Transmission Range Effects on AODV Multicast Communication

Elizabeth M. Royer[a] and Charles E. Perkins[b]

[a] *Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106*
[b] *Communications System Laboratory, Nokia Research Center, Mountain View, CA 94043*

As laptop computers begin to dominate the marketplace, wireless adapters with varying bandwidth and range capabilities are being developed by hardware vendors. To provide multihop communication between these computers, ad hoc mobile networking is receiving increasing research interest. While increasing a node's transmission range allows fewer hops between a source and destination and enhances overall network connectivity, it also increases the probability of collisions and reduces the effective bandwidth seen at individual nodes. To enable formation of multihop ad hoc networks, a routing protocol is needed to provide the communication and route finding capability in these networks. The Ad hoc On-Demand Distance Vector Routing protocol (AODV) has been designed to provide both unicast and multicast communication in ad hoc mobile networks. Because AODV uses broadcast to transmit multicast data packets between nodes, the transmission range plays a key role in determining the performance of AODV. This paper studies the effects of transmission range on AODV's multicast performance by examining the results achieved at varying transmission ranges and network configurations.

Keywords: ad hoc networks, wireless networks, mobile networking, multicast

1. Introduction

Within the last few years, mobile computing has gained popularity as laptop computers have become smaller, lighter, and more powerful. It has become commonplace for professionals to carry their computers with them as they travel. With this increase in popularity has also come a greater demand for connectivity. The idea of *anywhere/anytime* network access naturally appeals to mobile users. These users want to access the Internet and communicate with their associates, whatever their location. This provides the motivation for *ad hoc networking*, or the on-the-fly formation of networks. Protocols for managing such ad hoc networks must be able to formulate routes between any given source and destination, and then be able to maintain these routes as the location of the users changes.

As the number of mobile users has risen, a wide variety of applications have become available. Some of these new applications rely on multicast communication for their operation. While identical semantically to the corresponding concept in wired networks, multicast in ad hoc mobile networks has a distinguishing set of characteristics and constraints. These include limited power, limited bandwidth, and high error rates. An ad hoc multicast protocol must be able to connect all group members and then maintain this connectivity after topological changes in the network.

The Ad hoc On-Demand Distance Vector (AODV) routing protocol provides both unicast and multicast communication connectivity in an ad hoc mobile environment [11,16]. AODV is a reactive protocol in that it creates routes on-demand, or as needed. Both unicast and multicast routes are built using a route discovery cycle, which is initiated when a source node wishes to either find a route to some destination or join a multicast group. Nodes which are able to provide a route to the desired destination respond to the source node by sending it a reply packet. Once a route is established, it is maintained as long as it is needed; i.e., until either the source node stops sending packets, or until there are no longer any members of the multicast group. AODV is able to quickly repair link breaks in active routes whenever they occur.

Other ad hoc multicast protocols have recently been developed as this topic has attracted the attention of the research community. The On-Demand Multicast Rout-

ing Protocol (ODMRP) [9] is a mesh-based algorithm which calculates a forwarding group for each multicast group. The forwarding group is a set of nodes which forward multicast packets that they receive. The group is periodically refreshed by a network-wide Join Request message broadcast by each multicast source node. An alternative protocol is the Core-Assisted Mesh Protocol (CAMP) [6]. Like ODMRP, CAMP also creates a shared mesh per multicast group. CAMP uses core nodes to limit the amount of control traffic when nodes join a group, and it ensures that the shortest path from receivers to sources is a part of the mesh. Finally, the Lightweight Adaptive Multicast protocol (LAM) [7] is similar to AODV in that it is tightly coupled with a unicast protocol (TORA [10]), and also creates a shared tree for each multicast group. However, LAM bases this shared tree at a pre-selected core node.

Current wireless modems offer a wide range of transmission power and connectivity. For instance, the Wavelan IEEE Turbo card, which offers 2 Mb/s at a 400m transmission radius in open office conditions and a 90m radius in semi-open conditions. The 1 Mb/s data rate for this card offers a 550m and 115m range in the same conditions. Proxim's 1.6 Mb/s RangeLAN2 offers a 300m outdoor range and 150m indoor range. Breezecom's SA-PC Pro card provides data rates between 1 and 3 Mb/s, while transmitting at a range of 600m outdoors.

These products offer a variety of power levels, and thus transmission ranges, for fairly similar bandwidth. It might seem desirable to have the largest possible transmission radius, since this would provide the greatest amount of connectivity. However, high density networks suffer from channel access delays and an increased number of collisions. Moreover, applications sometimes have to run in constrained environments. For instance, in a conference scenario, attendees are likely to be confined within some fixed area, possibly in just a single large room. With that area, it is not necessarily the case that the largest transmission radius is the best solution for connectivity. A large transmission radius necessarily implies that more users are affected by each transmission, thereby limiting the effective bandwidth of neighboring users. The larger the transmission radius, the more users affected by transmissions.

In the case of unicast data, the effects of a large transmission radius in a confined area may be somewhat mitigated by channel access schemes such as the IEEE 802.11 Distributed Coordination Function (DCF) [4]. For unicast communication, DCF utilizes short control packets for acquiring the channel, and provides an acknowledgment to ensure data reception. Thus, while the number of collisions of the control packets may increase with larger transmission radii, thereby increasing channel access time, data throughput may not be notably affected. Since the number of hops to reach a destination is smaller for a larger transmission radius, the total delay for a data packet between source and destination might even decrease.

However, for multicast data, the situation is quite different. Multicast data packets are in some ways similar to broadcast traffic. In the case of AODV, when a node receives a data packet with a multicast destination address, it must send the packet up the protocol stack at least as far as the IP layer to determine whether to accept or forward the packet. For short transmission radii, the number of nodes affected by a single transmission is small, so a network node that does not belong to any multicast groups is less likely to waste significant processing power discarding useless packets. However, as the transmission range increases, the number of nodes which receive multicast data transmissions also increases, and, assuming that the membership of a multicast group is a relatively small percentage of the total network population, the number of nodes adversely affected by these multicast transmissions similarly rises. Hence there is a tradeoff between being able to reach multicast group members in a smaller number of hops, and keeping the set of nodes affected by multicast data transmissions to a minimum.

This paper investigates the nature of that tradeoff, examining the effects of varying transmission range within different confined network areas. A variety of results are examined, including the packet delivery ratio for multicast data packet delivery, and the number of multicast data packets non-group members must discard. The remainder of this paper is organized as follows. Section 2 takes an in-depth look at how AODV provides multicast connectivity for the lifetime of a multicast group, including the creation and maintenance of the multicast tree. Section 3 describes the forwarding mechanism used for the multicast data packets. Then, section 4 describes the simulations performed and examines the results of these simulations to determine

how the change in transmission range effects varying aspects of the protocol and the network connectivity. Section 5 describes directions for future work, and finally section 6 presents the conclusions from this study.

2. The Protocol

AODV's multicast operation is based on a route discovery cycle. When a node wishes to either join a multicast group or find a route to a group, it initiates route discovery by sending a *Route Request* (RREQ) packet. As nodes join the multicast group, a bidirectional tree is formed from the group members and the nodes connecting those group members (tree routers). There is only one tree per multicast group, and each multicast group has associated with it a multicast group leader. The multicast group leader's sole responsibility is to maintain and disseminate the multicast group sequence number. AODV utilizes sequence numbers to ensure relative freshness of routes, and thereby to prevent routing loops.

2.1. Routing Tables

Each node running AODV must potentially maintain two tables related to multicast. If a node is either a member of a multicast group or is a router for such a group, it must maintain a *Multicast Route Table* (MRT). The MRT is used by nodes to maintain next hop information for the multicast trees. The fields of the MRT are as follows:

- Multicast Group IP Address
- Multicast Group Leader IP Address
- Multicast Group Sequence Number
- HopCount to Multicast Group Leader
- Next hops, with the following data per hop:
  - Next Hop IP Address
  - Link Direction
  - Activated Flag

There is one entry in the MRT for each multicast group of which the node is either a member or a tree router. Each entry has associated with it a list of one or more next hops, or neighbors on the multicast tree. The next hops field is a linked list of structures, each of which contains the indicated information.

The link direction of a next hop is defined to be *upstream* if the link is towards the group leader, and *downstream* if it is away from the group leader. Because of the tree structure, a node should have at most one upstream link at any time. The Activated flag associated with each next hop is an indication of whether the link has been officially added to the multicast tree (see section 2.3.3). When a link is added to the tree, the flag is set, and only after that time can the link be used for receiving multicast data packets. In figure 1, node A's next hops on the multicast tree are enclosed by the dashed line.

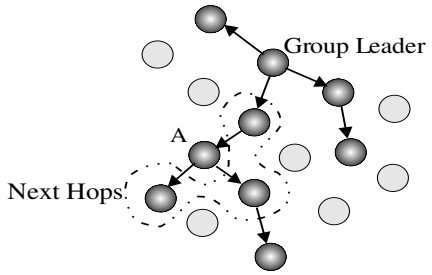


Figure 1. Sample Multicast Tree.

AODV also maintains a *Group Leader Table* (GLT) with two fields per entry:

- Multicast Group IP Address
- Multicast Group Leader IP Address

When a node receives a Group Hello (section 2.2), it updates its GLT to reflect the multicast group/group leader association indicated in the Group Hello message. If the node later wants to join a multicast group, it first checks its GLT for an entry for that group. If there is such an entry, and if the node has a route to the multicast group leader, it may unicast its Route Request to the group leader instead of broadcasting it across the network. This table is used only as an optimization; its elimination does not affect the correct operation of the protocol.

2.2. The Group Leader

Each multicast group has associated with it a group leader. When a node wishes to join a multicast group, it broadcasts a RREQ and then waits for a reply. If after some maximum number of attempts (`rreq_retries` + 1) it does not receive a reply, it may assume that there are no other members of the group in the connected partition of the network. It then becomes the group leader for that multicast group and initializes the sequence number to one. Once it becomes the group

leader, it broadcasts a *Group Hello* (GRPH) message. This message contains the following fields:

< *flags*, *hop_cnt*, *source_addr*, *mgroup_addr*, *mgroup_seqno* >

Currently, there are two flags defined. The first of these is the *Update* flag. This is set when there is a change in group leader information, as described in section 2.5.1. The second flag is *Offmtree*. When the group leader initiates the GRPH, it leaves this flag unset. Whenever a node *not* on the multicast tree receives the message, it sets this flag. This indicates that the GRPH message has traveled off the tree along this path. When a node on the multicast tree receives a GRPH with the *Offmtree* flag unset, it knows that the GRPH has traveled solely on tree links, and so the *hop_cnt* field can be used to update the node's current distance from the group leader. Otherwise, if a multicast tree node receives the packet with that flag set, it knows it cannot use the *hop_cnt* value as an indicator of its distance from the group leader, because the packet has not traveled only along the tree. The significance of this message being broadcast instead of multicast across the tree is shown in section 2.5.3.

The *hop_cnt* field is incremented each time the GRPH packet is forwarded. The *source_addr* field is set to the group leader's IP address, and the *mgroup_addr* and *mgroup_seqno* fields are set to the multicast group IP address and current sequence number, respectively. The group leader increments the group sequence number each time it initiates a new GRPH message.

When a node receives the GRPH packet, it records the multicast group IP address and sequence number before rebroadcasting the packet. If it later receives a GRPH with this same multicast group IP address/sequence number combination, it knows it has already seen this GRPH message and it can discard the packet. If, on the other hand, a node receives a GRPH packet it has not seen before, it updates its GLT to reflect the current group/group leader combination. If it is a member of the multicast tree, it also updates the multicast group sequence number.

A group leader change occurs when the current group leader either decides to unsubscribe from the group, or when the multicast tree becomes partitioned. These scenarios are described in sections 2.4 and 2.5.2, respectively.

2.3. Subscribing to the Multicast Group

A route discovery cycle is initiated each time a node would like to find a route to a multicast group. It may initiate route discovery in order to subscribe to a new group, or because it would like to begin sending to a group of which it is not already a member. The node initiates route discovery by broadcasting a RREQ. It then waits for the reception of a *Route Reply* (RREP) packet. After the discovery period, the node selects its next hop towards the multicast tree. It activates this link by unicasting this node a *Multicast Activation* (MACT) message.

2.3.1. Route Requests

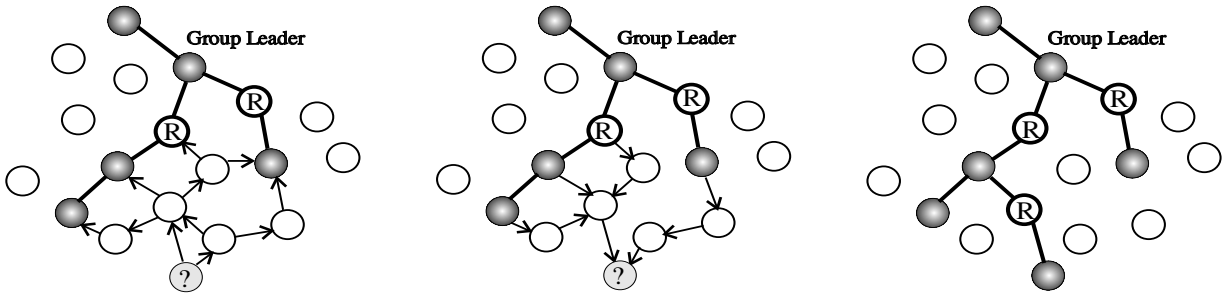
When a node wishes to subscribe to a multicast group or to find a route to a group of which it is not already a member, it initiates route discovery by broadcasting a RREQ. The RREQ has the following structure:

< *flags*, *hop_cnt*, *broadcast_ID*, *dest_addr*, *dest_seqno*, *source_addr*, *source_seqno* >

The currently defined flags are *Join* and *Repair*. *Join* is set when the node wishes to join the group, as opposed to just find a route to the group. The *Repair* flag is set when the RREQ is sent to repair the multicast tree (section 2.5.3). The *dest_addr* field is the IP address of the desired multicast group, and the *dest_seqno* is the source's record of the last known sequence number of the multicast group. Processing for the other RREQ fields follows the unicast algorithms as specified in [12] and reported in [11].

When a node receives the RREQ, it notes the node from which the RREQ arrived, and creates a next hop entry in its MRT for that previous hop. The Activated flag for that next hop is *false*. The node then determines whether it can send a RREP by the method described in section 2.3.2. If it cannot send a RREP, it rebroadcasts the request to its neighbors.

To reduce the impact of route discovery, the RREQ may be sent in an expanding ring search [12] for the destination. Figure 2(a) illustrates the propagation of a join RREQ throughout the network. In this figure, as well as in the multicast figures in the following sections, multicast group members are represented by shaded circles, and tree routers are represented by circles with the letter 'R'.



(a) RREQ Propagation (b) RREPs Returned to Source (c) Multicast Tree Branch Addition

Figure 2. Multicast Group Join.

2.3.2. Route Replies

If the RREQ is *not* a join request, any node with a current route to the multicast group can respond by sending a RREP. A current route is a route to the multicast group whose associated sequence number is no less than the *dest_seqno* of the RREQ. On the other hand, if the RREQ *is* a join RREQ, only a node that is a member of the multicast tree may respond to the RREQ. Since a RREP in response to a join request sets up a potential branch addition to the multicast tree, only members of the multicast tree are allowed to initiate this RREP. In either case, if the node determines it can respond to the RREQ, it creates a RREP and unicasts the RREP to the source node. RREP contains the following parameters:

$< flags,\ prefix\_size,\ hop\_cnt,\ dest\_addr,$
$dest\_seqno,\ source\_addr,\ lifetime >$

The only flag currently defined for the RREP is *Repair*. This flag is set when the RREP is in response to a repair request (section 2.5.1). The *prefix_size* field is utilized for subnet routing, as discussed in [12]. If the node generating the RREP is a member of the multicast tree, the *hop_cnt* field is initialized to zero. Otherwise, it is set to the responding node's distance from the multicast tree. This field is incremented each time the RREP is forwarded, so that when the source node receives the RREP it indicates the source's distance from the multicast tree. The *dest_addr* is set to the multicast group's IP address, and the *dest_seqno* is set to the responding node's record of the group's sequence number. The *source_addr* is the address of the node that originated the request. The *lifetime* field is used when the request is not a join request. It is set to the responding node's current lifetime for the multicast group route entry. For nodes on the multicast tree, the multicast group entry itself does not time out, and hence does not have a lifetime associated with it. Only the individual next hop links may time out.

If the RREQ was a join request, the RREP also contains an extension called the Multicast Group Information Extension. This extension contains the multicast group leader IP address and another hopcount field called *mgroup_hcnt*. This hopcount is set equal to the responding node's distance from the group leader. It is incremented each time the packet is forwarded, so that when the subscribing node (i.e., the node that sent the RREQ) receives the RREP, it indicates that node's distance from the group leader.

When a node receives a RREP, it stores the IP address of the node from which it received this packet. It also adds a next hop entry for the previous node to its multicast route table entry, and leaves the Activated flag associated with this next hop unset. The node then forwards the RREP towards the source. If an intermediate node later receives another RREP for the same subscribing node and multicast destination pair, it only forwards the new RREP if that RREP offers a better route than was previously known. A better route is one with either a greater destination sequence number or the same destination sequence number but a smaller hopcount to the multicast tree. Figure 2(b) shows the path of the RREPs sent back to the subscribing node.

After transmitting the RREQ, the subscribing node waits the discovery period (`route_discovery_timeout`) before selecting a route. During this period, it keeps track of the best route (greatest sequence number and smallest hopcount) to the multicast tree. At the end of the discovery period, the subscribing node selects its

next hop and activates that next hop, as described in the next section.

2.3.3. Multicast Activation

Once the discovery period has ended and the subscribing node has chosen its next hop, it activates this entry in its MRT by setting the Activated flag associated with that next hop. It then creates a *Multicast Activation* (MACT) message, and unicasts this message to its selected next hop. The MACT message contains the following fields:

< *flags*, *hop_cnt*, *mgroup_addr*, *source_addr*, *source_seqno* >

The currently defined flags for the MACT message are *Join*, *Prune*, *Grpldr*, and *Update*. The *Join* flag is set when the node is joining the multicast tree, while the *Prune* flag is used by a node when it wishes to prune itself from the tree (section 2.4). The *Grpldr* flag is used after a network partition when a new group leader must be selected (section 2.5.2), and the *Update* field is used after a tree branch repair (section 2.5.1).

The *hop_cnt* field is primarily used after a tree repair (section 2.5.1). For link activation, this field is set to one. The *mgroup_addr* field is set to the IP address of the multicast group, and the *source_addr* and *source_seqno* fields are set to the IP address and current sequence number of the node initiating the MACT, respectively.

When the next hop receives the MACT message, it activates the next hop entry for the sending node in its MRT. If this next hop was already a member of the multicast tree, the addition of the new branch to the tree is completed. Otherwise, if this next hop was not already a member of the multicast tree, then, like the subscribing node, it will also have been keeping track of the best next hop to the multicast tree. It activates this next hop in its MRT, and then unicasts a MACT message to this next hop. Processing continues in this manner until an existing member of the multicast tree is reached. Figure 2(c) shows the multicast tree after the join is completed.

2.4. Unsubscribing From the Multicast Group

Multicast group membership is dynamic; nodes can subscribe to or unsubscribe from a multicast group at any time. A node prunes itself from the multicast tree using a variation of the MACT message.

A multicast group member may unsubscribe from a multicast group of which it is a member at any time. However, it may only exit the *multicast tree* if it is a leaf node. If a non-leaf node attempted to exit the tree, the tree would then become partitioned. Hence, a non-leaf node that wishes to unsubscribe from the multicast group may change its member status internally, but it takes no overt action to notify any of the other tree members.

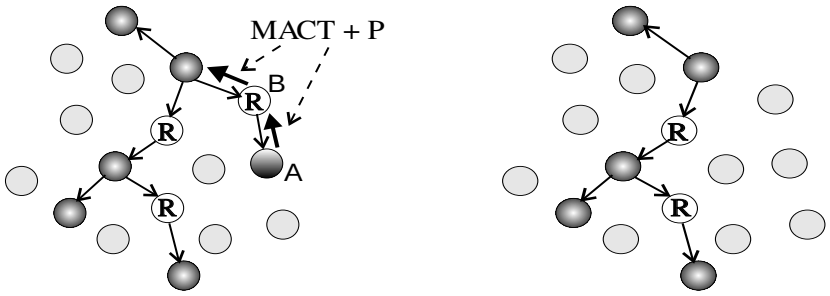
A leaf node unsubscribes from the multicast group by pruning itself from the multicast tree. It does this by first deleting the entry for that multicast group from its MRT, and then creating a MACT message with a set *Prune* flag. It then unicasts this message to its next hop.

When the next hop receives the prune message, it deletes the sending node's information from its MRT entry for that multicast group. If, due to the pruning of the sending node, the receiving node is now a leaf node, and if this node is *not* a multicast group member (only a tree router), it can in turn prune itself from the tree in the previously described manner. Otherwise, if it is not a leaf node, or if it is a member of the multicast group, then pruning ends at this node.

Figure 3 illustrates a pruning operation. In figure 3(a), node A is a multicast group member that wishes to unsubscribe from the group. Since it is a leaf node, it should also prune itself from the tree. It creates the MACT prune message and unicasts this message to its next hop, node B. When node B receives the message, it deletes node A from its next hop entries, and then notes that it is now a leaf node. Since it is a tree router and not a group member, it then prunes itself from the tree as well. Figure 3(b) illustrates the multicast tree after pruning.

When the group leader decides to unsubscribe the group, it operates in a similar manner. If it is a leaf node, it may prune itself from the tree. Otherwise, it must remain a router for the tree.

If it is a leaf node, it sends the prune message to its next hop and deletes the multicast group information from its MRT. When the next hop receives the prune message, it is handled as described in section 2.5.2. Otherwise, if the group leader is not a leaf node, it selects one of its next hops and sends this node a MACT mes-



(a) Pruning of Multicast Group Member

(b) Multicast Tree After Prune

Figure 3. Unsubscribing from the Multicast Group.

sage with set *Grpldr* flag, as is also described in section 2.5.2.

2.5. Multicast Tree Maintenance

Because the network nodes are mobile, links between nodes are likely to break. A multicast tree is maintained for the lifetime of the multicast group. Hence, there must be a way of maintaining the tree after topological changes in the network.

Multicast tree maintenance generally falls into one of three broad categories: (i) link break and repair; (ii) link break and subsequent network partition; and (iii) tree merge after a network partition.

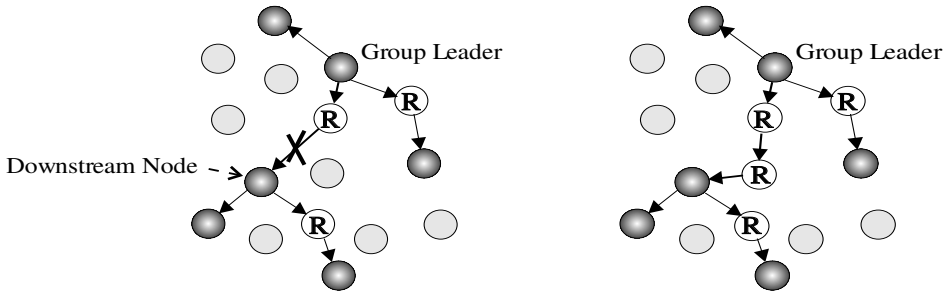
2.5.1. Repairing Link Breaks

Nodes determine that a link has broken in the same way as described in [11], whether or not the link is part of the multicast tree.

If the multicast tree has recently been used to send data packets, then a node on the tree must hear each of its next hops (except the node from which the packet was received) retransmit a data packet within `retransmit_time`, generally three times the propagation delay through a node. Because IP is a "best effort" network-layer protocol, a multicasting node does not need to hear each of its next hops retransmit each packet; it just must hear each next hop transmit *something* within that time frame. This special `retransmit_time` is used because waiting the full `hello_life` time period to detect a broken multicast tree link would often result in a large number of lost packets.

When a link break on the multicast tree occurs, the node *downstream* of the break is responsible for repairing the link. A node knows it is downstream of the break because it knows the direction of each of its next hops in relation to the multicast group leader. Only the downstream node should initiate the repair; if nodes on both sides of the break tried to repair the link, they might repair the link through different intermediate nodes, thus forming a loop. The downstream node initiates the repair by broadcasting a RREQ with the *Join* flag set and with a special extension included. This extension, called the Multicast Group Leader Extension, contains a *mgroup_hcnt* field, which is set equal to the node's current distance from the multicast group leader. In figure 4(a), the downstream node sets this field equal to two, since it is two hops away from the group leader. When this extension is included, only nodes that are *no farther* from the group leader can respond. This prevents nodes on the same side of the break as the downstream node from responding to the RREQ, which would form a loop. Because the two nodes are likely to still be close by, the downstream node can set the initial TTL value of the RREQ to be small, thereby allowing for a local repair and preventing the RREQ from being broadcast across the entire network.

Because the RREQ has the *Join* flag set, only a node on the multicast tree can respond. When such a node receives the RREQ with this extension field, it checks whether it is at least as close to the group leader as indicated by the *mgroup_hcnt* field. If so, and if its record of the group sequence number is at least as great as that contained in the RREQ, it can reply to the RREQ by unicasting a RREP back to the initiating node. RREP forwarding and subsequent route activation with the MACT message are handled as previously described. Figure 4(b) illustrates the multicast tree after the repair is completed.



(a) Link Break (b) Repaired Multicast Tree

Figure 4. Repair of Multicast Tree Branch.

Once the repair is finished, it is possible that the node which initiated the repair is now a new distance from the group leader. If this is the case, it must inform it downstream next hops of their new distance from the group leader. The node creates a MACT message, sets the *Update* flag, and sets the *hop_cnt* field equal to its distance from the group leader. It then multicasts this message to the multicast group. When the downstream nodes receive this message, they increment the *hop_cnt* value and then update their current distance from the group leader. If they are not leaf nodes, they in turn send this update message to their downstream next hops, and so on. Because the MACT is multicast, the node that is upstream of the multicasting node also receives the message. In this case, it notes that the packet came from its downstream link, and discards the message.

When a link break occurs, the node upstream of the break also notices the disconnection. It is possible that the tree branch will not be reconnected through that node. If this upstream node is not a group member, and if the loss of that link has made that node a leaf node, it sets a prune timer to wait for the repair. This prune timer should be longer than the route discovery period in order to allow time for the repair to be completed. The new leaf node may prune itself after the timer expires, if the next hop does not reactivate it (by sending it a MACT message).

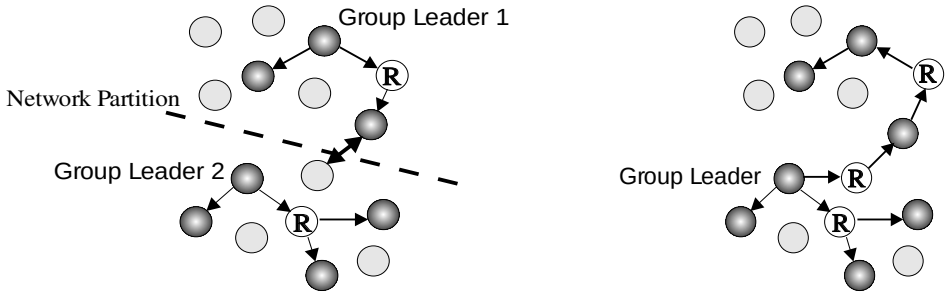
2.5.2. Network Partitions

If a node attempting to repair a broken tree link does not receive a RREP within the discovery period, it rebroadcasts its RREQ up to `rreq_retries` more times, using the same expanding ring search as indicated in section 2.3.1. If no response is received after this many attempts, the node must assume that the network has become partitioned and that the multicast tree cannot yet be repaired. If this is the case, the multicast tree partition that was downstream of the break is now left without a group leader.

If the node that was trying to repair the break is a multicast group member, then it becomes the new group leader. It broadcasts a GRPH message to announce the group leader change, and sets the *Update* flag in this message to indicate that the change has occurred.

If the node that was trying to repair the break is not a multicast group member, there are two possibilities. The first is that this node has only one downstream link. If this is the case, then the link loss has made this node a leaf on the tree, and so it can prune itself from the tree. It unicasts its next hop a prune message, and then deletes all the group information from its MRT. When the next hop receives the prune message, it deletes the sending node's next hop information from its MRT, and notes that the message came from its upstream link. It is then in the same position as the previous node. If it is a group member, it becomes the new group leader. Otherwise, if it has only one downstream next hop link, it prunes itself from the tree along this link. This process continues until a multicast group member is reached. This node becomes the new group leader.

The second possibility is that the node that was trying to repair the link has multiple downstream branches. In this case, it cannot prune itself from the tree, because doing so would disconnect the tree. It instead selects any one of its downstream links, and unicasts that next hop a MACT message with set *Grpldr* flag. This flag indicates that the next group member to receive this message should become the new group leader. After unicasting this message, it changes the direction associated with this next hop in its MRT so that the direction



(a) Partitioned Network before the Repair (b) Reconnected Network

Figure 5. Merge of Two Components of Multicast Tree.

is now upstream. If the next hop to receive the MACT is a group member, it becomes the new group leader. Otherwise, it in turn chooses one of its downstream links, and sends that next hop the MACT message with set *Grpldr* flag. It also updates the direction associated with that next hop to be upstream. This process ends once a multicast group member is reached.

Once the new group leader is determined, it broadcasts a GRPH message with set *Update* flag and incremented group sequence number to announce its new status as group leader.

2.5.3. Merging Two Disjoint Trees

Once a network partition has occurred, there are two group leaders for the same multicast group, each of which periodically broadcasts a GRPH message. If the two network partitions come back into contact with each other, group members learn of this occurrence through the reception of a GRPH message that has information about a new group leader. The two partitions of the multicast tree must then be reconnected. The only node which can initiate the repair of the tree is the multicast group leader with the lower IP address. This distinction is made because if more than one node tried to repair the tree, it is likely that it would be repaired through different intermediate nodes, and thus form a loop.

When the group leader with the lower IP address (GL_1) receives the GRPH, it creates a RREQ with set *Join* and *Repair* flags and *unicasts* this message to the other group leader (GL_2), using the node from which it received the GRPH as the next hop. As the RREQ travels to GL_2, nodes process the packet as they would a regular RREQ with the following exception. If a node that is a member of GL_2's tree receives the RREQ, it forwards the RREQ to GL_2 along its upstream multicast tree link. This prevents the formation of routing loops once the RREP is sent.

When GL_2 receives the RREQ, it notes the set *Repair* flag, and creates a RREP to send back to GL_1. It sets the *Repair* flag of this RREP, and then unicasts the RREP back to GL_1. It also updates the multicast group sequence number by taking the larger of its record of the group sequence number and that contained in the RREQ, and incrementing this value by one. The next time GL_2 broadcasts a GRPH message, it includes this new sequence number value, and sets the *Update* flag to indicate a group leader change has occurred.

As the RREP travels back to GL_1, nodes that receive the RREP create the next hop entries and *activate* these entries immediately. Because the RREQ was unicast, there is only one potential tree branch being added to the tree, and so a MACT message does not need to be sent. Hence the next hop entry can be activated without delay. If a node that is on the multicast tree of GL_1 receives the RREP message, it updates its group leader information for that multicast group to reflect GL_2 as the new group leader. This node forwards the RREP along its upstream tree link towards GL_1 to prevent routing loops. It then changes the direction of that link to downstream, and marks the link from which the packet arrived as upstream. This link direction change occurs because the new group leader is in a different direction than the previous one. Once GL_1 receives the RREP, it notes that it is no longer the group leader, makes the link addition or direction change, and the tree merge is complete. Figure 5 shows an example of a tree repaired in this manner.

3. Data Packet Forwarding

Data packets destined for the multicast group are transmitted as broadcast traffic, unless there is support for multicast at layer 2. When a node receives a multicast data packet, it checks whether it is a part of the multicast tree for that multicast group. If not, it discards the packet. If it is a member of the multicast tree, it then determines whether it has already received that packet. Nodes on the multicast tree keep a record of the source IP address, fragment id, and fragment offset of the multicast data packets they receive. If this source IP address/fragment id/offset combination is already represented in their records, they discard the packet. Otherwise, they create a new entry to represent the packet. In this way, if the node later receives the same data packet transmitted by another next hop, it knows not to reprocess the packet. If the node has not already received the data packet, the node processes the packet if it is a member of the multicast group to which the packet is addressed. Then, if the node is on the multicast tree for that group, it forwards (by broadcast or multicast) the packet to its next hops.

4. Simulations

The transmission range R_{xmit} is a key parameter in the interconnection pattern of a network. Its value affects a wide range of results, including:

- the neighbor degree of network nodes,
- the throughput,
- the probability of collision,
- the contention for channel access,
- battery lifetime of the transmitting node,
- average number of hops, and thus the delay, for message transmission,
- and the impact of transmissions on neighboring nodes.

In the following simulations, the effect of the transmission range is studied on different network topologies.

To investigate the effect of the transmission range on the AODV multicast protocol, a variety of results are examined. First, the packet delivery ratio is calculated by taking the number of data packets received, divided by the number of data packets transmitted. Control packets are not counted for the purposes of this calculation. The packet delivery ratio is a key indicator of how well the protocol performs under the given conditions. Since each data packet must be received by every member of the multicast group, the packet delivery ratio is then divided by the number of group members to yield the normalized overall packet delivery ratio.

To understand the packet delivery ratio results, various other results are examined. The average distance from a multicast group member to the group leader is directly affected by the transmission range. When transmissions have relatively small range, the path length to the multicast group leader is much longer. Longer path lengths lead to a higher probability of a packet collision before the packet reaches its destination, and also a higher probability of tree link breaks. Both of these events result in a lower packet delivery ratio. Since the amount of control message overhead is directly related to the number of breaks in the multicast tree, a shorter transmission radius is likely to result in a greater amount of control traffic.

A large transmission radius increases the average number of neighbors per node. This leads to shorter path lengths to reach multicast group members, but a greater number of nodes receive each data packet broadcast. Consequently, there is an increase in battery utilization at individual nodes, as well as an increase in the likelihood of data packet collisions.

4.1. Simulation Environment

The simulations were performed using the GloMoSim Network Simulator developed at UCLA [1]. This simulator models the OSI network architecture and includes models for IP and UDP. The simulator also allows for network node mobility, thereby enabling simulation of mobile ad hoc networks.

The MAC layer protocol used in the simulations is the IEEE standard 802.11 Distributed Coordination Function (DCF) [4]. This standard uses Request-To-Send (RTS) and Clear-To-Send (CTS) control packets for unicast data transmissions between neighboring nodes. A node wishing to unicast a data packet to its neighbor broadcasts a short RTS control packet. When its neighbor receives the packet, it responds with a CTS packet. Once the node receives the CTS, it transmits the data packet. After receiving this data packet, the neighbor then sends an acknowledgment (ACK) to the sender of the data packet, signifying reception of the packet. The use of the RTS-CTS

Parameter Name	Meaning	Value
allowed_hello_loss	# of Allowed Hello Losses	2
group_hello_interval	Frequency of Group Hello Broadcasts	5 sec
hello_interval	Frequency of Hello or Other Broadcasts	1 sec
hello_life	Maximum Time Allowed Between Hello Pkt Receptions	3 sec
pkt_id_save	Time to Buffer Data Packet Identifier	3 sec
prune_timeout	Time to Wait to Receive a MACT before Prune	3 sec
retransmit_time	Time to Wait for Data Packet Retransmissions	750 msec
rev_route_life	Time to Keep Reverse Route Entries	3 sec
rreq_retries	Max # of RREQ Retransmissions	2
route_discovery_timeout	Max Time to Wait for a RREP	1 sec

Table 1
Simulated Parameter Values.

control packets reduces the potential for the hidden-terminal problem [17]. Broadcast data packets and RTS control packets are sent using the unslotted Carrier Sense Multiple Access protocol with Collision Avoidance (CSMA/CA) [4]. When a node wishes to broadcast a packet, it first senses the channel. If it does not detect an on-going transmission, it sets a short timer and then re-senses the channel once the timer expires. If the channel is still idle, it broadcasts its packet. On the other hand, if it does detect a transmission, it calculates a backoff time and then waits this amount of time before reattempting the transmission.

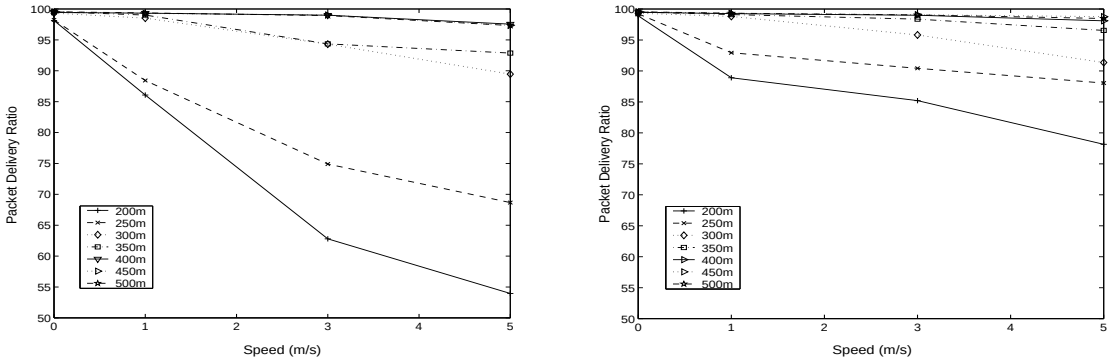
The bandwidth for the simulations is 2 Mb/sec. The propagation model used is the free space model [14] with threshold cutoff included in the GloMoSim simulation package. The free space model has a power signal attenuation of $1/d^2$, where d is the distance between nodes. The radio model used also has capture capability, where it can lock on to a strong signal during interference, and still receive the packet. Other interfering packets with weaker signal strength are dropped.

Node movement is modeled by the random direction mobility model [15]. In this model, nodes are initially placed randomly within the network simulation area. Each node chooses a random direction between 0 and 360 degrees, and then selects a destination on the border of the network area in that direction of travel. The node then moves to that destination at its pre-assigned speed (between 0 and 5 m/s). When the node reaches its destination, it rests for 30 seconds. It then chooses a new direction, this time between 0 and 180 degrees. The degree selected is adjusted relative to the boundary on which the node is located. The node then resumes movement. Except for the 0 m/s mobility scenario, this movement model causes continual changes in the network topology.

Each simulation simulates 300 seconds and models a network of 50 nodes. During each simulation, there is one multicast group which contains ten members. Nodes join the multicast group at the beginning of the simulation. Once all the nodes have joined the group and the tree is formed, data transmission begins. Data packets are sent by one of the group members at a constant rate of eight packets per second throughout the duration of the simulation. Each data packet is 64 bytes.

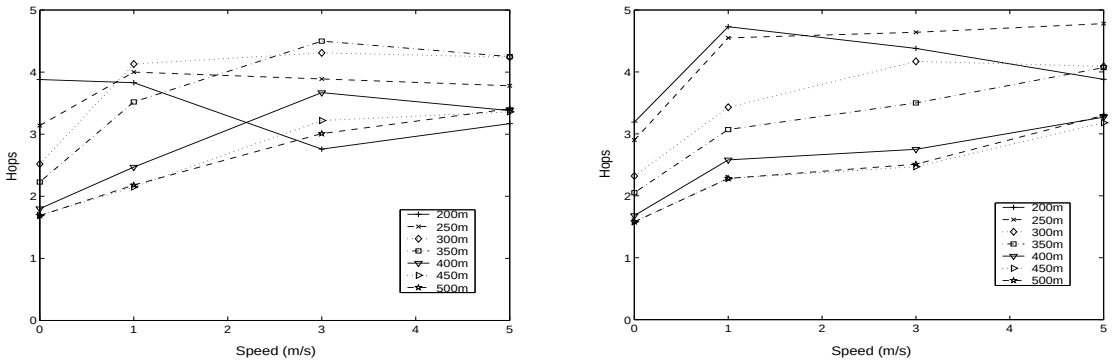
When a node receives a multicast data packet and is a member of that multicast group, it sends the packet to the application layer for processing and increments its count of the number of data packets it has received. It then rebroadcasts the packet. If the node is not a member of the multicast group but is on the multicast tree, it simply rebroadcasts the packet to allow reception of the packet by its next hops. In order to achieve 100% packet delivery, every member of the multicast group must receive the data packet. No layer 2 support for multicast is assumed.

AODV does not guarantee packet delivery; however, it does find good routes for IP's best-effort delivery. Because data packets are not buffered for retransmission, losses can occur. If a collision involving a data packet occurs at a node and the packet cannot be captured, the packet is lost. Typically, if a link break occurs on the multicast tree, data packets are lost before that break is noticed and while the link is being repaired. Hence, it is essential to take steps to monitor multicast tree links and provide for immediate repair so that link breaks result in minimum packet loss. Unfortu-



(a) 1000m×1000m Network (b) 1500m×300m Network

Figure 6. Packet Delivery Ratio.



(a) 1000m×1000m Network (b) 1500m×300m Network

Figure 7. Average Number of Hops to Multicast Group Leader.

nately, because multicast data packets are broadcast, MAC layer feedback provides no notification of broken links. For this reason, AODV provides a method of monitoring these active links, as described in section 2.5.1.

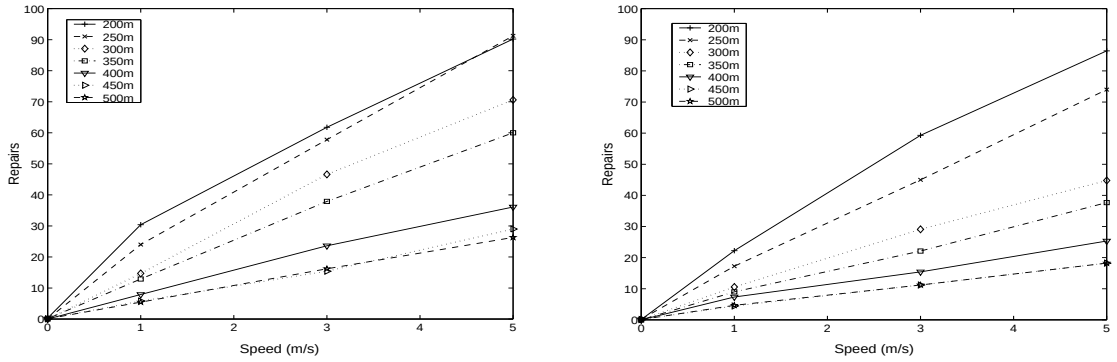
Table 1 shows the essential parameter values for these simulations. Note that the expanding ring search was not used in the simulations.

There are two different network roaming areas simulated. The first is a 1000m×1000m area, and the second is a 1500m×300m area. These two size areas have been used in several other ad hoc network simulations [2,3,8]. They are modeled here to determine the effect of transmission range in them.

In order to explore the effects of transmission range, seven different ranges, from 200m to 500m, are studied. Shorter than 200m, network connectivity is too sparse for an accurate comparison, as network partitions occur. The results of the 450m and 500m simulations are similar enough to be able to extrapolate the effects of further increasing the transmission range. Each transmission radius/speed combination was run for ten different initial network configurations.

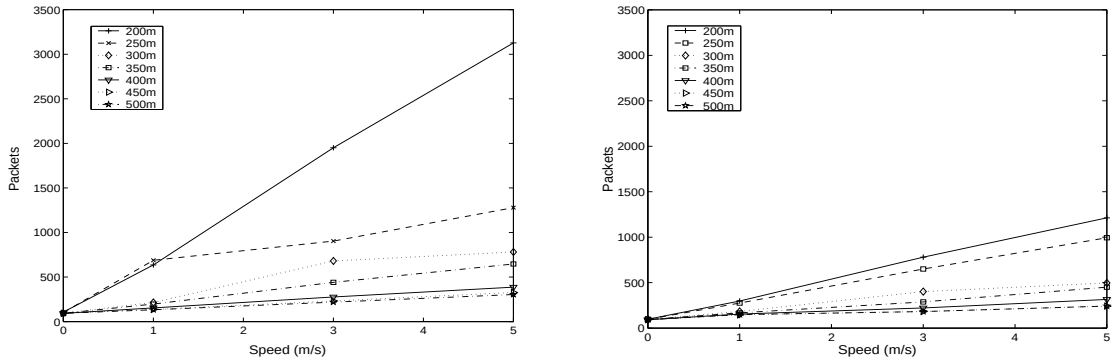
4.2. Results

First, consider the achieved packet delivery ratio. Figure 6(a) shows the results in the 1000m×1000m area for the seven different transmission ranges modeled, and illustrates how the packet delivery ratio is affected by the changing speed of the network nodes. Figure 6(b) shows the same results for the 1500m×300m simulations. For both network configurations, an increase in range yields an increase in the packet delivery ratio, or the number of data packets received by multicast group members. For the 1000m×1000m network, the increase in speed results in a decrease in the packet delivery ratio for the more sparsely connected networks. In the 1500m×300m network, the increase in speed has a smaller effect on the overall packet delivery.



(a) 1000m×1000m Network (b) 1500m×300m Network

Figure 8. Multicast Tree Repairs.



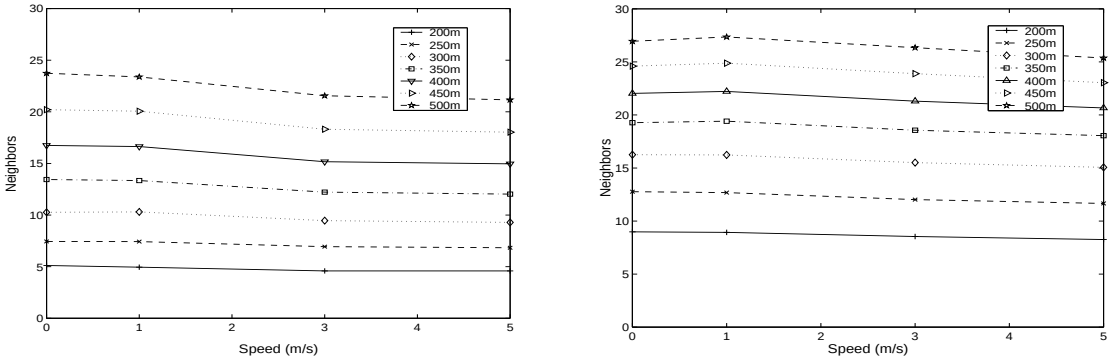
(a) 1000m×1000m Network (b) 1500m×300m Network

Figure 9. Control Packet Overhead.

To understand why the packet delivery ratio is affected by the transmission range, it is necessary to investigate how the transmission range affects other aspects of the network. Figures 7(a) and 7(b) illustrate the effect of the transmission range on the average distance of a multicast group member to its group leader. The distance to the group leader gives an indication of the size of the tree and how many hops data packets must traverse between destinations. The figures indicate that for smaller transmission ranges, the average number of hops to the group leader is greater. A larger distance to the group leader results in greater potential for packet collisions, as well as a higher chance of link breaks. Thus, a greater transmission range results in fewer hops on the multicast tree, which produces a better packet delivery ratio. It is interesting to note that there is not a notable difference in results between the two network sizes.

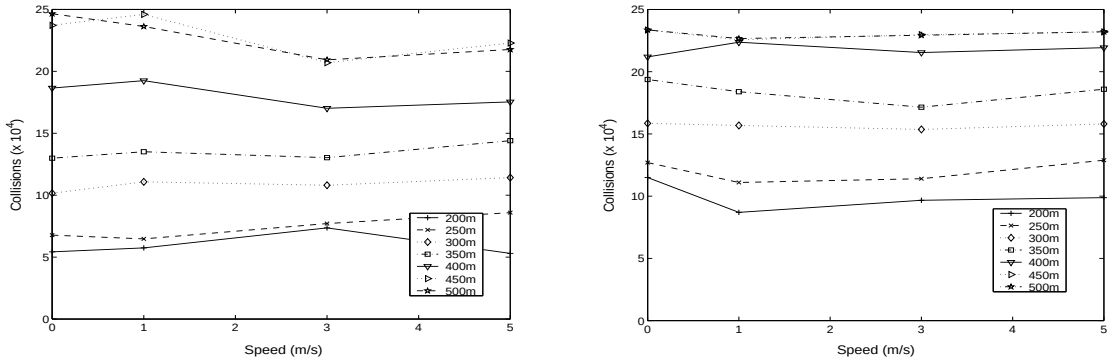
Because the average path length to the group leader is inversely proportional to the transmission radius, the number of repairs to the multicast tree is also likely to be inversely proportional. This is verified in figures 8(a) and 8(b). These figures show the average number of repairs needed to fix broken multicast tree links during the simulation. As expected, the number of repairs increases for increasing speed. Also as expected, the greatest transmission range requires the fewest number of repairs, again resulting in a better packet delivery ratio for these networks.

The amount of control overhead generated during the simulation directly corresponds to the number of repairs to the multicast tree. Figures 9(a) and 9(b) show the number of control packets produced during each of the simulations. The number of control packets represented here is found by summing the number of RREQ, RREP, MACT, and GRPH packets initiated. The figures show a reduction in the number of control messages as the transmission range is increased. For zero mobility, there are no repairs needed to the multicast tree. Figures 9(a) and 9(b) indicate that the number of control messages needed to initialize the multicast tree is approximately constant at the different trans-



(a) 1000m×1000m Network (b) 1500m×300m Network

Figure 10. Average Number of Neighbors.



(a) 1000m×1000m Network (b) 1500m×300m Network

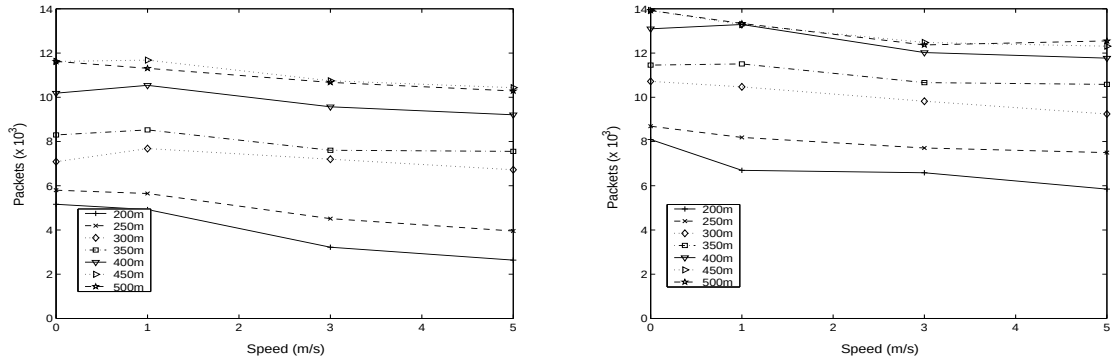
Figure 11. Collisions.

mission ranges. The variation in control overhead becomes more significant once the nodes begin moving. As nodes travel more quickly, there are more breaks and repairs to the multicast tree, and hence there are more control packets generated. Because there are fewer link breaks for the longer transmission ranges, there are subsequently fewer control packets generated in these simulations as well. The 200m transmission range in the 1000m×1000m network particularly suffers from the increase in mobility. The network topology and low connectivity in this network configuration often result in multiple attempts per repair to re-establish tree link connections.

Having examined only these results, it appears that increasing the transmission radius has a uniformly positive effect on the network. A larger transmission radius results in better packet delivery ratio, fewer link breaks in the multicast tree, and less control overhead. However, it is also necessary to examine the impact that increasing the neighborhood size has on the network. Increasing the transmission range also increases the number of neighboring nodes affected by each transmission. Figures 10(a) and 10(b) indicate the number of neighbors per node at the varying transmission ranges. For the purpose of the figure, two nodes are considered to be neighbors if the distance between them is less than or equal to the given transmission radius. As would be expected, the number of neighbors per node increases with increasing transmission range.

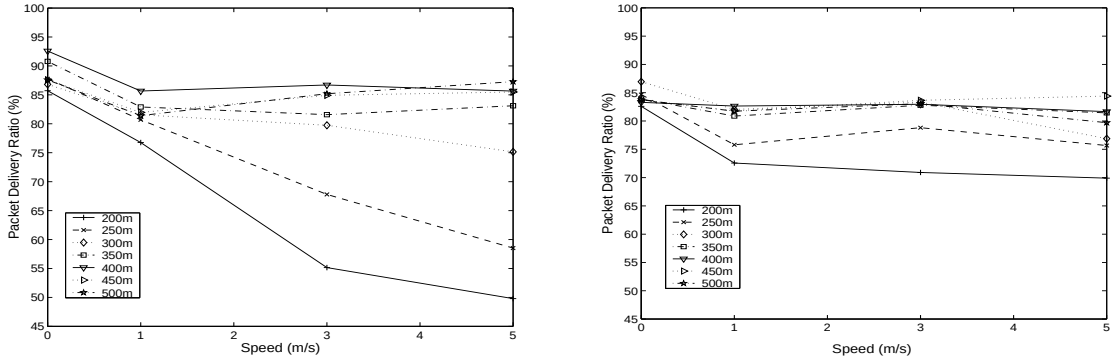
One of the primary effects of increasing the size of the neighborhood is the increase in the number of packet collisions. Figures 11(a) and 11(b) illustrate the total number of packet collisions in the networks. The figures show that the number of collisions rapidly increases as the transmission range grows. The number of collisions with 450m and 500m transmission ranges is nearly five times that of the 200m transmission range network.

To further explore the effect of the different neighborhood sizes, the number of multicast data packets received at nodes which are not group members is examined in figures 12(a) and 12(b). At a packet transmission frequency of eight packets per second and simulation length of 300 seconds, just under 2200 data packets



(a) 1000m×1000m Network (b) 1500m×300m Network

Figure 12. Multicast Data Packets Received By Non-Group Members.



(a) 1000m×1000m Network (b) 1500m×300m Network

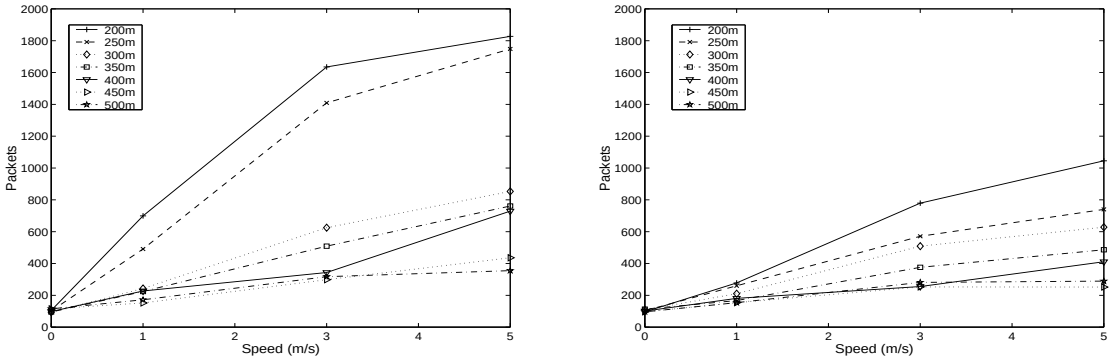
Figure 13. Packet Delivery Ratio for Increased Data Rate.

are initiated during the simulation. The figures indicate that for a transmission radius of only 200m, non-group member nodes receive, on average, each multicast data packet approximately twice. However, for the highest mobility and transmission radius combination, nodes receive each data packet approximately six times. Such a redundancy in packet reception is likely to have quite a negative effect on a node's battery lifetime, as the node will spend a large percentage of its battery power processing unnecessary packets. The higher nodal density engenders additional contention for slotted MAC schemes. This causes more collisions during the contention period, resulting in increased queuing delays as nodes are forced to wait longer periods of time between packet transmissions.

The results presented so far are based on one source with a moderate sending rate (eight packets per second). To determine the interaction between transmission range and network traffic, a second set of experiments was performed with the number of sources increased to two and with each source sending 20 packets per second. The data packet size in these simulations is 512 bytes, as opposed to the 64 byte packets in the previous experiments.

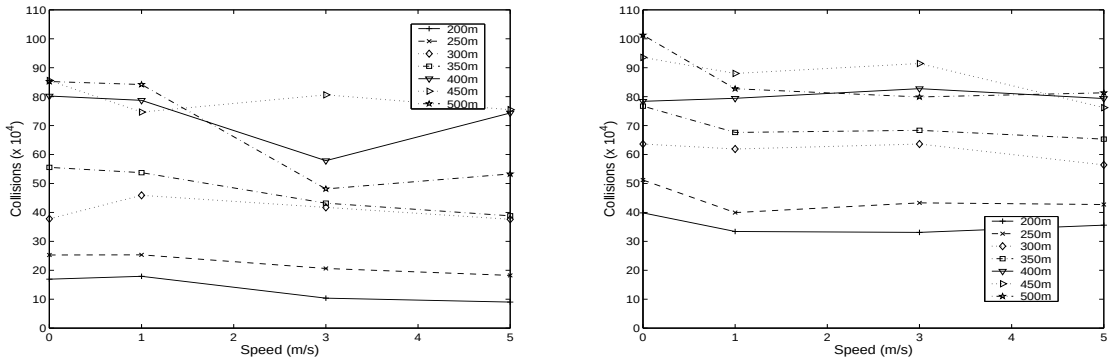
The packet delivery ratio for this set of experiments is shown in figure 13. The results here differ from the packet delivery ratio for the lower sending rate, shown in figure 6. Here, it is no longer the case that the longest transmission range produces the greatest packet delivery ratio. The nodes in these networks are not able to deliver as many data packets due to the increased contention for channel access and the increased likelihood of collisions. The transmission range of 400m in the 1000m×1000m networks and 400-450m in the 1500m×300m network results in more delivered data packets than does the 500m transmission range. The ranges of 200m and 250m still produce the lowest packet delivery ratio due to the lower connectivity in these networks.

The control packet overhead for the increased data rate is given in figure 14. These graphs do not vary significantly from the results shown in figure 9. The



(a) 1000m×1000m Network (b) 1500m×300m Network

Figure 14. Control Packet Overhead for Increased Data Rate.



(a) 1000m×1000m Network (b) 1500m×300m Network

Figure 15. Number of Collisions for Increased Data Rate.

control traffic in the 200 and 250m networks still dominates in both network sizes.

Finally, the number of collisions is shown in figure 15. Here, the results vary slightly from those in figure 11 for the large transmission range. Except for the static networks, the number of collisions is greatest in the 450m transmission range scenarios.

5. Future Work

Our work on multicast can be extended in many directions. We would like to investigate larger node populations, higher rates of mobility, and the effects of different parameter settings. We would also like to investigate algorithms for tunable power control. It is possible that certain broadcast multicast transmissions should be replaced by unicast (tunneled multicast) to the appropriate neighbors in the multicast tree. We would like to make the relevant comparisons and use the results for possible revisions to the way that multicast datagrams are handled. This information could be incorporated into the AODV protocol operation for multicast routing.

Other multicast algorithms for ad hoc networks have been proposed [6,7,9]. We would like to compare the power consumption vs. packet delivery ratio observed for AODV against the performance of the other algorithms. Feeney [5] has already made similar measurements for unicast algorithms, and her techniques should be easily adaptable to provide the necessary comparison data for multicast algorithms.

Power control represents an interesting area of research because the power level used at a mobile node has the effect of dynamically creating or destroying links. This dynamic link control can happen even without any node movement. Thus, there are two interacting mechanisms for changing the topology of the network. It seems likely that, at certain times, a mobile node should be able to beneficially reduce its power consumption if it has numerous links to nodes in its neighborhood. In this way, it would be able to conserve power. Alternatively, when the node has few other nodes in its neighborhood

and has a large percentage of its battery power remaining, it might be useful for it to increase its transmission range so that better network connectivity may be established. Ramanathan and Rosales-Hain have made steps towards this goal by developing a mechanism for dynamically adjusting the transmitter power at individual nodes in order to optimize the overall network topology [13]. Their scheme is able to adapt depending on the connectivity or bi-connectivity constraints.

The cases where battery power is diminished but more network connectivity is needed are not handled so easily. It is possible that AODV would benefit from acquiring information about neighborhoods once removed. Such information may be useful for power control algorithms. A more ambitious approach would be an attempt to find global information about links that may become useful to unreachable nodes. Perhaps exploratory, high-power probes should be transmitted occasionally, to get first-hand information about how the local neighborhood connectivity could be improved by the use of higher power for data transmissions.

6. Summary and Conclusions

In this paper, we have presented the AODV multicast routing algorithm, and have shown the effects of various transmission ranges and mobility rates on packet delivery and the number of repairs needed for maintenance of the multicast delivery tree. AODV handles the transmission of multicast and broadcast data in a natural way, maintaining compatibility with traditional IP route table mechanisms and the needs of unicast packet routing. The AODV routing protocol is able to provide multicast communication between group members in a variety of network configurations and mobility scenarios. By building bi-directional multicast trees between group members, AODV quickly connects group members, and is able to maintain these connections throughout the lifetime of the multicast group.

AODV's multicast routing algorithm is a straightforward extension to the algorithm used to discover unicast routes. The basic broadcast RREQ discovery mechanism, with unicast RREP messages, is adapted for use with multicast routing. Since the multicast IP address is not allocated to any specific network node, the responsibility for maintaining the sequence number for multicast routes has to be assigned to a distinguished node called the *group leader*. Aside from this, the main difference introduced by multicast routing is the need for maintaining multiple next hops per multicast route entry instead of just one next hop, as is the case for unicast routing. Having multiple next hops also creates new opportunities for route loops; however, this possibility is eliminated through the use of a new message type called the Multicast Activation (MACT) message. This message enables just one of several possible multicast tree paths; since only one path is enabled, route loops remain impossible even for multicast routing.

The transmission range and network size are key determinants of AODV's multicast performance. Increasing the transmission range has many benefits. The number of links on the multicast tree is reduced, resulting in fewer tree links which need to be maintained. Each multicast tree link repair requires control message overhead. Reducing the number of repairs has the advantage of also decreasing the amount of control overhead. For unloaded networks, the packet delivery ratio increases for longer transmission ranges due to the reduction in the number of hops between group members and the longer-lived tree links.

Despite these advantages, a large transmission range also causes more network nodes to be affected by multicast data transmissions, even when the nodes do not need to receive these packets. A large transmission radius therefore drains the battery not only of the transmitting node, but also of neighboring nodes within the source's transmission range. Worse, a large transmission radius reduces the effective bandwidth available to the individual nodes and increases the number of collisions seen throughout the network, as more nodes are competing for and utilizing the same network bandwidth. This increase in the number of collisions causes a reduction in the packet delivery ratio for traffic patterns that significantly load the wireless medium. Perhaps most significantly, increasing the transmission range places disproportionately greater demands on the power requirements of the (typically battery-powered) mobile nodes. Thus, it is crucial for the consumer market to find ways to minimize power consumption.

We hope that this exploratory work on the relationship between transmission ranges and multicast routing performance will lead the way towards improving the reliability of best-effort multicast packet delivery. We conclude that the transmission range should be adjusted

to meet the targeted throughput while minimizing battery power consumption. Our work shows that there are opportunities for power savings when nodes can get the same (or even better) performance by reducing the power drain caused by unnecessarily high transmission ranges.

References

[1] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla. GlomoSim: A Scalable Network Simulation Environment. Technical Report CSD Technical Report, #990027, UCLA, 1997.

[2] J. Broch, D. A. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. *Proceedings of the 4^{th} Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 85–97, Dallas, Texas, October 1998.

[3] S. R. Das, C. E. Perkins, and E. M. Royer. Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks. *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pages 3–12, Tel Aviv, Israel, March 2000.

[4] I. S. Department. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Standard 802.11–1997*, 1994.

[5] L. Feeney. An Energy Consumption Model of Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks. http://www.ietf.org/proceedings/99jul/slides/manet-feeney-99jul.pdf, July 1999.

[6] J. J. Garcia-Luna-Aceves and E. L. Madruga. A Multicast Routing Protocol for Ad-Hoc Networks. *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pages 784–792, New York, NY, March 1999.

[7] L. Ji and M. S. Corson. A Lightweight Adaptive Multicast Algorithm. *Proceedings of IEEE GLOBECOM*, pages 1036–1042, Sydney, Australia, December 1998.

[8] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks. *Proceedings of the 5^{th} ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 195–206, Seattle, WA, August 1999.

[9] S.-J. Lee, M. Gerla, and C.-C. Chiang. On-Demand Multicast Routing Protocol. *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1298–1304, New Orleans, LA, September 1999.

[10] V. D. Park and M. S. Corson. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pages 1405–1413, Kobe, Japan, April 1997.

[11] C. E. Perkins and E. M. Royer. Ad-hoc On-Demand Distance Vector Routing. *Proceedings of the 2^{nd} IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, New Orleans, LA, February 1999.

[12] C. E. Perkins, E. M. Royer, and S. R. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. *IETF Internet Draft, draft-ietf-manet-aodv-06.txt*, July 2000. (Work in Progress).

[13] R. Ramanathan and R. Rosales-Hain. Topology Control of Multihop Wireless Networks using Transmit Power Adjustment. *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pages 404–413, Tel Aviv, Israel, March 2000.

[14] T. S. Rappaport. *Wireless Communications, Principles & Practices*, chapter 3, pages 70–74. Prentice Hall, 1996.

[15] E. M. Royer, P. M. Melliar-Smith, and L. E. Moser. An Analysis of the Optimum Node Density for Ad hoc Mobile Networks. Submitted for publication.

[16] E. M. Royer and C. E. Perkins. Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Protocol. *Proceedings of the 5^{th} ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 207–218, Seattle, WA, August 1999.

[17] F. A. Tobagi and L. Kleinrock. Packet Switching in Radio Channels: Part-II - The Hidden Terminal Problem in Carrier Sense MultipleAccess Models and the BusyTone Solution. *IEEE Transactions on Communications*, 23(12):1417–1433–20, December 1975.

Elizabeth M. Royer received her B.S. degrees in both Computer Science and Applied Mathematics from Florida State University in April 1996. She obtained her M.S. degree in Electrical and Computer Engineering in December of 1997 from the University of California, Santa Barbara. She is currently completing her Ph.D. in Computer Engineering at the University of California, Santa Barbara. At UCSB, Elizabeth works in the Computer Networking and Distributed Systems Laboratory, where her research interests focus on mobile wireless networks. These interests include unicast routing, multicast routing, hierarchical routing, QoS management, service location, and auto configuration. Elizabeth is the recipient of a National Science Foundation Graduate Fellowship and a University of California Doctoral Scholars Fellowship. She an active participant in the IETF, and is a student member of the IEEE and ACM.

Charles E. Perkins is a Research Fellow at Nokia Research Center, investigating mobile wireless networking and dynamic configuration protocols. He is the editor for several ACM and IEEE journals for areas related to wireless networking. He is serving as document editor for the mobile-IP working group of the Internet Engineering Task Force (IETF), and is author or co-author

of standards-track documents in the mobileip, svrloc, dhc (Dynamic Host Configuration) and IPng working groups. Charles has served on the Internet Architecture Board (IAB) of the IETF and on various committees for the National Research Council. He is also associate editor for Mobile Communications and Computing Review, the official publication of ACM SIGMOBILE, and is on the editorial staff for IEEE Internet Computing magazine. Charles has authored a book on Mobile IP, and has published a number of papers and award winning articles in the areas of mobile networking, ad hoc networking, route optimization for mobile networking, resource discovery, and automatic configuration for mobile computers.