Wireless Ad hoc Multicast Routing with Mobility Prediction [*]

Sung-Ju Lee William Su [**] Mario Gerla
Computer Science Department, University of California, Los Angeles, CA 90095-1596, USA
E-mail: (sjlee, wsu, gerla)@cs.ucla.edu

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 destination(s) on a mesh topology using scoped flooding of data. We can apply a number of enhancements 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.

Keywords: Multicast and routing protocols, ad hoc networks, mobile computing, mobility prediction

1. Introduction

An ad hoc network is a dynamically reconfigurable wireless network with no fixed infrastructures. 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 such as Distance Vector Multicast Routing Protocol (DVMRP) [7], Multicast Open Shortest Path First (MOSPF) [19], Core Based Trees (CBT) [3], and Protocol Independent Multicast (PIM) [8], 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 [4,9,16,13,22,23,27]. In this study, we will use On-Demand Multicast Routing Protocol (ODMRP) [16,17] as the starting scheme. ODMRP applies *on-demand* routing techniques to avoid channel overhead and improve scalability. It uses the concept of *forwarding group* [5], 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, ODMRP avoids drawbacks of multicast trees in mobile wireless networks (for example, intermittent connectivity, traffic concentration, frequent tree reconfiguration, non-shortest path in a shared tree). ODMRP takes a *soft-state* approach to maintain multicast group members. No explicit control message transmission 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:

---

[*] 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.

[**] Now with eWings Technologies, Plano, TX.

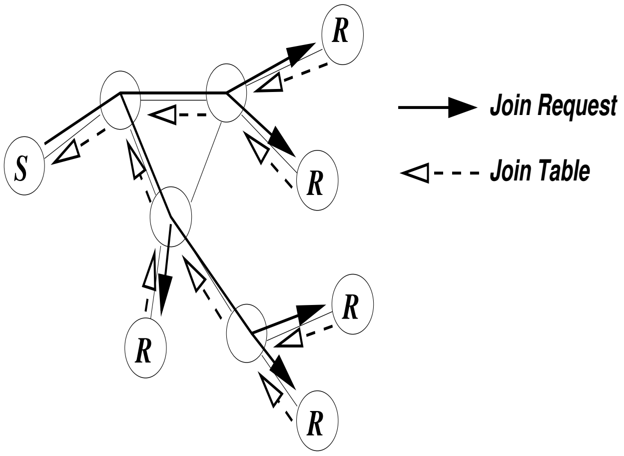


Figure 1. On-demand procedure for membership setup and maintenance.

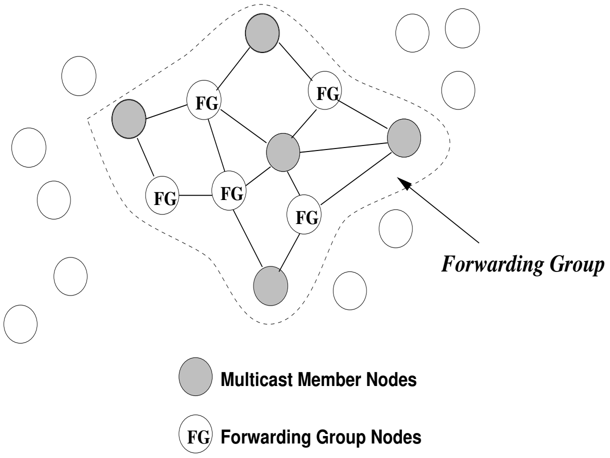


Figure 2. The forwarding group concept.

- 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 2 overviews the basic mechanism of ODMRP. Section 3 describes new enhancements applied to ODMRP. Section 4 follows with the simulation results and concluding remarks are made in Section 5.

2. ODMRP Overview

ODMRP establishes and maintains group membership and multicast routes by the source on demand. Similar to on-demand unicast routing protocols, a query phase and a reply phase comprise the protocol (see Figure 1). 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 address in its *route table* (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*. 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 address of one of the entries matches its own address. If matched, the node realizes that it is on the path to the source and thus is a part of the forwarding group. It then sets the `FG_FLAG` and broadcasts its own JOIN TABLE built upon matched entries. Each forwarding group member hence propagates the JOIN TABLE until the packet 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 visualize 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 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 with trees. Flooding redundancy among forwarding group helps overcome node displacements and channel fading. Hence, unlike trees, meshes do not require frequent reconfigurations.

An example in Figure 3 illustrates the robustness of a mesh configuration. Three sources (S_1, S_2, and S_3) send multicast data packets to three receivers (R_1, R_2, and R_3) via three forwarding group nodes (A, B, and C). Suppose the route from S_1 to R_2 is (S_1-A-B-R_2). In a tree configuration, if the link between nodes A and B breaks or fails, R_2 cannot receive any packets from S_1 until the tree is reconfigured. ODMRP, on the other hand, already has a redundant route (S_1-A-C-B-R_2) to

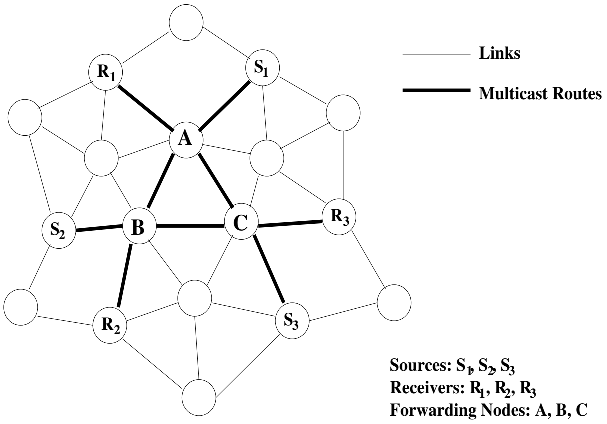


Figure 3. Why a mesh?

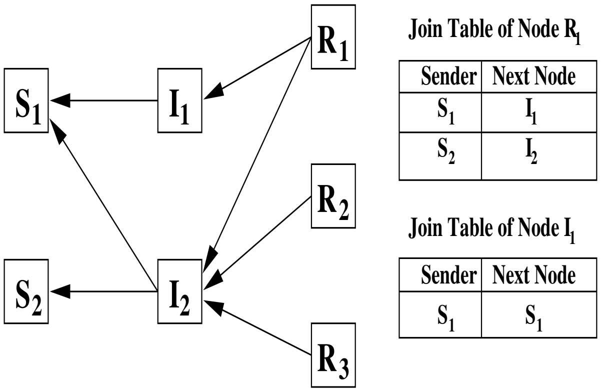


Figure 4. An example of a JOIN TABLE forwarding.

deliver packets without going through the disconnected link between nodes A and B.

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 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 address entry in the JOIN TABLE received from R_1 that matches its own address. 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 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 the packet to its neighbors. 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 the packet 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.

In ODMRP, nodes do not need to send explicit control packets to leave the group. If a multicast source wants to leave the group, it simply stops sending JOIN REQUEST 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 removes the corresponding entries from its *Member Table* and 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.

Unicast routing capability is one of the major strengths of ODMRP. Not only can ODMRP coexist with any unicast routing protocol, it can function as both multicast and unicast. Thus, ODMRP can run without any underlying unicast protocol. Other ad hoc multicast protocols such as Adhoc Multicast Routing Protocol (AMRoute) [4], Core Assisted Mesh Protocol (CAMP) [9], Reservation-Based Multicast (RBM) [6], and Lightweight Adaptive Multicast (LAM) [13] must be run on top of a unicast routing protocol. CAMP, RBM, and LAM in particular, only work with certain underlying unicast protocols.

3. Enhancements

3.1. 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 route refresh interval to mobility patterns and speeds [24]. By utilizing the location and mobility information provided by GPS (Global Positioning System) [15], 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 [21], where the received signal strength solely depends on its distance to the transmitter. We also assume that all nodes in the network have their clocks synchronized (for example, by using the NTP (Network Time Protocol) [18] or the GPS clock itself).[2] Therefore, if we know the motion parameters of two neighbors (such as speed, direction, radio propagation range), 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:

$$D_t = \frac{-(ab+cd) + \sqrt{(a^2+c^2)r^2 - (ad-bc)^2}}{a^2+c^2} \tag{3.1}$$

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, JOIN REQUEST and JOIN TABLE packets must add extra fields. 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 equation (3.1). The minimum between this value and the 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 received JOIN TABLES. Then the source builds new routes by flooding a JOIN REQUEST before the minimum RET approaches (i.e., route breaks). Note that multicast receivers need not periodically transmit JOIN TABLES. Since sources flood JOIN REQUESTS only when needed, receivers only send JOIN TABLES after receiving JOIN REQUESTS.

In addition to the estimated RET value, we need to consider other factors when choosing the route 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 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 we cannot reconstruct routes in time. Second, when a nonmember node located remotely from multicast members wants to join the group, it cannot inform the new membership or receive data until it receives a JOIN REQUEST. Hence, the `MAX_REFRESH_INTERVAL` should be

---

[1] We can obtain mobility speed and heading information from GPS or the node's own instruments and sensors (for example, 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 we can still perform periodic updates in large intervals).

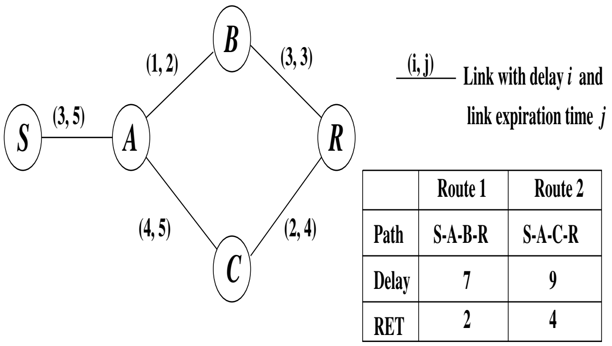
set. The selection of the `MIN_REFRESH_INTERVAL` and the `MAX_REFRESH_INTERVAL` should be adaptive to network situations (among others, traffic type, traffic load, mobility pattern, mobility speed, channel capacity).

3.1.1. Alternative Method of Prediction

Since GPS may not work properly in certain situations (for instance, indoor, fading), we are not always able to accurately predict the link expiration time for a particular link. Nevertheless, there is an alternative method to predict the LET. This method is based on a more realistic propagation model and is proposed in [1] and [20]. Basically, a node periodically measures transmission power samples from packets received from its neighbor. From this information, the node computes the change rate for a particular neighbor's transmission power level. Therefore, it can predict the time when the transmission power level will drop below the acceptable value (hysteresis region).

3.2. Route Selection Criteria

In the basic ODMRP, a multicast receiver selects routes based on the minimum delay (i.e., the route taken by the first received Join Request). We can apply a different route selection method when using the mobility prediction. The idea is inspired by the Associativity-Based Routing (ABR) protocol [25] 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 (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 it will know all possible routes and their RETs. 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 we use stable routes. An example that shows 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 a path of (S-A-B-R) and route 2 has a path of (S-A-C-R). If we use the minimum delay as the route selection metric, the receiver node R selects route 1. Route 1 has a delay of seven ($3 + 1 + 3 = 7$) and route 2 has a delay of nine ($3 + 4 + 2 = 9$). Since the Join Request that takes route 1 reaches the receiver first, node R chooses route 1. If we select the stable route instead, the receiver chooses route 2. The route expiration time of route 1 is two ($\min(5, 2, 3) = 2$) and that of route 2 is four ($\min(5, 5, 4) = 4$). The receiver selects the route with the maximum RET, and hence selects route 2. We evaluate different route selection methods by simulation in Section 4.



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

Figure 5. Route selection example.

3.3. Reliability

The reliable transmission of Join Tables plays an important role in establishing and refreshing multicast routes and forwarding groups. If Join Tables are not properly delivered, ODMRP cannot achieve effective multicast routing. The IEEE 802.11 MAC protocol [11], which is the emerging standard in wireless networks, performs reliable transmission by retransmitting the packet if no acknowledgment is received. If the packet is broadcasted, however, no acknowledgments or retransmissions are sent. In ODMRP, the transmission of Join Tables are broadcasted when there are multiple entries. Thus, ODMRP must perform the hop-by-hop Join Table delivery verification and retransmission.

We adopt a scheme that was used in [14]. Figure 6 illustrates 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 still within B's radio propagation range. The packet transmission by node B to node C is hence 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 the node does not receive any acknowledgment within the timeout interval,

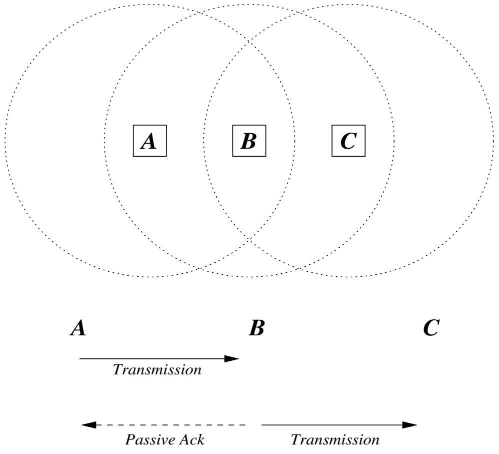


Figure 6. Passive acknowledgments.

it retransmits the message. If the node cannot verify the packet delivery after an appropriate number of retransmissions, it 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.

3.4. 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 the source does not have any multicast route information, it postpones the data transmission 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 it receives a Route Reply. In ODMRP, however, sources cannot transmit data 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 does not know any multicast route, it floods the data instead of the Join Request. The data packet also replaces the periodic transmission of Join Requests.[3] Basically, Join Data becomes a Join Request with data payload attached. 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 with Join Requests, route acquisition latency is eliminated.

4. Performance Evaluation

4.1. Simulation Environment

We implemented the simulator within the Global Mobile Simulation (GloMoSim) library [26]. The GloMoSim library is a scalable simulation environment for wireless network systems using the parallel discrete-event simulation capability provided by PARSEC [2]. 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. We conducted multiple runs with different seed numbers for each scenario and averaged collected data over those runs.

We used a free space propagation model [21] with a threshold cutoff 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) [21] was greater than the predefined threshold value, the arriving packet was received and other interfering packets were dropped. We used the IEEE 802.11 Distributed Coordination Function (DCF) [11] as the medium access control protocol. The scheme used was Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) with acknowledgments. We developed a traffic generator 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. We sim-

---

[3] 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.

ulated one multicast group with one source. 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.

4.2. Methodology

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

1. *Scheme A*: the basic ODMRP as specified in [10],
2. *Scheme B*: the enhanced ODMRP that uses the minimum delay as the route selection metric, and
3. *Scheme C*: the enhanced ODMRP that uses the route expiration time as the route selection metric.

Both enhanced schemes included reliable transmission and route acquisition latency elimination features. We evaluate the protocols as a function of speed and multicast group size. In the first set of experiments, we set the size of the multicast group constant to ten and vary the speed from 0 km/hr to 72 km/hr. In the second set of simulations, we set the node mobility speed constant at 18 km/hr and vary the multicast group size from two (unicast) to twenty. 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 (route acquisition latency) is included in the end-to-end delay.
- **Control overhead**: The total control bytes transmitted. We calculate bytes of data packet and JOIN DATA headers in addition to bytes of control packets (JOIN REQUESTS, JOIN TABLES, active acknowledgments) as control overhead.
- **Number of total packets transmitted per data packet delivered**: The number of all packets (data and control packets) transmitted divided by data packet delivered to destinations. This measure shows the efficiency in terms of channel access and is very important in ad hoc networks since link layer protocols are typically contention-based.

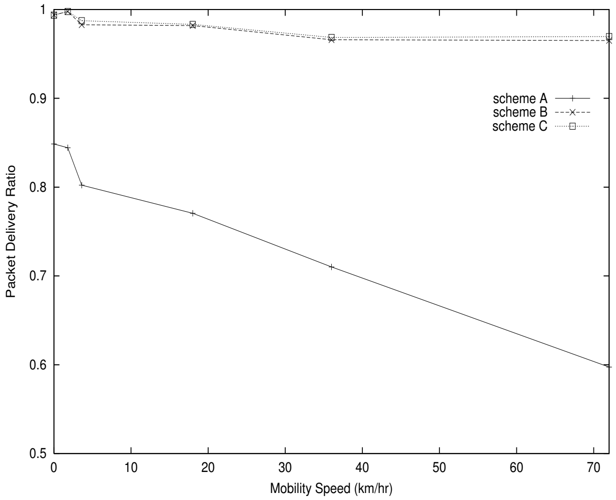


Figure 7. Packet delivery ratio as a function of speed.

4.3. Simulation Results

4.3.1. Packet Delivery Ratio

The packet delivery ratio as a function of the mobility speed and the multicast group size is shown in Figures 7 and 8, respectively. We can see from Figure 7 that as speed increases, the routing effectiveness of *scheme A* degrades rapidly compared with *schemes B* and *C*. Both *schemes B* and *C* have very high delivery ratios of over 96% regardless of speed. As they reconstruct the routes in advance of topology changes, most data are delivered to multicast receivers without being dropped. *Scheme A*, however, periodically transmits JOIN REQUESTS and JOIN TABLES (every 400 ms and 180 ms, 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 transmitted JOIN TABLES. Most JOIN TABLES failed to reach the source and establish the forwarding group. Thus, when the source sends the data, the multicast route is not properly built and packets can not be delivered. Both *schemes B* and *C* enforce reliable JOIN TABLE transmissions. The schemes appropriately establish and refresh the routes and forwarding group nodes even in high mobility situations and they proved to be robust to the mobility speed.

In Figure 8, *schemes B* and *C* outperform *scheme A* again. The result shows that our enhanced protocols are robust to multicast group size in addition to mobil-

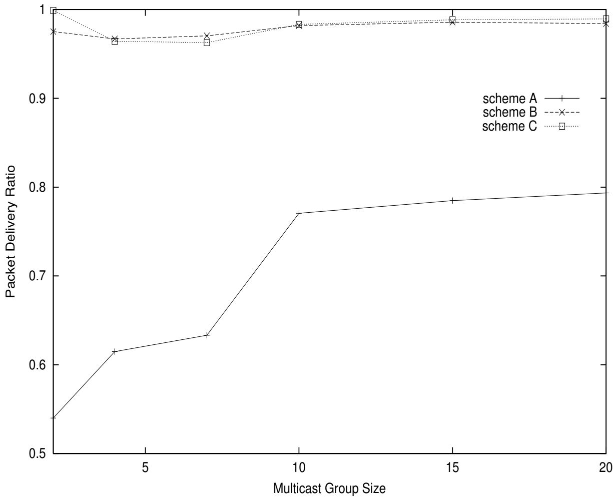


Figure 8. Packet delivery ratio as a function of number of multicast members.

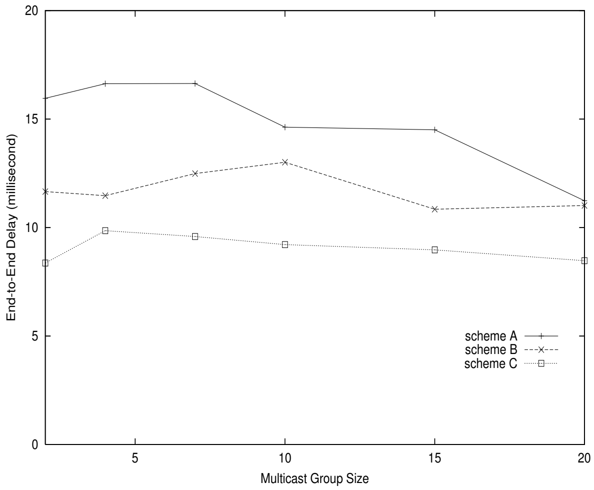


Figure 10. End-to-end delay as a function of number of multicast members.

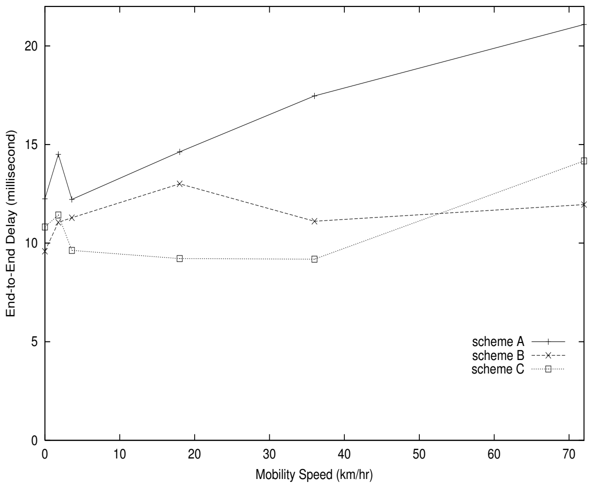


Figure 9. End-to-end delay as a function of speed.

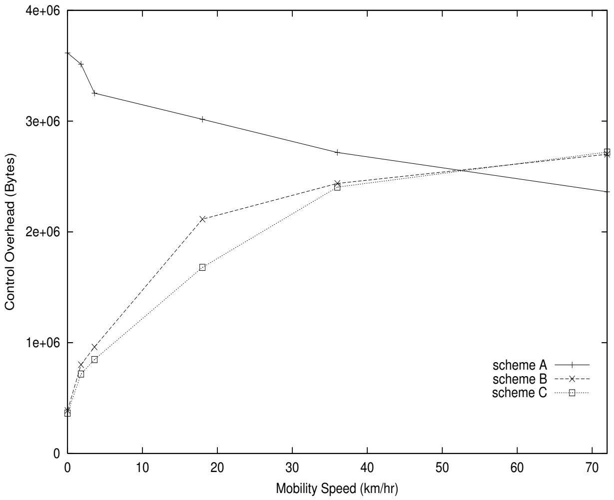


Figure 11. Control overhead as a function of speed.

ity speed. *Scheme A*'s performance improves as the size becomes larger. As the number of receivers increases, the number of forwarding group nodes increases accordingly. Hence, the connectivity of the multicast mesh becomes richer and the redundancy of the paths helps delivering data to destinations.

4.3.2. End-to-End Delay

Figures 9 and 10 show the end-to-end delay of each scheme. *Schemes B* and *C* have shorter delays compared with that of *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 they construct routes and forwarding group. The route acquisition latency is eliminated and packets are delivered to receivers in shorter delays. One may 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 with stable routes, the minimum delay routes disconnect more frequently which results in data packets traversing through alternate and longer routes formed by forwarding group nodes.

4.3.3. Control Overhead

Figure 11 shows the control byte overhead as a function of mobility speed for each protocol. Remember

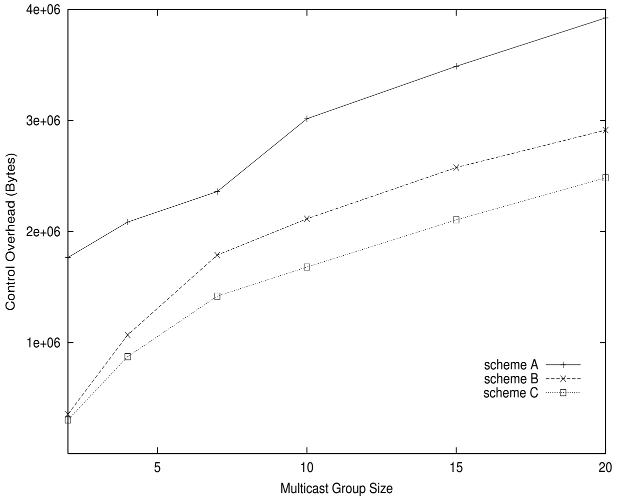


Figure 12. Control overhead as a function of number of multicast members.

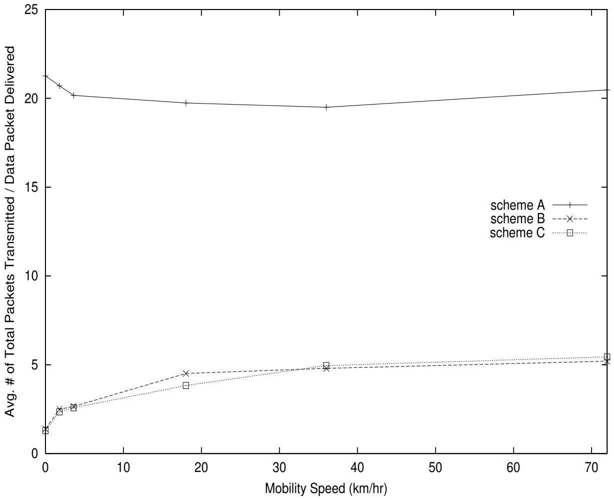


Figure 13. The Number of Total Packets Transmitted per Data Packet Delivered as a Function of Speed.

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. In fact, 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 fewer. Furthermore, data packets (whose header is calculated as control overhead), are transmitted fewer 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* goes up as mobility speed increases. Since they use mobility prediction to adapt to mobility speed, they send more Join Data and Join Tables 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 *schemes B* and *C* transmit control packets only when necessary. The enhanced schemes have more overhead when nodes move fast, but the extra control packets are used efficiently in delivering data (see Figure 7). When comparing *scheme B* with *scheme C*, we can see that *scheme B* yields more overhead in low mobility although both schemes produce nearly equal amount of overhead in high mobility. Since *scheme C* chooses a stable route, Join Data are flooded less often. When nodes move relatively fast (for example, 72 km/hr in our simulation), however, 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 overheads incurred by *schemes B* and *C* become almost identical.

In Figure 12, control overhead of all schemes increases when the number of multicast group increases. As there are more multicast receivers, more Join Tables are built and propagated. *Schemes B* and *C* have much less overhead than that of *scheme A*. *Scheme A* periodically sends Join Requests and Join Tables, but enhanced schemes send Join Data and Join Tables only in advance of topology changes. As expected, *scheme C* further improves *scheme B*. The number of control packet transmissions are less as *scheme C* uses stable routes.

4.3.4. Number of Total Packets Transmitted per Data Packet Delivered

The number of total packets (Join Requests, Join Tables, Join Data, Data, and active acknowledgments) transmitted per data packet delivered is presented in Figures 13 and 14. We have mentioned previously that this measure indicates the channel access efficiency. We can see the improvements made by enhanced schemes from the results. In Figure 13, the number for *scheme A* remains relatively constant to mobility speed. As shown in Figures 7 and 11, the number of data packets delivered and the amount of control bytes transmitted both decrease as mobility increases. The number for *scheme A* thus remains almost unchanging.

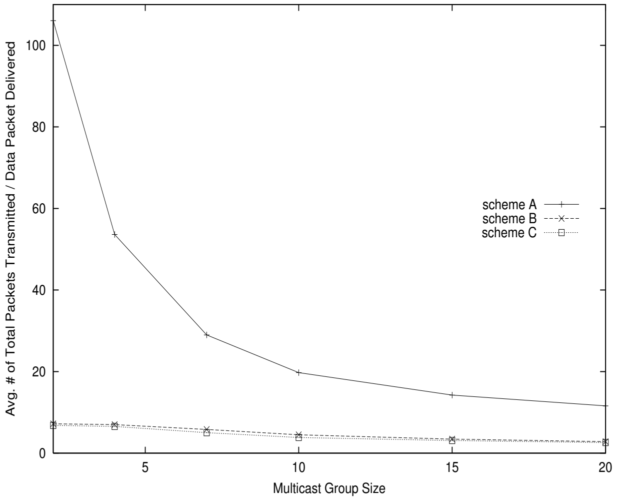


Figure 14. The Number of Total Packets Transmitted per Data Packet Delivered as a Function of Number of Multicast Members.

The measures for *schemes B* and *C* gradually increase with mobility speed. Both schemes deliver a high portion of the data to destinations regardless of speed (see Figure 7) and the number of data packets delivered remains similar. Nevertheless, more control packets must be sent in order to adapt to node mobility speed, and thus the total number of transmitted packets increases with speed.

In Figure 14, the number of all packets transmitted per data packet delivered decreases as the group size becomes larger for all schemes. This result is expected as the number of multicast members increases, the number of data packets received by members increases accordingly. Again, *schemes B* and *C* have greatly improved the efficiency of *scheme A*.

5. Conclusion

We presented new techniques to improve the performance of ODMRP. By using the mobility and link connectivity prediction, we reconstruct routes and forwarding groups in anticipation of topology changes. This adaptive selection of the refresh interval avoids the unnecessary control packet transmissions and the resulting bandwidth wastage. We 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. We used passive acknowledgments and retransmissions to improve the reliable JOIN TABLE delivery. 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 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, fewer control packets were produced in low mobility, control packets were utilized more efficiently in high mobility, and end-to-end delay was shorter. The enhanced ODMRP is scalable, robust to host mobility, and efficient in channel access.

Acknowledgements

Authors thank Dr. Ching-Chuan Chiang and Guangyu Pei for their contributions.

References

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

[2] 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.

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

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] J. J. Garcia-Luna-Aceves and E. L. Madruga, "A Multicast Routing Protocol for Ad-Hoc Networks," *Proceedings*

of IEEE INFOCOM'99, New York, NY, Mar. 1999, pp. 784-792.

[10] 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.

[11] 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.

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

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

[14] 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.

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

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

[17] S.-J. Lee, W. Su, and M. Gerla, "On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks," *Internet-Draft*, draft-ietf-manet-odmrp-02.txt, Jan. 2000, Work in progress.

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

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

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

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

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

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

[24] W. Su, S.-J. Lee, and M. Gerla, "Mobility Prediction and Routing in Ad Hoc Wireless Networks," *International Journal of Network Management*, to appear, Wiley & Sons, 2000.

[25] 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.

[26] 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`

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

Biographies

Sung-Ju Lee [ACM S] received his B.S. degree in Computer Science and Engineering from Hanyang University, Korea in February 1996. He obtained his M.S. degree in June 1998 and is expected to receive PhD degree in December 2000, both in Computer Science from University of California, Los Angeles. He is a graduate student researcher and a member of Wireless Adaptive Mobility (WAM) Laboratory at UCLA. His research interests include routing, multicasting, IP networking, mobile computing, wireless networks, ad hoc networks, congestion control, mobile IP, and network performance evaluation. He is a student member of ACM and IEEE. Email: sjlee@cs.ucla.edu

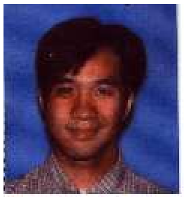
William Su received his B.S., M.S., and PhD in Computer Science from University of California, Los Angeles in 1994, 1996, and 2000, respectively. He joined the Southwestern Bell Communications in January of 2000 but left SBC to join the eWings Technologies in May 2000. His area of research includes ad hoc network protocol design and simulation, wireless ATM networks, and mobile IP. He is an IEEE member. Email: wsu@ewingstech.com

Mario Gerla received his graduate degree in Electrical Engineering from Politecnico di Milano, Italy in 1966 and his M.S. and Ph.D. degrees in Computer Science from University of California, Los Angeles in 1970 and 1973, respectively. From 1973 to 1976, Dr. Gerla was a manager in Network Analysis Corporation, Glen Cove, NY, where he was involved in several computer network design projects for both government and industry, including performance analysis and topological updating of the ARPANET under a contract from DoD. From 1976 to 1977, he was with Tran Telecommunication, Los Angeles, CA, where he participated in the development of an integrated packet and circuit network. Since 1977, he has been on the Faculty of the Computer Science Department of UCLA. His research interests include the design, performance evaluation, and control of distributed computer communication systems and networks. His current research projects cover the following areas: design and performance evaluation of protocols and control schemes for ad hoc wireless networks; routing, multicasting, congestion control and bandwidth allocation in wide area networks, and; traffic measurements and characterization. Dr. Gerla currently serves on the editorial board for IEEE/ACM Transactions on Networking and ACM/Baltzer Journal of Wireless Networks (WINET). Email: gerla@cs.ucla.edu

Photos

Sung-Ju Lee

William Su



Mario Gerla