Multicasting Techniques in Mobile Ad Hoc Networks

Xiao Chen
Computer Science Department
Southwest Texas State University
San Marcos, TX 78666

Jie Wu
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431

1 Introduction

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

In the wired environment, there are two popular network multicast schemes, namely, the shortest path multicast tree and core-based tree. The shortest path multicast tree guarantees shortest path to each destination. But each source needs to build a tree. Therefore, there exist too many trees in the network. The core-based tree cannot guarantee the shortest path from a source to a destination. But only one tree is constructed for each group. Therefore, the number of trees is greatly reduced.

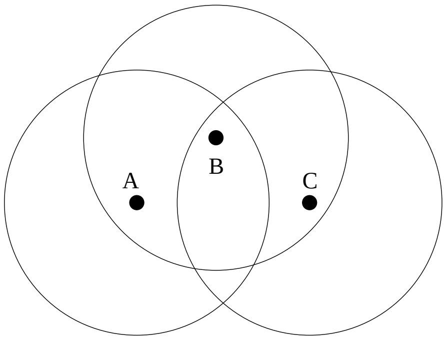

Figure 1: An example of a mobile ad hoc network.

Currently, one particularly challenging environment for multicast is a *mobile ad hoc network* (MANET). A MANET consists of a dynamic collection of nodes with sometimes rapidly changing multihop topologies that are composed of relatively low-bandwidth wireless links. There is no assumption of an underlying fixed infrastructure. Nodes are free to move arbitrarily. Since each node has a limited *transmission range*, not all packets may reach all the intended hosts. To provide communication through the whole network, a source-to-destination path could pass through several intermediate neighbor nodes. For example, two nodes can communicate directly with each other only if they are within each other's transmission range. Otherwise, the communication between them has to rely on other nodes. In the mobile ad hoc network shown in Figure 1, nodes A and B are within each other's transmission range (indicated by the circle around A and B respectively). If A needs to send a packet to B, it can send it directly. A and C are not within each other's range. If A wants to send a packet to C, it has to first forward the packet to B and then use B to route the packet to C.

Unlike typical wired routing protocols, routing protocols for mobile ad hoc networks must address a diverse range of issues. In general, the main characteristics of mobile computing are low bandwidth, mobility, and low power. Wireless networks deliver lower bandwidth than wired networks, and hence, the information collection during the formation of a routing table is expensive. Mobility of hosts, which causes topological changes of the underlying network, also increases the volatility of network information. In addition, the limitation of power leads users to disconnect mobile units frequently in order to save power consumption.

The goal of MANETs is to extend mobility into the realm of autonomous, mobile, wireless domains, where a set of nodes form the network routing infrastructure in an ad hoc fashion. The majority of applications for the MANET technology are in areas where rapid deployment and dynamic reconfiguration are necessary and the wired network is not available. These include military battlefields, emergency search and rescue sites, classrooms, and conventions where participants share information dynamically using their mobile devices. These applications lend themselves well to multicast operations. In addition, within a wireless medium, it is even more crucial to reduce the transmission overhead and power consumption. Multicasting can improve the efficiency of the wireless link when sending multiple copies of messages by exploiting the inherent broadcast property of wireless transmission. However, besides the issues for any ad hoc routing protocol listed above, wireless mobile multicasting faces several key challenges. Multicast group members move, thus precluding the use of a fixed multicast topology. Transient loops may form during multicast tree reconfiguration, so tree reconfiguration schemes should be simple to keep channel overhead low.

In mobile ad hoc networks, there are three basic categories of multicast algorithms. A naive approach is to simply flood the network. Every node receiving a message floods it to a list of neighbors. Flooding a network acts like a chain reaction that can result in exponential growth. The proactive approach precomputes paths to all possible destinations and stores this information in routing tables. To maintain an up-to-date database, routing information is periodically distributed throughout the network. The final approach is to create paths to other hosts on demand. The idea is based on a query-response mechanism or reactive multicast. In the query phase, a node explores the environment. Once the query reaches the destination, the response phase starts and establishes the path.

The rest of this chapter is organized as follows: in the next section, we review two multicast routing protocols: shortest path multicast tree and core-based tree methods, that are widely used in wired networks. In Section 3, we describe four extensions in mobile ad hoc networks: two distinct on-demand multicast protocols, forwarding group multicast protocol (FGMP) and core-assisted mesh protocol. Other multicast protocols used in mobile ad hoc networks are briefly summarized in Section 4. Section 5 discusses two related issues: QoS multicast and reliable multicast. The chapter concludes in Section 6.

2 Multicast Protocols in Wired Networks

In this section, we review two multicast protocols in wired networks, namely, the shortest path multicast tree protocol and the core-based tree multicast protocol. To facilitate the discussion, in the figures in the chapter, we use black nodes to represent group members, sources and destinations. Gray nodes for forwarding nodes and white for non-group members.

2.1 Shortest Path Multicast Tree

The single shortest path multicast tree can be constructed by applying Dijkstra's spanning tree algorithm [12]. Each path from the root of the tree to a destination is a shortest path.

In this protocol, to do multicast routing, each router computes a spanning tree covering all other routers in the network. For example, in Figure 2 (a) we have a network with two groups, 1 and 2. Some routers are attached to hosts that belong to one or both of these groups, as indicated in the figure. A spanning tree for node S is shown in Figure 2 (b).

When a process sends a multicast packet to a group, the first router examines its spanning tree and prunes it, removing all lines that do not lead to hosts that are members of the group. In our example, Figure 2 (c) shows the pruned spanning tree for group 1. Similarly, Figure 2 (d) shows the pruned spanning tree for group 2. Multicast packets are forwarded only along the appropriate spanning tree.

One potential disadvantage of this algorithm is that it scales poorly to large networks. Suppose that a network has n groups, each with an average of m members. For each group, m pruned spanning tree must be stored, for a total of mn trees. When many large groups exist, considerable storage is needed to store all the trees.

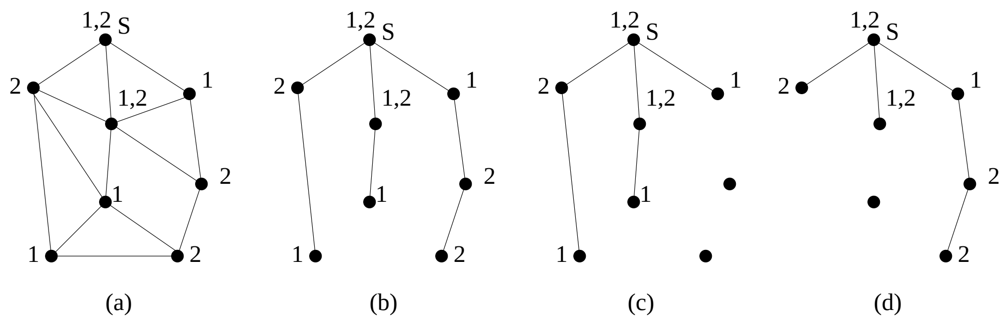

Figure 2: (a) A network. (b) A spanning tree for node S. (c) A multicast tree for group 1. (d) A multicast tree for group 2.

An alternative design uses *core-based trees* [3], which is introduced in the following subsection. Here, a single spanning tree per group is computed, with the root (the core) near the middle of the group. To send a multicast message, a host sends it to the core, which then does the multicast along the spanning tree. Although this tree will not be optimal for all sources, the reduction in storage costs from m trees to one tree per group is a major saving.

2.2 Core-Based Trees Multicast Protocol

A core-based tree involves having a single node (a router), known as the *core* of the tree, from which branches emanate. These branches are made up of other nodes, so-called *non-core* nodes, which form a shortest path between a member-host's directly attached node, and the core. A node at the end of a branch shall be known as a *leaf* node on the tree. The core need not be topologically centered between the nodes on the tree, since multicasts vary in nature, and so can the form of a core-based tree.

CBT involves having a single-core tree per group, with additional cores to add an element of robustness to the model. Since there exists no polynomial time algorithm that can find the center of a dynamic multicast spanning tree, a core should be "hand-picked", i.e., selected by external agreement based on a judgment of what is known about the network topology between the current members.

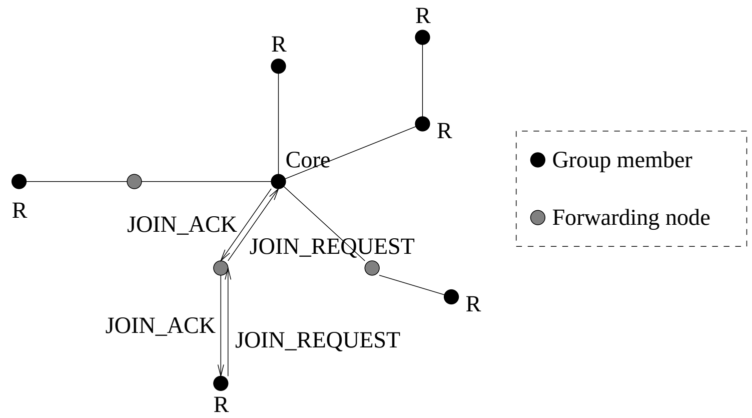

Figure 3: The member join procedure in CBT.

A node can join the group by sending a JOIN_REQUEST. This message is then forwarded to the next-hop node on the path to the core. The join continues its journey until it either reaches the core, or reaches a CBT-capable node that is already part of the tree. At this point, the join's journey is terminated by the receiving node, which normally sends back an acknowledgement by means of a JOIN_ACK. It is the JOIN_ACK that actually creates a tree branch. Figure 3 shows the procedure of a node joining a group.

A non-core node can leave the group by sending a QUIT_REQUEST. A QUIT_REQUEST may be sent by a node to detach itself from a tree if and only if it has no members for that group on any directly attached subnets, and it has received a QUIT_REQUEST on each of its child for that group. The QUIT_REQUEST is sent to the parent node. The parent immediately acknowledges the QUIT_REQUEST with a QUIT_ACK and removes that child from the tree. Any non-core node that sends a QUIT_ACK in response to receiving a QUIT_REQUEST should itself send a QUIT_REQUEST upstream if the criteria described above are satisfied.

For any non-core node, if its parent node or path to the parent fails, that non-core node has one of two options for failure recovery: it can either attempt to re-join the tree by sending a JOIN_REQUEST to the highest-priority reachable core or alternatively, the node subordinate to the failure can send a FLUSH_TREE message downstream, thus allowing each node to independently attempt to re-attach itself to the tree.

For reasons of robustness, we need to consider what happens when a primary core fails. There are two approaches we can take:

- *single-core CBT trees.* If paths or cores fail, a single tree can itself become partitioned. To cater for tree partitions, we have multiple "backup" cores to increase the probability that every network node can reach at least one of the cores of a CBT tree. At any one time, a non-core node is part of a single-core CBT tree.
- *multi-core CBT trees.* Multi-core CBT trees are most useful for groups that are topologically widespread. Each core is then strategically placed where the largest "pockets" of members are located so as to optimize the routes between those members. Each of the cores must be joined to at least one other, and a reachability/maintenance protocol must operate between them. There exists no ordering between the multiple cores, and senders send multicasts preferably to the nearest core.

3 Multicast Protocols in Mobile Ad Hoc Networks

In a highly dynamic environment of mobile ad hoc networks, traditional multicast approaches in wired networks are no longer suitable. Because nodes in these networks move arbitrarily, thus network topology changes frequently and unpredictably. Moreover, bandwidth and battery power are limited. These constraints, in combination with the dynamic network topology make multicasting in mobile ad hoc networks extremely challenging. The general solutions used in the protocols to solve these problems are: avoid global flooding and advertising, dynamically build routes and maintain memberships, etc. In this section, we introduce four extensions in the mobile ad hoc networks.

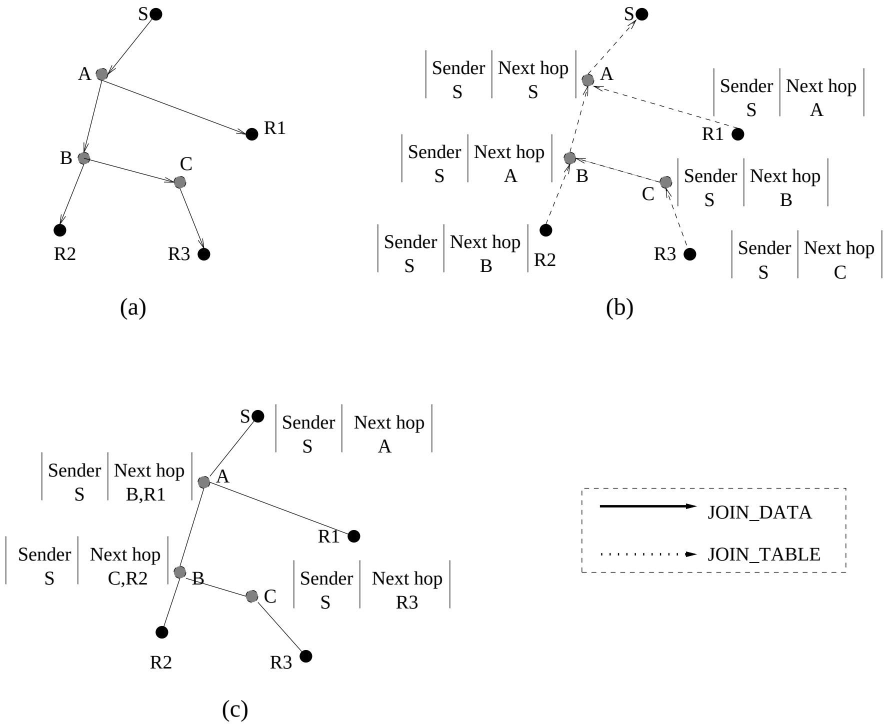

Figure 4: An example of ODMRP. (a) Propagation of JOIN_DATA packets. (b) Propagation of JOIN_TABLE packets. (c) The final multicast tables.

3.1 On-Demand Multicast Routing Protocol (ODMRP)

ODMRP (on-demand multicast routing protocol) [2] is mesh-based, and uses a forwarding group concept (only a subset of nodes forwards the multicast packets). A soft-state approach is taken in ODMRP to maintain multicast group members. No explicit control message is required to leave the group.

In ODMRP, group membership and multicast routes are established and updated by the source on demand. Consider the example in Figure 4 (a). The source S, desiring to send packets to a multicast group, but no route to the multicast group, will broadcast a JOIN_DATA control packet to the entire network. This JOIN_DATA packet is periodically broadcast to refresh the membership information and update routes.

When an intermediate node receives the JOIN_DATA packet, it stores the source ID and the sequence number in its message cache to detect any potential duplicates. The routing table is updated with the appropriate node ID (i.e., backward learning) from which the message was received for the reverse path back to the source node. If the message is not a duplicate and the time-to-live (TTL) is greater than zero, it is rebroadcast.

When the JOIN_DATA packet reaches a multicast receiver, it creates and broadcasts a "JOIN_TABLE" to its neighbors. When a node receives a JOIN_TABLE, it checks to see if the next hop node ID of one of the entries matches its own ID. If it does, the node realizes that it is on the path to the source and thus is part of the forwarding group, and sets the FG_FLAG (forwarding group flag). It then broadcasts its own join table built on matched entries. The next hop node ID field is filled by extracting information from its routing table. In this way, each forward group member propagates the JOIN_TABLE until it reaches the multicast source S via the selected path (shortest). Figure 4 (b) shows how these packets are forwarded to S. On receiving JOIN_TABLEs, a node also has to build its multicast table for forwarding future multicast packets. For example, when B receives R_2's JOIN_TABLE, it will add R_2 as its next hop. The final multicast table for each host is shown in Figure 4(c). This whole process constructs (or updates) the routes from sources to receivers and builds a mesh of nodes called the forwarding group.

After the forwarding group establishment and route construction process, sources can multicast packets to receivers via selected routes and forwarding groups. While it has data to send, the source periodically sends JOIN_DATA packets to refresh the forwarding group and routes. When receiving the multicast data packet, a node forwards it only when it is not a duplicate and the setting of the FG_FLAG for the multicast group has not expired. The procedure minimizes the traffic overhead and prevents sending packets through stale routes.

In ODMRP, no explicit control packets need to be sent to join or leave the group. If a multicast source wants to leave the group, it simply stops sending JOIN_DATA packets since it does not have any multicast data to send to the group. If a receiver no longer wants to receive from a particular multicast group, it does not send the join reply for that group. Nodes in the

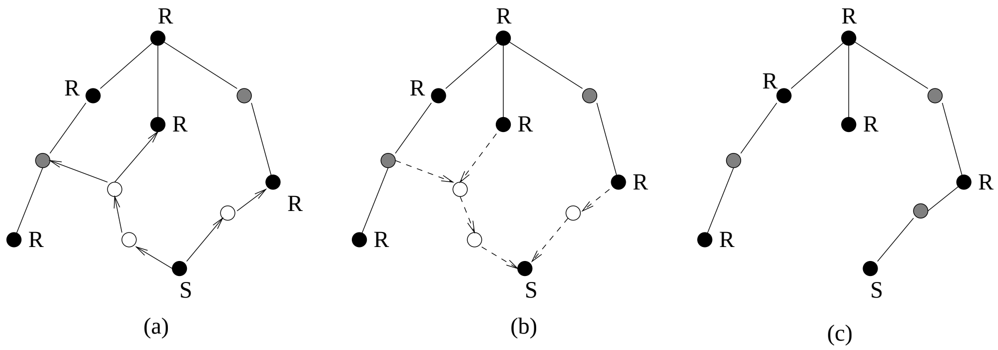

Figure 5: An example of MAODV protocol. (a) The propagation of RREQ packets. (b) The propagation of RREP packets. (c) The final multicast tree.

forwarding group are demoted to nonforwarding nodes if not refreshed (no join tables received) before they time out.

3.2 Multicast Ad Hoc On-demand Distance Vector Routing Protocol (Multicast AODV)

The MAODV routing protocol [24] discovers multicast routes on demand using a broadcast route-discovery mechanism. A mobile node originates a route request *RREQ* message when it wishes to join a multicast group, or when it has data to send to a multicast group but does not have a route to that group. Figure 5 (a) illustrates the propagation of RREQ (represented in the graph by solid arrow) from a host S. Only a member of the desired multicast group may respond to a join RREQ. If the RREQ is not a join request, any node with a fresh enough route (based on group sequence number) to the multicast group may respond. If an intermediate node receives a join RREQ for a multicast group of which it is not a member, or if it receives a RREQ and it does not have a route to that group, it rebroadcasts the RREQ to its neighbors.

As the RREQ is broadcast across the network, nodes set up pointers to establish the reverse route in their route tables. A node receiving a RREQ first updates its route table to reverse route entry may later be used to relay a response back to S. For join RREQs, an additional entry is added to the multicast route table. This entry is not activated unless the route is selected to be part of the multicast tree.

If a node receives a join RREQ for a multicast group, it may reply if it is a member of the multicast group's tree and its recorded sequence number for the multicast group is at least as great as that contained in the RREQ. The responding node updates its route and multicast route tables by placing the requesting node's next-hop information in the tables, and then unicasts a request response *RREP* (represented in the graph by dashed arrow) back to S [shown in Figure 5 (b)]. As nodes along the path to the source node S receive the RREP, they add both a route table and a multicast route table entry for the node from which they received the RREP, thereby creating the forward path.

When S broadcasts a RREQ for a multicast group, it often receives more than one reply. The source node S keeps the received route with the greatest sequence number and shortest hop count to the nearest member of the multicast tree for a specified period of time, and disregards other routes. At the end of this period, it enables the selected next hop in its multicast route table, and unicasts an activation message to this selected next hop. The next hop, on receiving this message, enables the entry for S in its multicast route table. If this node is a member of the multicast tree, it does not propagate the message any further. However, if this node is not a member of the multicast tree, it will have received one or more RREPs from its neighbors. It keeps the best next hop for its route to the multicast group, unicasts an activation message to that next hop, and enables the corresponding entry in its multicast route table. This process continues until the node that originated the RREP (member of tree) is reached. The activation message ensures that the multicast tree does not have multiple paths to any tree node. Nodes only forward data packets along activated routes in their multicast route tables. Figure 5 (c) illustrates the final multicast tree that is created.

The first member of the multicast group becomes the leader for that group. The multicast group leader is responsible for maintaining the multicast group sequence number and broadcasting this number to the multicast group. This is done through a group hello message. The group hello contains extensions that indicate the multicast group IP address and sequence numbers (incremented every group hello) of all multicast groups for which the node is the group leader. Nodes use the group hello information to update their request table.

Since AODV maintains hard state in its routing table, the protocol has to actively track and react to changes in this tree. If a member terminates its membership with the group, the multicast tree requires pruning. Links in the tree are monitored to detect link breakages. When a link breakage is detected, the node that is further from the multicast group leader (downstream of the break) is responsible for repairing the broken link. If the tree cannot be reconnected, a new leader for the disconnected downstream node is chosen as follows. If the node that initiated the route rebuilding is a multicast group member, it becomes the new multicast group leader. On the other hand, if it was not a group member and has only one next hop for the tree, it prunes itself from the tree by sending its next hop a prune message. This continues until a group member is reached. Once these two partitions reconnect, a node eventually receives a group hello for the multicast group that contains group leader information that differs from the information it already has. If this node is a member of the multicast group, and if it is a member of the partition whose group leader has the lower IP address, it can initiate reconnection of the multicast tree.

3.3 Forwarding Group Multicast Protocol (FGMP)

In a highly dynamic network such as mobile ad hoc network, multicast protocols based on upstream and downstream links (like CBT [3] and DVMRP [13]) are not efficient because creating and maintaining upstream and downstream link status in a wireless network cause a lot of overheads.

In [11], authors put forward *forwarding group multicast protocol*. The protocol keeps track not of links but of groups of nodes which participate in multicast packets forwarding. To each multicast group G is associated a forwarding group, FG. Any node in FG is in charge of forwarding (broadcast) multicast packets of G. That is, when a forwarding node (a node in FG) receives a multicast packet, it will broadcast this packet if it is not a duplicate. All neighbors can hear it, but only neighbors that are in FG will first determine if it is a duplicate and then broadcast it in turn. Figure 6

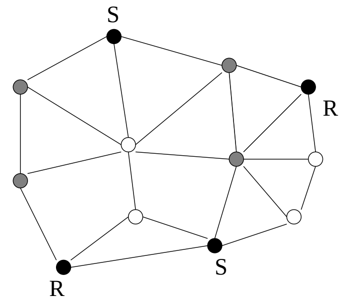

Figure 6: An example of FGMP.

shows an example of a multicast group containing two senders and two receivers. Four forwarding nodes take the responsibility to forward multicast packets. This scheme can be viewed as "limited scope" flooding. That is, flooding is contained within a properly selected forwarding set. It is interesting to note that with proper selection of the forwarding group, the FG scheme can emulate any of the existing schemes. For example, to produce global flooding, the FG must include all nodes in the network. For CBT, the FG is restricted to the nodes on the shared tree except the leaf nodes. In DVMRP, FG includes all the non-leaf nodes on the source trees.

Only one flag and a timer are needed for each forwarding node. When the forwarding flag is set, each node in FG forwards data packets belonging to G until the timer expires. Only the forwarding flag and timer are stored, thus reducing the storage overhead and increasing the flexibility and performance. In FGMP, only small size control messages are flooded and with less frequency.

The major problem of FGMP is how to elect and maintain the set FG of forwarding nodes. The size of FG should be as small as possible to reduce wireless channel overhead, and the forwarding path from senders to receivers should be as short as possible to get high throughput. Three schemes are discussed in the following three subsections.

One way to advertise the membership is to let each receiver periodically and globally flood its member information. When a sender receives the join request from receiver members, it updates the member table. Expired receiver entries will be deleted from the member table. The sender will broadcast multicast data packets only if the member table is not empty. After updating the member table, the sender creates from it the forwarding table FW. Next hop information is obtained from preexisting routing tables. The forwarding table FW is broadcast by the sender to all neighbors; only neighbors listed in the next hop list (next hop neighbors) accept this forwarding table (although all neighbors can hear it). Each neighbor in the next hop list creates its forwarding table by extracting the entries where it is the next hop neighbor and again using the preexisting routing table to find the next hops, etc. After the FW table is built, it is then broadcast again to neighbors and so on, until all receivers are reached. The forwarding group is created and maintained via the forwarding table FW exchanges. At each step nodes on the next hop neighbor list after receiving the forwarding table enable the forwarding flag and refresh the forwarding timer.

Another way to advertise the membership is to let senders flood sender information. Sender advertising is more efficient than receiver advertising if the number of senders is less than the number of receivers. Most multicast applications belong to this category. Like with receiver advertising, senders periodically flood the sender information. Receivers will collect senders' status, then periodically broadcast "joining tables" to create and maintain the forwarding group FG. The "joining table" has the same format as the "forwarding table" except that the joining table contains the sender ids while the forwarding table contains receiver ids. Forwarding flag and timer are set when a node receives the joining table. Forwarding group is maintained by the senders in the receiver advertising scheme and by the receivers in sender advertising scheme.

Wu [29] proposed a method of FW maintenance via connected dominating set. A set of dominating if all the nodes in the network are either in the set or neighbors of nodes in the set. Wu and Li [30] proposed a simple localized algorithm to define a connected dominating set: a host is selected as a dominating set if it has two unconnected neighbors. The size of the dominating set is reduced by two distributed pruning rules: a dominating node can be removed if its neighbor set is covered either by the neighbor set of another node with higher id or by the union of neighbor sets of two connected nodes both with higher ids. With connected dominating set as the basic "spine" of the network, FGMP is built and maintained on top of the set.

3.4 Core-Assisted Mesh Protocol

Many multicasting protocols today involve routing tree. Multicast trees can achieve the efficiency and simplicity by forcing a single path between any pair of routers. If multiple sources must transmit information to the same set of destinations, using routing trees requires that either a shared multicast tree be used for all sources, or that a separate multicast tree be established for each source. Using a shared multicast tree has the disadvantage that packets are distributed to the multicast group along paths that can be much longer than the shortest paths from sources to receivers. Using a separate multicast tree for each source of each multicast group forces the routers that participate in multiple multicast groups to maintain an entry for each source in each multicast group, which does not scale as the number of groups and sources per group increases. In addition, because trees provide minimal connectivity among the members of a multicast group, the failure of any link in the tree partitions the group and requires the routers involved to reconfigure the tree.

Due to these reasons, J. J. Garcia and E. L. Madruga put forward *multicast meshes* and the corresponding *Core-Assisted Mesh Protocol* (CAMP) [16]. CAMP builds and maintains a multicast mesh for information distribution within each multicast group. A multicast mesh is a subset of the network topology that provides at lease one path from each source to each receiver in the multicast group. CAMP ensures that the shortest paths from receivers to sources (called reverse shortest paths) are part of a group's mesh. Packets are forwarded through the mesh along the paths that first reach the routers from the sources, i.e., the shortest paths from sources to receivers that can be defined within the mesh. CAMP does not predefine such paths along the mesh. A router keeps a cache of the identifiers of those packets it has forwarded recently, and forwards a multicast packet received from a neighbor if the packet identifier is not in its cache and the router has been told by at least one neighbor that the router is the successor in a reverse shortest path to any group source.

Because a member router of a multicast mesh has redundant paths to any other router in the same mesh, topology changes are less likely to disrupt the flow of multicast data and to require the reconstruction of the routing structures that support packet forwarding. Figure 7 illustrates how packets

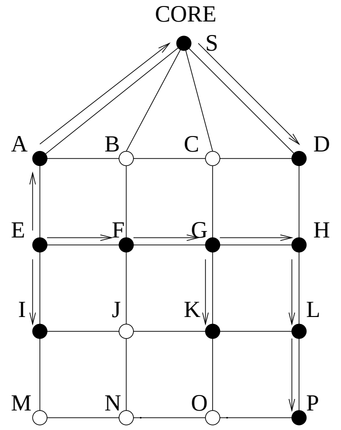

Figure 7: Traffic flow from router E in a multicast mesh

are forwarded from router E to the rest of the group in CAMP. In the figure, solid arrows indicate the flow of traffic along the reverse shortest path in CAMP. Note that CAMP delivers data along shorter paths than a core-based tree protocol. e.g. E to F takes only 1 hop in this example but 4 hops in core-based tree protocol. The length of paths incurred in multicasting over mobile ad hoc networks is very important, because longer paths require more routers forwarding packets.

In CAMP, cores are used to limit the control traffic needed for receivers to join multicast groups. One or multiple cores can be defined for each mesh. Cores need not be part of the mesh of their group. The use of cores in CAMP eliminates the need for flooding, unless all cores are unreachable from a connected component.

4 Other Multicast Protocols in Mobile Ad Hoc Networks

In the recent years, many new multicast protocols in mobile ad hoc networks have been proposed by considering GPS, IP multicast, prefix routing, zone routing, differential destination, real-time, bandwidth, and id-numbers. These protocols are explained as follows.

In [5], an on-demand location aware multicast protocol - OLAM, is introduced. The protocol assumes that, through the use of positioning system devices, such as Global Positioning System (GPS) devices, each node knows its own position and the current (global) time, and it is able to efficiently distribute these measures, including its current transmission radius, to all other nodes. As the measures are received, each node updates its local snapshot of the complete network topology. When a packet is to be multicast to a group, a heuristic is then used to locally compute the Steiner (i.e., multicast) tree for the addressed multicast group based on the snapshot rather than maintaining the tree in a distributed manner. The resulting Steiner tree is then optimally encoded by using its unique Prufer sequence and included along with the packet, extending the length of the header by no more than the header of packets in source routing (unicast) techniques. All local computations are executed using efficient (i.e., polynomial time) algorithms. The protocol has been simulated in mobile ad hoc networks with 30 and 60 nodes and with different multicast group sizes. The results show that OLAM delivers packets to all the nodes in a destination group in more than 85% of the cases.Furthermore, compared to flooding, OLAM achieves improvements of up to 50% on multicast completion delay.

In [6], an approach for robust IP multicast in mobile ad hoc networks by exploiting user-multicast tree and dynamic logical cores, called Ad hoc Multicast Routing Protocol (AMRoute) is presented. It creates a bi-directional, shared tree for data distribution using only group senders and receivers as tree nodes. Unicast tunnels are used as tree links to connect neighbors on the user-multicast tree. Thus, group state cost is incurred only by group senders and receivers and tree structure does not need to change even in case of a dynamic network topology. Certain tree nodes are designated by AMRoute as logical cores, and are responsible for initiating and managing the signaling component of AMRoute, such as detection of group members and tree setup. Simulation results demonstrate that AMRoute signaling traffic and join latency remain at relatively low levels for typical group sizes. The results also indicate that group members receive a high proportion of data multicast by a sender, even in the case of a dynamic network.

In [9], an efficient routing algorithm is discussed for MANETs. The fundamental idea is to build a spanning tree generation by generation with a generation table associated with each mobile station. Once the tree and generation tables are built, routing can proceed along the tree branches. This routing algorithm has a small transmission delay and needs only a small and easy-to-update generation table for each station. In addition, the communication channels in the routing algorithm are dynamically assigned, which allows a large MANET to use a limited number of communication channels. This routing algorithm is combined with *prefix routing* algorithm in [10]. Prefix routing is a special type of routing with a compact routing table (called generation table in the paper) associated with each station. Basically, each station is assigned a special label and it is selected as a forwarding node if its label is a prefix of the label of the destination node. The routing process follows a two-phase process of first going up and then going down the spanning tree, with a possible cross transmission (shortcut) between two branches of the tree between two phases.

In [14], a multicast routing protocol based on zone routing (MZR) is proposed. MZR is a source-initiated on-demand protocol, in which a multicast delivery tree is created using a concept called the zone routing mechanism. It is a source tree based protocol and does not depend on any underlying unicast protocol. The protocol's reaction to topological changes can be restricted to a node's neighborhood instead of propagating it throughout the network.

In [17], a protocol termed differential destination multicast (DDM) is proposed. It differs from other approaches in two ways. Firstly, instead of distributing membership control throughout the network, DDM concentrates this authority at the data sources (i.e. senders) thereby giving sources knowledge of group membership. Secondly, differentially-encoded, variable-length destination headers are inserted in data packets which are used in combination with unicast routing tables to forward multicast packets towards multicast receivers. Instead of requiring that multicast forwarding state to be stored in all participating nodes, this approach also provides the option of stateless multicasting. Each node independently has the choice of caching forwarding state or having its upstream neighbor to insert this state into self-routed data packets, or some combination thereof. The protocol is best suited for use with small multicast groups operating in dynamic networks of any size.

In [18], a protocol for multicasting real-time data called the Wireless

Ad Hoc Real-Time Multicast (WARM) protocol is proposed. The protocol is distributed, highly adaptive and flexible. Multicast affiliation is receiver initiated. The messaging is localized to the neighborhood of the receiving multicast member and thus the overhead consumed is low. The protocol enables spatial bandwidth reuse along a multicast mesh (a connected structure of multicast group members). The real time connection is guaranteed quality of service (QoS) in terms of bandwidth. For VBR traffic, a combination of reserved and random access mechanisms are used. The protocol is self-healing in the sense that the mesh structure has the ability to repair itself when members either move or relays fail. The simulation results show that the throughput is above 90% for pedestrian environments.

In [21], a bandwidth-efficient multicast routing protocol for mobile ad hoc networks is proposed and investigated. The proposed protocol achieves low communication overhead, namely, it requires a small number of control packet transmissions for route setup and maintenance. The proposed protocol also achieves high multicast efficiency, namely, it delivers multicast packets to receivers with a small number of transmissions. In order to achieve low communication overhead and high multicast efficiency, the proposed protocol employs the following mechanisms: (1) on-demand invocation of the route setup and route recovery processes to avoid periodic transmissions of control packets, (2) a new route setup process that allows a newly joining node to find the nearest forwarding node to minimize the number of forwarding nodes, and (3) a route optimization process that detects and removes unnecessary forwarding nodes to eliminate redundant and inefficient routes. The simulation results show that the proposed protocol achieves high multicast efficiency with low communication overhead compared with other existing multicast routing protocols, especially in the case where the number of receivers in a multicast group is large.

In [28], a multicast protocol called AMRIS, short for Ad hoc Multicast Routing protocol utilizing Increasing id-numbers, is introduced. The conceptual idea behind AMRIS is to assign every node in a multicast session with an id-number. A delivery tree rooted at a particular node called Sid joins up the nodes participating in the multicast session. The relationship between the id-numbers (and the node that owns it) and Sid is that the id-numbers increase in numerical value as they radiate from the root of the delivery tree. The significance of the Sid is that it has the smallest id-number within that multicast session. Utilizing the id-numbers, nodes are able to adapt rapidly to changes in link connectivity. Recovery messages due to link breakages are confined to the region where it occurred.

5 Related Issues

The main characteristic of a mobile ad hoc network is its ability to start and maintain the communication setup without the support of any existing wired or wireless infrastructure. However, scarce bandwidth, highly dynamic network topology, and unreliable communication medium pose special challenges on design of such a network. The traditional best effort traffic scenarios are no longer suitable for the mobile ad hoc network. In the following we will discuss two aspects of design related to multicast: QoS multicast and reliable multicast.

5.1 QoS Multicast

The notion of QoS (quality of service) was proposed to capture the qualitatively or quantitatively defined performance contract between the server and client. Specifically, QoS is a guarantee by the network to satisfy a set of predetermined service performance constraints for the client in terms of the end-to-end delay, available bandwidth, probability of packet loss, and so on. QoS in multicasting (routing) typically deals with multiple constraints on the selected routing tree (path). Assume $m(u, v)$ is the performance metric for the link (u, v) connecting host u to host v and a path $(u, u_1, u_2, ..., u_k, v)$ is a sequence of links in the multicast tree. Three types of constraints on the path are given in [26]:

1. *Additive constraints*: A constraint is additive if

$$m(u, v) = m(u, u_1) + m(u_1, u_2) + ... + m(u_k, v)$$

   For example, the end-to-end $delay(u, v)$ is an additive constraint which is equivalent to the summation of delays at each link.

2. *Multiplicative constraints*: A constraint is multiplicative if

$$m(u, v) = m(u, u_1) \times m(u_1, u_2) \times ... \times m(u_k, v)$$

   The probability $prob(u, v)$ for a packet to reach v from u is the product of individual link probabilities.

3. *Concave constraints*: A constraint is concave if

$$m(u, v) = \min\{m(u, u_1), m(u_1, u_2), ..., m(u_k, v)\}$$

   The bandwidth $band(u, v)$ available along path from u to v is the minimum bandwidth among the links on the path.

Based on the above classification of constraints, Wang and Hou [26] gave a list of twelve combinations with multiple constraints. It has been proved in [27] that any multiple constraints with two or more type 1. and/or 2. constraints is NP-complete; otherwise, it is tractable. Various approximation methods exist for QoS constraints that are NP-complete. One commonly used approach is *sequential filtering* where paths based on a single primary metric (say bandwidth) are selected first, and a subset of them are eliminated by optimizing over the secondary metric (say delay), and so on.

The mobility of mobile ad hoc networks adds another dimension of difficulty. Highly mobile hosts in the ad hoc network will make any QoS constraints unobtainable. Therefore, it is assumed that the mobile ad hoc network under consideration is *combinatorially stable* [7]: under a specific time window, the topology changes occur sufficiently slowly to allow successful propagation of all topology updates as necessary.

QoS routing (multicast) depends on the accurate availability of the current network state, which is expensive to maintain because of network dynamics and aggregation in large networks. The *imprecise network state model* [8] is a promising approach and provides a cost-effective solution for QoS routing (multicast) based on imprecise network information. Most of QoS routing (multicast) are reservation-based, probe signals are sent out to find QoS route(s) to the destination (connecting host(s) on the multicast or core tree). Because of network dynamics and imprecise state information, reserved QoS route(s) needs to be reaffirmed periodically by sending special control packets, called *refreshers*, from the destination (connecting host on the multicast or core tree) back to the source. Another approach is the use of *soft state* to tree/state maintenance: the state and reservation kept at each router periodically times out.

The main issues in providing QoS multicasting in mobile ad hoc networks are (a) locating a QoS route and (b) maintaining desired QoS on a multicast or core tree. We use the core-based tree as an example. Locating a QoS route is to find a QoS route to the core tree. In this case, the shortest path from a requesting host to the core may not be the best QoS route. In [4], a QoS route is found through a TTL-based bid-order broadcast. On-tree routes that receive the broadcast message become candidates and return bid messages. To maintain desired QoS on a multicast or core tree, each join request message carries relevant QoS parameters. The core/on-tree router conducts a set of eligibility tests to decide whether or not a new member can join.

Normally, routing and resource reservation are treated separately. It is an open problem whether or not to consider these two related issues at one stage, rather than two separated stages. The trade-off between the design complexity of QoS multicast protocols and the resulting performance improvement, especially in large-scale networks, still remains as a challenging issue. The models of imprecise information to support QoS routing/multicasting still need to be developed.

5.2 Reliable Multicast

The design of reliable multicast depends on the following three decisions [23]: (a) by whom errors are detected, (b) how error messages are signaled, and (c) how missing packets are retransmitted. (a) and (b) are normally handled jointly.

In the sender-initiated approach, the sender is responsible for the error detection. Error messages are signaled using ACK signals sent from each receiver. A missing data at a receiver is detected if the sender does not receive an ACK from the receiver. In this case, (c) is handled by retransmitting the missing data from the source through a unicast. When several receivers have missing packets, the sender may decide to re-multicast the missing packets to all receivers in the multicast group.

In the receiver-initiated approach, each receiver is responsible for the error detection. Instead of acknowledging each multicast packet, each receiver sends a NACK once it detects a missing packet. Suppose multicast packets are timestamped using a sequence number, a missing packet can be detected by a gap between sequence numbers of the receiving packets.

When the sender-initiated approach is applied, only the sender (that keeps the history of multicast packets) is responsible for re-transmitting the missing packet and the corresponding re-transmitting method is called sender-oriented. Note that when the sender receives ACK signals from all the receivers, the corresponding packet can be removed from the history.

There are three ways to retransmit the missing packet when the receiver-initiated approach is used: (1) sender-oriented, (2) neighborhood-oriented, and (3) fixed-neighborhood-oriented. These methods differ by the locations of the copies of missing packets. These locations are also called copy sites, which include the sender. Note that when there are several receivers that have the same missing packet, multicast NACK signals will be sent to the copy site(s). To ensure that at most one NACK is returned to the sender per packet transmission, when a receiver detects a missing error, it waits a random period of time before broadcasting a NACK to the sender and all other receivers. This process is called *NACK suppression* since a receiver will cancel its broadcast if it receives a NACK which corresponds to a packet it has missed.

In the sender-oriented approach, senders can either unicast to a receiver (that needs the missing packet) or multicast to all the receivers in the multicast group. In the neighborhood-oriented approach, the receiver that needs the missing packet searches its neighborhood for a group member that keeps a copy of the missing packet. The search process use a TTL-based unicast process or TTL-based broadcast process. The search space is either limited to the multicast tree (but now it is rooted at the receiver) or without limitation. In the fixed-neighborhood-oriented approach, the copy sites are fixed to a sub-group or each receiver has a “buddy” in the multicast group to backup each other.

Mobility of mobile ad hoc networks add complexity in achieving reliability. When a host moves from one neighborhood to another, proper hand-off protocols are needed. For example, when host u has just completed its forwarding process to its neighbor v, host w, a neighbor of v, moves away from the neighborhood of v and enters the neighborhood of u. To ensure that host w gets a copy of the packet, u needs to keep the copy for a while and will re-forward the packet (with a proper tag indicating this is a re-forwarding packet) whenever a change of its neighborhood is detected.

Reliability can be achieved through other means. For example, *forward error correction* [20] adds redundant information, which allows lost packets to be reconstructed from correctly received packets received from either a single path or multicast paths [25]. Two other requirements, ordering and delivery semantics that are commonly used in the traditional distributed system are still uncharted territory. Also, the way to integrate reliable multicast and QoS multicast still remains an open issue.

6 Conclusions

In this chapter, we have first reviewed two wired multicast protocols: shortest path multicast tree and core-based tree methods, that are used widely in wired networks. And then, we have described four extensions of these wired methods to mobile ad hoc networks: two distinct on-demand multicast protocols, forwarding group multicast protocol (FGMP) and core-assisted mesh protocol. Other multicast protocols used in mobile ad hoc networks have also been briefly summarized. And lastly, two related issues: QoS multicast and reliable multicast have also been discussed. There is a lot of work to be done in this field in the future. Namely, simulations need to be conducted to examine the various tradeoffs and alternatives of multicast routing algorithms suitable for mobile ad hoc networks; and the development of the multicast gateway for inter-connecting wired network multicast with ad hoc based multicast.

Acknowledgement

This work was supported in part by NSF grant CCR 9900646 and grant ANI 0073736.

References

[1] *citeseer.nj.nec.com/lucas95distributed.html*, 18.

[2] 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*, p. 70-77, Vol. 14, No. 1, January-February 2000.

[3] T. Ballardie, P. Francis, and J. Crowcroft, "Core Based Trees (CBT): An Architecture for Scalable Inter-Domain Multicast Routing,"*Proc. of ACM SIGCOMM'93,* p. 85, 1993.

[4] A. Banerjea, M. Faloutsos, and R. Pankaj, "Designing QoSMIC: a Quality of Service sensitive Multicast Internet protocol," *submitted as Internet Draft IETF in the IDMR working group*, 2000.

[5] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and R. Talebi, "On-demand Location Aware Multicast (OLAM) for Ad Hoc Networks," *Proc. of Wireless Communications and Networking Conference*, p. 1323-1328, Vol. 3, September 2000.

[6] E. Bommaiah, M. Liu, A. McAuley, and R. Talpade, "AMRoute: Ad Hoc Multicast Routing Protocol," Internet Draft, http://www.ietf.org/internet-drafts/draft-talpade-manet-amroute-00.txt.

[7] S. Chakrabarti and A. Mishra, "QoS Issues in Ad Hoc Networks," *IEEE Communications Magazine*, p. 142-148, Vol. 39, No. 2, February 2001.

[8] S. Chen, "Routing Support for Providing Guaranteed End-to-End Quality-of-Service," *http://www.cs.uiuc.edu/Dienst/UI/2.0/Describe /ncstrl.uiuc_cs/UIUCDCS-R-99-2090*, UIUCDCS-R-99-2090, University of Illinois at Urbana-Champaign, Urbana, Illinois, July 1999.

[9] X. Chen and X. D. Jia, "Package Routing Algorithms in Mobile Ad Hoc Networks," *Proc. of the Workshop on Wireless Networks and Mobile Computing held in conjunction with 2001 International Conference on Parallel Processing*, p. 485-490, September 2001.

[10] X. Chen, J. Wu, and X. D. Jia, "Prefix Routing in Wireless Ad Hoc Networks of Mobile Stations," January 2002, work in progress.

[11] C. -C. Chiang, M. Gerla, and L. Zhang, "Forwarding Group Multicast Protocol (FGMP) for Multihop, Mobile Wireless Networks," *Cluster Computing*, p. 187, 1998.

[12] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, Introduction to Algorithms, MIT Press, 1997.

[13] S. E. Deering and D. R. Cheriton, "Multicast Routing in Datagram Internetworks and Extended LANs," *ACM Transactions on Computer Systems,* p. 85-111, 1990.

[14] V. Devarapalli, and D. Sidhu, "MZR: a Multicast Protocol for Mobile Ad Hoc Networks," *Proc. of IEEE International Conference on Communications*, p. 886-891, Vol. 3, June 2001.

[15] M. Faloutsos, A. Banerjea, and R. Pankaj, "QoSMIC: Quality of Service Sensitive Multicast Internet Protocol," *SIGCOMM citeseer.nj.nec.com/faloutsos98qosmic.html*, p. 144-153, 1998.

[16] J. J. Garcia-Luna-Aceves and E. L. Madruga, "A Multicast Routing Protocol for Ad Hoc Networks," *Proc. of IEEE INFOCOM'99*, p. 784-792, March 1999.

[17] L. S. Ji, and M. S. Corson, "Differential Destination Multicast-a MANET Multicast Routing for Multihop, Ad Hoc Network," *Proc. of IEEE INFOCOM'01*, p. 1192-1201, Vol. 2, April 2001.

[18] G. D. Kondylis, S. V. Krishnamurthy, S. K. Dao, and G. J. Pottie, "Multicasting sustained CBR and VBR traffic in wireless ad hoc networks," *Prof. of IEEE International Conference on Communications*, p. 543-549, Vol. 1, June 2000.

[19] S. J. Lee, M. Gerla, and C. C. Chiang, "On-demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks," Internet draft, draft-retf-manet-odmrp-01.txt, June 1999, work in progress.

[20] M. T. Lucas, B. J. Dempsey, and A. C. Weaver, "Distributed Error Recovery for Continuous Media Data in Wide-Area Multicast," CS-95-52, University of Virginia, July 1995.

[21] T. Ozaki, J. B. Kim, and T. Suda, "Bandwidth-efficient Multicast Routing for Multihop, Ad Hoc Networks," *Proc. of IEEE INFOCOM'01*, p. 1182-1191, Vol. 2, April 2001.

[22] C. Perkins and E. M. Royer, "Ad Hoc on Demand Distance Vector (AODV) Routing (Internet draft)," August 1998.

[23] D. G. Petitt, "Reliable Multicast Protocol Design Choices," *MILCOM 97 Proceedings*, p. 242-246, Vol. 1, November 1997.

[24] E. M. Royer, C. E. Perkins, "Multicast Operation of the Ad Hoc On-Demand Distance Vector Routing Protocol," *Proc. of MobiCom*, p. 207-218, 1999.

[25] A. Tsirigos and Z. J. Haas, "Multipath Routing in the Presence of Frequent Topological Changes," *IEEE Communications Magazine*, p. 132–138, Vol. 39, No. 11, November 2001.

[26] B. Wang and C. -J. Hou, "A Survey on Multicast Routing and its QoS Extension: Problems, Algorithms, and Protocols," *IEEE Network Magazine*, p. 22-36, Vol. 14, No. 1, January/February 2000.

[27] Z. Wang and J. Crowcroft, "QoS Routing for Supporting Resource Reservation," *IEEE Journal on Selected Areas in Communications*, p. 1228-1234, Vol. 14, No. 7, September 1996.

[28] C. W. Wu, Y. C. Tay, and C. -K. Toh, "Ad hoc Multicast Routing protocol utilizing Increasing id-numbers (AMRIS) Functional Specification," *Internet-Draft*, draft-ietf-manet-amris-spec00.txt, November 1998, work in progress.

[29] J. Wu, "Dominating-Set-Based Routing in Ad Hoc Networks," in *Wireless Networks and Mobile Computing*, I. Stojmenovic (ed.), p. 425-460, 2002.

[30] J. Wu, H. Li, "A Dominating-Set-Based Routing Scheme in Ad Hoc Wireless Networks," special issue on *Wireless Networks* in the *Telecommunication Systems Journal*, p. 63-84, Vol. 3, 2001.