Comparing Multicast Protocols in Mobile Ad hoc Networks[1]

Robert C. Durst, Keith Scott, Mary Jo Zukoski
The MITRE Corporation
1820 Dolley Madison Blvd.
McLean, VA 22102
(703) 883-6547
{durst, kscott, mzukoski}@mitre.org

Cauligi S. Raghavendra
The Aerospace Corporation
P.O. Box 29257
Los Angeles, CA 90009
(310) 336-1686
raghu@aero.org

Abstract—This paper presents a novel method for comparing the performance of multicast protocols in mobile ad hoc networks (MANETs) with respect to delay, efficiency, and robustness. For each metric, we develop the notion of optimality that allows us to compare performance against a common set of reference points. Knowing where a protocol's performance diverges from optimal also indicates where efforts at improvement will provide the most benefit. Next we present a cross-platform physical layer model that allows researchers working with different simulation packages use a common physical model of the network. We then use simulations to illustrate the performance of a candidate protocol relative to optimal. An interesting result of the simulations is that MANET nodes with very high degrees of connectivity do not necessarily improve the performance of multicast routing protocols.

TABLE OF CONTENTS

1. INTRODUCTION
2. THE PROBLEM
3. OPTIMAL VALUES FOR DELAY, EFFICIENCY, AND ROBUSTNESS
4. CROSS-PLATFORM NETWORK SCENARIOS
5. SIMULATION RESULTS
6. CONCLUSIONS
7. FUTURE WORK

1. INTRODUCTION

Recent advances in wireless technologies have brought sweeping changes to computer communications. Small, low-power, lightweight wireless modems allow users with laptop or palmtop devices easy access to the Internet as well as to private networks at speeds that can outstrip conventional telephone modems. These devices have become small and inexpensive enough that they are available to the general public. This availability has, in turn, fostered deployment of wireless access networks such as Metricom's Ricochet [1].

In addition to providing access to fixed networks by mobile users, these advances have allowed entirely new types of networks to be envisioned – mobile ad hoc networks. Unlike wireline or cellular networks, ad hoc networks have little or no fixed infrastructure. Instead the network users, each generally mobile and acting as a router, cooperate among themselves to form and maintain the network. These Mobile, Ad hoc NETworks, or MANETs, are currently being investigated by the Internet Engineering Task Force's MANET working group, among others, for military, civilian (e.g. disaster relief, industrial floor communications, conferencing), and scientific (e.g. networks of semi-autonomous probes) applications.

Along with the growth of wireless technologies, we have seen in recent years an increased interest in multicasting, especially in MANETs. Multicasting allows a user to transmit the same information to a group of receivers simultaneously and in an efficient manner. Examples of multicast applications include audio- and video-teleconferencing among multiple users, shared whiteboards among multiple users, some subtasks of parallel computing, and "push" technology that distributes information from a central point to a set of receivers or caches.

While recent advances in wireless technology have enabled multicasting in MANETs, it is up to protocol designers to provide the rules that define how those communications will take place. As might be expected, advances at the physical layer have first been reflected in medium access and data link protocols. For example, the new wireless ethernet standards [2] have closely followed new advances in wireless transceiver design. The network layer, the layer at which routing decisions are made, is not far behind, and it is at this layer that critical decisions about multicasting need to be made.

Several competing criteria affect the design of a multicast protocol. For example, one can attempt to minimize the

[1] 0-7803-6599-2/01/$10.00 © 2001 IEEE

total number of transmissions needed to disseminate messages, to minimize the average or maximum time to deliver a message, or to maximize the probability that a message is received by all members of the group.

Generally speaking, the more robust a protocol is, the less efficient it is and vice-versa, since an efficient protocol transmits fewer messages and hence the loss of a single message is more detrimental. Conversely, a more robust protocol may transmit more copies of a message than are strictly necessary in order to ensure correct receipt of at least one copy, and thus suffers in terms of efficiency. Similarly, efficiency and delay may be at odds, as a protocol designed to minimize delay may do so by transmitting more copies of a message than are strictly needed.

In this paper we develop the notion of optimality with respect to three metrics providing quantitative measures of key performance properties: delay, efficiency, and robustness. The notion of optimality provides a number of advantages. First, protocol designers can compare their protocol's performance against a common reference point without simulating other protocols. Second, performance relative to optimal can provide suggestions as to where efforts at improvement will provide the most benefit; and a particular protocol can be compared against the best-case solutions for a variety of network sizes, yielding information about the protocol's scalability.

Because the various metrics have contradictory aims, the results must be considered as a whole when choosing a protocol for a particular situation. In particular, since the metric definitions are made in the context of 'rational' protocols (i.e. protocols that do indeed try to deliver data to all intended recipients) it is possible for a protocol to perform "better than optimal" with respect to certain metrics by sacrificing performance with respect to others.

The rest of the paper is organized as follows. Section 2 discusses multicast protocols for MANETs and the need for a common basis to compare the applicability of different protocols to specific scenarios. Section 3 introduces the metric set and defines the measurements of optimality that are used as the basis of comparison. Section 4 presents our cross-platform physical layer model. Section 5 uses the metrics and cross-platform physical model to evaluate the performance of a specific multicast protocol. Section 6 presents some conclusions, followed by planned extensions to the current work in section 7.

2. The Problem

The development of multicast algorithms for MANETs is still an area of research. Currently, protocols being developed from these algorithms [3, 4, 5, 6] are not necessarily suitable for all environments or all applications. Indeed it has been suggested that no one protocol will ever dominate in all situations due to the wide dynamic range of parameters in MANETs [7]. Instead it seems more promising, at least for the near-term, to be able to select an appropriate multicast protocol for a given set of parameters.

Unfortunately it can be difficult to compare multicast protocols against one another in an attempt to determine which one is best suited for a particular application. Generally the protocol architects provide some analytic and/or simulation results, and possibly information about the range of mobility the protocol is designed to support, the recommended denseness/sparseness of multicast nodes as a proportion of total nodes, etc. When comparisons are made to other protocols, however, subtle differences in simulation and/or modeling techniques including the choice of the mobility model (Brownian motion, constant motion along a vector, move-and-stop, etc.) and the RF model can greatly impact the results.

The delay, efficiency, and robustness metrics capture fundamental performance measures of multicast protocols, and can be used to compare uniformly the behavior of different multicast protocols. Note that the notion of using a set of quantitative metrics to compare networking protocols is not new. In [8] for example, the authors catalog a number of qualitative features desirable in MANET protocols, and list several quantitative measures that can be used to gauge performance, including delay and efficiency. Here in addition to presenting an example of applying such quantitative metrics to a concrete case, we introduce a notion of optimality that has several interesting applications.

Our notion of optimality provides a common basis for comparing protocols, as well as three added advantages. First, it allows protocol designers to compare their protocol's performance against a common reference point that does not require them to simulate other protocols. Second, it provides information about which areas of a protocol's performance have high potential for improvement. Finally, by comparing the performance of a protocol against optimality as network size increases, the optimal metric values provide information about scalability. Key to the method of comparison we suggest is the use of a common engine to generate topology changes in the network, and the use of this common "network evolution" in future comparison among different protocols.

Model Assumptions

We assume a graph $G = (V,E)$ where the nodes of V represent nodes of the mobile network, and links in E represent possible connectivity between nodes. We say possible connectivity because we assume that a node can receive from at most one neighbor concurrently. Thus if i transmits to j, node j receives the transmission iff the signal to noise ratio at j is above some threshold. If another neighbor of j, say k, is transmitting at the same time as i, node j receives neither j's nor k's transmission.

Calculation of the optimal metric values for a multicast protocol depends heavily on assumptions about the lower layers of the network. The optimal value for delay, for example, is affected by whether or not one assumes a broadcast capability at the link layer, the radio frequency (RF) interference model, and the medium access control (MAC) scheme, among others. For this paper we make the following assumptions about the underlying communications layers. These assumptions are used both to calculate the optimal metric values and in the simulations.

- The underlying link layer is broadcast, i.e. when one node wishes to send the same data to several of its neighbors, it only needs to transmit that data once, provided that this transmission does not conflict with what the neighbors are doing.
- A given node can receive a communication from at most one neighbor at a time. That is, two nodes cannot transmit to a third concurrently and be heard.
- When calculating the optimal metric values, we assume a perfect, zero-delay MAC layer.
- We assume that all radios have the same transmission rate.

The simulations make slightly more realistic assumptions about the lower network layers. A more detailed description of the simulation models is deferred until section 0.

3. Optimal Values for Delay, Efficiency, and Robustness

Affects of Link Layer Assumptions on Optimal Metric Calculations

As mentioned above, assumptions about the underlying link-layer technology affect the calculations for the optimal delay, efficiency, and robustness metrics. For example, in a broadcast environment, one node can transmit to multiple receivers at once, thus taking less time to forward a multicast packet through a fork in the multicast tree than if a point-to-point link layer is used. Similarly for efficiency, if a broadcast model is used then a node can, with one transmission, cover all next-hop nodes, whereas with a point-to-point model several such transmissions are needed. The drawback to assuming a broadcast model at the link layer is that if nodes can only receive from one source at a time then scheduling becomes a major issue in the delay calculation. Also, a broadcast model at the link layer prohibits the use of standard Steiner tree calculations that yield the optimally efficient multicast tree for unicast link-layer networks.

The next sections define the delay, efficiency, and robustness metric calculations we use in section 5 as well as lower bounds and heuristics for the optimal metric values.

Delay

Let the time the j^{th} packet is emitted be t^j, and let t_i^j be the time the packet is received by the i^{th} receiver of the multicast group, and let the number of members in the group be M. If a particular packet is never received by the i^{th} group member, we let t_i^j be $-\infty$. Thus protocols that are lax about assuring delivery of multicast packets will have low delay metrics. Such protocols will be punished, however, since they will have correspondingly low robustness metrics (section 3.3). We define the delay of the j^{th} packet as $\max_i \left(t_i^j - t^j\right)^+$ where $(X)^+$ is the maximum of X and 0. The average delay experienced by the multicast group during a specific time window W_i can thus be expressed as:

Delay during window W_i =

$$\frac{\sum_{j=1}^{\infty} \max_i \left(t_i^j - t^j\right)^+ * \mathrm{I}(t^j \in W_i)}{\sum_{j=1}^{\infty} \mathrm{I}(t^j \in W_i)} \qquad (1)$$

Where I( ) is the indicator function.

Note that the delay of the j^{th} packet is lower-bounded by relaxing the assumption that a node can receive from at most one neighbor at a time. In this case the delay is given by the packet transmission time T^j times the number of hops in the longest shortest path to any receiver in the group. Thus if there are h_i hops in the shortest path from the sender to receiver i, a lower bound on the optimal delay D^j is:

$$D^j \geq T^j * \max_i h_i \qquad (2)$$

Under the assumptions of a broadcast link layer where any node can receive from at most one other node at a time, this simplistic lower bound may not be tight. This situation is illustrated in Figure 1, where the maximal shortest path from the sender to a multicast receiver is 2, but it takes at least three separate transmissions to disseminate a message. The extra transmission is required because the two unmarked nodes each need to forward the message, but they cannot transmit concurrently as their transmissions would interfere with one another at the destinations at the top of the figure.

Note that the bound is not necessarily tight if we assume a unicast link layer technology, either. As a counterexample, consider a star network with a source directly connected to

N receivers. The maximal shortest path is 1 hop, but the source has to transmit to each of the N receivers serially, taking N^*T_j seconds.

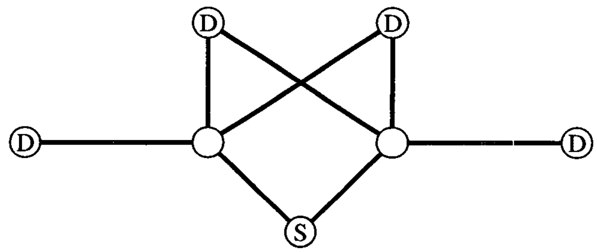


Figure 1: A case that requires more transmissions than there are links in the maximal shortest path from the sender (S) to any destination (D).

It is possible to compute the optimal delay with the assumption that a node can receive from at most one neighbor at a time [9, 10]. This is a rather complex optimization problem that we have not yet implemented, though we plan to do so in the future.

Efficiency

The efficiency metric measures how well a multicast protocol uses the underlying network and consists of two parts. First, there are the transmissions needed to actually disseminate multicast messages across the fabric. Second, there may be control messages needed to set up and to maintain that fabric. Let the number of information bits in the j^{th} multicast message be I^j. In general, each multicast message will be transmitted several times in order to traverse different links in the network. Let the j^{th} message be transmitted M^j times with the number of bits in the m^{th} transmission (including overhead) being N_m^j. Here we are counting each transmission of the message, and multiple simultaneous transmissions (by different nodes) are counted separately. Similarly, let the number of copies of a control message originally sent at time t^k be K^k, with the message's size (in bits, including any overhead) C_k. Again, if a particular control message traverses multiple links it is counted multiple times, once per transmission, and multiple simultaneous transmissions of a control message by different nodes are counted separately. The reason for the difference in notation is to allow for the possibility that different copies of the data-bearing packet have different lengths.

With these definitions the efficiency of a multicast protocol during a particular time window W_i is then defined as:

Efficiency during window W_i =

$$\frac{\sum_{j=1}^{\infty} I^j * \mathrm{I}(t^j \in W_i)}{\sum_{j=1}^{\infty}\sum_{m=1}^{M^j} N_m^j * \mathrm{I}(t^j \in W_i) + \sum_{k=1}^{\infty} C_k * K^k * \mathrm{I}(t^k \in W_i)} \tag{3}$$

We can lower bound the optimal efficiency of a multicast protocol by assuming that it needs no control messages, that nodes can receive multiple messages simultaneously, and that the per-packet overhead is zero. In this case, we can compute the minimum number of bits needed per multicast transmission by computing the Steiner network tree for a graph describing the wireless network and assuming that the multicast protocol transmits across the links in this tree.

Recall that the Steiner network problem is: given a weighted graph $G = (V, E)$ and a set $T \subseteq V$ of target nodes, find the minimum cost subtree of G that spans all nodes of T. If a unicast link layer mechanism is used, this yields the minimum number of links (and hence the minimum number of transmissions) that must be used to distribute multicast packet. If a broadcast link layer mechanism is used then we want to count the number of packet transmissions, not the number of links used. In this case we can modify the graph used to compute the Steiner tree by inserting fictitious unidirectional links with unity cost between the input and output of each node, and assigning no cost to the links in the original graph G. Unfortunately, computing the Steiner tree for graphs with unidirectional links is difficult, and for our optimality results presented later we resorted to a heuristic that computes optimal link-based (as opposed to broadcast-based) Steiner trees and converts them to broadcast-based trees by pruning.

Robustness

The robustness metric attempts to quantify a protocol's ability to deliver packets to all members of the multicast group in the presence of unreliable links and interference. There are a number of possible definitions for the optimal robustness value of a multicast protocol. One could compare packets received using a particular protocol to those received when flooding is used, for example.

Here we define the optimal robustness of a multicast group g with sender s as the fraction of nodes in the network graph G that are reachable from s at the time the multicast message is first transmitted by the sender.

We define the robustness of a multicast protocol averaged over a time window W_i as:

Robustness during window W_i =

$$\frac{\sum_{j=1}^{\infty} I(t_i^j > 0) I(t^j \in W_i)}{M * \sum_{j=1}^{\infty} I(t^j \in W_i)} \quad (4)$$

Equation 4 simply counts the number of received copies of multicast packets divided by the number of multicast members times the number of packets transmitted.

4. Cross-Platform Network Scenarios

One thing that drives the lack of continuity between different researchers' simulation results is the lack of a common physical model for how the network evolves with time. This modeling task is usually broken into two parts: 1) modeling how the nodes move relative to one another and 2) modeling the RF characteristics given the nodes' positions. Early approaches to modeling node movement include Brownian motion and move-pause-move models. Approaches to modeling the RF propagation range from simple free-space propagation models to r^4 loss models, to more complicated Rician fading models. The result is that most simulations use slightly different physical models, complicating the job of comparing results.

We have developed a set of cross-platform network scenarios designed to capture interactions among different nodes of a MANET. The model allows us to calculate the RF characteristics of users in highly realistic scenarios using a combination of two tools, Modular Semi Automated Forces (MODSAF) and Terrain Integrated Rough Earth Model (TIMREM). MODSAF simulates movement of military units working together to accomplish an objective. MODSAF scenarios range from vehicles travelling down a road, to groups of infantry units moving through urban terrain, and MODSAF's output is a sampled version of the node position information. We then feed this position information into TIREM, which determines the RF characteristics between points taking into account such factors as: the heights of the transmitting and receiving antennas, transmission frequency, antenna polarization, permittivity, terrain, refractivity and conductivity of the surface over which communications are taking place, and humidity [11]. TIREM accounts for losses due to diffraction and tropospheric scatter, and determines the dominant (least loss) path between all transmitter/receiver pairs.

Thus for each sampled instant in time, TIREM outputs a path loss matrix describing the path gains (losses) between all pairs of nodes in the simulation, whose positions are given by the ModSAF output. These path loss files can be read by OPNET[12], GLOMOSIM[13], and soon by ns, providing true cross-platform physical layer models.

This model provides very realistic models for the RF characteristics between nodes. For example, if users A, B, and C are in a North-South line with A the Northernmost and B in the center, and if B moves North, the path loss matrices will reflect a decreased path loss between A and B, while showing a correlated increased loss between B and C. If A then passes behind an obstruction with respect to B, the path loss between A and B may very well exceed that between B and C, even though A and B are physically closer together.

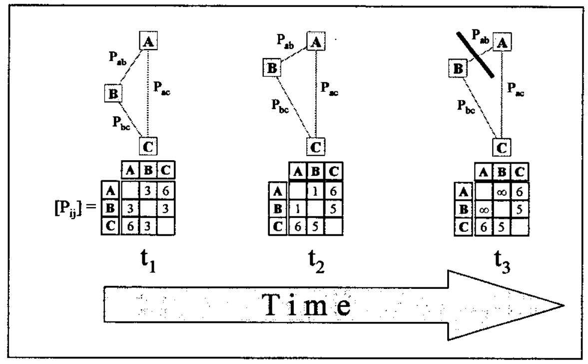


Figure 2: Illustration of the time-evolution of a path loss matrix.

Even with a common physical layer model, the behavior of a particular protocol depends on how the matrices are used by the particular simulation engine. We are working to find parameter settings to equate the GLOMOSIM and OPNET modeling environments.

5. Simulation Results

This section compares the performance of an ODMRP model for UCLA's GLOMOSIM simulation environment to the optimal metric values. Both the simulation and the optimal metric calculations are driven by the physical-layer model of the previous section.

Network Scenarios

The example presented here uses an infantry urban terrain (IUT) scenario consisting of five squads of five members each moving through flat, urban terrain at a rate of roughly 3 km/hr. All of the nodes are clustered within 1 square kilometer, and their positions and RF path losses are sampled every 10 seconds. Using the design parameters of the simulation (transmission powers and required received signal strengths), all members of a particular squad are always fully connected.

Figure 3 shows the connectivity of the IUT scenario at time 0. If the pair of nodes (i,j) can communicate with one another (i.e. the path loss between them is less than or equal to 130dB), then the [i,j] entry in the grid has a black border. If the path loss between two nodes is such that they cannot

0	111	120	108	109	131	136	135	134	130	133	134	138	139	137	138	141	143	143	139	143	146	145	145	142
111	0	112	108	120	125	131	130	130	123	133	133	137	139	137	135	138	141	141	137	141	143	142	142	139
120	112	0	109	122	125	131	131	126	119	128	127	133	135	134	131	134	138	138	132	140	143	141	141	138
108	108	109	0	111	129	134	134	131	126	129	129	135	136	134	135	138	141	140	135	142	145	144	143	141
109	120	122	111	0	134	138	138	136	132	129	132	136	137	134	138	141	143	142	138	145	147	146	146	144
131	125	125	129	134	0	112	109	120	112	138	136	139	142	141	133	134	138	140	136	132	136	135	135	130
136	131	131	134	138	112	0	92	120	121	140	138	140	143	143	134	133	138	140	137	125	131	130	130	124
135	130	131	134	138	109	92	0	122	121	141	139	141	143	143	135	135	139	140	138	127	132	131	132	126
134	130	126	131	136	120	120	122	0	111	135	132	134	138	138	123	124	131	133	129	132	135	133	132	128
130	123	119	126	132	112	121	121	111	0	134	131	135	138	138	128	130	135	136	132	134	138	136	135	132
133	133	128	129	129	138	140	141	135	134	0	111	120	120	111	131	136	137	135	128	146	147	146	145	144
134	133	127	129	132	136	138	139	132	131	111	0	112	120	120	125	131	133	130	121	144	145	144	143	141
138	137	133	135	136	139	140	141	134	135	120	112	0	111	120	125	131	131	126	117	145	146	145	143	142
139	139	135	136	137	142	143	143	138	138	120	120	111	0	111	131	135	135	131	126	147	148	147	146	145
137	137	134	134	134	141	143	143	138	138	111	120	120	111	0	133	138	138	135	130	147	149	148	147	146
138	135	131	135	138	133	134	135	123	128	131	125	125	131	133	0	111	120	120	109	139	140	138	136	135
141	138	134	138	141	134	133	135	124	130	136	131	131	135	138	111	0	112	120	120	136	138	135	132	132
143	141	138	141	143	138	138	139	131	135	137	133	131	135	138	120	112	0	112	122	139	140	138	135	136
143	141	138	140	142	140	140	140	133	136	135	130	126	131	135	120	120	112	0	116	142	143	141	139	139
139	137	132	135	138	136	137	138	129	132	128	121	117	126	130	109	120	122	116	0	141	143	141	139	138
143	141	140	142	145	132	125	127	132	134	146	144	145	147	147	139	136	139	142	141	0	110	110	119	111
146	143	143	145	147	136	131	132	135	138	147	145	146	148	149	140	138	140	143	143	110	0	104	117	118
145	142	141	144	146	135	130	131	133	136	146	144	145	147	148	138	135	138	141	141	110	104	0	107	110
145	142	141	143	146	135	130	132	132	135	145	143	143	146	147	136	132	135	139	139	119	117	107	0	111
142	139	138	141	144	130	124	126	128	132	144	141	142	145	146	135	132	136	139	138	111	118	110	111	0

Figure 3: Path loss matrix at time 0 for the IUT Scenario.

communicate, there is no border.

In addition, we have made the following assumptions:

- Transmission power: 10dBm
- Required received signal strength: -119dBm
- Transmission rate: 250kbps
- MAC protocol: MACA
- Unicast Routing protocol: WIRP. Note: ODMRP does not depend on the unicast routing protocol. In our simulations, WIRP was simply a source of background unicast traffic.
- No AWGN ($\eta = 0$). The only interference is from other transmissions.

For voice calls, we assumed constant-bit-rate encoding using 63 bytes of data every 100ms. We implemented the voice calls using GLOMOSIM's constant-bit-rate (CBR) client/server models, so that each 63 byte packet acquired 8 bytes of UDP header, 20 bytes of IP header, and 18 bytes of ODMRP header, for a total of 109 bytes, by the time it reached the medium access layer. We chose to run over MACA, which added another 20 bytes of header. Thus a single voice call provided:

(63 [Application Data] + 8 [UDP] + 20 [IP] + 18 [ODMRP] +20 [MACA]) * 8 * 10 = 10,320

bits per second of load from the source. Note that packets might need to be forwarded some number of times in order t reach all the receivers. Thus the effective offered load of a voice call could be much higher. For example, if a particular sender's packets need to be forwarded four times in order to reach all receivers, this would cause roughly 41 kbps of network traffic.

Situation awareness (SA) data conveys information about a unit's position. We modeled an SA update as a 50-byte UDP packet, and each unit sent SA updates once every 30 seconds. This caused only

(50 [Application Data] + 8 [UDP] + 20 [IP] + 18 [ODMRP] + 20 [MACA])*8 / 30 ~= 31

bits per second of data to be sourced by each node, but the infrequency of the data coupled with some ODMRP timer values actually caused all situation awareness messages to be flooded.

Table 1 lists the multicast groups defined in the simulations. We defined 5 intra-squad groups, each consisting of the five members of one of the fully-connected squads, a squad leaders group containing one member from each of the five squads, and a group containing all nodes in the network for distributing SA data.

Table 1: Multicast Groups Used in the Simulations

<table>
<tr><th>Group</th><th>Nodes</th><th>Description</th></tr>
<tr><td>Squad Leaders</td><td>0, 5, 10, 15, 20</td><td rowspan="6">Each of these multicast groups supported voice calls. Any node could source data to the entire group.</td></tr>
<tr><td>Squad 1</td><td>0, 1, 2, 3, 4</td></tr>
<tr><td>Squad 2</td><td>5, 6, 7, 8, 9</td></tr>
<tr><td>Squad 3</td><td>10, 11, 12, 13, 14</td></tr>
<tr><td>Squad 4</td><td>15, 16, 17, 18, 19</td></tr>
<tr><td>Squad 5</td><td>20, 21, 22, 23, 24</td></tr>
<tr><td>Situation Awareness (SA)</td><td>All (0—24)</td><td>This multicast group supported situation awareness data. All nodes sourced data to the entire group.</td></tr>
</table>

GLOMOSIM ODMRP Model

This subsection describes the GLOMOSIM ODMRP model and parameters used.

We obtained the GLOMOSIM ODMRP model from the author through our research partners at The Aerospace Corporation. To run ODMRP in our target environment including unicast background traffic and multiple multicast groups required modifications to both the ODMRP model and to the GLOMOSIM environment:

ODMRP under GLOMOSIM 1.2.3: We extended GLOMOSIM 1.2.3's capabilities to allow the use of both a (single) unicast protocol and a (single) multicast protocol concurrently. That is, with our extended version it is possible to specify one unicast and one multicast routing protocol. All addresses above a certain compile-time threshold are assumed to be multicast addresses. Our intent was to use traffic profiles containing both the multicast traffic of interest and unicast background traffic. Our initial results showed that simply maintaining adequate robustness in the face of the multicast traffic was difficult enough, and we have postponed simulations with unicast traffic.

Radio Capture Model: We wanted to use the GLOMOSIM radio model that includes capture, the phenomenon that once locked onto a radio signal, a receiver can continue to receive that signal even if the interference increases substantially. In version 1.2.3 of GLOMOSIM, however, it appears that the radio model was in a state of flux between representing received signal and noise powers in Watts and representing them in dBm. The values passed into the various functions are definitely in dBm, but they are operated on as if they are in Watts. This problem was independently noted by GLOMOSIM's maintainers, who have reworked the radio model for the next release. This problem was also corrected for the results presented here.

Message Queueing Fix: GLOMOSIM contains a notion of resource locking within a node. The radio (or Network Interface Card, NIC) is a scarce resource, for example. The radio can only transmit one packet at a time, and cannot transmit while receiving, for example. To ensure that such resources are in fact used by only one process at a time within the node, GLOMOSIM's basic event processing mechanism queues messages that require a resource that is unavailable. Queued messages are then delivered when the resource becomes free. Unfortunately, while some message types (i.e. a node wanting to transmit a packet) can be queued, others, such as a node receiving a signal that a packet is incoming, can not. To remedy this we added a field to the GLOMOSIM message that explicitly states whether or not the message should obey the normal queueing mechanisms. This ensured that messages corresponding to 'external' events (arrival of a packet, end of a packet, etc.) were never queued. This problem was also independently noted by the GLOMOSIM maintainers, who will fix it in the next revision.

Receiver Synchronization and Delays: There were a number of places in the ODMRP model that imposed random delays in attempts to prevent collisions when multiple nodes receive a packet and then all attempt to forward it at the same time. These delays were generally set to be uniformly distributed over [0, 100ms), and were invoked any time a node retransmitted a packet it had received. For the multi-hop paths of the squad leaders multicast group, this caused delays on the order of 300-500 milliseconds, clearly unacceptable for voice calls. We needed to remove these delays without introducing an undue number of collisions caused by multiple receivers all trying to forward packets at the same time. To accomplish this we modified the aforementioned resource locking mechanism to include a small (~20-40 bit times) random interval after transmission and reception during which a node would remain inactive. Thus if a particular data packet is received by multiple nodes that all need to forward it, as is the case with ODMRP Join Request packets, the different receivers will choose different random wait times up to 20 bit times. Using this method, it is very likely that a single receiver begins retransmission of the packet before others have finished waiting. Once it begins transmitting, the 'early' node captures the channel and other nodes wait before retransmitting their copies of the packet.

Further ODMRP runs revealed that this simplistic approach could not be applied uniformly across all data rates. In particular, at 2Mbps, receiver synchronization of exactly the type we were trying to avoid cropped up again. We believe that the correct value for the transceiver jitter should be a function of the data rate and the parameters of the underlying MAC and physical layers. Until we understand this in greater detail we have lower bounded the jitter range to [0, 1ms) (i.e. if 20 bit times is less than 1ms, we force the range to [0, 1ms)).

Metric Calculation: We instrumented the ODMRP model to output information to allow us to calculate the delay, efficiency, and robustness metrics mentioned earlier. For each data packet, we record the send time and the receive times at each multicast destination, and this information is dumped to a file. From this we can calculate the maximum / average / minimum delay for each packet. We also instrumented the ODMRP model to track both the number of data bytes presented by the transport layer as well as the total number of bytes transmitted. These were used to calculate efficiency values that were periodically dumped to a file.

Graphical Event-Log: Finally, while GLOMOSIM contains an excellent graphical user interface (GUI) that displays transmissions in "real-time" as the simulation progresses, it is not a good tool for post-mortem analysis. By inserting some diagnostic calls into various parts of GLOMOSIM, we were able to create "strip-charts" that describe the entire system and that graphically depict certain events such as transmissions, receptions (both successful and failed), data passed between ODMRP and the transport layer, etc.

Figure 4 shows a simplified view of such an event log showing the evolution of an ODMRP simulation over 300ms of activity across 10 nodes. The timeline for what happens at each node is shown as a thin horizontal line, with filled boxes above the line indicating transmissions and filled boxes below the line indicating successful receptions. The gray horizontal arrows indicate *failed* receptions, i.e. cases where a signal was detected but was not locked onto. The larger arrows that point down and to the right show the injection of a particular packet by node 0 and its reception by group members 5 and 10.

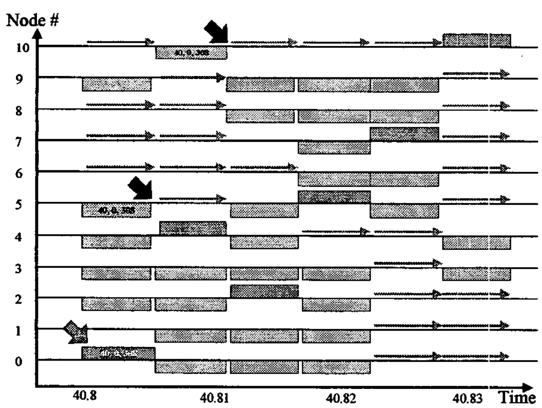


Figure 4: Graphical event log incorporated into GLOMOSIM.

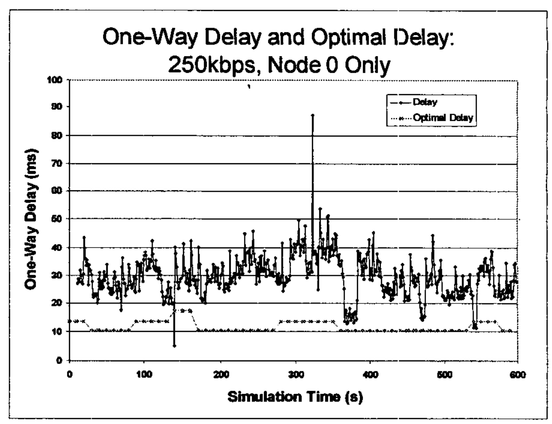
The following parameters were used for the simulations

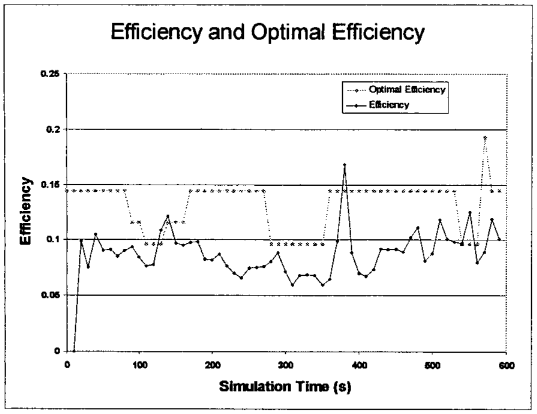
- GLOMOSIM's RADIO-CAPTURE model with the following parameters:
  - Receiver threshold: -119dB
  - Propagation Limit: -131dB
  - KEEP_CURRENT_LOCK_THRESH, TRY_NEW_LOCK_THRESH: 10dB
- ODMRP Join Table Refresh Time: 10s
- ODMRP Forwarding Group Timeout Value: 22s

Node 0 Sourcing a Single Voice Call to the Squad Leaders Group

Figure 5 shows the delay, efficiency, and robustness metrics for the case of node 0 transmitting to the squad leaders group. The optimal metric results are overlaid in gray. There are two interesting points about ODMRP's performance. First, the delay for ODMRP dips slightly below the optimal delay around time 140s and again around 540s. This is because ODMRP's robustness at these times fell precipitously, and since the delay metric is defined as the difference between when a packet is injected and the time the last receiver *to get the packet* receives it, ODMRP is able to sacrifice robustness for delay. This underscores the necessity to examine the entire suite of metrics before passing judgement on a particular protocol.

For the simulation results, we actually plot the delay and robustness of each data packet. The window size used to calculate the efficiency values is 10 seconds.





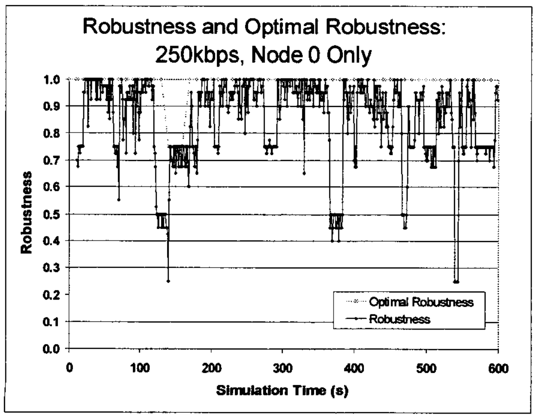


Figure 5: Delay, Efficiency, and Robustness for the ODMRP model with node 0 transmitting to the squad leaders multicast group.

Performance With All Multicast Calls Enabled

The previous section looked at performance when there was a single multi-hop, multicast voice call in the network. This section examines performance when there can be multiple concurrent calls in multiple groups. That is, for this section we have turned on the squad-leaders group, each of the five intra-squad groups, and situation awareness traffic. The results presented here are the averages over five runs with different traffic profiles (call start/stop times). The various traffic profiles were all generated to provide an average of 25% of offered load to the network.

Figure 5-4 shows the offered, effective offered (i.e. offered load when using the optimally efficient multicast tree), and carried loads when all multicast groups of Table 1 are enabled. Most of the time the carried load is greater than 1, indicating that there is usually at least one node transmitting somewhere in the network at all times. It is interesting to note that the infrequency with which SA data were transmitted actually worked against ODMRP for most reasonable configurations. Recall that ODMRP sources will send join requests to rebuild the mesh whenever: 1) more than the Join Table Refresh Time has passed since the last data join was sent and 2) no join table was received for the last MULTICASTROUTING_ODMRP_CONGESTION_TIME seconds. The model as provided had MULTICASTROUTING_ODMRP_CONGESTION_TIME set to 250ms, ensuring that an SA source would send a new join request after the join table refresh period. Thus if the refresh period is less than 30 seconds (as seems reasonable in a network with quickly changing topology), all SA data end up being flooded.

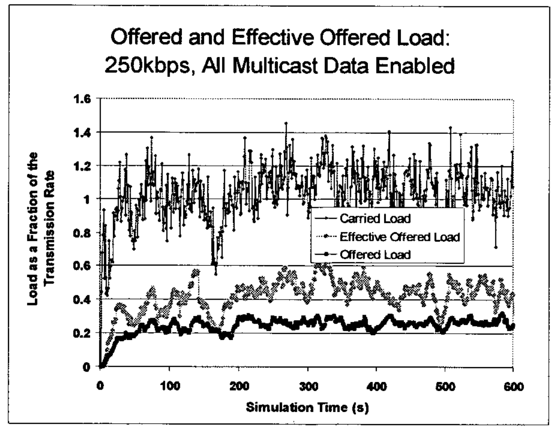


Figure 6: Offered and effective offered load when all multicast groups are active.

Figure 7 shows the average robustness metric for the different multicast groups as functions of time during the simulations. In Figure 7 we have plotted the different multicast groups in different ranges of the Y-axis, so that the robustness of squad 1 falls between 1.0 and 2.0, the robustness of squad 3 between 3.0 and 4.0, etc. For reference, the optimal robustness for the squad leaders group is shown in gray and reflects that squad 5 becomes disconnected from the rest of the simulation between 140s and 170s.

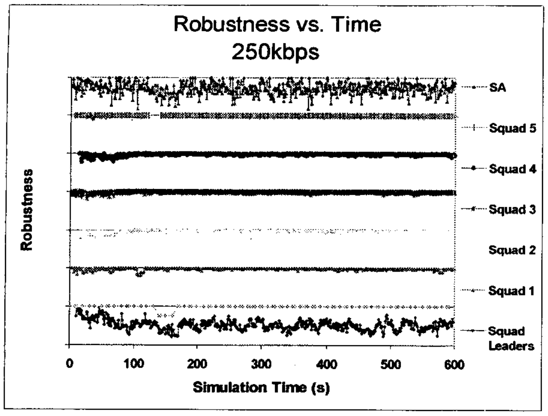


Figure 7: Robustness of the various multicast groups.

While the robustness of the intra-squad voice calls was very good, both the squad leaders calls and the situation awareness traffic performed poorly. That the SA traffic did poorly was particularly surprising in light of the fact that all SA data were flooded, as mentioned above. The reason for the degradations lies in ODMRP's assumption of a broadcast channel coupled with MACA's unreliable broadcast at the link layer. These two factors leave our ODMRP implementation open to the hidden terminal problem. Floods of SA data generated large amounts of traffic in the network, which actually self-congested the network and caused packet loss.

Figure 8 shows the one-way delays when all multicast groups are active.

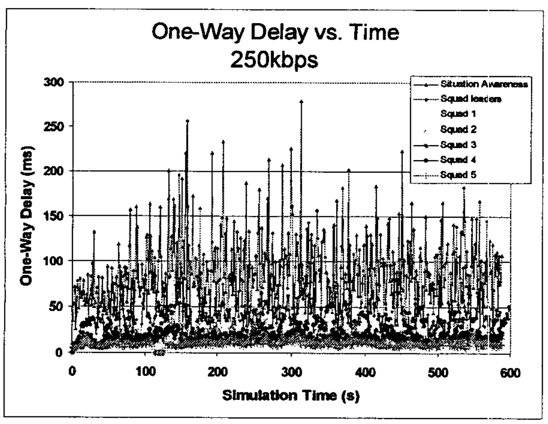


Figure 8: One-way delays when all multicast groups are active.

To understand why the performance of the squad leaders multicast group was so poor, we looked at the fraction of time an average network node had available to transmit when there was a single call on the squad leaders group (i.e. using the setup of the previous section). Figure 9 shows the capacity available to each radio in the network. That is, the figure shows the fraction of time when the nodes' radios were neither transmitting nor receiving a packet. It was somewhat surprising that when a single multicast voice call in the squad leaders group was active, roughly 50% of the network bandwidth was consumed *at every node*. While this is a function of the simulation parameters for received signal and interference thresholds, it is relatively constant with these parameters. The contributions of the unicast routing traffic and the multicast voice call are not broken out in Figure 9. In fact the unicast routing traffic alone used roughly 10-15% of the network capacity, so that the multicast voice call was actually using only 35-40% of the network.

For each of the five different traffic profiles we used, there were many times when multiple squad leaders were transmitting simultaneously. While this might not occur in practice with voice calls, it could certainly be the case if the network were carrying data instead. In any case, even two such calls active at the same time required more resources than the network had. This, coupled with the added intra-squad and SA traffic caused squad leader traffic to be lost.

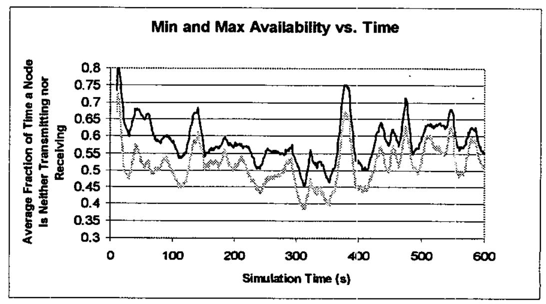


Figure 9: Minimum and maximum average availability when node 0 is sourcing a single voice call to the squad leaders group.

The Affects of Data Rate

One way to increase the performance, particularly the robustness of the squad leaders multicast group, is to increase the radio transmission rate. Figure 10 shows the difference in robustness between a 250kbps data rate and a 2Mbps rate for one simulation run with node 0 transmitting to the squad leaders group. Figure 11 shows the robustness of the squad leaders and situation awareness multicast groups vs. the radio channel data rate, averaged over 5 runs.

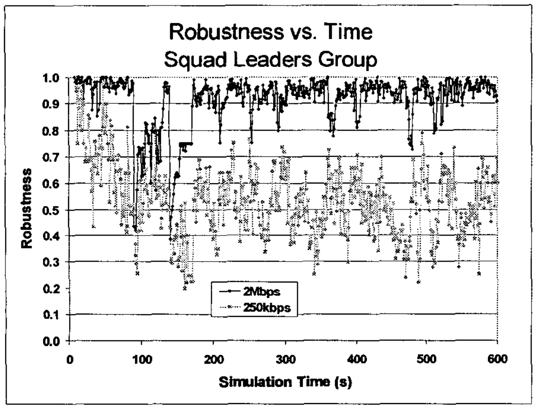


Figure 10: Robustness of the squad leaders group with a 2Mbps data rate.

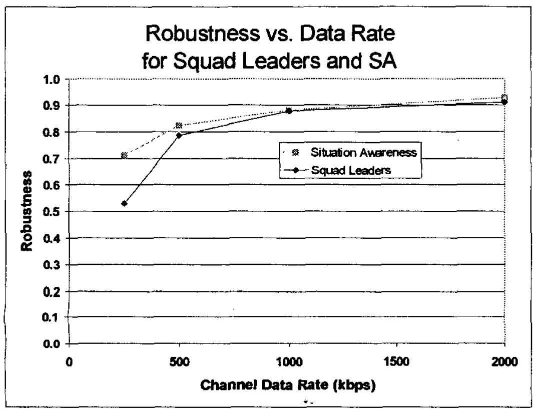


Figure 11: Robustness as a function of data rate.

Performance With an Advantaged Node

We also examined performance of the IUT scenario with an unpiloted aerial vehicle (UAV) that had global connectivity. Specifically, the path loss from the UAV to each of the terrestrial nodes was 111dBm. We hoped that routing data for the squad leaders and perhaps the SA groups over the UAV would alleviate congestion in the terrestrial network. While this was the case, improvements were not nearly as dramatic as we had hoped.

We first added the UAV as a regular node in the network. While it did not source or sink any traffic, the UAV participated in the unicast routing protocol and responded to all ODMRP packets as if it were a terrestrial node. We next imposed policy on the UAV's participation in ODMRP mesh formation by restricting it to only respond to the squad leaders and SA multicast groups. Finally, we restricted the UAV to only respond to the squad leaders group.

Figure 12 shows the robustness of the various multicast groups as a function of the UAV presence and policy. While the presence of the UAV improved performance of the squad leaders group, it had a slight detrimental affect on the intra-squad traffic. This is understandable; because the UAV had a relatively low path loss to all terrestrial nodes, data transmitted by the UAV interfered with traffic on the ground. Situation awareness traffic benefited if it was allowed to use the UAV, and was penalized when it was not.

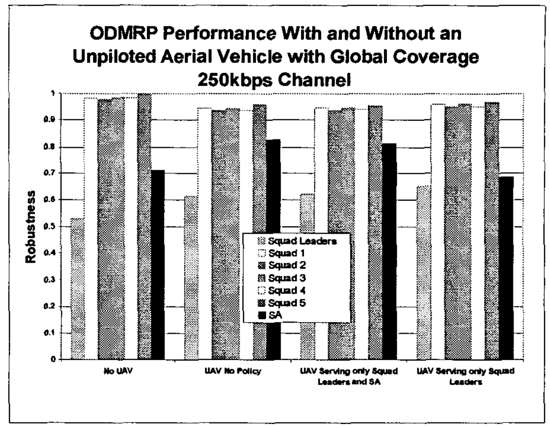


Figure 12: ODMRP performance both with and without an unpiloted aerial vehicle with global coverage.

6. CONCLUSIONS

Comparing protocol performance to optimal in terms of delay, efficiency, and robustness will allow protocol designers to target areas where multicast protocols can most benefit from improvement. For example, we have shown that the ODMRP implementation we have needs improvement particularly in the area of robustness. Similarly, protocol performance can be compared to optimal for a range of network sizes, multicast group sizes, and node densities to provide information about how a protocol's performance diverges from optimal as these parameters are varied. This allows one to describe an "operating range" that describes networks to which the protocol is well-suited.

The cross-platform physical layer method described will allow protocol designers to compare multicast (and unicast) protocols on a level playing field. Although the Infantry Urban Terrain scenario presented here sampled the RF environment every 10s, higher sampling rates are possible, up to several Hz. Finally, the path loss matrices capture the dynamics of correlated node movements and the interrelationships between nodes to a higher fidelity than simple free-space or r^4 models.

For the network scenarios we considered, ODMRP performance does *not* benefit from an "advantaged node" (the UAV in our case) with global connectivity. This is somewhat counterintuitive, since the advantaged node provides an excellent relay point, particularly when there is little traffic in the network. As it turns out, there are two related downsides to MANET nodes with very high degress of connectivity. First, it is difficult for such nodes to receive any data, since there are so many possibilities for interference. Second, such nodes have a difficult time *transmitting* because they are constantly barraged with packets, leaving them very little opportunity to send traffic.

We conjecture that putting such advantaged nodes on a separate, non-interfering radio channel would provide an enormous benefit.

7. Future Work

The optimal versions of the efficiency, delay, and robustness metrics discussed in this paper all apply to a single multicast group with a single sender operating in isolation. A future paper will expand the metric set to include metrics that gauge the interactions between multiple multicast groups (the node and link concentration metrics mentioned in the introduction), and between multicast and unicast traffic.

We are seeking multicast models for both GLOMOSIM and OPNET. Having a common model to run in both simulation environments will allow us to fine-tune parameters so as to further "level the field" between simulation packages.

We are developing an advantaged resource discovery and use protocol whose goal is to make efficient use of advantaged nodes such as the UAV in section 0. In this model, not all nodes have radios equipped to communicate with the advantaged node, so that communications may have to use both terrestrial and advantaged resources. The advantaged resources will be channelized, however, so that they do not suffer interference from all nodes in the terrestrial network.

We are currently using the MODSAF and TIREM tools to gather data on a much larger network scenario. By sampling the scenario we will be able to generate a number of sub-scenarios with different numbers of nodes. We can then use information from the delay, efficiency, and robustness metrics to evaluate protocol scalability as network size increases.

References

[1] Metricom's web site: http://www.metricom.com

[2] LAN/MAN Standards Committee of the IEEE Communications Society, *Information Technology--Telecommunications and Information Exchange Between Systems--Local and Metropolitan Area Networks--Specific Requirements--Part 11:Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, June 1997, Institute of Electrical and Electronics Engineeers, Inc., ISBN 1-55937-935-9 (IEEE Std 802.11-1997).

[3] T. Pusateri, "draft-ietf-idmr-dvmrp-v3-07," Work in progress, August 1998

[4] Bommaiah, McAuley, Talpade, and Liu, "draft-talpade-manet-amroute-00.txt: AMRoute: Adhoc Multicast Routing Protocol," Work in progress, August 1998

[5] Mario Gerla, Guangyu Pei, Sung-Ju Lee, and Ching-Chuan Chiang, "draft-ietf-manet-odmrp-00.txt: On-Demand Multicast Routing Protocol (ODMRP) for Ad-Hoc Networks," Work in progress, November 1998.

[6] Charles E. Perkins and Elizabeth M. Royer, "draft-ietf-manet-aodv-02.txt: Ad Hoc On Demand Distance Vector (AODV) Routing," Work in progress, November 1998.

[7] Scott Corson, "draft-ietf-manet-appl-00.txt: MANET Routing Protocol Applicability Statement," Work in progress, November 1998.

[8] Scott Corson and J. Macker, "draft-ietf-manet-issues-02.txt: Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," Work in progress, October 1998.

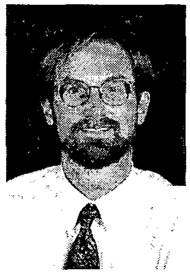
[9] Bruce Hajek and Galen Sasaki, Link scheduling in polynomial time. *IEEE Transactions on Information Theory*, 34(5):910—917, September 1988.

[10] Leandros Tassiulas and Anthony Ephremides, Jointly optimal routing and scheduling in packet radio networks. *IEEE Transactions on Information Theory*, 38(1), January 1992.

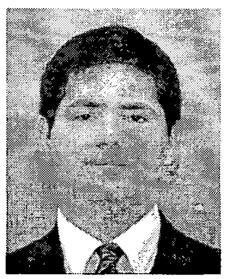
[11] http://www.jmtk.org/pages/capabilities/definitions.html

[12] http://www.opnet.com/

[13] Lokesh Bajaj, Mineo Takai, Rajat Ahuja, Ken Tang, Rajive Bagrodia, and Mario Gerla, "GloMoSim: A Scalable Network Simulation Environment," Technical Report, UCLA Computer Science Department - 990027.



Robert C. Durst is a Principal Networking and Communications Engineer at the MITRE Corporation in Reston, VA. He has been involved in research and development of communication protocols for use in mobile environments since 1993. He has been active in developing protocols for use in military tactical networking environments and in spacecraft communication environments. His research interests include multicasting in mobile networks, congestion control algorithms for mixed-loss environments, and congestion control in his two-year-old son. Prior to joining MITRE in 1986, Mr. Durst was a Member of the Group Technical Staff at Texas Instruments, Inc. He received a B.S. in electrical engineering from the University of Missouri in 1978.



Keith Scott is a Lead Engineer with the MITRE Corporation in Reston, VA. He received a B.S. in Electrical Engineering and Computer Science from the University of California at Berkeley in 1990 and his M.S. and Ph.D. degrees in Electrical Engineering from the University of California at Los Angeles in 1991 and 1997. From 1997 to 1998 he worked at NASA's Jet Propulsion Laboratory in Pasadena, California. Keith joined The MITRE Corporation in September of 1998. His research interests include self-organizing networks and routing in both ad hoc and space environments.

Dr. Cauligi Raghavendra is a Research Professor in the Electrical Engineering-Systems Department at the University of Southern California. For the last two years he was a Senior Engineering Specialist in the Computer Science Research Department at the Aerospace Corporation. He has taught for over 15 years, as a faculty with the University of Southern California (1982-92) and as Boeing Chair Professor at Washington State University (1992-97). He received his Ph.D degree in Computer Science from UCLA in 1982. His research interests include wireless networks and parallel and distributed processing. He is a subject area editor for Journal of Parallel and Distributed Computing and a co-Editor-in-Chief for Cluster Computing journal published by Baltzer Science. Dr. Raghavendra received the Presidential Young Investigator Award in 1985 and became an IEEE Fellow in 1997.