Ad hoc Wireless Multicast with Mobility Prediction

Sung-Ju Lee, William Su, and Mario Gerla
Wireless Adaptive Mobility Laboratory
Computer Science Department
University of California
Los Angeles, CA 90095-1596
http://www.cs.ucla.edu/NRL/wireless

Abstract – ***An ad hoc wireless network is an infrastructureless network composed of mobile hosts. The primary concerns in ad hoc networks are bandwidth limitations and unpredictable topology changes. Thus, efficient utilization of routing packets and immediate recovery of route breaks are critical in routing and multicasting protocols. A multicast scheme, On-Demand Multicast Routing Protocol (ODMRP), has been recently proposed for mobile ad hoc networks. ODMRP is a reactive (on-demand) protocol that delivers packets to destinations on a mesh topology using scoped flooding of data. A number of enhancements can be applied to improve the performance of ODMRP. In this paper, we propose a mobility prediction scheme to help select stable routes and to perform rerouting in anticipation of topology changes. We also introduce techniques to improve transmission reliability and eliminate route acquisition latency. The impact of our improvements is evaluated via simulation.***

I. INTRODUCTION

An ad hoc network [10] is a dynamically reconfigurable wireless network with no fixed infrastructure. Each host acts as a router and moves in an arbitrary manner. Ad hoc networks are deployed in applications such as disaster recovery and distributed collaborative computing, where routes are mostly multihop and network hosts communicate via packet radios. In a typical ad hoc environment, network hosts work in groups to carry out the given task. Hence, multicast plays an important role in ad hoc networks. Multicast routing protocols used in static networks (e.g., DVMRP [5], MOSPF [15], CBT [2], and PIM [6]), however, do not perform well in ad hoc networks. Multicast tree structures are fragile and must be readjusted continuously 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. Limited bandwidth, constrained power, and mobility of network hosts make the multicast protocol design particularly challenging.

To overcome these limitations, several multicast protocols have been proposed [4], [7], [8], [11], [17]. In this study, we use On-Demand Multicast Routing Protocol (ODMRP) [8] as the starting scheme. ODMRP applies *on-demand* routing techniques to avoid channel overhead and improve scalability. It uses the concept of *forwarding group* [3], a set of nodes which is responsible for forwarding multicast data on shortest paths between any member pairs, to build a forwarding *mesh* for each multicast group. By maintaining and using a mesh instead of a tree, 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.

The major strengths of ODMRP are its simplicity and scalability. We can further improve its performance by several enhancements. In this paper, we propose new techniques to enhance the effectiveness and efficiency of ODMRP. Our primary goals are the following:

- Improve adaptivity to node movement patterns
- Transmit control packets only when necessary
- Reconstruct routes in anticipation of topology changes
- Improve hop-by-hop transmission reliability
- Eliminate route acquisition latency
- Select stable routes

The remainder of the paper is organized as follows. Section II overviews the basic mechanism of ODMRP. Section III describes new enhancements applied to ODMRP. Section IV follows with the simulation results and concluding remarks are made in Section V.

II. ODMRP OVERVIEW

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 comprise the protocol. While a multicast source has packets to send, it periodically broadcasts to the entire network a member advertising packet, called JOIN REQUEST. This periodic transmission refreshes the membership information and updates the routes as follows. When a node receives a non-duplicate JOIN REQUEST, it stores the upstream node ID (i.e., backward learning) and rebroadcasts the packet. When the JOIN REQUEST packet reaches a multicast receiver, the receiver creates or updates the source entry in its *Member Table*.

This work was funded in part by the Defense Advanced Research Projects Agency (DARPA) under contract DAAB07-97-C-D321, as a part of the Global Mobile Information Systems (GloMo) program.

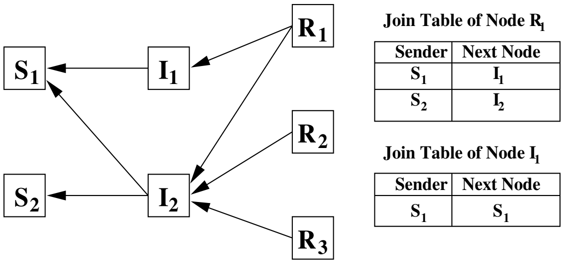


Fig. 1. An Example of a Join Table Forwarding.

While valid entries exist in the *Member Table*, JOIN TABLES are broadcasted periodically to the 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` 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*.

Let us consider Fig. 1 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 and S_2 via I_2. R_1 sends its packet 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.

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

III. ENHANCEMENTS

A. *Adapting the Refresh Interval via Mobility Prediction*

ODMRP requires periodic flooding of JOIN REQUESTS to build and refresh routes. 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) [13], we predict the duration of time routes will remain valid.[1] With the predicted time of route disconnection, JOIN REQUESTS are only flooded when route breaks of ongoing data sessions are imminent.

In our prediction method, we assume a free space propagation model [16], 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) [14] or the GPS clock itself). Therefore, if the motion parameters of two neighbors (e.g., speed, direction, radio propagation range, etc.) 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 host j. Also let v_i and v_j be the speeds, and θ_i and θ_j $(0 \leq \theta_i, \theta_j < 2\pi)$ 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 REQUEST and JOIN TABLE packets. When a source sends JOIN REQUESTS, 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 REQUEST, predicts the link expiration time between itself and the previous hop using the above equation. The minimum between this value and the

---

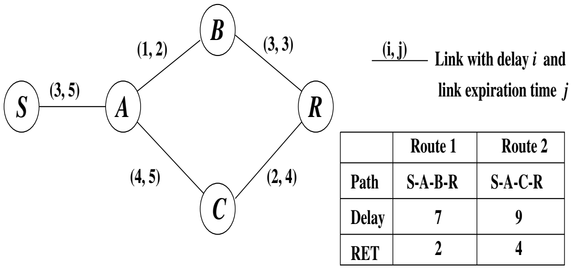
[1] Mobility speed and heading information can be obtained from GPS or the node's own instruments and sensors (e.g., campus, odometer, speed sensors, etc.).

MIN_LET indicated by the JOIN REQUEST 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 REQUEST, 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 REQUEST is the RET (Route Expiration Time). This RET value is enclosed in the JOIN TABLE and broadcasted. 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 REQUEST before the minimum RET approaches (i.e., route breaks). Note that JOIN TABLES need not be periodically transmitted by multicast receivers. Since sources flood JOIN REQUESTS only when needed, receivers only send JOIN TABLES after receiving JOIN REQUESTS.

In addition to the estimated RET value, other factors need to be considered when choosing the flooding interval of JOIN REQUESTS. If the node mobility rate is high and the topology changes frequently, routes will expire quickly and often. The source may propagate JOIN REQUESTS 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 REQUESTS. 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 REQUEST is received. Hence, the `MAX_REFRESH_INTERVAL` should be set. The selection of the `MIN_REFRESH_INTERVAL` and the `MAX_REFRESH_INTERVAL` values should be adaptive to network situations (e.g., traffic type, traffic load, mobility pattern, mobility speed, channel capacity, etc.).

B. 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 REQUEST 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 [18] which chooses associatively stable routes. In our new algorithm, instead of using the minimum delay path, we can 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 REQUEST 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 REQUEST propagation will reduce because stable routes are used. An example showing the difference between two route selection algorithms is presented in Fig. 2. Two routes are available from the source S to the receiver R. Route 1 has a path of S-A-B-R and route 2 has a 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 REQUEST that takes route 1 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, and hence route 2 is selected. We will evaluate different route selection methods by simulation in Section IV.



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

Fig. 2. Route Selection Example.

C. Reliability

The reliable transmission of JOIN TABLES plays an important role in establishing and refreshing multicast routes and forwarding groups. Hence, if JOIN TABLES are not properly delivered, effective multicast routing cannot be achieved by ODMRP. The IEEE 802.11 MAC protocol [9], which is the emerging standard in wireless networks, performs reliable transmission by retransmitting the packet if no acknowledgment is received. However, if the packet is broadcasted, no acknowledgments or retransmissions are sent. In ODMRP, the transmission of JOIN TABLES are mostly broadcasted. Thus, the hop-by-hop verification of JOIN TABLE delivery and the retransmission must be done by ODMRP.

We adopt a scheme that was used in [12]. Fig. 3 is shown to illustrate the mechanism. When node B transmits a packet to node C after receiving a packet from node A, node A can hear the transmission of node B if it is within B's radio propagation

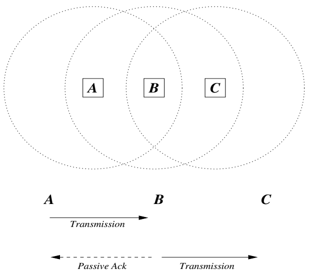


Fig. 3. Passive Acknowledgments.

range. Hence, the packet transmission by node B to node C is used as a *passive acknowledgment* to node A. We can utilize this passive acknowledgment to verify the delivery of a Join Table. Multicast sources must send active acknowledgments to the previous hops since they do not have any next hops to send Join Tables to unless they are forwarding group nodes. When no acknowledgment is received within the timeout interval, the node retransmits the message. If packet delivery cannot be verified after an appropriate number of retransmissions, the node considers the route to be invalidated. The node then broadcasts a message to its neighbors specifying that the next hop to the source cannot be reached. Upon receiving this packet, each neighbor builds and unicasts the Join Table to its next hop if it has a route to the multicast source. If no route is known, it simply broadcasts the packet specifying the next hop is not available. In both cases, the node sets its `FG_FLAG`. The `FG_FLAG` setting of every neighbor may create excessive redundancy, but most of these settings will expire because only necessary forwarding group nodes will be refreshed in the next Join Table propagation phase.

D. Elimination of Route Acquisition Latency

The major drawback of on-demand routing protocols is the delay required to obtain a route. This route acquisition latency makes on-demand protocols less attractive in networks where real-time traffic is exchanged. In the basic ODMRP, when no multicast route information is known by the source, data transmission is delayed for a certain period of time. In contrast to unicast routing, the selection of the waiting time is not straightforward. In unicast, the source can send data as soon as a Route Reply is received. In ODMRP, however, the data transmission cannot be made immediately after receiving the first Join Table since routes to receivers that are farther away may not yet have been established.

To eliminate these problems, when a source has data to send but no multicast route is known, it floods the data instead of the Join Request. The periodic transmission of Join Requests is also replaced by data.[2] Basically, Join Data becomes a Join Request with data payload attached. Thus, the flooding of Join Data achieves data delivery in addition to constructing and refreshing the routes. Although the size of the flooded packet is larger compared to Join Requests, route acquisition latency is eliminated.

IV. PERFORMANCE EVALUATION

A. Simulation Environment

The simulator was implemented within the Global Mobile Simulation (GloMoSim) library [19]. The GloMoSim library is a scalable simulation environment for wireless network systems using the parallel discrete-event simulation capability provided by PARSEC [1]. Our simulation modeled a network of 50 mobile hosts placed randomly within a $1000m \times 1000m$ area. Radio propagation range for each node was 250 meters and channel capacity was 2 Mbits/sec. Each simulation executed for 600 seconds of simulation time. Multiple runs with different seed numbers were conducted for each scenario and collected data were averaged over those runs.

A free space propagation model [16] with a threshold cutoff was used in our experiments. In the free space model, the power of a signal attenuates as $1/d^2$ where d is the distance between radios. In the radio model, we assumed the ability of a radio to lock on to a sufficiently strong signal in the presence of interfering signals, i.e., radio capture. If the capture ratio (the minimum ratio of an arriving packet's signal strength relative to those of other colliding packets) [16] was greater than the predefined threshold value, the arriving packet was received while other interfering packets were dropped. The IEEE 802.11 Distributed Coordination Function (DCF) [9] was used as the medium access control protocol. The scheme used was Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) with acknowledgments. A traffic generator was developed to simulate constant bit rate sources. The size of data payload was 512 bytes. Each node moved constantly with the predefined speed. Moving direction was selected randomly, and when nodes reached the simulation terrain boundary, they bounced back and continued to move. One multicast group of size ten with one source was simulated. The multicast members and the source were chosen randomly with uniform probabilities. Members joined the group at the start of the simulation and remained as members throughout the simulation.

B. Methodology

To investigate the impact of our enhancements, we simulated the following three schemes:

1. *Scheme* A: the basic ODMRP as specified in [8]
2. *Scheme* B: the enhanced ODMRP that uses the minimum delay as the route selection metric
3. *Scheme* C: the enhanced ODMRP that uses the route expiration time as the route selection metric

---

[2] To differentiate between the flooded data that performs the Join Request role and the ordinary data, we term the flooded data packet as Join Data.

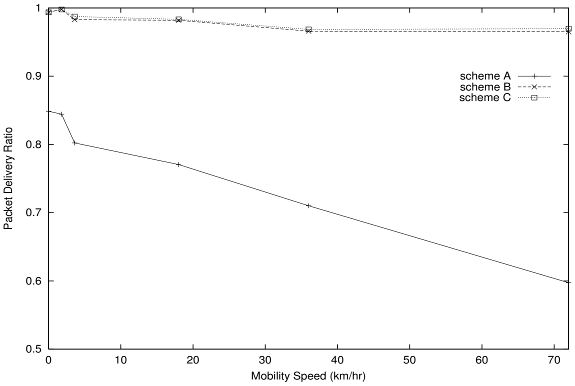


Fig. 4. Packet Delivery Ratio as a Function of Speed.

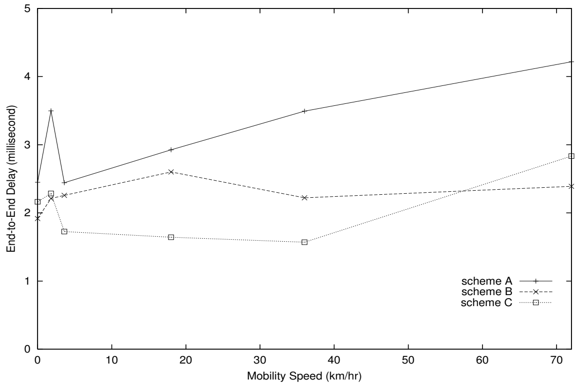


Fig. 5. End-to-End Delay as a Function of Speed.

Both enhanced schemes included reliable transmission and route acquisition latency elimination features. The protocols were evaluated as a function of speed. The metrics of interest are:

- **Packet delivery ratio**: The number of data packets actually received by multicast members over the number of data packets supposed to be received by multicast members.
- **End-to-end delay**: The time elapsed between the instant when the source has data packet to send and the instant when the destination receives the data. Note that if no route is available, the time spent in building a route (i.e., route acquisition latency) is included in the end-to-end delay.
- **Control overhead**: The total control bytes transmitted. Bytes of data packet and JOIN DATA headers in addition to bytes of control packets (i.e., JOIN REQUESTS, JOIN TABLES, active acknowledgments) are calculated as control overhead.

C. Simulation Results

C.1 Packet Delivery Ratio

The packet delivery ratio as a function of the mobility speed is shown in Fig. 4. We can observe that as speed increases, the routing effectiveness of *scheme A* degrades rapidly compared to *schemes B* and *C*. Both *schemes B* and *C* have very high delivery ratios of over 96% regardless of speed. As the routes are reconstructed in advance of topology changes, most data are delivered to multicast receivers without being dropped. In *scheme A*, however, JOIN REQUESTS and JOIN TABLES are transmitted periodically (every 400 msec and 180 msec, respectively) without adapting to mobility speed and direction. Frequent flooding resulted in collisions and congestion, leading to packet drops even in low mobility rates. At high speed, routes that are taken at the JOIN REQUEST phase may already be broken when JOIN TABLES are propagated. In *scheme A*, nodes do not verify the reception of JOIN TABLES transmitted. Most JOIN TABLES failed to reach the source and establish the forwarding group. Thus, when data is sent by the source, the multicast route is not properly built and packets can not be delivered. Both *schemes B* and *C* enforce reliable transmissions of JOIN TABLES. Routes and forwarding group nodes are established and refreshed appropriately even in high mobility situations and the schemes proved to be robust to the mobility speed.

C.2 End-to-End Delay

Fig. 5 shows the end-to-end delay of each scheme. *Schemes B* and *C* have shorter delay compared to *scheme A*. In *scheme A*, sources flood JOIN REQUESTS and must wait for a certain amount of time to send data until routes are established among multicast members. In *schemes B* and *C*, on the contrary, sources flood JOIN DATA immediately even before routes and forwarding group are constructed. The route acquisition latency is eliminated and packets are delivered to receivers in shorter delays. One might be surprised to see that the delay of *scheme B* which uses the minimum delay route is larger than that of *scheme C* which uses the stable (and possibly longer delay) route. Even though the route taken by JOIN DATA is the shortest delay route at that instant, it may not be the minimum delay route later on as nodes move. In addition, compared to stable routes, the minimum delay routes break more frequently and data may need to traverse through longer redundant routes formed by forwarding group nodes.

C.3 Control Overhead

Fig. 6 shows the control byte overhead as a function of mobility speed for each scheme. Remember that the transmission of control packets in *scheme A* is time triggered only without adapting to mobility speed. Hence, the amount of control overhead does not increase as the mobility speed increases. Actually, control overhead decreases as nodes move faster. As JOIN TABLES are less likely to reach the target nodes in a highly mobile environment, the JOIN TABLE propagations by the next nodes are triggered less. Furthermore, data packets

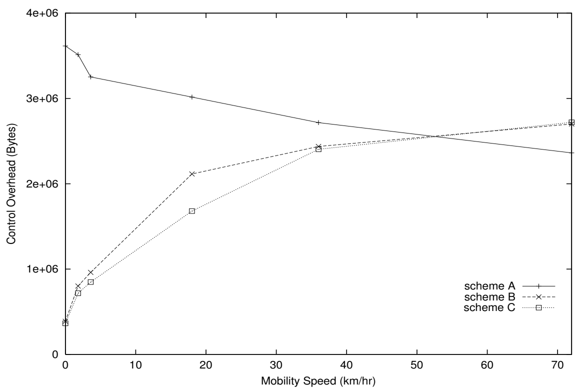


Fig. 6. Control Overhead as a Function of Speed.

(whose header is calculated as control overhead), are transmitted less because forwarding group nodes and routes are not established or refreshed appropriately as the speed increases. On the other hand, the overhead of *schemes B* and *C* go up as mobility speed increases. Since mobility prediction is used to adapt to mobility speed, more JOIN DATA and JOIN TABLES are sent when mobility is high. In addition, JOIN TABLE retransmission and active acknowledgment propagation also increase with mobility and add to the control overhead. It is important to observe that the overhead of *schemes B* and *C* are both significantly less than that of *scheme A* in low mobility cases because control packets are transmitted only when necessary in *schemes B* and *C*. The enhanced schemes have more overhead when nodes move fast, but the extra control packets are used efficiently in delivering data (see Fig. 4). When comparing *scheme B* with *scheme C*, we can see that *scheme B* yields more overhead in low mobility while both schemes produce nearly equal amount of overhead in high mobility. Since *scheme C* chooses a stable route, JOIN DATA are flooded less often. However, when nodes move relatively fast (e.g., 72 km/hr in our simulation), routes are broken often and links will remain connected for a short duration of time. Sources are thus likely to use `MIN_REFRESH_INTERVAL` and the overhead incurred by both *schemes B* and *C* become almost identical.

V. CONCLUSIONS

We have presented new techniques to improve the performance of ODMRP. By using the mobility and link connectivity prediction, routes and forwarding groups are reconstructed in anticipation of topology changes. This adaptive selection of the refresh interval avoids the transmission of unnecessary control packets and the resulting bandwidth wastage. We have applied a new route selection algorithm to choose routes that will stay valid for the longest duration of time. The usage of stable routes further reduces the control overhead. Passive acknowledgments and retransmissions have been used to improve the reliable delivery of JOIN TABLES. The improved reliability plays a factor in protocol enhancement since the delivery of JOIN TABLES is critical in establishing the routes and forwarding group nodes. We have also introduced a method to eliminate the route acquisition latency.

Simulation results showed that our new methods improved the basic scheme significantly. More data packets were delivered to destinations, less control packets were produced in low mobility, control packets were utilized more efficiently in high mobility, and end-to-end delay was shorter. These enhancements enabled ODMRP to be more robust to host mobility.

References

[1] R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, and H.Y. Song, "PARSEC: A Parallel Simulation Environment for Complex Systems," *IEEE Computer*, vol. 31, no. 10, Oct. 1998, pp.77-85.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[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] 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.

[8] M. Gerla, G. Pei, S.-J. Lee, and C.-C. Chiang, "On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks," *Internet Draft*, `draft-ietf-manet-odmrp-00.txt`, Nov. 1998, Work in progress.

[9] IEEE Computer Society LAN MAN Standards Committee, *Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification*, IEEE Std 802.11-1997. The Institute of Electrical and Electronics Engineers, New York, NY, 1997.

[10] Internet Engineering Task Force (IETF) Mobile Ad Hoc Networks (MANET) Working Group Charter.

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

[12] 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.

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

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

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

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

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

[18] 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.

[19] UCLA Parallel Computing Laboratory and Wireless Adaptive Mobility Laboratory, *GloMoSim: A Scalable Simulation Environment for Wireless and Wired Network Systems*, http://pcl.cs.ucla.edu/projects/domains/glomosim.html.