The Design, Implementation, and Performance Evaluation of the On-Demand Multicast Routing Protocol in Multihop Wireless Networks*

Sang Ho Bae, Sung-Ju Lee, William Su, and Mario Gerla

Computer Science Department
University of California
Los Angeles, CA 90095-1596
{sbae, sjlee, wsu, gerla}@cs.ucla.edu

Abstract

Multicasting has emerged as one of the most focused areas in the field of networking. As the technology and popularity of Internet grow, applications, such as video conferencing, that require multicast feature are becoming more widespread. Another interesting recent development has been the emergence of dynamically reconfigurable wireless ad hoc networks to interconnect mobile users for applications ranging from disaster recovery to distributed collaborative computing. In this article we describe the On-Demand Multicast Routing Protocol for mobile ad hoc networks. ODMRP is a mesh-based, rather than a conventional tree-based, multicast scheme and uses a forwarding group concept (only a subset of nodes forwards the multicast packets via scoped flooding). It applies on-demand procedures to dynamically build routes and maintain multicast group membership. We also describe our implementation of the protocol in a real laptop testbed.

1 Introduction

An ad hoc network [1, 2] is a dynamically reconfigurable wireless network with no fixed infrastructure or central administration. Due to the limited radio propagation range of wireless devices, routes are often multihop. Applications such as disaster recovery, crowd control, search and rescue, and automated battlefields are typical examples of where ad hoc networks are deployed. 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 routing and multicasting in ad hoc networks extremely challenging.

In a typical ad hoc environment, network hosts work in groups to carry out a given task. Hence, multicast plays an important role in ad hoc networks. Multicast protocols used in static networks — for example, Distance Vector Multicast Routing Protocol (DVMRP) [3], Multicast Open Shortest Path First (MOSPF) [4], Core Based Trees (CBT) [5], and Protocol Independent Multicast (PIM) [6] — do not perform well in ad hoc networks because multicast tree structures are fragile and must be readjusted as connectivity changes. Furthermore, multicast trees usually require a global routing substructure such as link state or distance vector. The frequent exchange of routing vectors or link state tables, triggered by continuous topology changes, yields excessive channel and processing overhead.

To overcome these limitations, we have developed the On-Demand Multicast Routing Protocol (ODMRP) [7, 8, 9]. ODMRP applies *on-demand* routing techniques to avoid channel overhead and improve scalability. It uses the concept of *forwarding group* [10], a set of nodes responsible for forwarding multicast data on shortest paths

---

*This work was funded in part by the Defense Advanced Research Projects Agency (DARPA) DOMAINS, Intel Corporation, and National Science Foundation (NSF).

between any member pairs, to build a forwarding *mesh* for each multicast group. By maintaining and using a mesh instead of a tree, the drawbacks of multicast trees in mobile wireless networks (e.g., intermittent connectivity, traffic concentration, frequent tree reconfiguration, non-shortest path in a shared tree, etc.) are avoided. A *soft-state* approach is taken in ODMRP to maintain multicast group members. No explicit control message is required to leave the group. We believe the reduction of channel/storage overhead and the relaxed connectivity make ODMRP more scalable for large networks and more stable for mobile wireless networks.

The performance of ODMRP has been tested using a detailed simulator [7, 9]. The results obtained from simulation led us to take the next step in development: implement ODMRP in a real testbed to validate and fine tune the protocol. No prior work exists in building a *multicast* protocol in an ad hoc network testbed. Our implementation utilizes the multicast extension built into the Linux kernel. A wireless mobile testbed consisting of laptops with the Linux operating system has been organized to test the implementation.

A few other multicasting protocols have been recently proposed for ad hoc networks [11, 12, 13, 10, 14, 15, 16, 17, 18, 19, 20, 21, 22]. The Reservation-Based Multicast (RBM) routing protocol [14] builds a core (or a Rendezvous Point) based tree for each multicast group. RBM is a combination of multicast, resource reservation, and admission control protocol where users specify requirements and constraints. The Lightweight Adaptive Multicast (LAM) algorithm [17] is a group shared tree protocol that does not require timer-based messaging. Similar to other core-based protocols, it suffers from disadvantages of traffic concentration and vulnerability of the core. The Adhoc Multicast Routing Protocol (AMRoute) [11] is also a shared-tree protocol which allows dynamic core migration based on group membership and network configuration. The Ad hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS) [22] builds a shared-tree to deliver multicast data. Each node in the multicast session is assigned an ID number and it adapts to connectivity changes by utilizing the ID numbers. A multicast extension of Ad Hoc On Demand Distance Vector (AODV) routing protocol has been newly proposed in [20]. Its uniqueness stems from the use of a destination sequence number for each multicast entry. The sequence number is generated by the multicast grouphead to prevent loops and to discard stale routes. Similar to ODMRP, the Core-Assisted Mesh Protocol (CAMP) [15] uses a mesh. However, a conventional routing infrastructure based on enhanced distance vector algorithm, such as Wireless Routing Protocol (WRP) [23], is required for CAMP to operate. Core nodes are used to limit the traffic required when a node joins a multicast group.

The rest of this article is organized as follows. We describe the design of ODMRP, followed by the protocol implementation description. Performance evaluation results and analyses are presented, and concluding remarks are made.

2 ODMRP Illustrated

2.1 Multicast Route and Membership Maintenance

In ODMRP, group membership and multicast routes are established and updated by the source *on demand*. Similar to on-demand unicast routing protocols, a request phase and a reply phase constitute the protocol (see Figure 1). While a multicast source has packets to send, it floods a member advertising packet with data payload attached. This packet, called JOIN DATA, is periodically broadcast to the entire network to refresh the membership information and update the routes as follows. When a node receives a non-duplicate JOIN DATA, it stores the upstream node ID (i.e., backward learning) into the routing table and rebroadcasts the packet. When the JOIN DATA packet reaches a multicast receiver, the receiver creates and broadcasts a JOIN TABLE to its neighbors. When a node receives a JOIN TABLE, it checks if the next node ID of one of the entries matches its own ID. If it does, the node realizes that it is on the path to the source and thus is part of the forwarding group. It then sets the `FG_FLAG` (Forwarding Group Flag) and broadcasts its own JOIN TABLE built upon matched entries. The JOIN TABLE is thus propagated by each forwarding group member until it reaches the multicast source via the shortest path. This process constructs (or updates) the routes from sources to receivers and builds a mesh of nodes, the *forwarding group*.

We have visualized the forwarding group concept in Figure 2. The forwarding group is a set of nodes which is in charge of forwarding multicast packets. It supports shortest paths between any member pairs. All nodes inside the "bubble" (multicast members and forwarding group nodes) forward multicast data packets. Note that

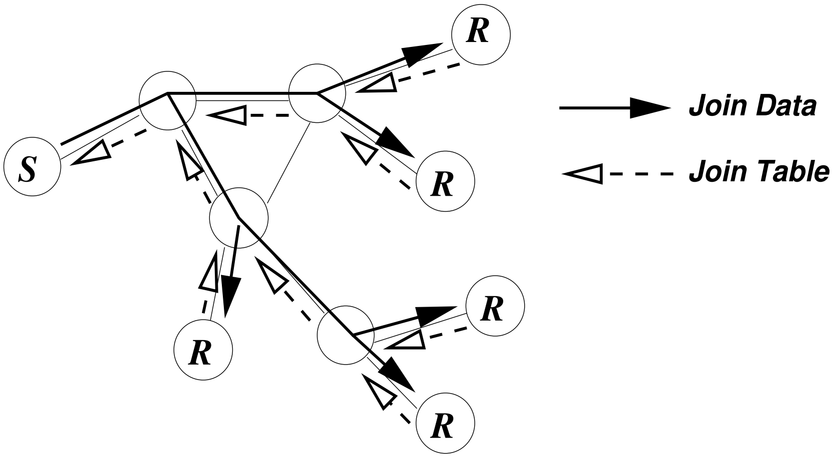


Figure 1: On-Demand Procedure for Membership Setup and Maintenance.

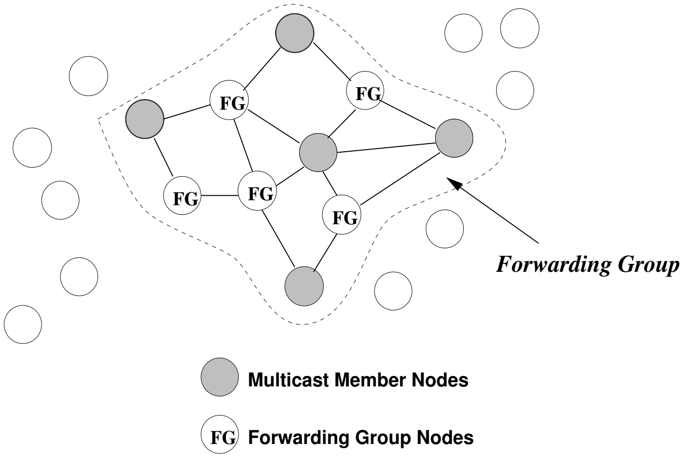


Figure 2: The Forwarding Group Concept.

a multicast receiver also can be a forwarding group node if it is on the path between a multicast source and another receiver. The mesh provides richer connectivity among multicast members compared to trees. Flooding redundancy among forwarding group helps overcome node displacements and channel fading. Hence, unlike for trees, frequent reconfigurations are not required.

Figure 3 is an example to show the robustness of a mesh configuration. Three sources (S_1, S_2, and S_3) send multicast data packets to three receivers (R_1, R_2, and R_3) via three forwarding group nodes (A, B, and C). Suppose the route from S_1 to R_2 is S_1-A-B-R_2. In a tree configuration, if the link between nodes A and B breaks or fails, R_2 cannot receive any packets from S_1 until the tree is reconfigured. ODMRP, on the other hand, already has a redundant route (e.g., S_1-A-C-B-R_2) to deliver packets without going through the broken link between nodes A and B.

2.2 Example

Let us consider Figure 4 as an example of a Join Table forwarding process. Nodes S_1 and S_2 are multicast sources, and nodes R_1, R_2, and R_3 are multicast receivers. Nodes R_2 and R_3 send their Join Tables to both S_1

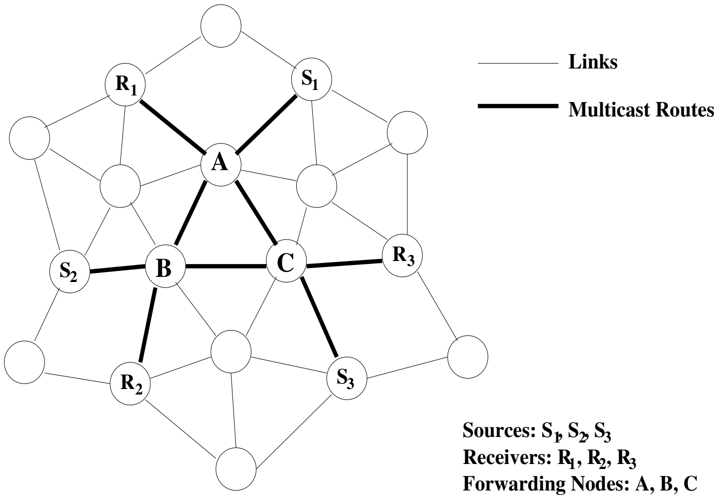
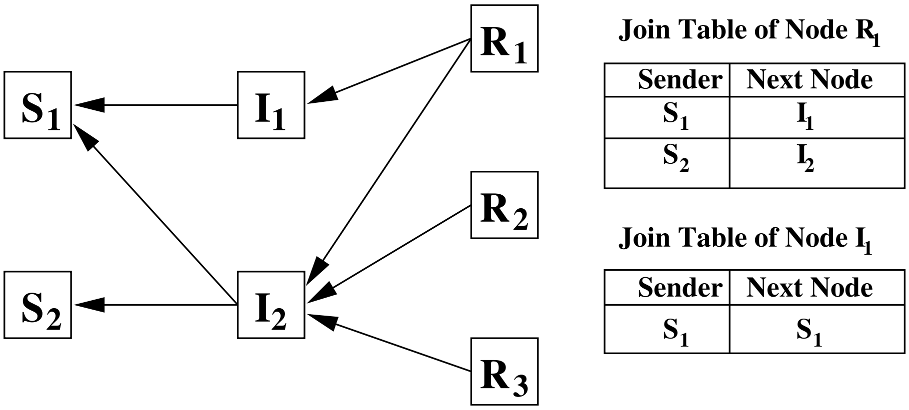


Figure 3: Why a Mesh?



Join Table of Node R_1

Sender	Next Node
S_1	I_1
S_2	I_2

Join Table of Node I_1

Sender	Next Node
S_1	S_1

Figure 4: An Example of a Join Table Forwarding.

and S_2 via I_2. R_1 sends its JOIN TABLE to S_1 via I_1 and to S_2 via I_2. When receivers send their JOIN TABLES to next-hop nodes, an intermediate node I_1 sets the `FG_FLAG` and builds its own JOIN TABLE since there is a next-node ID entry in the JOIN TABLE received from R_1 that matches its ID. Note that the JOIN TABLE built by I_1 has an entry for sender S_1 but not for S_2 because the next-node ID for S_2 in the received JOIN TABLE is not I_1. In the meantime, node I_2 sets the `FG_FLAG`, constructs its own JOIN TABLE and sends it to its neighbors. Note that even though I_2 receives three JOIN TABLES from the receivers, it broadcasts the JOIN TABLE only once because the second and third table arrivals carry no new source information. Channel overhead is thus reduced dramatically in cases where numerous multicast receivers share the same links to the source.

2.3 Data Forwarding

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

2.4 Soft State

In ODMRP, no explicit control packets need 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 TABLE for that group. Nodes in the forwarding group are demoted to non-forwarding nodes if not refreshed (no JOIN TABLES received) before they timeout.

2.5 Selection of Timer Values

Timer values for route refresh interval and forwarding group timeout interval can have impacts on ODMRP performance. The selection of these soft state timers should be adaptive to network environment (e.g., traffic type, traffic load, mobility pattern, mobility speed, channel capacity). When small route refresh interval values are used, fresh route and membership information can be obtained frequently at the expense of producing more packets and causing network congestion. On the other hand, when large route refresh values are selected, even though less control traffic will be generated, nodes may not know up-to-date route and multicast membership. Thus in highly mobile networks, using large route refresh interval values can yield poor protocol performance. The forwarding group timeout interval should also be carefully selected. In heavy traffic load, small values should be used so that unnecessary nodes can timeout quickly and not create excessive redundancy. In situations with high mobility, however, large values should be chosen so that more alternative paths can be provided. It is important to note that the forwarding group timeout value must be larger (e.g., 3 to 5 times) than the value of route refresh interval.

2.6 Unicast Capability

One of the major strengths of ODMRP is its unicast routing capability. Not only can ODMRP coexist with any unicast routing protocol, it can also operate very efficiently as an unicast routing protocol. Thus, a network equipped with ODMRP does not require a separate unicast protocol. Other ad hoc multicast routing protocols such as AMRoute [11], CAMP [15], RBM [14], and LAM [17] must be run on top of a unicast routing protocol. CAMP, RBM, and LAM in particular only work with certain underlying unicast protocols.

2.7 Enhancements

2.7.1 Adapting the Refresh Internal via Mobility Prediction

ODMRP requires periodic flooding of JOIN DATA to refresh routes and group membership. Excessive flooding, however, is not desirable in ad hoc networks because of bandwidth constraints. Furthermore, flooding often causes congestion, contention, and collisions. Finding the optimal refresh interval is critical in ODMRP performance. Here we propose a scheme that adapts the refresh interval to mobility patterns and speeds. By utilizing the location and mobility information provided by GPS (Global Positioning System) [24], we predict the duration of time routes will remain valid.[1] With the predicted time of route disconnection, JOIN DATA are only flooded when route breaks of ongoing data sessions are imminent.

In our prediction method, we assume a free space propagation model [25], where the received signal strength solely depends on its distance to the transmitter. We also assume that all nodes in the network have their clock synchronized (e.g., by using the NTP (Network Time Protocol) [26] or the GPS clock itself).[2] Therefore, if the motion parameters of two neighbors (e.g., speed, direction, radio propagation range) are known, we can determine the duration of time these two nodes will remain connected. Assume two nodes i and j are within the transmission range r of each other. Let (x_i, y_i) be the coordinate of mobile host i and (x_j, y_j) be that of mobile

---

[1] Mobility speed and heading information can be obtained from GPS or the node's own instruments and sensors (e.g., campus, odometer, speed sensors).

[2] Time synchronization of the nodes is done only at the boot time. Once nodes have powered up and their clocks are synchronized, it is not required to perform periodic updates (although periodic updates can still be done in large intervals).

host j. Also let v_i and v_j be the speeds, and θ_i and θ_j $(0 \le \theta_i, \theta_j < 2\pi)$ be the moving directions of nodes i and j, respectively. Then, the amount of time that they will stay connected, D_t, is predicted by:

$$D_t = \frac{-(ab+cd) + \sqrt{(a^2+c^2)r^2 - (ad-bc)^2}}{a^2+c^2}$$

where

$a = v_i \cos\theta_i - v_j \cos\theta_j$,
$b = x_i - x_j$,
$c = v_i \sin\theta_i - v_j \sin\theta_j$, and
$d = y_i - y_j$.

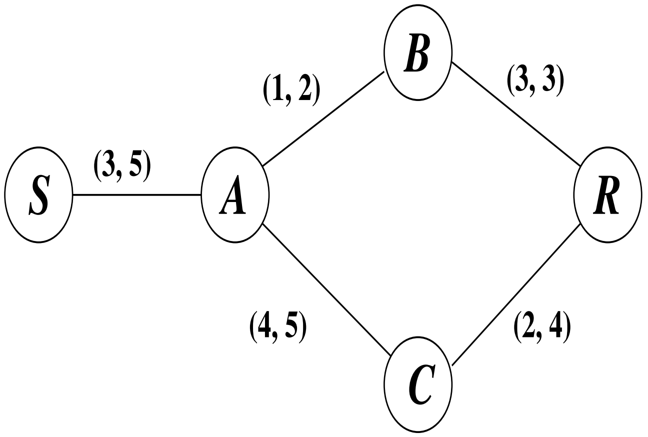
Note that when $v_i = v_j$ and $\theta_i = \theta_j$, D_t is set to ∞ without applying the above equation.

To utilize the information obtained from the prediction, extra fields must be added into JOIN DATA and JOIN TABLE packets. When a source sends JOIN DATA, it appends its location, speed, and direction. It sets the MIN_LET (Minimum Link Expiration Time) field to the `MAX_LET_VALUE` since the source does not have any previous-hop node. The next-hop neighbor, upon receiving a JOIN DATA, predicts the link expiration time between itself and the previous hop using the above equation. The minimum between this value and the MIN_LET indicated by the JOIN DATA is included in the packet. The rationale is that as soon as a single link on a path is disconnected, the entire path is invalidated. The node also overwrites the location and mobility information field written by the previous node with its own information. When a multicast member receives the JOIN DATA, it calculates the predicted LET of the last link of the path. The minimum between the last link expiration time and the MIN_LET value specified in the JOIN DATA is the RET (Route Expiration Time). This RET value is enclosed in the JOIN TABLE and broadcast. If a forwarding group node receives multiple JOIN TABLES with different RET values (i.e., lies in paths from the same source to multiple receivers), it selects the minimum RET among them and sends its own JOIN TABLE with the chosen RET value attached. When the source receives JOIN TABLES, it selects the minimum RET among all the JOIN TABLES received. Then the source can build new routes by flooding a JOIN DATA before the minimum RET approaches (i.e., the route breaks).

In addition to the estimated RET value, other factors need to be considered when choosing the refresh interval. If the node mobility rate is high and the topology changes frequently, routes will expire quickly and often. The source may propagate JOIN DATA excessively and this excessive flooding can cause collisions and congestion, and clogs the network with control packets. Thus, the `MIN_REFRESH_INTERVAL` should be enforced to avoid control message overflow. On the other hand, if nodes are stationary or move slowly and link connectivity remains unchanged for a long duration of time, routes will hardly expire and the source will rarely send JOIN DATA. A few problems arise in this situation. First, if a node in the route suddenly changes its movement direction or speed, the predicted RET value becomes obsolete and routes will not be reconstructed in time. Second, when a non-member node which is located remotely to multicast members wants to join the group, it cannot inform the new membership or receive data until a JOIN DATA is received. Hence, the `MAX_REFRESH_INTERVAL` should be set.

2.7.2 Route Selection Criteria

In the basic ODMRP, a multicast receiver selects routes based on the minimum delay (i.e., routes taken by the first JOIN DATA received). A different route selection method is applied when we use the mobility prediction. The idea is inspired by the Associativity-Based Routing (ABR) protocol [27] which chooses associatively stable routes. In our new algorithm, instead of using the minimum delay path, we choose a route that is the most stable (i.e., the one with the largest RET). To select a route, a multicast receiver must wait for an appropriate amount of time after receiving the first JOIN DATA so that all possible routes and their RETs will be known. The receiver then chooses the most stable route and broadcasts a JOIN TABLE. Route breaks will occur less often and the number of JOIN DATA propagation will reduce because stable routes are used. An example showing the difference between two route selection algorithms is presented in Figure 5. Two routes are available from the source S to the receiver R. Route 1 has the path of S-A-B-R and route 2 has the path of S-A-C-R. If the minimum delay is used as the route selection metric, the receiver node R selects route 1. Route 1 has a delay of 7 $(3+1+3=7)$ while route 2 has a delay of 9 $(3+4+2=9)$. Since the JOIN DATA that takes route 1



(i, j) Link with delay *i* and link expiration time *j*

	Route 1	Route 2
Path	S-A-B-R	S-A-C-R
Delay	7	9
RET	2	4

Figure 5: Route Selection Example.

reaches the receiver first, node R chooses route 1. If the stable route is selected instead, route 2 is chosen by the receiver. The route expiration time of route 1 is 2 ($\min(5, 2, 3) = 2$) while that of route 2 is 4 ($\min(5, 5, 4) = 4$). The receiver selects the route with the maximum RET; hence route 2 is selected.

2.7.3 Alternative Method of Prediction

Since GPS may not work properly in certain situations (e.g., indoor, fading), we may not always be able to accurately predict the link expiration time for a particular link. However, there is an alternative method to predict the LET. This method is based on a more realistic propagation model and has been proposed in [28] and [29]. Basically, transmission power samples are measured periodically from packets received from a mobile's neighbor. From this information it is possible to compute the rate of change for a particular neighbor's transmission power level. Therefore, the time when the transmission power level will drop below the acceptable value (i.e., hysteresis region) can be predicted. We plan to investigate this option in our future work.

3 Implementation

3.1 The Implementation Platform

3.1.1 The Operating System and Software

ODMRP is developed on Linux kernel version 2.0.36, the kernel version provided by the Red Hat Linux version 5.2. All tools and software packages that are used in our development originate from software bundle incorporated within the Red Hat Linux version 5.2 operating system package with the singular exception of Lucent WaveLan IEEE 802.11 device driver [30]. The Linux operating system is chosen for its availability, familiarity, and most importantly, kernel level support for multicasting. The kernel support for multicast allows fast kernel level multicast packet switching and minimizes expensive delays caused by kernel-to-application and application-to-kernel level crossing.

Later, the bandwidth utilization of DVMRP [3] in a wireless environment is studied by routing the Multicast Backbone (MBone) [31] traffic from wired to wireless network with the Linux version of *mrouted*, a DVMRP based multicast routing daemon.

3.1.2 Hardware

Ad hoc network nodes consist of Intel Pentium II based Hewlett Packard Omnibook 7150 laptops equipped with Lucent IEEE 802.11 WaveLan radio devices [30]. The WaveLan devices operate on 2.4 GHz bandwidth and communicate at the maximum capacity of 2 Mbps with the semi-open space range of 150 meters. The WaveLan devices are operated in an ad hoc mode.

3.2 Software Architecture

ODMRP uses the kernel level multicast support option built into the Linux operating system. With the exception of a minor alteration made to allow single device forwarding, no changes were made at the kernel level. Linux kernel supports multicast by altering the forwarding destination interface of the IP packets and rapidly forwarding them without sending them to user level. The destination interface is changed in accordance with the listings in the kernel level multicast routing table. The kernel level routing table is updated and maintained by a user level routing daemon which keeps its own user level routing table. The user level table is copied to the kernel level table as the updates are made. We have used this basic routing table interface to build and maintain ODMRP routing tables both on kernel and user level. In the following sections, our schemes to manage the control packets and the routing table are described and a forwarding scheme based on virtual interfaces is discussed.

3.2.1 Packet and Table Management

There are two types of control packets in ODMRP. JOIN DATA packets are used to advertise the multicast session and JOIN TABLE packets are used to establish the path and the forwarding group. These packets are implemented as new types of IGMP (Internet Group Management Protocol) [32] packet with a data section. Existing IGMP packet structure and handler function are expanded to include functionalities for JOIN DATA and JOIN TABLE.

When a JOIN DATA packet arrives at the router, the content of the packet is cached into a temporary routing table `tr_table` and the timer for the entry is started. If the router does not receive a corresponding JOIN TABLE in time, the timer expires and the cached entry is removed. If a JOIN TABLE which has a corresponding entry in the `tr_table` arrives before the timeout, the user level routing table `routing_table` is searched to find the (source, multicast group) pair that matches the `tr_table` entry. If such a pair is found, the soft state timer for the entry is reset and the router waits for the next event. If the pair can not be found in the `routing_table`, a new entry is created and inserted into the table. The `routing_table` is periodically checked for timer expiration and expired entries are removed. The trigger for the update of the kernel level routing table `kr_table` is activated whenever an entry is inserted or deleted.

3.2.2 Forwarding on Virtual Interfaces

The DVMRP [3], PIM [6], and CBT [5] based multicast routers are all built to be used over wired networks. Therefore, their frameworks are designed for routers with multiple network interfaces. The forwarding capability of these systems is limited strictly to passing the packets from one interface to another. In wired networks, having multiple interfaces does not cause any problems since the devices do not interfere with one another. This is not the case with our wireless network testbed because we use omni directional antennas and a common broadcast channel. Having multiple wireless interfaces does not improve the performance. Unless specifically configured, the devices interfere with one another. However, the framework in Linux does allow the forwarding between virtual interfaces (VIF) to support the tunneling among the multicast islands. Single device forwarding is made possible by creating VIFs by aliasing the existing hardware interface and then enabling the forwarding of the multicast packets to VIFs corresponding to the `routing_table` entries.

3.3 ODMRP Implementation

The enhancement mechanisms utilizing GPS have *not* been incorporated in our current implementation. While the implementation of our enhancements with GPS is in progress, the results reported in this article correspond to the basic version.

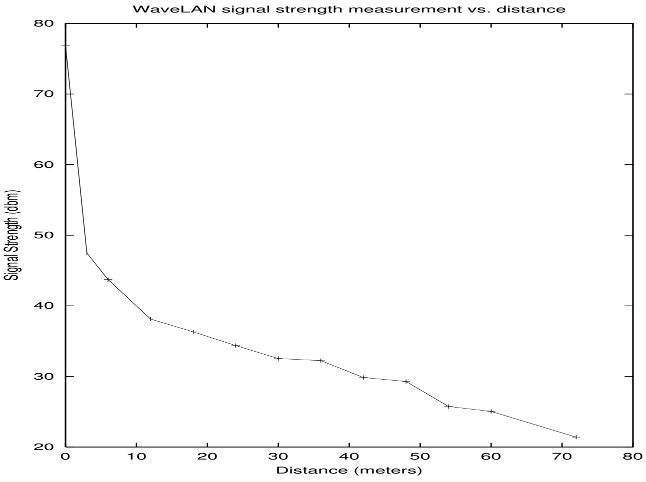


Figure 6: Signal-to-Noise Ratio vs. Distance with WaveLAN Radio Device.

For the ODMRP soft state timer values, we selected 1 s for route refresh interval and 5 s for forwarding group timeout interval.

4 Performance Evaluation

The radio channel performance of a basic wireless link with WaveLAN devices is presented next along with our channel experimental results. For multicast experiments, we created a testbed consisting of four nodes. The bandwidth utilizations of DVMRP and ODMRP are studied and the results are presented later.

4.1 Radio Channel Evaluation

The channel data was collected by initiating a large scale UDP (User Datagram Protocol) packet transfer from one station to another. We chose the UDP as the transport layer protocol instead of TCP (Transmission Control Protocol) in order to study the behavior of packet loss. The experiments were repeated at every 6-m interval until it reached the maximum line of sight distance within the building. At each trial, 13600 UDP packets of size 532 bytes are sent. The receivers are programmed to collect the number of packets received and at each successful reception, record the signal strength. The signal strength is the signal to noise ratio in dbm. Measuring the pure signal strength alone is meaningless unless the ambient noise level is known. The average signal strength values at each location are plotted in Figure 6. As expected, the signal strength degraded with the increase in distance, but the packet loss rate remained nearly the same throughout the experiment. The packet loss rate, on average, remained below 0.5 percent. The packet loss depends more on the position of the node (next to a steal beam, near the elevator, etc.) than the strength of the signal so long as the signal to noise ratio remains above zero. This result indicates that in our environment, wireless communications suffer more losses from random noise than from signal degradation. Some of the effects of random channel loss are explained in the next section.

4.2 DVMRP Bandwidth Utilization

The basic topology of the testbed is shown in Figure 7. On each node, DVMRP based multicast routing daemon *mrouted* is installed. The base station on the lower left corner of the figure links the wired and wireless network, and through this node, we extended the Internet multicast backbone (MBone) to the wireless network. Initially, the multicast daemons on each node are activated one by one. The time it takes for the routing daemon to stabilize as the routing updates from the parent node in the multicast tree are forwarded is variable. The

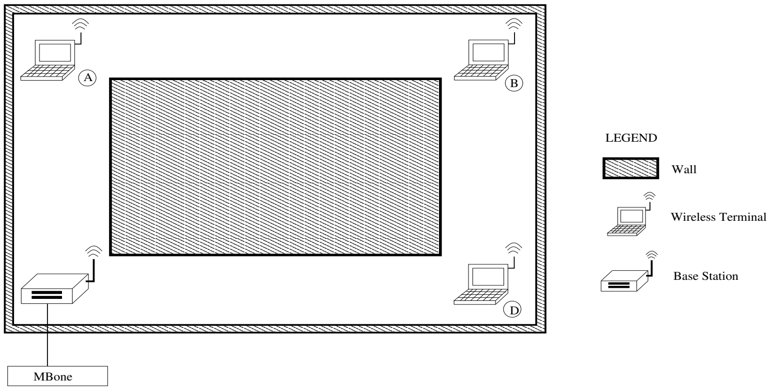


Figure 7: Testbed Topology for DVMRP Traffic Traces.

Table 1: DVMRP Bandwidth Distribution

	Value	% of total
Average session length	178 s	N/A
IGMP control packet overhead	15.58 kb/s	6
Average number of active multicast channels	2	N/A
Overhead caused by multicast channels	26.41 kb/s	10.29
Bandwidth used by data	214.71 kb/s	83.71

duration of time depends on the number of sender and destination pairs in the network. The testbed is ready for the measurement once the waves of initial control packets subside and a regular control packet traffic pattern emerges.

The experiments are carried out in the following steps. First, the traffic traces are started at each router node using *tcpdump*. The receivers at node B and node D then join an audio and a video multicast session. The multicast is monitored for approximately 3 min. The results are compiled into Table 1.

DVMRP operates by allowing the multicast streams to be forwarded downstream and then pruning the unwanted stream from bottom-up with IGMP messages. This practice works well in wired networks since very few control packets are lost. In wireless networks however, packet losses are frequent. When a prune message is lost, the corresponding multicast group is allowed to continue forwarding on the router until another prune message is sent. This causes the control overhead shown in the Table 1 to be relatively high. If a more complex topology is deployed, there will be more control packet losses and more bandwidth wastage.

4.3 ODMRP Bandwidth Utilization

In ODMRP experiments, the base station node of the DVMRP experiments is replaced with a wireless source. The rest of the topology is kept the same as in DVMRP experiments for comparison purposes (see Figure 8). The ODMRP multicast routing daemon does not need the stabilizing period required by the DVMRP since no control packets are switched between routers to establish the initial state. Currently, the MBone traffic cannot be forwarded onto the ODMRP testbed, so the observation was made at the forwarding group nodes while FTP packets are multicasted from node C to nodes B and D. The results are shown in Table 2.

The sessions in this experiment last only until the file is transfered, so the session length field in the table is left blank. Since the forwarding nodes relay the packet only when Join Table packets set/refresh the `FG_FLAG`, unnecessary forwarding does not exist. This results in zero overhead caused by channels. On the other hand, the control packet overhead is much larger than in DVMRP since ODMRP control overhead increases in proportion to the number of active channels. Unfortunately, this overhead can not be eliminated totally,

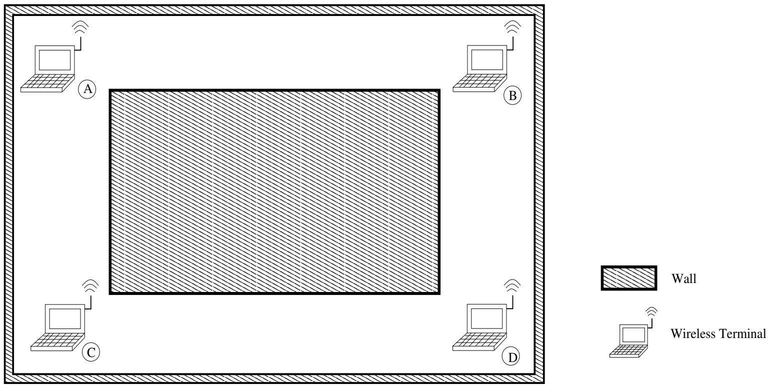


Figure 8: Testbed Topology for ODMRP Traffic Traces.

Table 2: ODMRP Bandwidth Distribution

	Value	% of total
Average session length	N/A	N/A
Control packet overhead	6.68 kb/s	3.4
Average number of active multicast channels	1	N/A
Overhead caused by multicast channels	0 kb/s	0
Bandwidth used by data	170.3 kb/s	96.6

although the amount of control overhead can be adjusted to an optimal value in time. It is essential for the dynamic adaptation scheme in ODMRP to use control packets efficiently to adapt to the frequent route changes due to mobility or changes in intermediate link quality. In contrast, recall that DVMRP, when there is a severe change in link condition, simply discontinues the forwarding of the multicast streams. DVMRP routing table update frequency is inadequate for handling rapid topology changes. Because of these limitations, DVMRP can be operated only in a relatively stable environment.

5 Conclusions

We present ODMRP (On-Demand Multicast Routing Protocol) for a mobile ad hoc wireless network. ODMRP is based on mesh (instead of tree) forwarding. It applies on-demand (as opposed to periodic) multicast route construction and membership maintenance. The advantages of ODMRP are:

- Low channel and storage overhead
- Usage of up-to-date shortest routes
- Robustness to host mobility
- Maintenance and exploitation of multiple redundant paths
- Exploitation of the broadcast nature of the wireless environment
- Unicast routing capability

We have studied the performance of ODMRP and DVMRP in a real ad hoc network testbed with four network hosts. From our experiments, we discovered that DVMRP suffered from high channel overhead due to control message loss in the wireless channel. Our study showed that ODMRP is more suitable in multihop ad hoc wireless environment than DVMRP. Our on going work includes implementing ODMRP enhancements utilizing GPS, building the network for an outdoor environment, and increasing the number of hosts in our ad hoc network testbed.

References

[1] Internet Engineering Task Force (IETF) Mobile Ad Hoc Networks (MANET) Working Group Charter. `http://www.ietf.org/html.charters/manet-charter.html`.

[2] J. Jubin and J.D. Tornow, "The DARPA Packet Radio Network Protocols," *Proceedings of the IEEE*, vol. 75, no. 1, Jan. 1987, pp. 21-32.

[3] S.E. Deering and D.R. Cheriton, "Multicast Routing in Datagram Internetworks and Extended LANs," *ACM Transactions on Computer Systems*, vol. 8, no. 2, May 1990, pp. 85-110.

[4] J. Moy, "Multicast Routing Extensions for OSPF," *Communications of the ACM*, vol. 37, no. 8, Aug. 1994, pp. 61-66, 114.

[5] T. Ballardie, P. Francis, and J. Crowcroft, "Core Based Trees (CBT) - An Architecture for Scalable Inter-Domain Multicast Routing," In *Proceedings of ACM SIGCOMM'93*, San Francisco, CA, Oct. 1993, pp. 85-95.

[6] S. Deering, D.L. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu, and L. Wei, "The PIM Architecture for Wide-Area Multicast Routing," *IEEE/ACM Transactions on Networking*, vol. 4, no. 2, Apr. 1996, pp. 153-162.

[7] S.-J. Lee, M. Gerla, and C.-C. Chiang, "On-Demand Multicast Routing Protocol," In *Proceedings of IEEE WCNC'99*, New Orleans, LA, Sep. 1999, pp. 1298-1302.

[8] S.-J. Lee, W. Su, and M. Gerla, "On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks," *Internet Draft*, `draft-ietf-manet-odmrp-01.txt`, Jun. 1999, Work in progress.

[9] S.-J. Lee, W. Su, and M. Gerla, "Ad hoc Wireless Multicast with Mobility Prediction," In *Proceedings of IEEE ICCCN'99*, Boston, MA, Oct. 1999, pp. 4-9.

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

[11] E. Bommaiah, M. Liu, A. McAuley, and R. Talpade, "AMRoute: Adhoc Multicast Routing Protocol," *Internet-Draft*, `draft-talpade-manet-amroute-00.txt`, Aug. 1998, Work in progress.

[12] C.-C. Chiang and M. Gerla, "On-Demand Multicast in Mobile Wireless Networks," In *Proceedings of IEEE ICNP'98*, Austin, TX, Oct. 1998, pp. 262-270.

[13] C.-C. Chiang, M. Gerla, and L. Zhang, "Adaptive Shared Tree Multicast in Mobile Wireless Networks," In *Proceedings of IEEE GLOBECOM'98*, Sydney, Australia, Nov. 1998, pp. 1817-1822.

[14] M.S. Corson and S.G. Batsell, "A Reservation-Based Multicast (RBM) Routing Protocol for Mobile Networks: Initial Route Construction Phase," *ACM/Baltzer Wireless Networks*, vol. 1, no. 4, Dec. 1995, pp. 427-450.

[15] J.J. Garcia-Luna-Aceves and E.L. Madruga, "A Multicast Routing Protocol for Ad-Hoc Networks," In *Proceedings of IEEE INFOCOM'99*, New York, NY, Mar. 1999, pp. 784-792.

[16] S.K.S. Gupta and P.K. Srimani, "An Adaptive Protocol for Reliable Multicast in Mobile Multi-hop Radio Networks," In *Proceedings of IEEE WMCSA'99*, New Orleans, LA, Feb. 1998, pp. 111-122.

[17] L. Ji and M.S. Corson, "A Lightweight Adaptive Multicast Algorithm," In *Proceedings of IEEE GLOBECOM'98*, Sydney, Australia, Nov. 1998, pp. 1036-1042.

[18] Y.-B. Ko and N.H. Vaidya, "Geocasting in Mobile Ad Hoc Networks: Location-Based Multicast Algorithms," In *Proceedings of IEEE WMCSA'99*, New Orleans, LA, Feb. 1999, pp. 101-110.

[19] T. Ozaki, J.B. Kim, and T. Suda, "Bandwidth-Efficient Multicast Routing Protocol for Ad-Hoc Networks," In *Proceedings of IEEE ICCCN'99*, Boston, MA, Oct. 1999, pp. 10-17.

[20] E.M. Royer and C.E. Perkins, "Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Protocol," In *Proceedings of ACM/IEEE MOBICOM'99*, Seattle, WA, Aug. 1999, pp. 207-218.

[21] P. Sinha, R. Sivakumar, and V. Bharghavan, "MCEDAR: Multicast Core-Extraction Distributed Ad hoc Routing," In *Proceedings of IEEE WCNC'99*, New Orleans, LA, Sep. 1999, pp. 1313-1317.

[22] C.W. Wu and Y.C. Tay, "AMRIS: A Multicast Protocol for Ad hoc Wireless Networks," In *Proceedings of IEEE MILCOM'99*, Atlantic City, NJ, Nov. 1999.

[23] S. Murthy and J.J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks," *ACM/Baltzer Mobile Networks and Applications*, vol. 1, no. 2, Oct. 1996, pp. 183-197.

[24] E.D. Kaplan (Editor), *Understanding the GPS: Principles and Applications*, Artech House, Boston, MA, Feb. 1996.

[25] T.S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, Upper Saddle River, NJ, Oct. 1995.

[26] D.L. Mills, "Internet Time Synchronization: the Network Time Protocol," *IEEE Transactions on Communications*, vol. 39, no. 10, Oct. 1991, pp. 1482-1493.

[27] C.-K. Toh, "Associativity-Based Routing for Ad-Hoc Mobile Networks," *Wireless Personal Communications Journal*, Special Issue on Mobile Networking and Computing Systems, Kluwer Academic Publishers, vol. 4, no. 2, Mar. 1997, pp. 103-139.

[28] P. Agrawal, D.K. Anvekar, and B. Narendran, "Optimal Prioritization of Handovers in Mobile Cellular Networks," In *Proceedings of IEEE PIMRC'94*, The Hague, Netherlands, Sep. 1994, pp. 1393-1398.

[29] B. Narendran, P. Agrawal, and D.K. Anvekar, "Minimizing Cellular Handover Failures Without Channel Utilization Loss," In *Proceedings of IEEE GLOBECOM'94*, San Francisco, CA, Dec. 1994, pp. 1679-1685.

[30] W. van der Moolen, "IEEE 802.11 WaveLAN PC Card User's Guide," Lucent Technologies, Jun. 1998.

[31] H. Eriksson, "MBONE: The Multicast Backbone," *Communications of the ACM*, vol. 37, no. 8, Aug. 1994, pp. 54-60.

[32] W. Fenner, "Internet Group Management Protocol, Version 2," *Request For Comments 2236*, Internet Engineering Task Force, Nov. 1997.

A Protocol Specification

Protocol operation specification is presented as a pseudo code in Figure 9. For more detail operations, readers are referred to [8].

```
Procedure OriginateData
    if (route exists to the destination) and
        (Route_Refresh_Timer has not expired)
    then send data;
    else originate JOIN DATA;

Procedure ProcessData
    if (the received packet is not a duplicate) and
        (FG_Flag for the multicast group is set)
    then forward data;

Procedure CheckFgFlagTimeout
    if FG_Flag_Timer expires
    then reset FG_Flag;

Procedure ProcessJoinData
    if the received packet is not a duplicate
    then {
        insert (source address, sequence no) into message cache;
        insert (source address, last hop address) into route table;
        if the node is a member of the multicast group
        then {
            originate JOIN TABLE;
            if TTL > 0
            then forward JOIN DATA;
        }
        else if TTL > 0
        then forward JOIN DATA;
    }

Procedure ProcessJoinTable
    for each entry in the received JOIN TABLE {
        if the node address matches the next hop address field
        then {
            set FG_Flag for the multicast group;
            if the node address does not match the source address field
            then build a JOIN TABLE entry with a new next hop address;
        }
    }

    if number of newly built JOIN TABLE entry > 0
    then send JOIN TABLE;
```

Figure 9: Protocol Specification.