluster-heads are decided, the cluster-head broadcasts the following parameters:

- *Report Time (TR)*: This is the time period between successive reports sent by a node.
- *Attribute*s *(A)*: This is a set of physical parameters which the user is interested in obtaining data about.

At every report time, the cluster members sense the parameters specified in the attributes and send the data to the cluster-head. The cluster-head aggregates this data and sends it to the base station or a higher level cluster-head. This ensures that the user has a complete picture of the entire area covered by the network. Important features of this scheme are as below:

- Since the nodes switch off their sensors and transmitters at all times except the report times, the energy of the network is conserved.
- At every cluster change time, TR and A are transmitted afresh and so, can be changed. Thus, by changing A and TR, the user can decide what parameters to sense and how often to sense them. It is also possible that different clusters sense different attributes for different TR.

This scheme, however, has an important drawback. Because of the periodicity with which the data is sensed, it is possible that time critical data may reach the user only after the report time. Thus, this scheme may not be adequate for time-critical data sensing applications. In this text we will cover both proactive and reactive protocols, while highlighting that the protocol to be chosen is directly related to application requirements.

*6.3 Fundamentals of MAC Protocol for Wireless Sensor Networks*

Wireless medium is mostly a broadcast medium. All nodes within radio range of a node can hear its transmission. This can be used as a unicast medium by specifically addressing a particular node and all other nodes 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 nodes, the bandwidth is divided into N equal portions either in frequency (FDMA: frequency division multiple access), in time (TDMA: time division multiple access), in code (CDMA: code division multiple access), in space (SDMA: space division multiple access) or OFDM (orthogonal frequency division multiplexing). Since each node is assigned a private portion, there is no interference between multiple users. These protocols works very well with efficient allocation mechanisms, when there are only a small and fixed number of users, 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 users changes dynamically and data is bursty at arbitrary nodes, it is most advisable to use dynamic channel allocation scheme. These are contention-based schemes, where nodes contend for the channel when they have data while minimizing collisions with other nodes' transmissions. When there is a collision, the nodes are forced to retransmit data, thus leading to increased wastage of energy of the nodes and unbounded delay. Example protocols are: CSMA (persistent and non-persistent) [Tanenbaum 1996], MACAW [Bharghavan 1994], IEEE 802.11 [Crow 1997], etc.

As we will see shortly, in a hierarchical clustering model, once clusters have been formed, the number of nodes in the cluster is fixed and due to hierarchical clustering, the number of nodes per cluster is also not large. So, with such a scenario, it is better to use one of the static channel allocation schemes. Studies [Heinzelman 2000a, Intanagonwiwat 2000] have pointed out the uses of TDMA for wireless sensor networks. In this scheme all the nodes transmit data in their slot to the cluster head and at all other times the radio can be switched off thereby saving valuable energy. When it is not possible to use TDMA, the nodes can use non-persistent CSMA since the data packets are of fixed size.

TDMA is suitable for either type of networks. In proactive networks since 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 never reaches 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 some slots might be empty, it is better than the delay and energy consumption incurred due to dynamic channel allocation schemes.

CDMA is used to avoid inter cluster collisions. Though this means that more data needs to be transmitted per bit, it allows for multiple transmissions using the same frequency. A number of advantages have been pointed out for using TDMA/CDMA combination to avoid intra/inter cluster collision in ad hoc and sensor networks [Heinzelman 2000b].

### *6.4 Flat Routing in Sensor Networks*

Routing in wireless sensor networks is very different from the traditional wired or wireless networks. Sensor networks are data centric, requesting information satisfying certain attributes and thus do not require routing of data between specific nodes. Also since adjacent nodes have almost similar data and might almost always satisfy the same attributes, rather than sending data separately from each node to the requesting node, it is desirable to aggregate similar data in a certain region before sending it. This aggregation is also known as "data fusion" [Brooks 1998, Varshney 1997]. Many protocols have been proposed that collect data based on the queries injected by the user or which always collect data so that the network is ready to answer any query the user asks. These protocols are based on the same concept as ad hoc networks where a route is set up only when needed (on-demand/reactive routing) or have a route from each node to every other node so that when it is needed, it is immediately available (proactive).

We now look into protocols which collect data to answer queries injected by the user.

#### *6.4.1 Directed Diffusion*

Directed Diffusion [Intanagonwiwat 2000] is a data dissemination paradigm for sensor networks. It is a data-centric paradigm and is very useful to query dissemination and processing applications. The query is disseminated (flooded) throughout the network and gradients are setup to draw data satisfying the query towards the requesting node. Events (data) start flowing towards the requesting node from multiple paths. A small number of paths can be reinforced so as to prevent further flooding.

Such type of information retrieval is well suited only for persistent queries where requesting nodes are expecting data that satisfy a query for a duration of time. This makes it unsuitable for historical or one-time queries as it is not worth setting up gradients for queries which employ the path only once. Also this type of data collection does not fully exploit the feature of sensor networks, that adjacent nodes have similar data, as it uses a flat topology. At most, in this protocol data can be aggregated at the intermediate nodes.

### *6.4.2 Spin*

In [Heinzelman 1999], a family of adaptive protocols called SPIN (Sensor Protocols for Information via Negotiation) has been proposed that disseminates all the information at each node to every node in the network. This enables a user to query any node and get the required information immediately. These protocols make use of the property that near-by nodes have similar data and thus distribute only the data which other nodes do not have. These protocols work pro-actively and distribute the information all over the network, even when a user does not request any data.

### *6.4.3 Cougar*

Distributed Query processing results in several orders of magnitude times less message traffic and higher sensor lifetime than centralized query processing. In [Bonnet 2000, Bonnet 2001], it is discussed the application of distributed query execution techniques to improve communication efficiency in sensor and device networks. They discussed 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 (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 model for sensor database systems known as COUGAR has been proposed, with appropriate user representation and internal representation of queries. The representation of sensor queries is also considered so that it is easier to aggregate the data and to combine two or more queries. Routing of queries is not being handled. Cougar has a three-tier architecture:

- The Query Proxy: A small database component running on the sensor nodes to interpret and execute queries.
- A Front end Component: A powerful query-proxy that allows the sensor network to connect to the outside world. Each front-end includes a full-fledged database server.

- A Graphical User Interface (GUI): Through the GUI, users can pose ad hoc and long running queries on the sensor network. A map component allows the user to query by region and visualize the topology of sensors in the network.

Queries are formulated regardless of the physical structure or the organization of the sensor network. Sensor data is different from the traditional relational data since it is not stored in a database server and it varies over time. Aggregate queries or correlation queries that give a bird eye's view of the environment also zoom on a particular region of interest. Each long running query defines a persistent view which it maintains during a given time interval. In addition, a sensor database should account for sensor and communication failures. It should consider sensor data as measurements with an associated uncertainty, and not as facts; the abstract data type represents data from physical sensors through representation by continuous distribution, thus capturing the uncertainty hidden in the sensor measurement. Finally, it should be able to establish and run a distributed query execution plan without assuming global knowledge of the sensor network.

In summary, the protocols we have seen so far use a flat topology which is not suitable for some applications of wireless sensor networks since through this topology one can not aggregate data from a number of near-by nodes and does not take full advantage of the specific features in sensor nodes. There are a number of clustering algorithms in literature and we discuss some of them in the next section. It is always important to keep in mind that different algorithms, whether viewing the topology as flat or hierarchical, are best suitable for different application environments.

### *6.5 Hierarchical Routing in Sensor Networks*

Some authors suggest that a hierarchical clustering scheme is the most suitable for wireless sensor networks, as this model enables us to take advantage of all the features that are specific to sensor networks. The network is assumed to consist of a base station (*BS*), away from the nodes, through which the end user can access data from the sensor network. All the nodes in the network are homogeneous and begin with the same initial energy. The *BS* however has a constant power supply and so, has no energy constraints. Hence, it can be used to perform functions that are energy intensive. It can transmit with high power to all the nodes. Thus, there is no need for routing from the BS to any specific node. However, the nodes cannot always reply to the BS directly due to their power constraints, resulting in asymmetric communication. BS can also be used as a database to hold past data.

This model uses a hierarchical clustering scheme. Consider the partial network structure shown in Figure 19. Each cluster has a cluster head (CH) which collects data from its cluster members, aggregates it and sends it to the BS or an upper level cluster head. For example, nodes 1.1.1, 1.1.2, 1.1.3, 1.1.4, 1.1.5 and 1.1 form a cluster with node 1.1 as the cluster head. Similarly, there exist other cluster heads such as 1.2, etc. These cluster-heads, in turn, form a cluster with node 1 as their cluster-head. So, node 1 becomes a second level cluster head 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. The main features of such architecture are:

- All the nodes transmit only to their immediate cluster-head, thus saving energy.
- Only the cluster head needs to perform additional computations on the data such as aggregation etc. So, energy is again conserved.
- The cluster members of a cluster are mostly adjacent to each other and have similar data. Since the cluster-heads aggregate similar data, aggregation can said to be more effective
- Cluster-heads at increasing levels in the hierarchy need to transmit data over relatively longer distances. As they need to perform extra computations, they end up consuming energy faster than the other lower level nodes. In order to evenly distribute this consumption, all the nodes take turns, becoming the cluster head for a time interval T, called the cluster period.
- Now since only the cluster-heads need to know how to route the data towards its own cluster-head or BS, it reduces complexity in data routing.

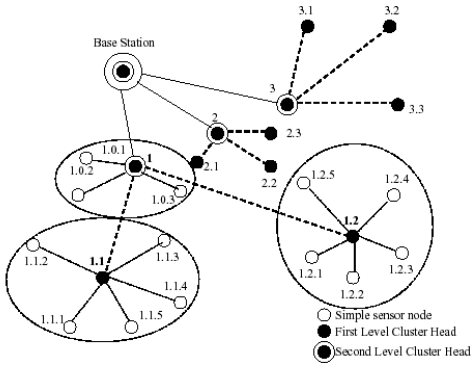


Figure 19 – Hierarchical Clustering

For applications which need to collect data for analysis of the situation/circumstances, it is adequate if we get data when the sensors are able to send it. But in applications that get data when something critical happens, such as “temperature going beyond 100ºF”, “more than 20

tanks passing by a region", etc., but do not really care what happens in the network at other times, it is not desirable to waste sensors' energy transmitting all the data they have collected. Ideally, it would be better if we could have flexibility in the network so that the user could decide how the network should behave based on the requirements and/or expectations.

### *6.5.1 Cluster Based Routing Protocol*

A cluster based routing protocol (CBRP) has been proposed in [Jiang 1998] for sensor networks. It divides the network nodes into a number of overlapping or disjoint two-hop-diameter clusters in a distributed manner. Here, the cluster members just send the data to the cluster head (CH) and the CH routes the data to the destination. But this protocol is not suitable for wireless sensor networks as, due to high mobility, it requires a lot of "hello" messages to maintain the clusters. The sensor nodes do not have as much mobility and 2-hop-diameter clusters are not adequate to exploit the underlying feature of "adjacent nodes have similar data" in sensor networks.

### *6.5.2 Scalable Coordination*

In [Estrin 1999], a hierarchical clustering method is discussed with emphasis on localized behavior and the need for asymmetric communication and energy conservation in sensor networks. In this method (no experimental results are provided) the cluster formation appears to require considerable amount of energy. Periodic advertisements are needed to form the hierarchy. Also, any changes in the network conditions or sensor energy level result in re-clustering which is not quite acceptable as some parameters tend to change dynamically.

### *6.5.3 Leach*

It is introduced in [Heinzelman 2000b] a hierarchical clustering algorithm for sensor networks, called LEACH (Low-Energy Adaptive Clustering Hierarchy). LEACH is actually a family of protocols [Heinzelman 2000b] which suggests two schemes, distributed and centralized, that have minimal setup time and are also very energy efficient. One important feature of LEACH is that it utilizes randomized rotation of local cluster base stations (cluster-heads) to evenly distribute the energy load among the sensors in the network. They also make use of TDMA/CDMA MAC to reduce inter-cluster and intra-cluster collisions. LEACH is a good approximation of a proactive network protocol, with some minor differences.

Once the clusters are formed, the cluster heads broadcast a TDMA schedule giving the order in which the cluster members can transmit their data. Every node in the cluster is assigned a slot in the frame, during which it transmits data to the cluster head. When the last node in the schedule has transmitted its data, the schedule is repeated.

The *report time* discussed earlier is equivalent to the *frame time* in LEACH. The *frame time* is not broadcasted by the cluster head, but is derived from the TDMA schedule. However, it is not under user control. Also, the attributes are predetermined and are not changed after initial installation. This network can be used to monitor machinery for fault detection and diagnosis. It can also be used to collect data about temperature (or pressure, moisture, etc.) change patterns over a particular area.

But data collection is centralized and done periodically. This can be said to be most appropriate only for constant monitoring of networks. The user mostly always does not need all that data (immediately). So, periodic data transmissions are unnecessary, which it is saying as though very limited energy is consumed drains the limited energy from the sensors. This approach is similar to the warehousing approach.

### *6.5.4 Threshold sensitive Energy Efficient Network (TEEN)*

In this section, we present the reactive network protocol called TEEN (Threshold sensitive Energy Efficient sensor Network protocol) [Manjeshwar 2001] with its time line depicted in Figure 20. In this scheme, at every cluster change time, in addition to the attributes, the cluster-head broadcasts the following to its members:

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

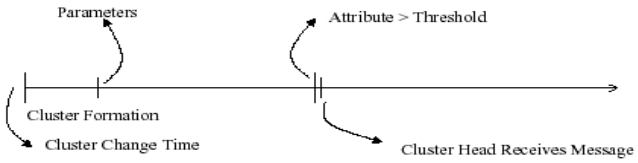


Figure 20 – Time line for TEEN

The nodes sense their environment continuously. The first time a parameter from the attribute set reaches its hard threshold value, the node switches on its transmitter and sends the sensed data. The sensed value is also stored in an internal variable in the node, called the

*sensed value* (SV). The nodes will next transmit data in the current cluster period, only when *both* the following conditions are true:

- The current value of the sensed attribute is greater than the hard threshold.
- The current value of the sensed attribute differs from SV by an amount equal to or greater than the soft threshold.

Whenever a node transmits data, SV is set equal to the current value of the sensed attribute. Thus, the hard threshold tries to reduce the number of transmissions by allowing the nodes to transmit only when the sensed attribute is in the range of interest. The soft threshold further reduces the number of transmissions by eliminating all the transmissions which might have otherwise occurred when there is little or no change in the sensed attribute once the hard threshold is reached. The main features of this scheme are as follows:

- Time critical data reaches the user almost instantaneously. So, this scheme is eminently suited for time-critical data sensing applications.
- Message transmission consumes much more energy than data sensing. So, even though the nodes sense continuously, the energy consumption in this scheme can potentially be much less than in the proactive network, because data transmission is done less frequently.
- The soft threshold can be varied, depending on the criticality of the sensed attribute and the target application.
- A smaller value of the soft threshold gives a more accurate picture of the network, at the expense of increased energy consumption. Thus, the user can control the trade-off between energy efficiency and accuracy.
- At every cluster change time, the parameters are broadcast afresh and so, the user can change them as required.

The main drawback of this scheme is that, if the thresholds are not reached, the nodes will never communicate, the user will not get any data from the network at all and will never be able to know even if all the nodes die. Thus, this scheme is not well suited for applications where the user needs to get data on a regular basis. Another possible problem with this scheme is that a practical implementation would have to ensure that there are no collisions in the cluster. TDMA scheduling of the nodes can be used to avoid this problem. This will, however, introduce a delay in reporting of time-critical data. CDMA is another possible solution to this problem. This protocol is best suited for time critical applications such as intrusion and explosion detection.

### *6.5.5 Adaptive Periodic Threshold-sensitive Energy Efficient Sensor Network Protocol*

There are applications in which the user wants time-critical data and also wants to query the network for analysis on conditions other than collecting time-critical data. In other words, the user might need a network that reacts immediately to time-critical situations and also gives an overall picture of the network at periodic intervals, so that it is able to answer analysis queries. Neither of the above networks can do both the jobs satisfactorily since they have their own limitations.

APTEEN (Adaptive Periodic Threshold-sensitive Energy Efficient Sensor Network Protocol) [Manjeshwar 2002] is able to combine the best features of proactive and reactive networks while minimizing their limitations to create a new type of network called a *Hybrid network*. In this network, the nodes not only send data periodically, they also respond to sudden changes in attribute values. This uses the same model as the above protocols with following changes. In APTEEN, once the cluster heads are decided the following events take place, in each cluster period. The cluster head first broadcasts the following parameters:

- *Attributes* (A): This is a set of physical parameters which the user is interested in obtaining data about.
- *Thresholds*: This parameter consists of a hard threshold (HT) and a soft threshold (ST). HT is a value of an attribute beyond which a node can be triggered to transmit data. ST is a small change in the value of an attribute which can trigger a node to transmit.
- *Schedule*: This is a TDMA schedule similar to the one used in [Heinzelman 2000b], assigning a slot to each node.
- *Count Time* (CT): It is the maximum time period between two successive reports sent by a node. It can be a multiple of the TDMA schedule length and it introduces the proactive component in the protocol.

The nodes sense their environment continuously. However, only those nodes which sense a data value at or beyond the hard threshold, transmit. Furthermore, once a node senses a value beyond HT, it next transmits data only when the value of that attribute changes by an amount equal to or greater than the soft threshold ST. The exception to this rule is that if a node does not send data for a time period equal to the count time, it is forced to sense and transmit the data, irrespective of the sensed value of the attribute. Since nodes near to each other may fall in the same cluster and sense similar data, they may try sending their data simultaneously, leading to collisions between their messages. Hence, a TDMA schedule is

used and each node in the cluster is assigned a transmission slot, as shown in Figure 21. In the sections to follow, we will refer to data values exceeding the threshold value as critical data.

The main features of this scheme are as follows:

- It combines both proactive and reactive policies. By sending periodic data, it gives the user a complete picture of the network, like a proactive scheme. It also senses data continuously and responds immediately to drastic changes, thus making it responsive to time critical situations. It, thus, behaves as a reactive network also.
- It offers a lot of flexibility by allowing the user to set the time interval (CT) and the threshold values for the attributes.
- Changing the count time as well as the threshold values can control energy consumption.
- The hybrid network can emulate a proactive network or a reactive network, based on the application, by suitably setting the count time and the threshold values.

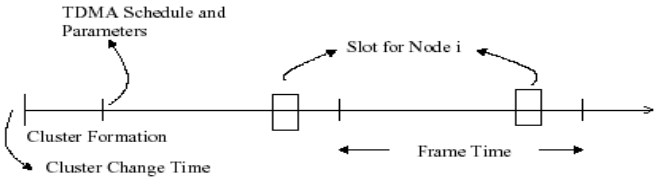


Figure 21 – Time line for APTEEN

The main drawback of this scheme is the additional complexity required to implement the threshold functions and the count time. However, this might be seen as a trade-off.

### *6.6 Comparison Table*

Table 3 illustrates the characteristics of hierarchical and flat topologies for WSN.

Table 3 – Hierarchical versus Flat topologies for WSN

| Hierarchical | Flat |
|---|---|
| Reservation-based scheduling | Contention-based scheduling |
| Collisions avoided | Collision overhead present |
| Reduced duty cycle due to periodic sleeping | Variable duty cycle by controlling sleep time of nodes |
| Data aggregation by cluster head | Node on multi-hop path aggregates incoming data from neighbors |
| Simple but non-optimal routing | Routing is complex but optimal |
| Requires global and local synchronization | Links formed in the fly, without synchronization |
| Overhead of cluster formation throughout the network | Routes formed only in regions that have data for transmission |
| Lower latency as multi-hop network formed by cluster-heads is always available | Latency in waking up intermediate nodes and setting up the multi-hop path |
| Energy dissipation is uniform | Energy dissipation depends on traffic patterns |
| Energy dissipation cannot be controlled | Energy dissipation adapts to traffic pattern |
| Fair channel allocation | Fairness not guaranteed |

*6.7 Adapting to the Inherent Dynamic Nature of Wireless Sensor Networks*

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

- Exploit spatial diversity and density of sensor/actuator nodes to, for example, build an adaptive node sleep schedule. Characterize the relationship between deployment density and network size. Explore of the trade-off between data redundancy and bandwidth consumption.
- The nodes on deployment should 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 traffic changes. Some nodes may detect an event which triggers a big sensor, like a camera, generating heavy traffic. But when sensing activity is low, traffic is light.
- Allowing finer control over an algorithm than simply turning off or on. Nodes should be capable of dynamically trading precision for energy or scope for convergence time based on incoming data.

The SCADDS Project (Scalable Coordination Architectures for Deeply Distributed Systems) [SCADDS], also a part of DARPA SensIT program, 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 listen schedules; there is no clustering and no inter-cluster communications and interference.

Adaptive Self-Configuring sEnsor Network Topologies (Ascent) [Cerpa 2001], 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. In Ascent each node assesses its connectivity and adapts its participation in its multi-hop network topology based on the measured operating region. 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 rely 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. It avoids transmitting dynamic state information repeatedly across the network.

*6.8 MAC Layer Design Issues in Wireless Sensor Networks*

As with MAC protocols for traditional mobile ad hoc networks, sensor networks have their own issues that must be addressed. Below we will discuss some of the most important issues involved in the design of MAC protocols for wireless sensor networks.

### *6.8.1 Fighting Node Failure*

When many nodes have failed, the MAC and routing protocols must accommodate formation of new links and routes to the sink nodes. 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 left.

### *6.8.2 Sources of Resource Consumption at the MAC Layer*

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

- Collisions – When a transmitted packet is corrupted it has to be discarded. The follow-on retransmissions increase energy consumption and increase latency.
- Overhearing – Nodes listen to transmissions that are destined to other nodes.
- Control packets overhead – Sending and receiving control packets consumes energy, and less useful data packets can be transmitted. This overhead increases linearly with node density. Moreover, as more nodes fail in the network, more control messages are required to self configure the system resulting in more energy consumption.
- Idle Listening – Listening to receive possible traffic that is not sent. This is especially true in many sensor network applications. If nothing is sensed, nodes are in idle mode for most of the time.

### *6.8.3 Measures to Reduce Energy Consumption*

One of the most cited methods to conserve energy in sensor networks is by avoiding to listen to idle channels, that is, neighboring nodes periodically sleep (radio off) auto synchronizing on sleep schedules. It is important to note that in wireless sensor networks fairness, latency, throughput and bandwidth utilization are secondary.

S-MAC (Sensor-MAC) [Ye 2002] is new MAC protocol specifically designed for wireless sensor networks. The main goal of the S-MAC protocol design is to reduce energy consumption, while supporting good scalability and collision avoidance. It tries to reduce

energy consumption from almost all the sources that we have identified to cause energy waste, *i.e.*, idle listening, collision, overhearing and control overhead. S-MAC consists of three major components: periodic listen and sleep, collision and overhearing avoidance, and message passing. S-MAC assumes that nodes are able to turn their radios off and on, and tune carrier frequency to a large number of available bands. It is a distributed protocol with flat topology that enables collection of nodes to discover their neighbors and establish transmission or reception schedules for communicating with them, without the need for any local or global master nodes. Links are formed on fly because non-synchronous slots are assigned. This concept is known as Non-synchronous scheduled communication (after link establishment each node knows ahead of time when to turn its transceiver on). This is to quickly retrieve information "trapped" in the low-duty cycle network as getting information to its ultimate destination in a timely manner is difficult when routes are blocked, nodes are turned off, and large fractions of the network are not available for long periods.

### *6.8.4 Comparison of Scheduling & Reservation based and Contention based MAC Design*

One approach of MAC design for sensor networks is based on reservation and scheduling, for example TDMA based protocols that conserve more energy compared to contention based protocols like the IEEE 80211 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 its key challenges.

## 7. Standard Activities

### *7.1 Internet Engineering Task Force (IETF) Activities*

The MANET Working Group [MANET] is a chartered working group established within the Internet Engineering Task Force (IETF) [IETF] to investigate and develop candidate standard Internet routing support for mobile, wireless IP autonomous segments. The primary focus of the working group is to develop and evolve MANET routing specifications and introduce them to the Internet Standards track. The goal is to support networks scaling up to hundreds of routers. If this proves successful, future work may include development of other protocols to support additional routing functionality. The working group also examines related security issues around a MANET.

Along with the MANET working group, the IETF has also established the mobileip (IP Routing for Wireless/Mobile Hosts) Working Group (WG) [MOBILEIP]. This WG has developed routing support to permit IP nodes (hosts and routers) using either IPv4 or IPv6 to seamlessly "roam" among IP subnetworks. The Mobile IP method supports transparency above the IP layer, including the maintenance of active TCP connections and UDP port bindings. Wherever this level of transparency is not required, solutions such as DHCP and dynamic DNS updates may be adequate and techniques such as Mobile IP not needed.

Future work is expected to focus on deployment issues in Mobile IP and provide appropriate protocol solutions to address known deficiencies and shortcomings. For example, the wireless/cellular industry is considering using Mobile IP as one technique for IP mobility for wireless data. The working group will endeavor to gain an understanding of data service in cellular systems such as GPRS, UMTS, CDMA2000, and interact with other standards bodies that are trying to adopt and deploy Mobile IP WG protocols in these context.

### *7.2 Bluetooth and Wireless PANs*

In the past quarter century we have seen the rollout of three generations of wireless cellular systems attracting end-users by providing efficient mobile communications. On another front, wireless technology became an important component in providing networking infrastructure for localized data delivery. This later revolution was made possible by the induction of new networking technologies and paradigms, such as wireless local area networks (WLAN) and wireless personal area networks (WPAN).

Wireless personal area networks (WPANs) are short to very short-range (from a couple centimeters to a couple of meters) wireless networks that can be used to exchange information between devices in the reach of a person. WPANs can be used to replace cables between computers and their peripherals, to establish communities helping people do their everyday chores making them more productive, or to establish location aware services. Wireless local area networks (WLANs) on the other hand provide with a larger transmission range. Although WLAN equipment usually carries the capability to be set up for ad hoc networking, the premier choice of deployment is yet a cellular like infrastructure mode to interface wireless users with the Internet. The best example representing WPANs is the recent industry standard: Bluetooth [Bluetooth], other examples include Spike [Spike], and in the broad sense HomeRF [Negus 2000]. For WLANs, the most well known representatives are based on the standards IEEE 802.11 [Crow 1997] and HiperLAN [HiperLAN 1995] with all their variations.

The IEEE 802 committee has also realized the importance of short-range wireless networking and initiated the establishment of the IEEE 802.15 WG for WPANs [WPAN] to standardize protocols and interfaces for wireless personal area networking. Altogether, the 802.15 working group is formed by four Task Groups (TG):

- IEEE 802.15 WPAN/Bluetooth TG 1 – The TG 1 was established to support applications which require medium-rate WPANs (such as Bluetooth). These WPANs will handle a variety of tasks ranging from cell phones to PDA communications and have a QoS suitable for voice applications.
- IEEE 802.15 Coexistence TG 2 – Several wireless standards, such as Bluetooth and IEEE 802.11b, and appliances, such as microwaves operate, in the unlicensed 2.4 GHz ISM (Industrial-Scientific-Medical) frequency band. The TG 2 is developing specifications on the ISM band due to the unlicensed nature and available bandwidth. Thus, the IEEE 802.15 Coexistence TG 2 (802.15.2) for Wireless Personal Area Networks is developing Recommended Practices to facilitate coexistence of Wireless Personal Area Networks (802.15) and Wireless Local Area Networks (802.11).
- IEEE 802.15 WPAN/High Rate TG 3 – The TG3 for WPANs is chartered to draft and publish a new standard for high-rate (20Mbit/s or greater) WPANs. Besides a high data rate, the new standard will provide for low power, low cost solutions addressing the needs of portable consumer digital imaging and multimedia applications.
- IEEE 802.15 WPAN/Low Rate TG 4 – The goal of the TG 4 is to provide a standard having ultra-low complexity, cost, and power for a low-data-rate (200Kb/s or less) wireless connectivity among inexpensive fixed, portable, and moving devices. Location awareness is being considered as a unique capability of the standard. The scope of the TG 4 is to define the physical and media access control (MAC) layer specifications. Potential applications are sensors, interactive toys, smart badges, remote controls, and home automation.

One key issue in the feasibility of WPANs is the inter-working of wireless technologies to create heterogeneous wireless networks. For instance, WPANs and WLANs will enable an extension of the third generation (3G) cellular networks (i.e., UMTS and cdma2000) into devices without direct cellular access. Moreover, devices interconnected in a WPAN may be able to utilize a combination of 3G access and WLAN access by selecting the access that is best for the moment. In such networks 3G, WLAN and WPAN technologies do not compete against each other but enable the user to select the best connectivity for his/her purposes.

Figure 22 clearly shows the operating space of the various 802 wireless standards and activities still in progress.

Given the importance within the WPAN operating space, intensive research activities, and availability of devices, we will now devote a little time in, first, giving a brief introduction on Bluetooth and then provide an overview of the Bluetooth standard as defined by the Bluetooth SIG (Special Interest Group) [Bluetooth].

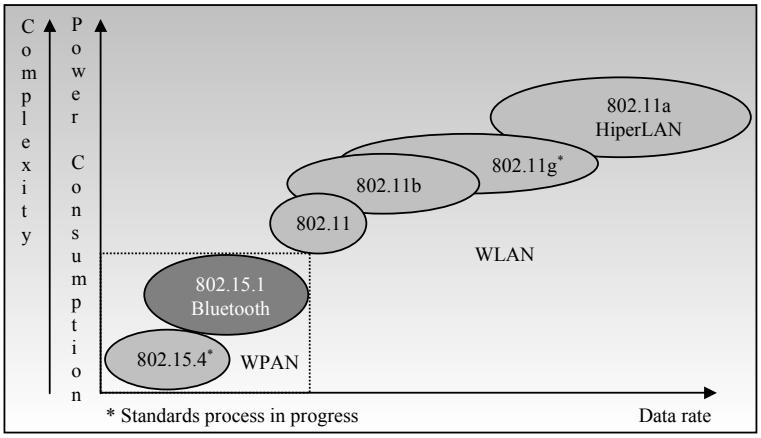



Figure 22 – The scope of the various WLAN and WPAN standards

*7.2.1 Brief History and Applications of Bluetooth*

In the context of ad hoc wireless networks, the Bluetooth technology came to light in May 1998, and since then the Bluetooth SIG has steered the development of the technology through the development of an open industry specification, including both protocols and applications scenarios. It is predicted that in 2006 Bluetooth will be present in 73 percent of phones and 44 percent of PDAs. It will provide device-to-device communication, enabling seamless communication between phones, printers, PDAs and scanners in the office and between phones, smart home control units, TVs and VCRs in the home. The Bluetooth specification comprises of an end-to-end description for both protocols and application profiles that guarantee value-added to its users right out-of-the-box. As per the current specification (version 1.1), it consists of the following two parts:

- The core specification defining the radio characteristics and the communication protocols for exchanging data between devices and Bluetooth radio links.
- The profile specification that defines how the Bluetooth protocols are to be used to realize a number of selected applications.

Bluetooth has a tremendous potential in moving and synchronizing information in a localized setting. Potential for Bluetooth applications is huge, because we do business

transactions and communicate more frequently with the people who are close by as compared to those who are far away - a natural phenomenon of human interaction.

### *7.2.2 An Overview of the Bluetooth Wireless Technology*

While we give here only an overview of Bluetooth, its system, architecture and protocols are defined in detail in [Bluetooth]. Bluetooth operates in the ISM frequency band starting at 2.402 GHz and ending at 2.483 GHz in USA and most countries of Europe. A total of 79 RF channels of 1 MHz width are defined, where the raw data rate is 1 Mbit/s. A Time Division Multiplexing (TDD) technique divides the channel into 625μs slots and, with a 1Mbit/s symbol rate, a slot can carry up to 625 bits. Transmission in Bluetooth occurs in packets that occupy 1, 3 or 5 slots. Each packet is transmitted on a different hop frequency with a maximum frequency hopping rate of 1600 hops/s.

Bluetooth operates on a Master-Slave concept whereby the Master device controls data transmissions through a polling scheme. The Master periodically polls the Slave devices for information and only after receiving such a poll is a Slave allowed to transmit. A Master device can directly control seven active Slave devices in what is defined as a piconet. Multiple piconets can be linked together through common Bluetooth devices to form a scatternet. Figure 23 illustrates a scatternet composed of four piconets, where each piconet has several slaves (indicated by the letter $S_{i,j}$) and one master (indicated by the letter $M_i$).

Figure 24 depicts the Bluetooth protocol stack, which also shows the application "layer" where the profiles would reside. The protocols that belong to the core specification are:

- The Radio – The radio layer, which resides below the Baseband layer, defines the technical characteristics of the Bluetooth radios. The Bluetooth radios come in three power classes, depending on their transmit power. Class 1 radios have transmit power of 20 dBm (100mW); class 2 radios have transmit power of 4 dBm (2.5mW); class 3 radios have transmit power of only 0 dBm (1mW).
- The Baseband – The baseband defines the key procedures that enable devices to communicate with each other using the Bluetooth wireless technology. The baseband defines the Bluetooth piconets (see Figure 24) and how they are created, and the Bluetooth links. It also defines the low-level packet types.
- The Link Manager Protocol (LMP) – The LMP is a transactional protocol between two link management entities in communicating Bluetooth devices whose responsibilities is to setup the properties of the link.

- The Logical Link Control & Adaptation Protocol (L2CAP) – The L2CAP layer shields the specifics of the Bluetooth lower layers and provides a packet interface to higher layers. At L2CAP, the concepts of master and slave devices do not exist anymore.

The Bluetooth specification defines two distinct types of links for the support of voice and data applications, namely, SCO (*Synchronous connection-oriented*) and ACL (*Asynchronous connectionless*). The first link type supports point to point voice switched circuits while the latter supports symmetric as well as asymmetric data transmission. ACL packets are intended to support data applications and do not have prescribed time slot allocations as opposed to SCO packets, which support periodic audio transmission at 64Kb/s in each direction.

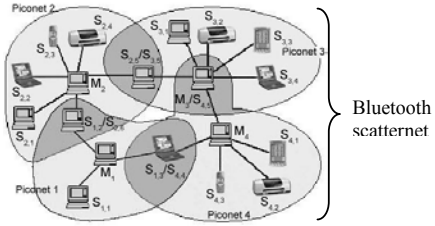



Figure 23 – Four piconets forming a scatternet

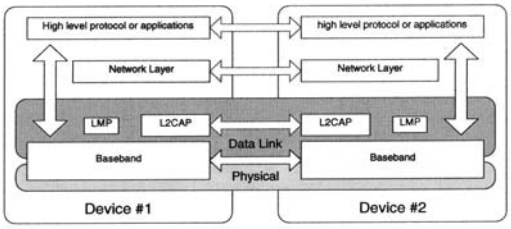



Figure 24 – Bluetooth protocol architecture

## 8. Open Problems

As we have already mentioned, the research in the area of Mobile Ad hoc Networking is far from being exhaustive. Much of the effort so far has been on devising routing protocols to support the effective and efficient communication between nodes that are part of the network. However, there are still many topics that deserve further investigation such as:

- Scalability – To what extent can an ad hoc network grow?
- Address configuration – The address scheme used in wired networks (e.g., DHCP), as well as in Mobile IP, might not be adequate in a MANET. A new addressing approach may be required for MANETs.
- Interoperation with the Internet – How can ad hoc networks seamlessly and efficiently access the Internet in order to obtain advanced services?
- Improvement of interaction between layers – Would it be better to interact layers in order to achieve better performance?
- Quality of service (QoS) – Is it feasible for bandwidth/delay-constrained applications to run well in a MANET?
- Applications for MANET – Have we found a killer application?
- Security – How can the network secure itself from malicious or compromised hosts?
- Power control – How can battery life be maximized?

The research community is already investigating some answers to these questions, however there is still a lot more work to be done.

**References**


[Aggelou 1999] G. Aggelou and R. Tafazolli, "RDMAR: A bandwidth-efficient routing protocol for mobile ad hoc networks," in *ACM International Workshop on Wireless Mobile Multimedia (WoWMoM)*, August 1999.

[Bae 2000] S.H. Bae, S.-J. Lee, W. Su, and M. Gerla, "The design, implementation, and performance evaluation of the on-demand multicast routing protocol in multihop wireless networks," IEEE Network, vol. 14, January 2000, 70-77.

[Basagni 1998] S. Basagni, I. Chlamtac, V.R. Syrotiuk, and B.A. Woodward, "A distance routing effect algorithm for mobility (DREAM)," in ACM/IEEE International Conference on Mobile Computing and Networking, October 1998, 76-84.

[Bharghavan 1994] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A media access protocol for wireless LANs," in *ACM SIGCOMM*, August 1994, 212-225.

[Bluetooth] Bluetooth SIG, http://www.bluetooth.com/.

[Bommaiah 1998] E. Bommaiah, M. Liu, A. McAuley, and R. Talpade, "AMRoute: Adhoc Multicast Routing Protocol," *Internet-Draft*, August 1998.

[Bonnet 2000] P. Bonnet, J. Gehrke, and P. Seshadri, "Querying the Physical World," In *IEEE Personal Communications,* October 2000.

[Bonnet 2001] P. Bonnet, J. Gehrke, and P. Seshadri, "Towards Sensor Database Systems," In *2nd Int. Conference on Mobile Data Management*, January 2001.

[Brooks 1998] R. Brooks and S. Iyengar, "Multi-Sensor Fusion" Prentice Hall PTR, 1998.

[Chandran 2001] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, "A feedback-based scheme for improving TCP performance in ad hoc wireless networks," In *IEEE Personal Communications Magazine*, vol. 8, no. 1, pp. 34–39, February 2001.

[Chiang 1998] C.-C. Chiang, M. Gerla, and L. Zhang, "Forwarding Group Multicast Protocol (FGMP) for Multihop, Mobile Wireless Networks," Baltzer Cluster Computing, special Issue on Mobile Computing, vol. 1, no. 2, 1998, 187-196.

[Cerpa 2001] A. Cerpa and D. Estrin, "Adaptive Self-Configuring sEnsor Networks Topologies," *UCLA CS Department Tech. Report UCLA/CSD-TR-01-0009,* May 2001.

[Chlamtac 1986] L. Chlamtac and A. Lerner, "Link allocation in mobile radio networks with noisy channel," In *Proc. of the IEEE INFOCOM*, April 1986.

[Corson 1996] M.S. Corson, J. Macker, and S. Batsell, "Architectural Considerations for Mobile Mesh Networking," In *Proceedings of the IEEE MILCOM*, October 1996.

[Corson 1997] M.S. Corson and V.D. Park, "An Internet MANET Encapsulation Protocol (IMEP) Specification," *Internet-Draft*, Nov. 1997.

[Crow 1997] B.P. Crow, I. Wadjaja, J.G. Kim, and P.T. Sakai, "IEEE 802.11 Wireless Local Area Networks," In *IEEE Communications Magazine*, September 1997, 116-126.

[Dube 1997] R. Dube, "Signal stability based adaptive routing for ad hoc mobile networks," In *Proc. of IEEE Personal Communications*, February 1997, 36-45.

[Duggirala 2000] R. Duggirala, "A Novel Route Maitenence Techinque for Ad Hoc Routing Protocols," *M.S. Thesis*, University of Cincinnati, November 2000.

[Estrin 1999] D. Estrin et al, "New Century Challenges: Scalable Coordination in Sensor Networks," *ACM Mobicom*, 1999.

[Garcia-Luna-Aceves 1999a] J.J. Garcia-Luna-Aceves and E.L. Madruga, "The Core-Assisted Mesh Protocol," *IEEE Journal on Selected Areas in Comm.*, August 1999, 1380-1394.

[Garcia-Luna-Aceves 1999b] J. J. Garcia-Luna-Aceves, "Reversing the collision-avoidance handshake in wireless networks," in *ACM MOBICOM*, August 1999, 120-131.

[Haarsten 1998] J. Haarsten, "Bluetooth – The Universal Radio Interface for Ad Hoc Wireless Connectivity," *Ericsson Review (3)*, 1998.

[Haas 1998] Z. Haas et al, "The performance of query control schemes for the zone routing protocol," in *ACM SIGCOMM*, 1998.

[Heinzelman 1999] W. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks," In *ACM/IEEE Mobicom,* August 1999.

[Heinzelman 2000a] W.B. Heinzelman, "Application-Specific Protocol Architectures for Wireless Networks," PhD Thesis, Massachusetts Institute of Technology, June 2000.

[Heinzelman 2000b] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *Proceedings of the 33rd Hawaii International Conference on System Sciences*, January 2000.

[HiperLAN 1995] ETSI, "Hiperlan Functional Specification," ETSI Draft Standard, July 1995.

[Holland 1999] G. Holland and N. H. Vaidya, "Analysis of TCP Performance over Mobile Ad Hoc Networks", in *ACM MOBICOM*, August 1999.

[Holland 2001] G. Holland, N.H. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for wireless multi-hop networks," in *ACM MOBICOM*, July 2001.

[IETF] Internet Engineering Task Force (IETF), http://www.ietf.org/.

[Intanagonwiwat 2000] C. Intanagonwiwat, R. Govindan, and D. Estrin. "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," In *ACM/IEEE* MOBICOM, August 2000, 56-67.

[Iwata 1999] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen, "Scalable Routing Strategies for Ad Hoc Wireless Networks," *IEEE Journal on Selected Areas of Communications*, August 1999, 1369-1379.

[Jiang 1998] M. Jiang, J. Li and Y. Tay, "Cluster Based Routing Protocol (CBRP) Functional Specification," Internet Draft, 1998.

[Jin 2000] K.T. Jin and D.H. Cho, "A MAC Algorithm for Energy-limited Ad Hoc Networks," In *Proceedings of Fall VTC 2000*, September 2000, 219-222.

[Johnson 1996] D. Johnson and D. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing*, Kulwer Academic, 1996, 153-181.

[Jubin 1987] J. Jubin and T. Truong, "Distributed Algorithm for Efficient and Interference-free Broadcasting in Radio Networks," in Proceedings of INFOCOM, January 1987, 21-32.

[Kahn 1999] J.M. Kahn, "New Century Challenges: Mobile Networking for Smart Dust," *ACM Mobicom*, 1999.

[Karn 1990] P. Karn, "MACA - A new channel access method for packet radio," In *Proc. of ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, September 1990.

[Ko 1998] Y.-B. Ko and N.H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," in *ACM MOBICOM*, November 1998.

[Lee 2000a] S.H. Lee and D.H. Cho, "A new adaptive routing scheme based on the traffic characteristics in mobile ad hoc networks," In *Proc. of Fall VTC 2000*, September 2000, 2911-2914.

[Lee 2000b] S.-J. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagrodia, "A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols," In *Proc. of IEEE INFOCOM 2000,* March 2000, 565-574.

[Liu 2001] J. Liu and S. Singh, "ATCP: TCP for mobile ad hoc networks," In *IEEE J-SAC*, vol. 19, no. 7, pp. 1300–1315, July 2001.

[MANET] IETF MANET Working Group, http://www.ietf.org/html.charters/manet-charter.html.

[Manjeshwar 2001] A. Manjeshwar and D.P. Agrawal, "TEEN: A protocol for Enhanced Efficiency in Wireless Sensor Networks," *Proceedings of the 1st Int. Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, April 2001.

[Manjeshwar 2002] A. Manjeshwar and D. P. Agrawal, "APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks," *Proceedings of the 2nd Int. Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, April 2002.

[MOBILEIP] IETF MOBILEIP Working Group, http://www.ietf.org/html.charters/mobileip-charter.html.

[Murthy 1996] S. Murthy and J.J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks," In *ACM Mobile Networks and Applications Journal*, October 1996, 183-197.

[Negus 2000] K.J. Negus, A.P. Stephens, and J.Lansford, "HomeRF: Wireless networking for the connected home", *in the IEEE Personal Communications*, February 2000, 20-27.

[Ozugur 1998] T. Ozugur, M. Naghshineh, P. Kermani, C.M. Olsen, B. Rezvani, and J.A. Copeland, "Balanced media access methods for wireless networks," in *ACM MOBICOM*, October 1998.

[Pei 2000] G. Pei, M. Gerla, and X. Hong, "Lanmar: Landmark routing for large scale wireless ad hoc networks with group mobility," In *ACM MobiHoc*, August 2000.

[Park 1997] V.D. Park and M.S. Corson, "A highly adaptive distributed routing algorithm for mobile and wireless networks," In *Proceeding of IEEE INFOCOM,* April 1997, 103-112.

[Perkins 1994] C.E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," In *Computer Comm. Review*, October 1994, 234-244.

[Perkins 1999] C.E. Perkins and E. Royer, "Ad hoc on-demand distance vector routing," *IEEE Workshop on Mobile Computing Systems and Applications*, February 1999, 90-100.

[Perkins 2001] C.E. Perkins, Ad Hoc Networking, Addison-Wesley, ISBN: 0201309769, 2001.

[Prakash 1999] R. Prakash, "Unidirectional Links Prove Costly in Wireless Ad Hoc Networks," In *Proceedings of the Third International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, August 1999, 15-22.

[Royer 1999] E.M. Royer and C.E. Perkins, "Multicast operation of the ad-hoc on-demand distance vector routing protocol," in *ACM MOBICOM*, August 1999, 207-218.

[Royer 2000] E.M. Royer, S-J. Lee, and C.E. Perkins, "The Effects of MAC Protocols on Ad hoc Communication Protocols," In *Proceedings of IEEE WCNC 2000*, September 2000.

[SCADDS] SCADDS Project, http://www.isi.edu/scadds/.

[SensIT] DARPA SensIT Program, http://www.darpa.mil/ito/research/sensit.

[Singh 1998] S. Singh, M. Woo, and C.S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks," In *Proceedings of Mobihoc*, 1998, 181-190.

[Spike] Spike, http://www.spikebroadband.net/.

[Tanenbaum 1996] Andrew Tanenbaum, "Computer Networks," Prentice Hall PTR, 1996.

[Toh 1997] C.-K. Toh, "Associativity based routing for ad hoc mobile networks," Wireless Personal Communications, March 1997.

[Tsuchiya 1988] P.F. Tsuchiya, "The Landmark Hierarchy: a new hierarchy for routing in very large networks," In *Computer Communication Review*, vol.18, no.4, Aug. 1988, 35-42.

[Vaidya 2000] N.H. Vaidya, P. Bahl, and S. Gupta, "Distributed fair scheduling in a wireless LAN," in *ACM MOBICOM*, August 2000.

[Varshney 1997] P. Varshney, "Distributed Detection and Data Fusion," Springer-Verlag, 1997.

[WPAN] IEEE 802.15 Working Group for WPANs, http://grouper.ieee.org/groups/802/15/.

[Wu 1998] C.W. Wu, Y.C. Tay, and C.-K. Toh, "Ad hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS) Functional Specification," *Internet-Draft*, November 1998.

[Ye 2002] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," In *INFOCOM 2002*, June 2002.