SANDIA REPORT

SAND2003-8762
Unlimited Release
Printed January 2004

Robust Message Routing for Mobile (Wireless) Ad Hoc Networks

M.E. Goldsby, R.P. Tsang, M.M. Johnson, H.Y. Chen,
N.R. Bierbaum, D. Kilman, H.R. Ammerlahn and D.M. Nicol

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online order:	http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2003-8762
Unlimited Release
Printed January 2004

Robust Message Routing for Mobile (Wireless) Ad Hoc Networks

Michael E. Goldsby, Rose P. Tsang and Michael M. Johnson
System Studies Department

Helen Y. Chen and Neal R. Bierbaum
High Performance Computing and Networking Department

Heidi R. Ammerlahn
Systems Research Department

Sandia National Laboratories
P. O. Box 969
Livermore, California 94551-9201

Dominique Kilman
Networked Systems Survivability and Assurance Department
Sandia National Laboratories
P. O. Box 5800
Albuquerque, New Mexico 87185-0785

David M. Nicol
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801-2307

Abstract

This report describes the results of research targeting improvements in the robustness of message transport in wireless ad hoc networks. The first section of the report provides an analysis of throughput and latency in the wireless medium access control (MAC) layer and relates the analysis to the commonly used 802.11 protocol. The second section describes enhancements made to several existing models of wireless MAC and ad hoc routing protocols; the models were used in support of the work described in the following section. The third section of the report presents a lightweight transport layer protocol that is superior to TCP for use in wireless networks. In addition, it introduces techniques that improve the performance of any ad hoc source routing protocol. The fourth section presents a novel, highly scalable ad hoc routing protocol that is based on geographic principles but requires no localization hardware.

Contents

Introduction

This report describes four complementary efforts coordinated around the common theme of robust messaging in wireless ad hoc networks. Considered in terms of the seven-layer OSI model for network protocols [Walrand 2000], the efforts address issues in Layers 2, 3 and 4, the Link, Network and Transport layers respectively. The first section of this report provides an analysis of throughput and latency in wireless medium access control (MAC) protocols (Layer 2) and develops guidelines for achieving good ad hoc network performance. The second section describes enhancements made to existing simulation models of wireless MAC and routing protocols (Layer 3). The enhanced models were used in support of the work described in the third section, which presents a new transport protocol (Layer 4) that is superior to TCP for use in wireless ad hoc networks. The third section also introduces several innovative techniques for improving performance in the route discovery phase of ad hoc routing protocols. The fourth section presents a novel, highly scalable ad hoc routing protocol based on geographic principles.

Throughput and Latency Analysis of MAC Protocols for Ad Hoc Wireless Networks

The feasibility of supporting applications with performance requirements, or Quality-of-Service (QoS), is dependent upon the underlying limits of the network. The performance limits of a network are usually expressed in terms of throughput (channel capacity) and latency.

In ad hoc wireless networks, available bandwidth and latency are closely related to the particular media access control techniques. In this study, we present simulation results of the performance of several common ad hoc network topologies. We also present analytical discussions to further explain the simulation results as well as shed light on the performance of ad hoc wireless networks in general.

Types of MAC Protocols

Ad hoc wireless networks usually rely on a common transmission medium. Thus, among multiple nodes on a network, access to the medium must be controlled by a media access (MAC) protocol. In this section, we discuss the major proposed MAC protocols. Since they developed in an evolutionary manner, with each new protocol subsuming the benefits of the prior protocol, we present them from the most basic to the most sophisticated (most widely used – 802.11b) protocol.

CSMA

In CSMA [Kleinrock 1975], every node senses the carrier before transmitting. If a node senses a transmission on the carrier (by testing the signal strength in its vicinity), then it defers its own transmission. CSMA is effective for avoiding collisions in the vicinity of the transmitter. However, unless the receiver is co-located with the transmitter, collisions may also occur at the receiver.

There is a well-known situation, the Hidden Terminal scenario, where collisions may occur at the receiver. CSMA does not detect this case. In the Hidden Terminal problem, two nodes which are outside of each other's transmission range try to communicate with the same node (which is within each of their transmission ranges). This results in a collision at the receiving node. In Figure 1, the collision occurs at node 2.

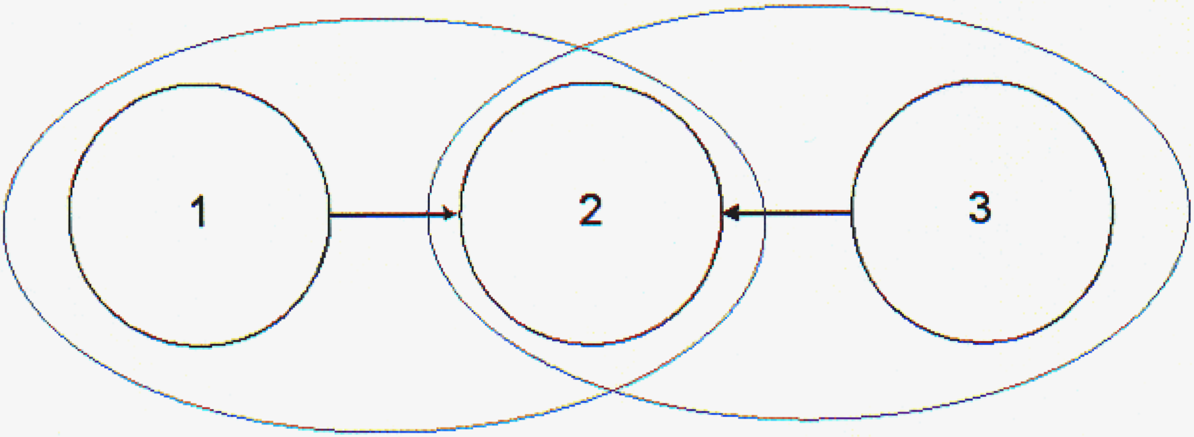

Figure 1. Hidden Terminal problem.

MACA

MACA was proposed in [Karn 1990] for use in packet radio as an alternative to CSMA. MACA uses two types of short fixed-size signaling packets. When node A desires to send to node B, it first sends a Request-To-Send (RTS) packet to B. The RTS packet also contains the length of the proposed transmission. When node B hears the RTS, it may immediately reply with a Clear-To-Send (CTS) packet. Upon receiving the CTS, node A immediately sends its data. Any node overhearing an RTS defers its transmissions until some time after the associated CTS packet would have finished. Any node overhearing a CTS defers its transmissions for the length of the expected data transmission which is contained in both the RTS and CTS packets. (Recall, we are assuming that the sender and receiver are not co-located.) This RTS/CTS/Data exchange solves the Hidden Terminal problem.

If node A does not hear a CTS in response from node B, it eventually times out, and then schedules the packet for retransmission. MACA uses a binary exponential backoff (BEB) algorithm to select the retransmission time.

MACAW

MACAW differs from MACA in two aspects: the RTS-CTS exchange, and the backoff algorithm.

[Bharghavan 1994] found that the binary exponential backoff scheme produces wild oscillations resulting in different nodes experiencing dramatically different latencies. They propose a milder adjustment algorithm. In this algorithm, upon a collision, the backoff interval is increased by a multiplicative factor (1.5) and upon a success, it is decreased by 1 (linearly). This multiplicative increase and linear decrease (MILD) provides quick response (escalation) in the backoffs when contention is high. By not resetting the backoff counter to the minimum it avoids having to repeat the escalation of the backoff counters after every successful transmission.

MACAW appends an ACK to the end of the RTS/CTS/Data exchange; thus the exchange is RTS/CTS/Data/ACK. The ACK increases throughput because it detects at the link-layer when packets are corrupted by intermittent noise or suffer a collision. Intermittent noise is much more frequent in wireless networks. Using MACA (RTS/CTS/Data), if TCP is used, the error has to be recovered at the Transport Layer, resulting in decreased channel utilization.

802.11 MAC Protocol

The 802.11 MAC protocol [IEEE 1997] uses a RTS/CTS/Data/ACK exchange as in the MACAW protocol. Each node has a Network Allocation Vector (NAV). The RTS and CTS packets include the amount of time the medium will be busy for the remainder of the exchange. Each node reads these values in the RTS and CTS frames as they pass by and use them to update their NAV vector.

The 802.11 specification defines five timing intervals for the MAC protocol. The Short InterFrame Space (SIPS) and the slot time are the two determined by the physical layer. The Priority InterFrame Space (PIFS), the Distributed InterFrame Space (DIFS) and the Extended InterFrame Space (EIFS) are defined based upon the SIPS and the slot time. The SIFS is the shortest time interval followed by the slot time. The slot time can be viewed as the time unit for the MAC protocol operations.

For 802.11b networks (based upon the DSSS physical layer), the SIFS and slot time are 10μs and 20μs, respectively. 20μs was chosen for the slot time to account for signal propagation and processing delays. The PIFS time interval is equal to the SIFS time interval plus one time slot. The DIFS time interval is equal to the SIFS time interval plus two time slots. The EIFS is much longer than any of the other time intervals; it is used if a data frame is received in error. The sequencing of these time intervals is discussed in greater detail in the next section.

Analysis of MAC Protocols

It has been noted that the actual effective capacity of an ad hoc wireless network is much less than the theoretical capacity [Li 2001]. It is important to understand the reasons for this because it also affects latency, which is a performance measure that has not received much attention in the current literature. Reasons for the limited capacity include the following.

1) The effectiveness of the MAC layer protocol to control collisions and fairly distribute the resources (medium) among competing sources. Message retransmission due to collision subtracts channel bandwidth from that available to successful transmissions.

2) A node's interference distance is greater than the node's transmission capability. For example, in the Lucent Wavelan card, the effective transmission range is 250 meters, and the interfering range is about 550 meters.

Figure 1 below depicts 9 nodes in sequence. The circles around each node depict the nodes' transmission range. For clarity, circles around all of the nodes are not shown. In this example topology, the interference range is the same as the transmission range. The darkened arrows in Figure 1 show which links can be used simultaneously; only every third link can be used simultaneously. Thus the maximum channel utilization for this case is 1/3.

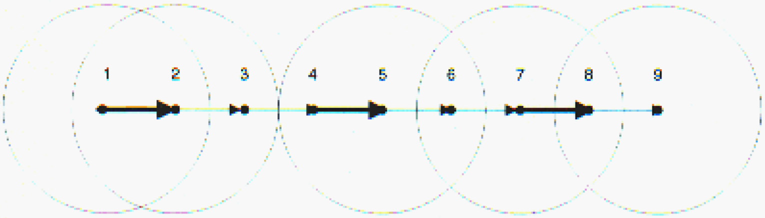

Figure 2. A linear sequence of nodes where the transmission range is the same as the interference range. The maximum channel utilization is 1/3.

Figure 1 depicts another 9 nodes in sequence. Once again, the black circles around each node depict the nodes' transmission range. The grey larger circles depict the interference range. For clarity, circles around each node are not all shown. This is the more realistic scenario where the interference range is larger than the transmission range. The darkened arrows in Figure 2 show which links can be used simultaneously; only every fourth link can be used simultaneously. Thus the maximum channel utilization for this case is 1/4.

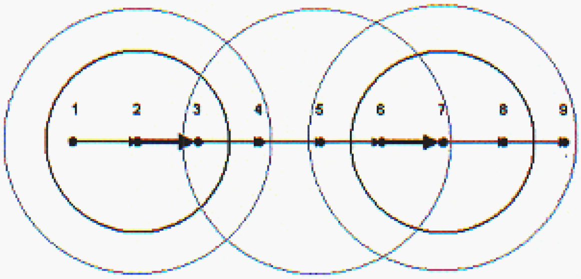

Figure 3. A linear sequence of nodes where the interference range is greater than the transmission range. The maximum channel utilization is 1/4.

3) Any contention-based MAC protocol must provide mechanisms to avoid collisions. The overhead involved may be non-trivial especially if multiple retransmissions are necessary before a successful transmission.

RTS (40B), CTS, ACK (39B) and MAC header of a data packet is 47B. For 500B packets, efficiency = 1500/(1500+40+39+47) = 0.9. With inter-frame timings it is reduced to 0.85.

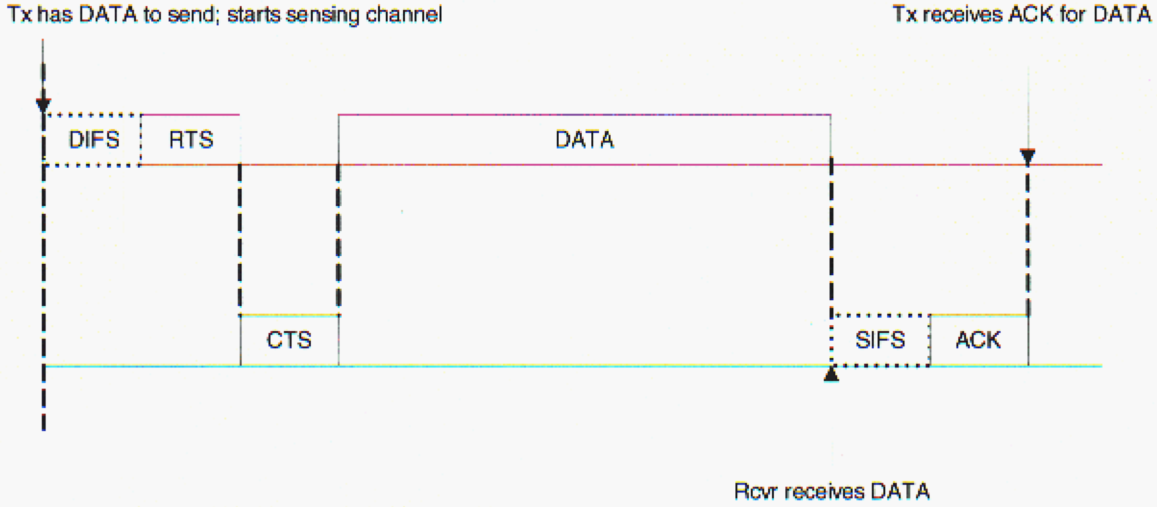

Figure 4. 802.11 Access Mechanism.

Figure 4 shows the components of the 802.11 access mechanism. Let t_{single} denote the time it takes for a single successful transmission. Then, according to Figure 4:

$$t_{single} = t_{DIFS} + t_{RTS} + t_{CTS} + t_{DATA} + t_{SIFS} + t_{ACK} + 4\tau, \qquad \text{where} \qquad \text{(d1)}$$

- t_{DIFS} is the Distributed InterFrame Space (DIFS). This is the interval a sending node must wait for the channel to be idle before beginning its transmission. 128µ is the value recommended by [IEEE 1997].
- t_{RTS} is the time it takes the sending node to transmit the RTS message.
- t_{CTS} is the time it takes the receiving node to transmit the CTS message
- t_{DATA} is the time it takes the sending node to transmit the DATA message.
- t_{SIFS} is the Short InterFrame Space (SIFS). This is the interval a sending node must wait before sending an ACK for the DATA message it just received. This interval is smaller than the DIFS. 28µ is the value recommended by [IEEE 1997].
- t_{ACK} is the time it takes the receiving node to transmit the ACK message

- τ is the propagation delay between the sending and receiving node. In a single successful transmission, there are 4 propagation time delays between the sender and the receiver. The first one occurs when the RTS packet is sent from the sender to the receiver. The second one occurs when the CTS packet is sent from the receiver to the sender. The third one occurs when the DATA message is sent from the sender to the receiver, and the fourth one occurs when the receiver sends the sender the ACK packet.

When there are multiple nodes active in the same transmission range, collisions are likely to occur. Thus, the actual latency may consist of a number of collision intervals followed by t_{single}, the time it takes to send a successful transmission. The collision interval, or CollisionDelay, consists of the following.

$$\text{CollisionDelay}_i = \text{idle}_i + \text{collision}_i + \tau + \text{DIFS}, \text{ where} \qquad \text{(d2)}$$

- idle_i is an idle period due to the backoff algorithm.
- collision_i is the length of the collision (which depends upon the maximum length of the colliding packets)
- τ is the time it takes the node to hear the signal from the collision
- DIFS is the DIFS time interval

The actual delay, t_{actual}, consists of the series of collision interval delays (retransmission delays) and the final successful transmission, i.e.,

$$t_{actual} = E[t_{single}] + E[\Sigma_{i=1,\ldots N}\text{CollisionDelay}_i] \qquad \text{(d3)}$$

Simulation Results

Measures of MAC protocol excellence include channel capacity (or channel utilization), latency and fairness. In this study, we focus on channel capacity and latency.

The results of this study are based on simulations using the NS-2 (Network Simulator) [Fall 1998] developed at Lawrence Berkeley National Laboratories (LBNL) with extensions for wireless ad hoc networks from the MONARCH project developed at Carnegie Mellon.

Simulation Parameters

- **Control Packet Sizes.**
  RTS (40B), CTS (40B), ACK (39B), MAC header of data packet (47B)
- **Packet Sizes.**
  The simulations were run for three common packet sizes: 1500 bytes, which is the maximum size for an Ethernet frame, 600 bytes, which is a typical size packet generated by web browsers, and 60 bytes, which is the size of a TCP ACK message.
- **Time intervals.**
  SIFS (28 μs), DIFS (128 μs), slot time (50 μs), propagation delay (1 μs)
- **Channel Bit Rate.**
  2 Mbps
- **CWmin, CWmax.**
  8, 256. The backoff schemes used are the BEB and MILD as described in the previous section.
- **Transmission probability per node.**
  Unless explicitly stated, the simulations were run under asymptotic conditions; all network stations always have a packet ready for transmission

Simulation Scenarios

In the first scenario, all nodes are within each other's transmission range. Figure 5 depicts this topology. In this scenario, each node sends packets as fast as the medium allows to a randomly selected node.

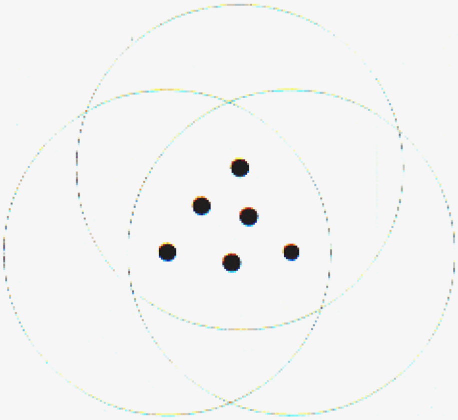

Figure 5. Topology where all nodes are within each other's radio ranges.

Figure 6 and Figure 7 show the channel throughput/channel utilization as the number of nodes increases. As can be noted, especially in Figure 6 with the 1500 byte packets, contention is the factor which diminishes throughput. Thus when there are two nodes, there is the least amount of contention so the throughput is the highest. As the number of nodes grows, the amount of contention grows, and hence the throughput decreases.

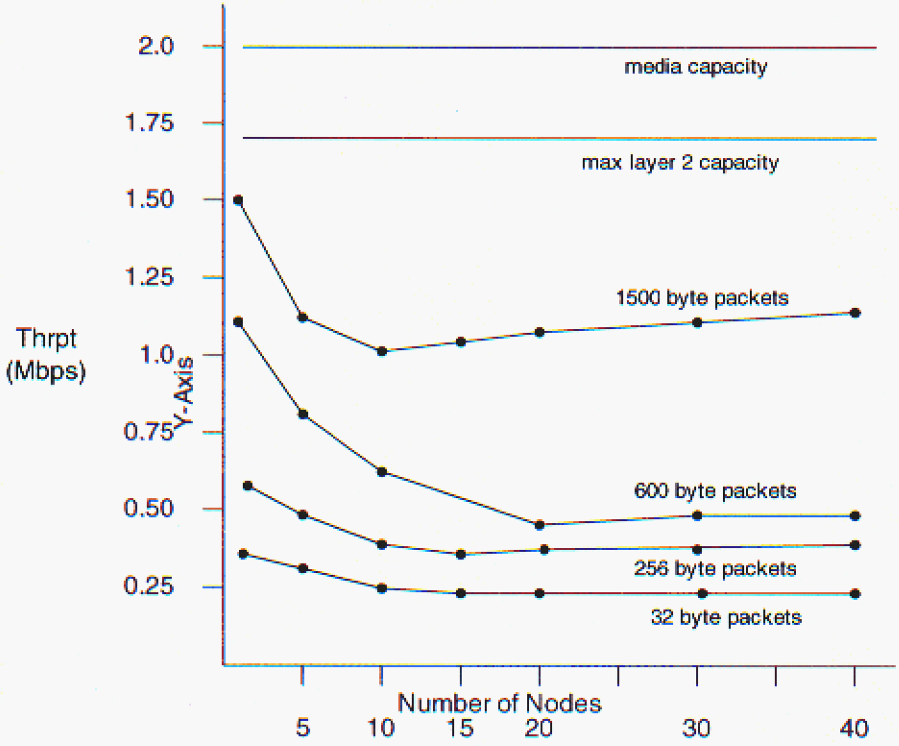

Figure 6. Throughput versus load for a topology where all nodes are within each other's transmission range.

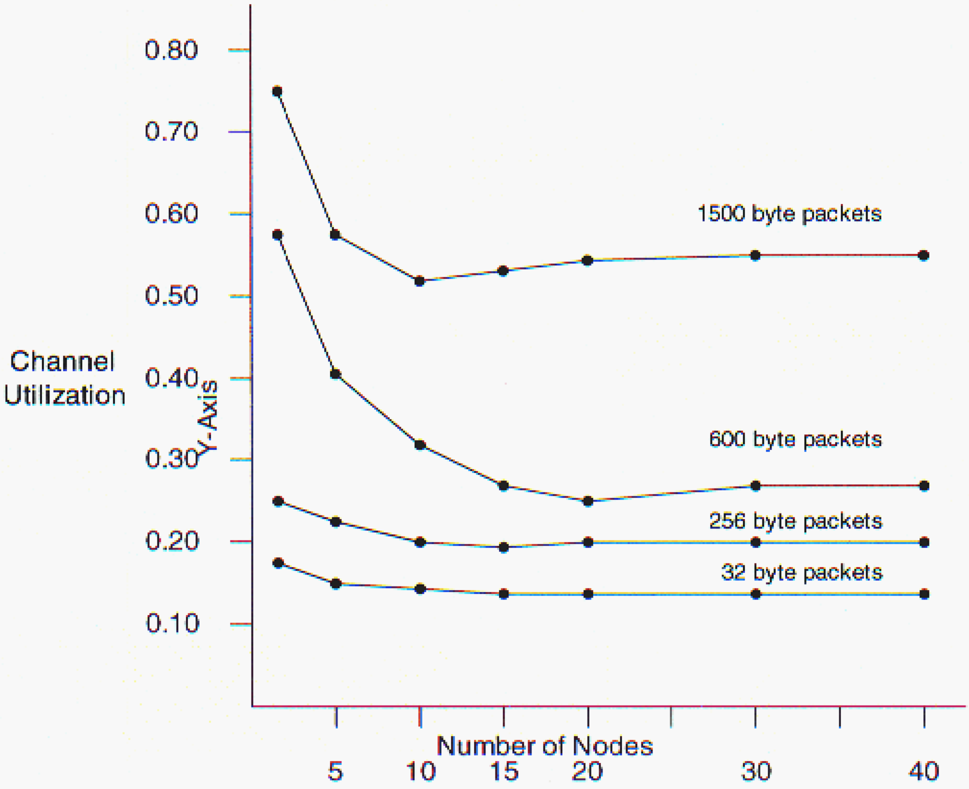

Figure 7. Utilization versus load for a topology where all nodes are within each other's transmission range.

As mentioned in the previous section, latency is the accumulation of collision intervals followed by a successful transmission.

Figure 8 (below) depicts latency as a function of contention. The "P" in the graph corresponds to the probability that each node has a packet ready to transmit. P=1 means that every node always has a packet to transmit. P=0.5 means that every node has a packet to transmit ½ of the time

The actual latency values where obtained by simulating the number of collisions (under the specified conditions: probability of transmission, number of nodes, backoff scheme), and then computing the latency using the equations in Section 3.0.

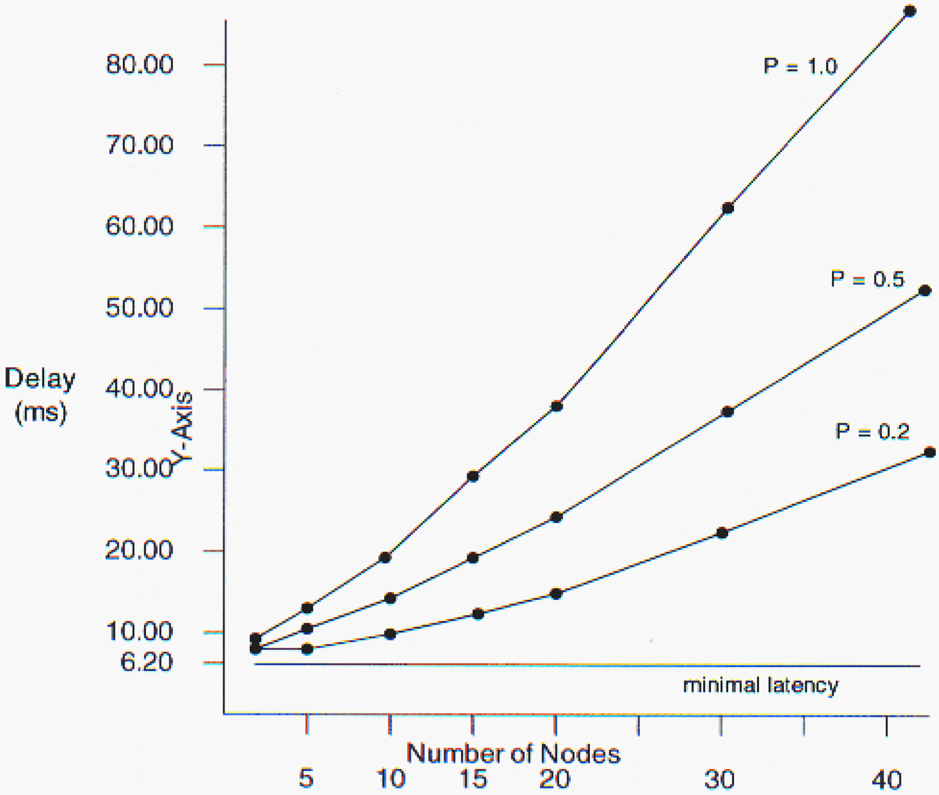

Figure 8. Latency as a function of contention.

In the second scenario, we examine the throughput and latency for nodes arranged in a sequential topology. Traffic was generated between nodes which were randomly paired. This is in contrast to the work in [Li 2001] where they simulated the same topology but under the traffic condition where the first node in the sequence generates all the traffic and the last node in the sequence acts as the traffic sink.

A linear sequence of nodes is an important topology to examine because in ad hoc wireless networks nodes may roam in and out of the transmission/interference range of other nodes so the traffic pattern flows between sequences of nodes.

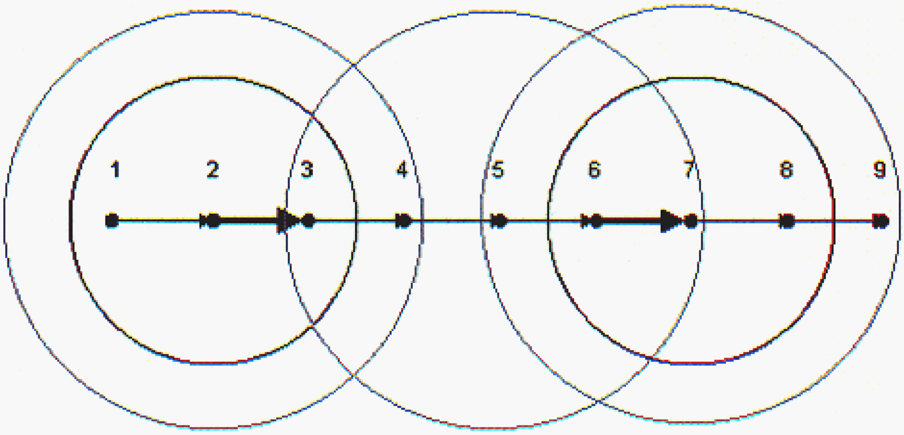

Figure 9. A linear sequence of 9 nodes.

The above figure depicts the topology used in the simulations. The darkened circles represent the transmission range of the node inside the circle. The light grey circles represent the interference range of the node inside the circle. In order not to clutter the diagram, not all circles are shown. The circles not depicted can be inferred from the shown circles. For example, node 5 can directly transmit to node 4 and node 6. However, when node 5 transmits to either node 4 or node 6, it also produces interference which prevents node 3 and node 7 from transmitting or receiving.

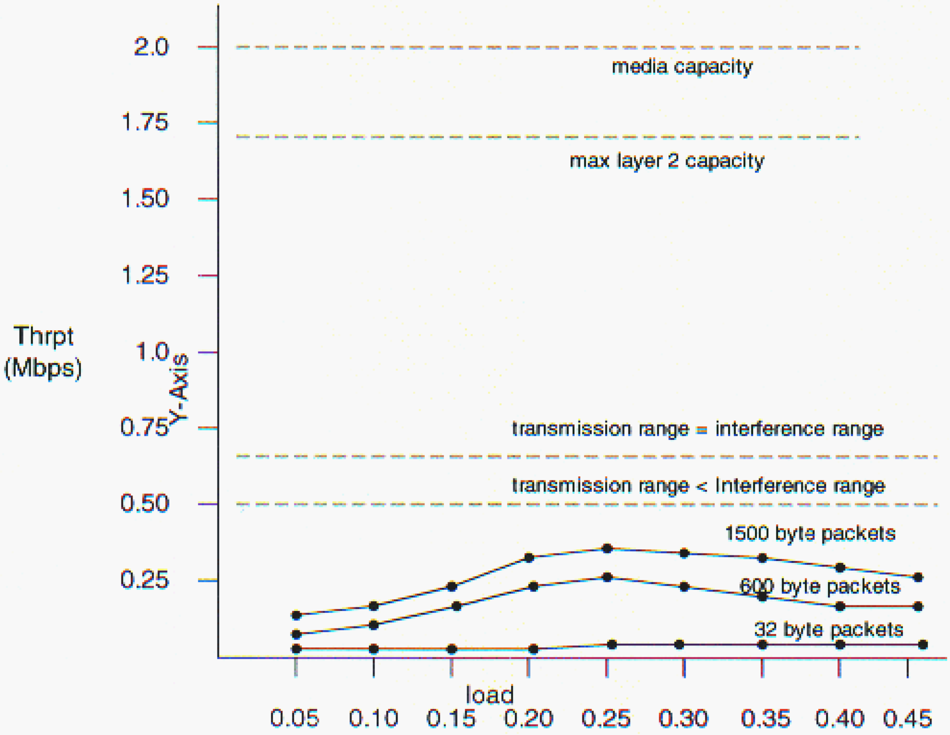

Figure 10. Throughput versus load for a sequential series of 9 nodes.

Figure 10 depicts the throughput as a function of load. The load is represented by the probability that a node has a packet to send. Asymptotic conditions are considered to be reached when the load reaches 1.0. The maximum throughput is achieved with the 1500 byte message when each node has an approximate 35% chance of transmitting. This translates into a maximal channel utilization of approximately 17%.

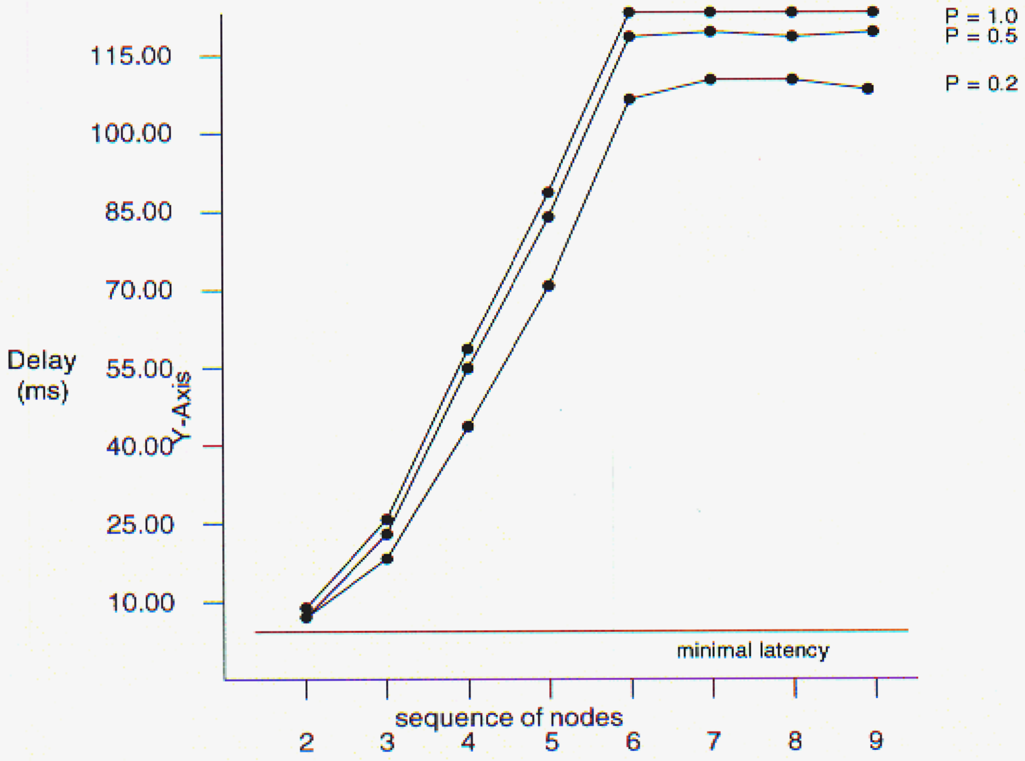

Figure 11. Latency as a function of length of sequence of nodes.

Figure 11 depicts latency as a function of the length of the sequence of nodes. The "P" in the graph corresponds to the probability that each node has a packet ready to transmit. P=1 means that every node always has a packet to transmit. P=0.5 means that every node has a packet to transmit ½ of the time. The latency measured is not end-to-end latency; it is the latency between adjacent nodes. The latency grows quickly as the sequence of nodes increases from 2 to 6. This is because as the number of nodes increases beyond 2 the number of nodes residing in the interference range of other nodes increases too. However, once the number of sequential nodes reaches 6, the latency stabilizes because the number of simultaneous transmissions stabilizes as well (see **Analysis of MAC Protocols** above).

As in the previous latency measurements for the single shared medium, the actual latency values where obtained by simulating the number of collisions (under the specified conditions: probability of transmission, number of nodes, backoff scheme), and then computing the latency using the equations under **Analysis of MAC Protocols**.

Throughput and Latency Analysis

Assume an infinite population of nodes which generate packets according to a Poisson distribution with mean P frames per frame time. Note that $P \leq 1$. Packets which are generated may also include packets retransmissions. Let G be the number of new and retransmitted frames generated per frame time. Thus $G \geq P$. The probability that k frames are generated during a given time frame is given by:

$$P_k = G^k e^{-G}/k!, \text{ where } k \geq 0, G\geq 0 \qquad \text{(t1)}$$

Throughput Analysis

Let p_i be the probability of a successful transmission by node i. Let G_i be the total transmission probability (per slot) of node i. It follows that

$$p_i = G_i \, \Pi(1-G_j), \text{ for } j\neq i \qquad \text{(t2)}$$

Now, for M identical users, using the above equation,

$$P = G\,(1 - G/M)^{M-1}, \qquad \text{(t3)}$$

where $P= Mp_{i,}$ and $G=MG_{i,}$ $(G=\Sigma G_i)$.

As $M\rightarrow\infty$, we know that

$$\lim_{k\rightarrow\infty} (1 + x/k)^k = e^x \qquad \text{(t4)}$$

so

$$P = G\, e^{-G} \qquad \text{(t5)}$$

Figure 12 depicts channel utilization versus load in the most general case. From the graph, we can see that the optimal channel utilization occurs when there is exactly one frame generated per slot. If there is more than one frame generated per slot, collisions will occur, decreasing the channel utilization. If there is less than one frame generated per slot, then there will be unused slots, also resulting in less than full utilization. It is interesting to note the channel utilization of approximately 37%. This relatively low channel utilization is a key characteristic of shared medium networks where many nodes are allowed to transmit frequently.

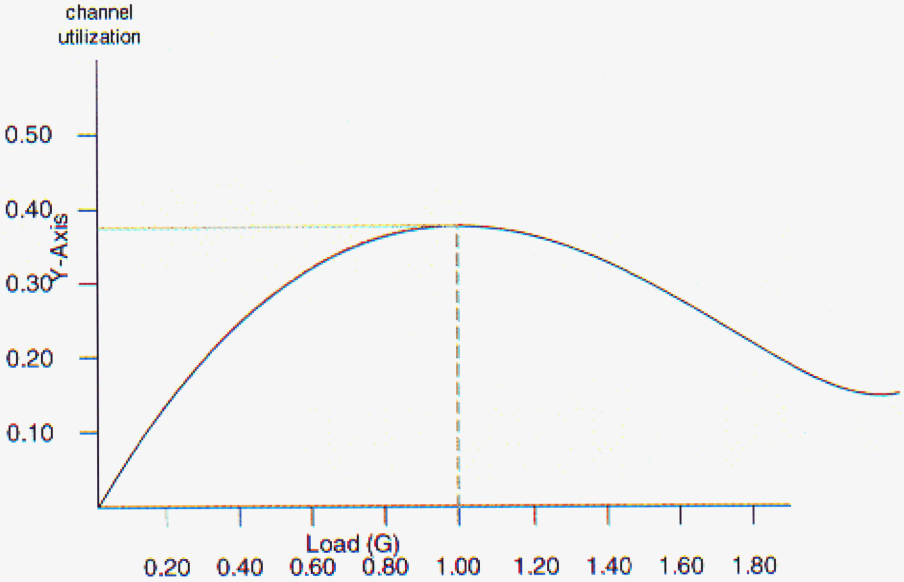

Figure 12. Channel utilization versus Load.

Now to illustrate the effects of contention in a simpler case, consider just two users. Then using equation (t2), $p_1=G_1^2$ and $p_2 = (1 - G_1)^2$.

p_1, p_2 and $P=p_1+p_2$ are depicted graphically in Figure 13.

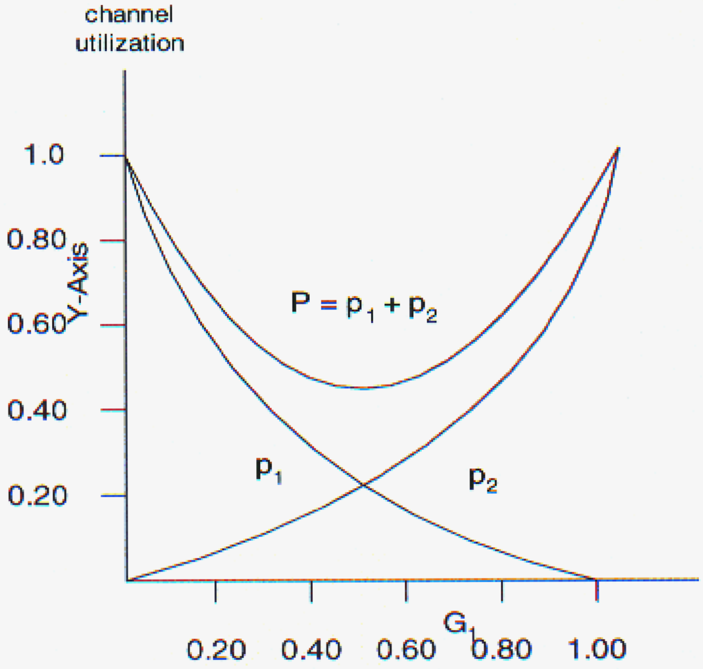

Figure 13. Channel utilization versus load for a 2 user system.

Figure 13 shows that the throughput is the highest in the following cases: (i) User 1 is transmitting at every slot and User 2 is silent, and (ii) User 2 is transmitting at every slot and User 1 is silent. Throughput is at its worst when both User 1 and User 2 are transmitting with equal probability (50%) of the time. Thus throughput is worse when contention is at its highest. This corresponds to our throughput experiments described in the previous section.

Latency Analysis

As we discussed in the previous sections, collisions are the dominant factor in increased latency. Here we present an analytical argument to show why this is true.

From equation (t1), P_0 is the probability that no traffic is initiated; $P_0 = e^{-G}$. Thus $P = G\, e^{-G}$ So e^{-G} is the probability of no collision , and $(1 - e^{-G})$ is the probability of a collision. The probability of a transmission requiring exactly k attempts is the following.

$$P_k = e^{-G} (1 - e^{-G})^{k-1} \qquad \text{(t12)}$$

The expected number of transmissions, E_k, is

$$E_k = \Sigma_{k=1,2,\dots}\ k\ P_k = \Sigma_{k=1,2,\dots}\ k\ e^{-G}\ (1 - e^{-G}\ )^{k-1} = e^G \qquad \text{(t13)}$$

Thus latency is exponentially dependent upon G, which implies that small changes in load can greatly increase the latency.

MAC Performance and Routing Performance

For wireless ad hoc networks, MAC protocol performance issues are closely tied to routing protocol performance. Some obvious ways in which the two layers interact include the following. The ability of the MAC layer to efficiently control access to the shared medium directly impacts the latency of packets. Any packet collision results in at least one packet retransmission, thus immediately increasing the latency by a factor of two. Routing protocol updates which are incessantly delayed will result in routing decisions based upon out-of-date information. Channel capacity in wireless ad hoc networks is constrained, especially between sequences of nodes which do not share the same transmission range. Furthermore, channel capacity must be shared between control and data traffic. For mobile ad hoc networks, the routing layer relies upon the MAC layer for notifications when a node moves out of range. MAC-level performance depends upon the type of mobility. [Barrett 2002] shows the interaction between the MAC and routing layer for mobile networks by examining the variation in the number of control packets generated by each layer.

Conclusions

This study has shown the effects of load on the throughput and latency of ad hoc mobile networks. The effects of load are manifested by the increased number of collisions. Each collision necessitates a retransmission. In a heavily loaded network, a node may attempt several retransmissions before achieving a successful transmission. This translates into lower network utilization (due to the capacity wasted by retransmitting the same packet) as well as sharp increases in network latency. It is difficult to specify an exact network load which will be appropriate for achieving high network utilization and low packet latency. As shown in this study, the performance of an ad hoc network is dependent upon many factors – including network topology, packet size, the ratio of interference range to transmission range, and the particular MAC protocol implementation. However, general guidelines for achieving good performance in ad hoc network include the following: (i) use larger packet sizes (600 bytes +), (ii) use a locality-based topology where nodes which communicate more frequently are within each other's transmission range, and (iii) maintain network loading of less than 50% between nodes in the same transmission range, and less than 40% between nodes in different transmission ranges.

Simulation Models for Ad Hoc Networking

MANET Routing Protocols

There are several ad hoc routing protocols that are currently being investigated for use in mobile ad hoc networks (MANETs). Two of the more popular protocols are Dynamic Source Routing (DSR) and Ad Hoc On-demand Distance Vector Routing (AODV). Both of these protocols were investigated for this LDRD to enable flow control at the MAC layer of the network.

AODV creates routes through the network in a purely on-demand fashion. When a source node needs to communicate, Route Request (RREQ) messages are broadcast through the network. The destination node or a node with an existing path to the destination will send a Route Reply (RREP) message back to the source. The routes set up in this manner will remain active as long as data is traversing the route. Once data stops traversing the route, the path will time out and the intermediate nodes will remove the hop information from their routing tables.

DSR works by storing information regarding routes through the network in cache. This information is kept in the cache whether or not the path is in active use. If a node wishes to communicate with a destination which it does not have a route to, broadcast RREQ messages are sent. Similar to AODV, the destination or intermediate node with a valid path will respond using a RREP message.

Comparison

Each routing protocol has plusses and minuses depending on the environment in which it is deployed. For example, AODV performs better in stressful situations [Das 2000] (those in which there is a high degree of mobility, large number of sources, etc.) DSR has a lower routing load due to the cached routes. Because of the need to store routes in DSR, scalability for DSR is not as good as that for AODV, but DSR handles frequent topology changes better.

The final decision to use DSR for the simulation model was influenced more by the mechanics of the simulation models than by the constraints of either routing protocol. The existing AODV model was unreliable, did not perform well under stress conditions, and did not allow for the buffer overflows which would call for MAC level flow control. See detailed discussion in Simulation Decisions.

Simulation Decisions

The initial starting point for the simulation was the NIST OpNet models for AODV [NIST 2003a] and DSR [NIST 2003b]. At the time the LDRD started, these were the only ad hoc wireless models available for the OpNet simulator.

The AODV routing protocol contained in the model is based on what was then a draft specification from the IETF for AODV. The draft has become RFC 3561 as of July 2003 [Perkins 2003]. The latest version of the NIST model is dated August 2001.

The DSR routing protocol is based on the IETF internet draft version 4 written in November, 2000. The latest version of the DSR internet draft is version 9 written in April 2003 [Johnson 2003]. The latest DSR model is dated April 2002.

The NIST models used abbreviated versions of a network stack. They did not have a true transport layer, so there was no opportunity to use TCP or UDP. There was also a simplified version of the application layer which simply created dummy data packets to be sent. The MAC/Link layer of the model was the most important process for our simulation purposes since this is where flow control was being added.

The NIST models provided a baseline to learn about the workings of both DSR and AODV in the OpNet simulation environment. Unfortunately, the NIST models were not designed for the type of situation we were trying to simulate: mainly an environment where there was congestion due to buffer size limitations at some or all nodes in the network. In trying to simulate this situation, the NIST models started showing evidence of errors. AODV would not allow for high data rates (this caused collision for every packet) and DSR would start displaying errors because ACKs did not correctly correspond to the sent packets. This was a result of packets being dropped at the MAC layer, but not at the routing layer. The errors being reported were tagged “Amazing Errors” by the NIST developers because they should not occur in reasonable network simulations, or should never occur at all.

OpNet developed a beta version of a DSR model in early 2003. Because of the problems with the NIST models, we switched to the OpNet supported model. The OpNet model is much more complex than the NIST models and includes all levels of the networking stack including TCP, UDP, and ARP. This added complexity made simulating the environment more difficult, but the reliability of the model made the tradeoff worthwhile.

Why DSR?

DSR was chosen as the routing protocol for many reasons, one of which was OpNet support of the new DSR model. With a supported model, problems which appeared in the simulation could be addressed quickly. The NIST models were older and had less technical support for problems.

The NIST model also had some unexpected implementation side effects which would cause varying results for the same simulation. At times the model would completely crash due to memory access errors. To achieve the desired congestion in the network, the simulation parameters in NIST AODV had to be extreme. The buffer had to be extremely small (2200 bytes), the data rate 1 mbps, with no mobility in the network. All the nodes were required to send data to a single destination in order to cause congestion. Changes to the settings could cause a multitude of problems, including memory access errors, zero throughput in the network, or no congestion.

A problem with using AODV apart from the simulation issues was the way in which congestion occurred at all. In AODV, data packets are not buffered at the MAC layer but at the routing layer. Because of this feature, congestion in the MAC layer would only occur when control packets were buffered. This was one of the reasons that the simulation constrains for AODV were so severe. Implementing MAC layer flow control in AODV did not yield any gain in network performance.

Simulation Changes

In order to support MAC layer flow control in the OpNet-supported DSR model, changes were required in the simulation model, especially in the MAC process. The most significant modification was adding MAC flow control parameters. In order to compare different methods of flow control, a new attribute was added to the wireless nodes: a CTS Flag. This attribute tells the node how to react when it receives an RTS message, but the node is congested. The node may perform one of three actions: normal operation, no CTS sent, wait message sent. In normal operation, a CTS message will be sent despite the congestion at the node.

A node determines that it is congested when it is forced to drop a packet due to buffer constraints. To remove this constraint flag, the node must wait until is has an empty buffer.

In order to determine more accurately what was happening when a node became congested, new statistics were created: *data dropped by full buffer* and *data dropped by retry*. This divides the data dropped statistic into the two situations which cause packets to be dropped. Data will be dropped due to a full buffer when there is congestion at the node. Data will be dropped due to retry at the sending node when the sending node does not receive an acknowledgment that the destination received the packet. So *data dropped by retry* is also the result of congestion, but this congestion is at the destination.

Simulation Results

The following graphs show the packet drop performance of the three different CTS response scenarios. As can be seen from the graphs, adding a WAIT message to the traffic in the network degrades performance. This is a result of the added traffic in the network. As can be seen from the Load comparison graph, adding the WAIT message predictably increases the load on the network due to the extra messages being sent. Ignoring the RTS request reduces the network load because the data packets are not being sent as often because the sender does not receive the CTS message.

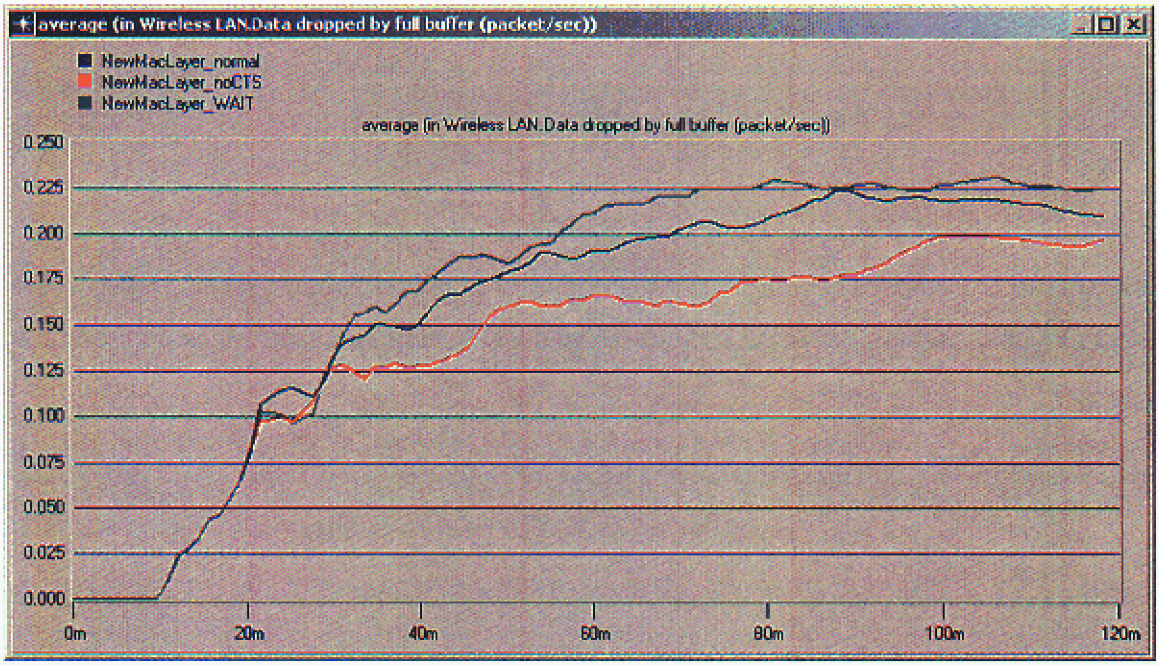

Figure 14. Average data dropped by full buffer.

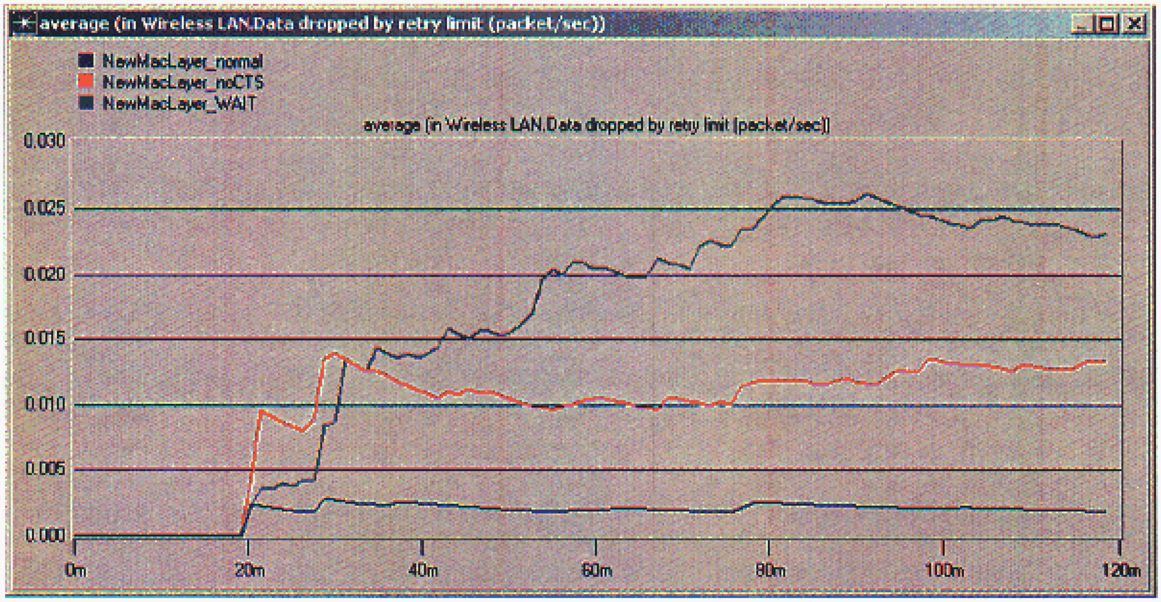

Figure 15. Average data dropped by retry limit.

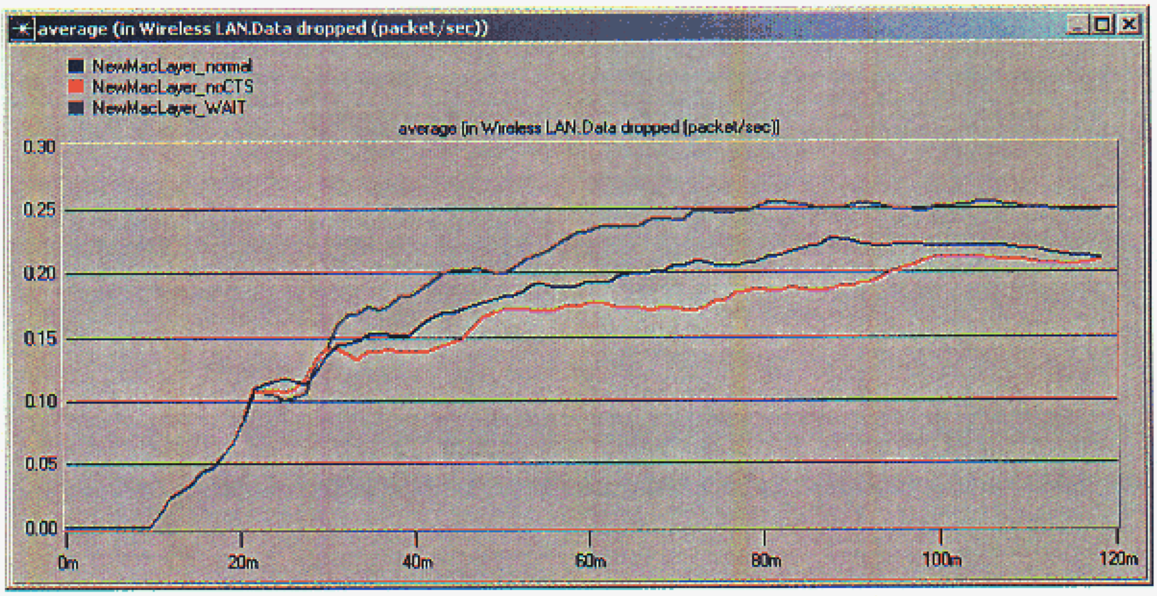

Figure 16. Average data dropped.

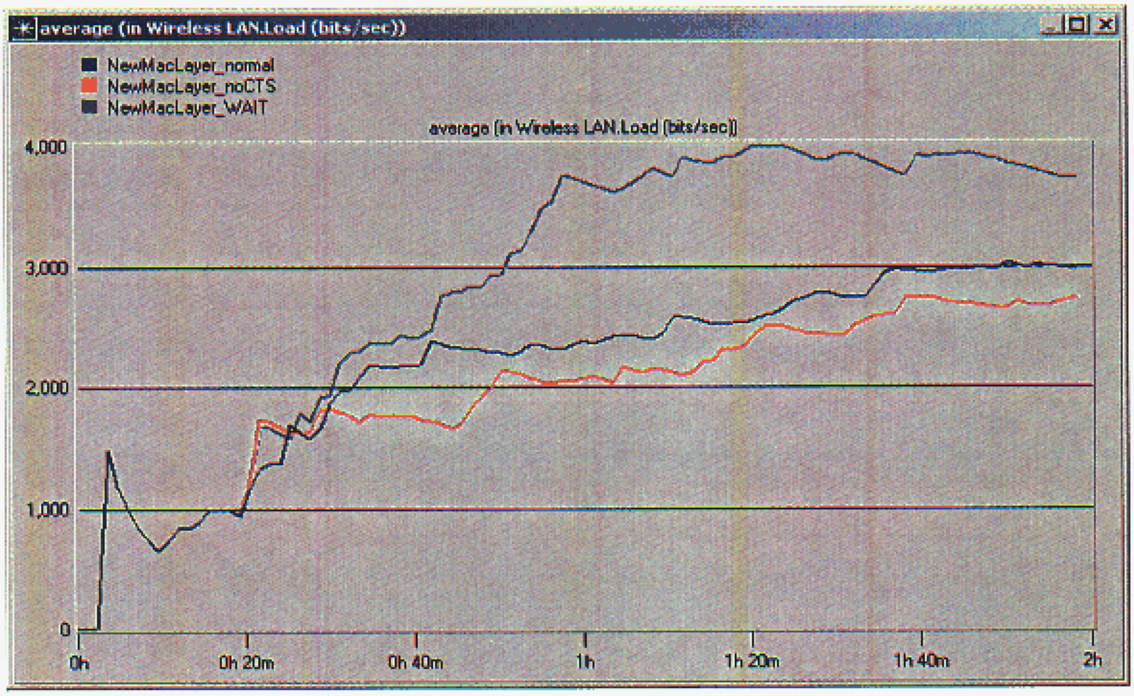

Figure 17. Average load

Simulation and Evaluation of a Custom Messaging Protocol and Enhanced Source Routing for Wireless Communications

Introduction

This project consisted of the development, enhancement, and simulation of the key elements of a dynamic wireless communications fabric with multiple nodes. These elements included a reliable messaging protocol based upon the design described in [Lockhart 2000], a source routing protocol with extensions, and the 802.11b wireless protocol. All elements were evaluated by simulation in the OPNET simulation framework. All of the messaging and routing protocols required completely custom-written OPNET models; the wireless protocol used an existing model with corrections and enhancements to correctly model the standards.

The simulations and evaluation showed the basic viability of the design for several specific environments. These environments are characterized by fairly small, infrequent messages passed between either peer-to-peer or master-slave nodes. They may also include changing connectivity between nodes caused by node movement or failure.

Messaging Protocol Design and Simulation

The protocol used for messaging was derived from the design described in [Lockhart 2001]. This protocol was originally designed for high speed communications via Ethernet within computer clusters by cooperating processes. The initial simulation was created to match that environment because 802.11b was not yet available at the start of the study. Furthermore, OPNET's Ethernet model is robust and simple to use with well-defined parameters such as bit error and transmission rate. Therefore, this stage of modeling served an important role in the overall simulation development by isolating simulation errors to the protocol elements.

Description of the Messaging Protocol with a Comparison to TCP/IP

The protocol as described within [Lockhart 2000] was incomplete and required further design to become functional. Some further extensions were added to improve performance and efficiency. The enhanced protocol and a comparison to TCP/IP are described in the following paragraphs.

The messaging protocol is significantly different from TCP/IP. The primary unit of connection is a single message, often of limited size, between remote processes. TCP/IP uses a continuous stream of binary data with no differentiation between messages within the protocol itself. This means that less long term state information must be maintained for the messaging protocol. If the messaging protocol will be used between hosts supporting several simultaneous processes then each process must have an identifying port, perhaps provided by using the messaging protocol over UDP. TCP must have a unique pair of ports for each stream. In contrast the messaging protocol can use a single port number per process on a host as a destination point for any number of inbound connections from other hosts. When processes, such as long term services or data receivers, are defined in a fixed manner, these ports can be defined as constant values known by all nodes. This allows all state information about the communications to be maintained for only the duration of the message exchange. This makes the messaging protocol more resilient when the processes, hosts, or communication links experience continuous dynamic change.

The protocol works in the following manner:

The message sender initiates message transfer by sending the first segment of the message to the remote destination process. The tuple of source host address, source port, message id, destination host address, and destination port completely identifies the individual message. The message id is an integer which must be unique only as long as a message has not been completely transferred. Each message segment within a packet is identified by an index and a segment size. This allows a long message to be fragmented over multiple packets while retaining individual information about each fragment. The sender flags the last segment of the message to indicate completion of the message. If all segments of the message have been correctly received, the receiving process passes the completed message to the destination process, returns a final special acknowledgment (ack) to indicate that the message has been correctly received, and clears all state information about the message. When the sender receives this final ack, it informs the sending processor of the successful send and also clears state and buffers. To efficiently handle short messages and the tiny acks, each packet may contain several different messages, message segments, acks, and negative acknowledgments (nacks).

The protocol's error handling is rapid even over error prone channels that have significant delay. Each message segment is acked individually rather than as an update to a stream index as is done by TCP. This allows the retransmission of only the missed or errored segments. If the transmission window is greater than a single packet and the delay between packets is low, the sender may delay the ack slightly. This can reduce the number of ack messages sent in return by combining multiple acks in a single return packet. When the information transfer is bi-directional, any number of acks can be piggybacked with the message data. If multiple messages are queued for sending, the first portion of the next message may be added to a packet that contains the tail portion of the prior message. A message segment that must be retransmitted can be inserted in any packet destined for the receiver – even packets containing segments of the next message. As a packet is constructed, acks and nacks receive highest priority, retransmitted segments the next, and new data the last. Each message segment is acked as it is received even if there are prior segments that have not yet being received. If the communications channel has only a single path so that packets will not be received out of order, a nack will be sent by the receiving process immediately when it sees a message segment with an index higher than the next expected. If the sender receives acks out of order, it will immediately retransmit the intermediate segments not acked. These actions facilitate rapid retransmission of missed segments. There are, of course, timers on the sender to support retransmission of unacknowledged segments and on the receiver to send nacks if message segments cease to be received. If the sender retransmits a segment without an acknowledgment too many times, it assumes failure. It notifies the requesting process and frees all resources associated with the message. The receiver acts in a similar way when it receives no more packets and must send too many nacks.

In contrast to TCP, this protocol does not require any continuing state information about stream position counters and does not need periodic "keep-alive" packets to maintain this connection. In addition, the messaging protocol maintains no state information about the channel performance after the completion of the message, and therefore, should work better than TCP in environments like mobile ad hoc networks [de Oliveira 2002][Wang 2002]. However, this protocol does not support congestion control mechanisms such as the TCP slow-start to allow fair share of links in an efficient manner. Thus it is not suitable for general internet communications. Two rate control mechanisms are available: first, the number of outstanding message segments (the "window size") may be set to a small value to control the rate to a fraction of the link capacity. If the window size is only a single packet, the delay between packets is at least the round trip time. Then the transmission rate cannot exceed the available channel capacity even over a heavily loaded link. While slow, this configuration allows the protocol to be used in channels which aggregate a number of message senders. Variable delays between packet-transmission can also be used to ensure that the average data rate does not exceed a defined limit. This assures fairness in the use of network resources.

Timeouts, retransmission limits, immediate nack and ack, and ack coalescing were enhancements added to the original protocol by the Sandia development team as a part of this project. These optimizations aid in much more efficient retransmits of dropped packets. For the single path routing found in any source routed or bridged network, the protocol's active nack and out-of-order retransmit significantly speeds up the total message relay of multi packet messages in comparison to simple timeouts. The protocol was incomplete without at least a subset of these enhancements.

Simulation of Messaging Protocol

The model used for the messaging protocol simulation was completely custom written for OPNET. For initial development, OPNET's standard Ethernet model was used for link level connectivity. This stage of modeling served an important role in the overall simulation development. The easy to use and stable Ethernet model assured that all simulation errors were within the protocol model. The configurable link bandwidth and error rates of the Ethernet model were used to evaluate and confirm the protocol's error handling.

The custom protocol model is highly complex with 15 separate OPNET processes, more than 50 states, hundreds of variables, and well over 1000 lines of C code spread across the OPNET process definitions. The model includes scripted traffic generators, the protocol itself, and simple modeling of the hardware interaction within the interface card and across the internal bus. The simulation has a large number of run-time configuration variables to control the protocol's features and function. Further variables define hardware and kernel performance.

All elements, except the interface to Ethernet of the initial model, were applied to the wireless system model. As defined in project requirements, the simulated wireless environment has a high error rate, moderate delay, infrequent and smaller messages between well known remote processes, and dynamic topology changes. Simulations with this integrated model showed the basic viability of the enhanced messaging protocol and its robust error handling.

Wireless and Source Routing Design and Simulation

The wireless environment this project addresses creates unique challenges for routing:

- The single-hop wireless relay associated with the standard access point or cell techniques cannot be used. These techniques create a costly and fragile hierarchy of node capabilities, mobility, and link type useful only in friendly and stable areas. Communications must use multi-hop directed transmissions among peers in a mesh topology to either relay information or to an aggregating node of greater capability several hops away.
- The nodes may move at arbitrary intervals so that routing protocols that use regular updates may fail unpredictably.
- If messages between node pairs are reasonably infrequent in comparison to the dynamic nature of the network, maintaining full routing information between all nodes may be unjustifiably expensive.

These characteristics suggest that source routing between nodes on an as-required basis may be an attractive alternative. New routes are first required upon the initial attempt at communication. If intermediate nodes move or fail, repeated packet failures or message failure signal that the route must be updated. These factors allow route computations to be performed only as required and as often as the mesh environment changes.

A basic implementation of source routing can consume so much of the network communications capability that it renders the network unusable for some time. Modifications and enhancements can improve the efficiency by more than an order of magnitude. The most effective enhancements were developed as a part of this project. Simulation using the 802 wireless protocol and the messaging protocol allowed a comparison of the source routing enhancements in a realistic environment. By simulation, the comparison across the implementations could be made exact by performing the tests with a common set of topologies and traffic profiles.

Source Routing in a Challenging Wireless Environment

All simulations were performed in an environment of nodes connected by the 802.x wireless links without an Access Point. The messaging protocol with its unique performance and error handling capabilities was used to provide the message transfer. Messages were generated with controlled normal distributions for message size and rate per host, and either a uniformly distributed or single destination. The message sequence and timing could be reproduced by using the same pseudo-random seed for each simulation. The simulations measured the "worst case" situation beginning with all routing information invalid.

The basic concept of source routing works as follows: When one node, the sender, wishes to communicate with another, the receiver, it initiates a source routing action. The sender sends a broadcast packet requesting the route to the receiver. As each node in the ever expanding pool of nodes receives the broadcast, it adds its own address to the list of nodes that the broadcast has traversed. When this broadcast request reaches the receiver node, it sends this list of nodes back to the sender using inversed path. Although the receiver does not continue the broadcast relay, all other nodes do – perhaps several times as the broadcasts reach it via different paths. Each node must perform this costly action whenever it wishes to send to a node whose route is not already determined. These problems make this basic solution unacceptable.

There are obvious improvements to the basic form of source routing that virtually all source routing protocols (such as the IETF DSR [Johnson 2003]) define:

- The broadcast request can only continue for a limited number of hops. This prevents the request from infinitely echoing through the network. This limit should be slightly larger than the longest expected path to assure that the requests will reach the destination.
- Each node checks the partial route in the broadcast request packet for its own address. If the address is there, the broadcast is not relayed. No route which returns twice to the same intermediate node is valid.
- If a node receives a broadcast request that is at the hop limit, it sends a failed message to the sender via the route determined so far. This can be used by the source to determine if the destination node is unreachable.
- The sender uses a minimum delay period before it can initiate another source routing request to the same node. This prevents a stream of routing requests as each packet of a multi-packet message is queued for sending. This does not restrict requests for routes to other nodes. If the network has very limited capacity, the delay may be applied to all requests to throttle the rate of routing request generation by a node.
- The destination node compares the route in each route request packet it receives with its current route to the requester. Each time a better route is seen, the new route is returned to the source node. This assures the best routing at the potential cost of multiple routing replies.

As a part of this project, several further subtle or even seemingly contradictory enhancements were developed and evaluated by simulation. These enhancements can provide an order of magnitude reduction in cost with the protocol described above, and make response time to network topology changes engendered by node failure or movement.

1. Delay relay of the broadcast request by a short time. The minimum hop broadcast will reach all next hop nodes first assuring that the first route learned is the optimum route.
2. If 1 is used, each node should suppress other requests for the same destination by the same host by a period of time slightly longer than the request will take to propagate through the network. This greatly cuts down on the number of broadcasts by assuring that each node repeats a request only once and that the relayed broadcast contains the optimum route.

The following enhancements are based upon the concept that a node may learn a guaranteed route to the source of the packet by simply inverting the path that the packet has taken so far. This is obviously a functional up-to-date path because the packet has just come to the node via that path. This means that a node may passively learn many routes simply by examining all packets that transit it whether they are broadcast requests or just normal data packets relayed by the node. This concept can be used even further. Routes to all of the nodes that the packet has transited can be obtained by the same method by examining each hop of the route so far. This means that routes can be established between many nodes just by a single broadcast route-request as it expands through the mesh, and that all nodes can learn a route to the source of the broadcast. This simple concept allows enhancements that can reduce routing requests by more than an order of magnitude, and cheaply and rapidly correct out-of-date routing in a dynamic environment.

The inverse situation which at first seems obvious is, in fact, incorrect. The node can learn nothing valid about a route to the destination from the routing information in a packet transiting it. That information was simply the source node's idea of a route to the destination and may well be obsolete or incorrect. This also means that a node cannot "short circuit" a routing request by responding with its own information about a route to the destination because its own information about the continuing route may be obsolete.

These concepts lead to two further optimizations; the first is also adapted by IETF's DSR proposal:

3. As described in the paragraph above, learn routes to each node that has relayed the packet to the current node.

The final optimization seems to contradict the source routing concept. Simulation shows that it should not be used in networks with a communication pattern of many to many. In a network with communications characterized by a general flow of data from many peripheral nodes into a limited number of primary aggregating nodes, this enhancement yields remarkable results with an optimum response to network changes.

4. When a routing request reaches its destination node, the destination node should respond with a broadcast routing request to a nonexistent node. This is most useful when the primary traffic pattern is messages inbound from minor nodes to a central or aggregating node. As described above, all nodes within the radius of the max hops will learn a current route to the broadcast source. If enhancement (1) is used then the route learned will be the optimum route. If (2) is applied as well, the entire spanning tree across all nodes to the

sender will be established with the absolute minimum of packets. This broadcast should only be done after the request arrives; this implies that there is a change in connectivity within the network where at least the route between the source of the request and this node is no longer correct and must be updated. If this enhancement is used between the destination node and all other nodes, it will be updated only when the network topology has changed, and as soon as this change affects communications. Topology changes that do not affect actual communications will be ignored. A final benefit to this approach is the complete elimination of return route relay. The requesting node learns the route in the same manner as do all other nodes – the broadcast packet which contains the newly created spanning tree to the source and all nodes in between passes through the requester.

Simulation Results for Various Routing Enhancements

Three simple metrics provide excellent means of comparing the effect of these enhancements: the sum of broadcasts across all nodes, the number of route-response packets relayed, and the number of simulation events. Broadcasts are very costly because they must be directly processed by all adjacent nodes, consume bandwidth, and require special handling to prevent infinite echoing across the network. Route-relay packets have the same cost as data packets but do not transfer user data. The last metric, simulation events, is unique to this method of simulation. The number of events is proportional to the "work" going on in the network – the creation and transfer of messages, the determination of routing, and the number of actions taken by the wireless simulator as it attempts to secure the wireless channel, broadcast, and rebroadcast upon failure. There are a minimum number of events that must be performed no matter how efficiently the network is operating; this prevents a ratio comparison. Yet this metric is perhaps the most accurate metric of overall network performance.

The simulations reported here had a number of common factors. The topology had 20 hosts (or nodes) arrayed in a somewhat irregular mesh format. Each node could communicate with at least two others. The destination node used for the single-node test was near the center of the mesh. The nodes used for the limited-number-of-node tests were toward the inner portion of the mesh. All tests used a common rate for message creation for each node, where inter-message-delay times were randomly generated using a normal distribution with a mean of 9 seconds and a standard deviation of 3. This low rate was necessary to prevent overloading of the network that used 1 mbps wireless links to simulate worst case delays and conflicts for channel access. The message destinations for each message had a uniform distribution across the set of destination nodes. All tests used the same seed for the pseudo-random number generator to assure that the traffic pattern was identical for every test that used the same number of destination nodes

The simulations measured the worst case scenario of all topologies, routes are not known by any nodes. The initial distribution of message-timing for each node to simulate the ongoing traffic was the same as the continuing inter-message rate after a massive network disruption.

Each row in the tables below represents a specific combination of the enhancements described above:

A) Basic Source Routing

B) Basic Source Routing, Record Routes (3.)

C) Basic Source Routing, Suppress Broadcasts (2.)

D) Basic Source Routing, Suppress Broadcasts (2.), Record Routes (3.)

E) Basic Source Routing, Suppress Broadcasts (2.), Record Routes (3.), Broadcast On Request (4.)

The metrics represented in the columns are:

1. Broadcasts: The total number of individual broadcasts performed. A single source route request can generate tens of broadcasts
2. Routing Packets Relayed: The total number of relays through intermediate node of routing replies. The single response of a destination node to a routing request may be forwarded through several nodes before reaching the requester.
3. Simulation Events: See above
4. % Baseline Cost: The "cost" is defined as the sum of broadcasts and routing packets relayed. This is compared to the cost of basic source routing.

The first table shows all to all messaging:

Table 1. All to all messaging.

	Broadcasts	Route Relays	Simulation Events	% Baseline Cost
A.	3321	1347	5654877	100.00%
B.	843	680	2773917	32.63%
C.	1674	287	1596572	42.01%
D.	239	51	576622	6.21%
E.	469	0	1079771	10.05%

The second table shows the other extreme, all to one messaging. Note the extreme efficiency gained by using Broadcast On Request (4):

Table 2. All to one messaging.

	Broadcasts	Route Relays	Simulation Events	% Baseline Cost
A.	309	137	1105597	100.00%
B.	236	166	1032366	90.13%
C.	155	41	648361	43.95%
D.	155	41	513120	43.95%
E	34	0	365002	7.62%

The third table shows an intermediate test of messaging from all to eight. Note that the routing is almost as efficient with or without Broadcast On Request (4). A test with seven aggregation nodes (not shown) indicated that E was slightly better. This ratio of 20 to eight represents the break-over point for using this enhancement. Only the last two options sets were used; they were the only relevant configurations for this test:

Table 3. All to eight messaging.

D.	212	32	510436	100.00%
E	290	0	564841	118.85%

Summary

The simulation created for this project showed the basic viability of the original design approach. The enhancements designed as an extension to the project make the approach significantly more attractive. All elements of the simulation are available for further use by an extension to this project or support for new associated projects.

Virtual Geographic Routing

This section of the report introduces a novel routing protocol called virtual geographic routing (VGR). It is highly scalable and suitable for wireless nodes in self-configuring networks.

Scalability of ad hoc routing protocols is only a meaningful issue if ad hoc networks themselves are scalable. Shepard [Shepard 1995] has shown that, under certain conditions, wireless networks are indefinitely scalable. He undertakes an analysis of propagation and interference models and shows that the noise level of spread-spectrum systems is manageable even in systems of billions of nodes. He also presents a pseudo-random time-division MAC protocol that requires no central coordination and can scale arbitrarily. He observes that the throughput of the network scales linearly with the number of nodes, provided the path length is bounded to a maximum number of hops. Li *et al* [Li 2001] present statistical arguments and make the stronger statement that if the expected path length approaches a constant as the number of nodes increases, the per-node communication capacity approaches a constant. They apply their analysis to the 802.11 MAC protocol. Note that this criterion places a restriction on average path length but not on the number of nodes or the size of the space they inhabit.

Geographic Routing

Geographic routing [Giordano 2003] works on a simple principle. At every hop, a message is forwarded to a neighboring node that is geographically closer to the destination until the message finally reaches the destination. The advantage of geographic routing is that it requires only a small, constant amount of storage at each node [Karp 2000b]. Therefore it is massively scalable.

Geographic routing works best when the nodes are densely enough placed that the network path approximates a straight line between the source and the destination. It fails when it encounters a local minimum in the distance function, a node at which there is no neighboring node closer to the destination. The node configuration facing such a node is called a "void" (see Figure 18). In Figure 18 and elsewhere we will consider a communications network to be represented by a graph whose vertices are the radio nodes and whose edges are the communication links between neighboring nodes; neighboring nodes are precisely those that can communicate directly with each other.

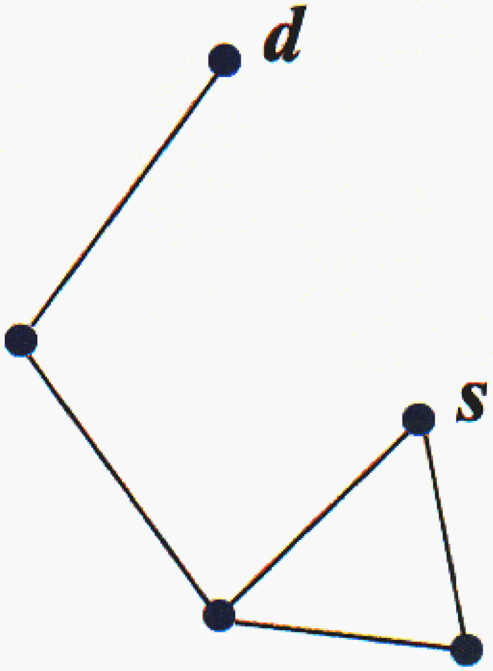

Figure 18. A *void* or local minimum of the distance function. A message traveling from ***s*** to ***d*** encounters a void at ***s*** since no neighbor of ***s*** is closer to ***d*** than ***s*** itself is.

A workaround for local minima is perimeter routing [Karp 2000a], which uses the *right-hand rule* to traverse the perimeter of the polygon whose interior constitutes the void, as one might do in traversing a maze. Perimeter routing only works if none of the communication graph's edges cross, where edges are represented by straight lines (that is, if the communication graph is planar and the geographic node layout is a planar embedding of a straight line representation of the communication graph [Bollabás 1998]). Normally, such a planar embedding of the full graph does not exist, but a connected subgraph having such an embedding can be produced provided the graph satisfies the *unit graph condition*. The unit graph condition specifies that a link exists between two nodes if and only if they are less than a fixed distance apart. (The subgraph is produced by removing selected edges from the communication graph, and the unit graph condition ensures that the resulting subgraph is still connected.)

When the radio ranges vary (at least by more than a small amount [Barrière 2001]), the unit graph condition is not satisfied and perimeter routing cannot be applied. Radio ranges may vary for a variety of reasons, for instance because there are obstacles in the communication path. The transmission ranges may also be varied intentionally in order to conserve power or minimize noise in the network. Another limitation of perimeter routing is that it cannot be applied to three-dimensional node layouts.

A disadvantage of geographic routing is that the nodes must know their own locations, which typically means that special equipment such as Global Positioning System (GPS) receivers must be used. GPS receivers add expense, size and weight, increase power consumption, and do not work wherever they cannot receive the signals of geopositioning satellites, for instance indoors.

Virtual Geographic Routing

Virtual geographic routing replaces the physical location of a node with a virtual location developed solely using information about the connectivity of the nodes. In this report, we take the virtual coordinates of a node to be the number of hops from a small number of distinguished nodes, which we will call *anchor nodes*. Thus a node's virtual coordinates give its location in a *K*-dimensional virtual space, where *K* is the number of anchors. The number of hops to each anchor node can be developed by a simple distributed algorithm, similar to the distributed Bellman-Ford algorithm [Lynch 1996] but using messages that include only the hops counts to the anchor nodes rather than to all nodes.

The simple geographic routing algorithm (at each step move to a node closer to the destination) can be used with virtual coordinates as well as physical location coordinates. Various distance metrics can be used; the familiar Euclidean metric works well. Our research has shown that geographic routing using the virtual coordinates described here is as effective as geographic routing with physical coordinates and more effective in the presence of barriers to transmission.

Virtual geographic routing (VGR) is stalled by local minima of the distance metric, just as real geographic routing is. In addition, it displays a phenomenon that we will call *aliasing*, in which two different nodes have the same virtual coordinates. Since such nodes can be several hops apart, aliasing causes a problem for the routing algorithm.

VGR works best when there are three or more anchors, since using only two anchors produces unacceptable numbers of aliased nodes. Perimeter routing, which applies only in two dimensions (real or virtual), cannot be used to deal with voids. At any rate, it is desirable to find a way of dealing with voids that works in the presence of obstacles and with variable radio ranges in general.

This report presents a highly scalable routing algorithm that uses virtual geographic routing supplemented by other routing techniques for dealing with voids and aliasing.

Location Service

For some applications, such as geographically based data storage and retrieval [Ratnasamy 2002][Shenker 2003], it is not necessary to associate locations with particular nodes. Often, however, the nodes are distinguished by unique node identifiers, and a message must be routed to the node with a particular identifier. In that case, for either real or virtual geographic routing, it is necessary to have a means of discovering a node's location given its identifier.

A service that supplies the location of a node given its node identifier is called a *location service*. A location service that is particularly suited to systems using geographic routing was presented in [Li 2000]. The service is completely distributed, letting every node play the role of location server for a number of other nodes. The service is itself based on geographic principles and works without any node knowing the identity of any of its location servers. It is presented in connection with real geographic routing and based on a recursive subdivision of the unit square, but the analysis and the algorithms carry over to K-dimensional virtual coordinate space. It requires an average amount of storage at each node that is proportional to the total number of nodes with a very small constant of proportionality. The service has a communication pattern such that the expected path length approaches a constant as the number of nodes increases [Li 2001], so it meets the scalability criterion mentioned above.

The location service will not be further discussed in this report.

Handling Voids

We tried a number of routing techniques for handling the cases in which virtual geographic routing fails. The one that showed the best scalability was the simplest one: doing an expanding ring search to find a node geographically closer to the destination and caching the results. Use of expanding ring search in connection with real geographic routing was suggested in [Morris 2000].

Note that the term "broadcast" will be used here to mean a transmission by a node to all of its one-hop neighbors. Such a broadcast is the natural means of communication in the wireless medium.

Escape from a void can be found by flooding a search to find a node that is closer to the destination in the virtual geographic metric. In order to keep the communication load on the network low, and especially since a closer node may be found in relatively few hops, an expanding ring search is used. Other ad hoc routing techniques, such as AODV [Perkins 2003] and DSR [Johnson 2003] use flooding or expanding ring search for discovery of the entire route.

An expanding ring search broadcasts a search request with a maximum search radius attached. The request is broadcast by all nodes that receive it if they cannot satisfy the request and the search radius is not exhausted. (Such a radius is often called "time to live" or TTL.) Positive results are sent back to the node that originated the request and can be consolidated along the way to avoid congestion of the links near the requesting node. If no positive results are found at the current radius, the search is repeated with an increased radius. In a static, connected network, such a search will always eventually be successful.

The path traveled by the request is accumulated along the way and included in the request. The response is returned along the reverse path, and at each of the nodes on the path, the next hop from the originating node toward the satisfying node is cached in a table of "escape hops". Such escape hops are associated with the satisfying node in the table.

When a void is encountered in the course of routing a message, the escape table is searched to see if it contains any entry associated with a node that is closer to the destination. If it does, the next hop in that entry is used, and the closer node becomes an intermediate routing destination. Before it is forwarded to the next hop, the message is marked to show that it is no longer in geographic mode, and the identity of the intermediate destination and the virtual geographic coordinates of the node at which the message left geographic mode are included in it.

A node receiving such a message first checks to see if it is itself closer to the destination than the node at which the message left geographic mode. If it is, it restores the message to geographic mode and continues the routing. If it is not, it looks in its escape table to see if it has an entry associated with the intermediate destination given in the message. If it does, it forwards the message to the next hop in that entry. Note that in a static network, it will always find such an entry. If it does not, it acts as if were the node that first encountered the void, as explained in the previous paragraph.

Handling Aliases

An alias is recognized when the message reaches a node that has the same virtual geographic coordinates as the destination but is not the destination (that is, it has a different node identifier). An expanding ring search is used here, too. The search succeeds when the search request message reaches the node with the destination's node identifier. In this case, the response to the search returns the entire path from the originating node to the destination, and the originating node caches the path upon receiving the response, associating it with the destination's identifier. As an optimization, the destination caches the reverse path, associating it with the originating node.

When a node recognizes an alias in the course of routing a message, the alias cache is consulted. If the path to the destination is already present in the cache, no search need be done. Before the message is forwarded to the first hop in the path, the message is marked to show that it is in alias mode and the remainder of the path is placed in the message. At each step on the path, the next hop is removed from the message and the message is forwarded to it. That is, source routing is used between the two aliased nodes.

Cache Considerations

Note that in practice, a tradeoff can be made between the size of the cache used for void and alias routing and routing overhead messages. If the cache is restricted to a fixed size and is full when a new entry must be made, old entries can be removed according to a least-recently-used scheme to make room for a new entry.

Placement of Anchor Nodes

The placement of the anchor nodes is important. An uneven distribution of the anchors among the other nodes tends to produce a greater number of local minima and aliases. Methods of placing anchors can be divided into three categories, *a priori*, random and algorithmic. A good placement can be assured if it is possible to position the anchor nodes *a priori*.

The anchors can be assigned randomly by allowing each node to draw a random number and become an anchor if the number is less than some threshold value. The overhead of random placement is extremely low, but random placement may result in an uneven anchor distribution. Also, some knowledge of the likely size of the network is implied in choosing the threshold value. The probability of suffering the effects of an uneven distribution can be lowered by using a greater number of anchors (raising the threshold value), but doing so costs more overhead in the form of storage to hold virtual addresses and bandwidth to transmit them.

With random placement, there is always the chance that too few or too many anchors will select themselves. If there are too few anchors, it is necessary to have another round of anchor selection, to replace the current anchors (possibly using a higher threshold value) or to add to them. If there are too many anchors, some can be discarded during the virtual address computation (say those with lowest node identifiers).

If the anchors are chosen algorithmically, it is desirable to make this computation require as little overhead as possible. One simple expedient that seems to work well in practice is to allow those nodes to become anchors for which there are no nodes with lower node identifiers within an L-hop radius. If the size of the network is known, L can be chosen to produce the desired probable number of anchors. The expected number of anchors in a random layout is a complicated function of L and the number of nodes, but it can be estimated experimentally. To select anchors in this way, each node must send and receive L messages before the anchors can be determined and the computation of the virtual coordinates for the nodes can begin.

If the size of the network is not known, a value of L that is likely to produce at least enough anchors can be used and the superfluous anchors can be discarded during the virtual address computation, as described above. In deciding which anchors to discard, it is helpful to keep the anchors as evenly distributed among the nodes as possible. A simple algorithm for doing this follows. It has the advantage that it can be combined with the computation of the virtual coordinates but the disadvantage that it requires longer messages. Start with *L* small, say 2. Include in the messages sent by the nodes the distances in hops between each pair of anchors, as soon as they are available (note that the distances between anchors that are near each other will be available sooner than distances between distant anchors). Let there also be a limit *U* on the total number of anchors. A node in any network with more than a few nodes will soon accumulate more than *U* anchors and have to discard some. In order to select which anchors to discard, the node increases *L* by 1 and discards one of any two anchors that are closer than *L* hops to each other (choosing, say, the one with the lower node identifier to discard), repeating this step until the number of anchors is again no greater than *U*.

A variation is to discard one of the two anchors that are closest to each other, say the one with the lower identifier. If the internode distances are correct, all nodes will discard the same anchors, since the *U* anchors with the greatest internode distances will be retained no matter in what order the node receives information about the various anchors. This technique has the advantage that the size of the placement protocol messages can be decreased, since for each anchor, only the smallest internode distance to any anchor with a greater node identifier need be retained.

Both of the algorithmic techniques described above require that some additional messages be sent beyond those required to compute the virtual coordinates. Enough additional messages are required to compute the path lengths between the anchors, at least between those that turn out to be close enough to each other that one is discarded. The internode distances (like the virtual coordinates) may not stabilize immediately. If the interanchor distances are propagated before they stabilize, it could lead to different anchors being discarded by different nodes. As long as the nodes retain a sufficiently large set of common anchors, however, routing can still be done effectively.

Implementation

The main factor affecting the scalability of geographic routing is the amount of local storage required at each node. Each node must keep information about its neighbors, so that it can choose the neighbor through which to forward a message. In the implementation presented here, each node also keeps an escape table, a table of hops used to escape from local minima of the distance function. In addition, VGR requires an alias table to contain the paths between aliased nodes.

For each neighbor, real geographic routing (RGR) requires the neighbor's node identifier and its physical location. Thus in a two-dimensional node layout, three memory cells per neighbor suffice (identifier, X and Y) for each neighbor, and in a three-dimensional layout, four cells suffice. VGR requires one cell for the neighbor's node identifier and one cell for each anchor used in the virtual coordinates of the neighbor. VGR typically requires a greater amount of storage per neighbor than RGR, since there are likely to be more than three anchors. If there is the possibility that the anchors sets used by the nodes may differ for different nodes (as explained above in the section on anchor placement), two memory cells per anchor are required, one for the anchor's node identifier and one for the number of hops from the node to the anchor. Note, however, that the amount of storage devoted to neighbor information at a node, for both RGR and VGR, depends only on the number of the node's neighbors (the node degree), not on the size of the network.

Geographic routing tends to perform better in networks with high average node degree. However, such networks require greater amounts of local storage for neighbor information. In exceptionally dense networks, for both RGR and VGR, it may be desirable to route through only a subset of a node's neighbors in order to minimize the amount of local storage used.

The nodes must exchange messages in order to inform each node of the identities and locations of its neighbors. In a static network, this can be done at start-up. In a dynamic network, such messages must be exchanged periodically. A network qualifies as dynamic if nodes can move, fail, or sleep to conserve power, or if the changing transmission characteristics of the medium can cause communication links to fail.

The likelihood of encountering a local minimum can be reduced somewhat by expanding the definition of a node's neighbors to include all k-hop neighbors. The node considers all neighbors within k hops when trying to find a node closer to the destination. The disadvantage is the requirement for more local storage to hold neighbor information and larger messages to exchange neighbor information.

Performance

Performance studies were done using random two-dimensional node layouts in the unit square, with and without the presence of transmission barriers. Two such layouts are shown in Figures 19 and 20. The barriers shown in Figure 20 were used in the performance studies. For ease of computation, the barriers are treated as areas that no nodes may inhabit rather than areas through which no communication may pass; the effect is nearly identical. The transmission range was adjusted to produce varying numbers of neighbors. Statistics were developed over all source/destination pairs for each layout. Only connected graphs were used, since it is impossible to route to disconnected nodes.

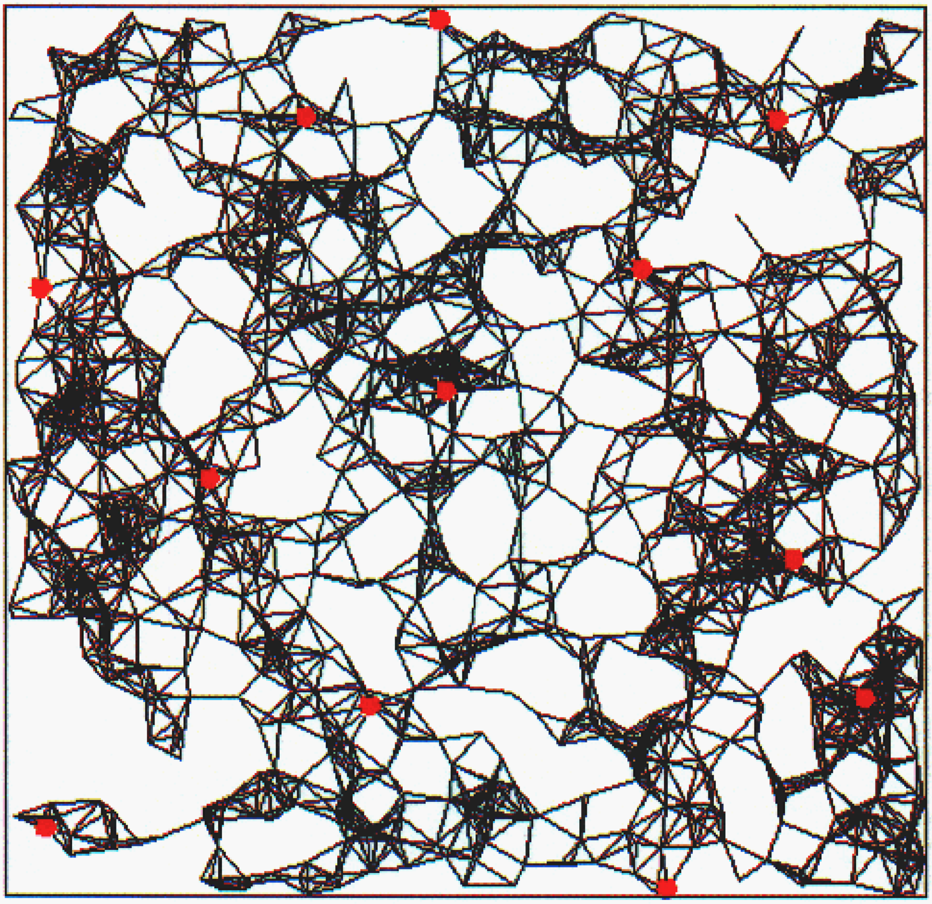

Figure 19. Network graph without barriers. 1016 nodes, average node degree 9.32. The red dots are anchor nodes.

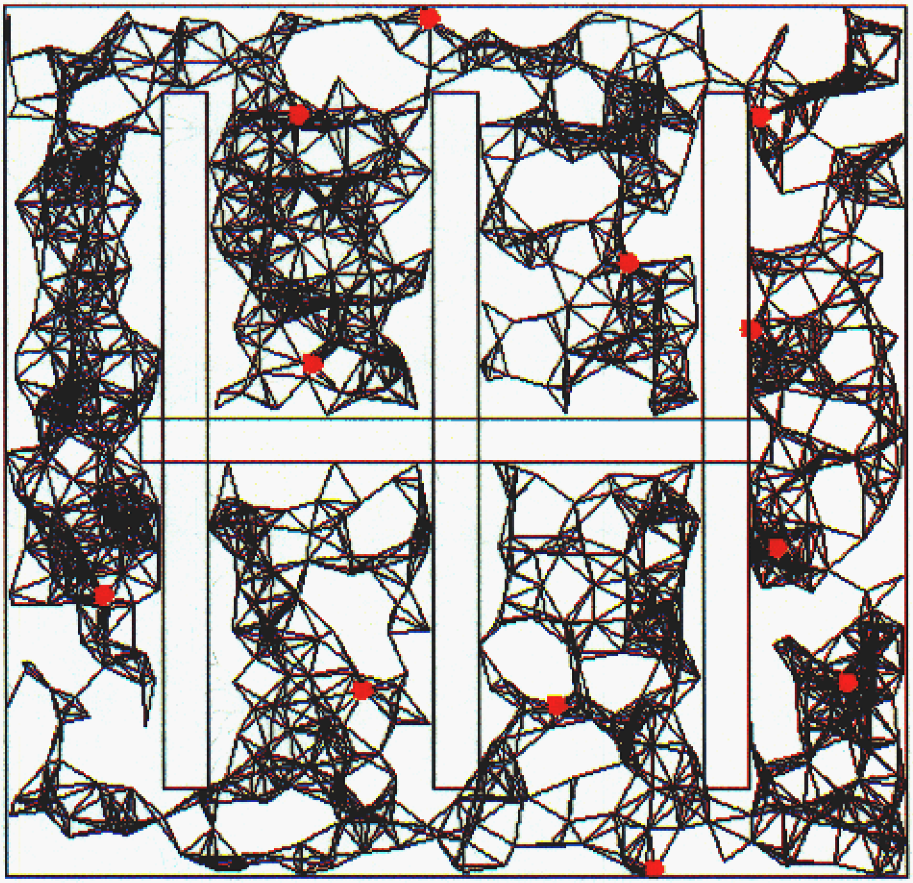

Figure 20. Network graph with barriers. 1024 nodes, average node degree 9.93. The red dots are anchor nodes.

The graphs and tabulated results compare VGR, RGR and ideal shortest-path routing. Neither VGR nor RGR finds shortest paths between source and destination, though VGR does no worse than RGR in this respect. Both VGR and RGR approach shortest paths as the average number of neighbors increases.

Both RGR and VGR use one-hop neighborhoods in determining the next hop toward the destination.

VGR and RGR both use the same expanding ring search mechanism to deal with voids, and VGR uses it to deal with aliases. The initial search radius is two hops (the least power of two greater than the neighborhood radius), and the search radius doubles each time the request fails to be satisfied within the current radius. If the local cache holds the next hop around a void or toward an alias, no search is done. The numbers of search and reply packets are tabulated. These numbers would be more significant in a dynamic network, where they would be considered as routing overhead and amortized over all transmissions.

VGR anchors were selected by the algorithm described above that constrains the number of anchors by discarding one of the two closest anchors when the number of anchors rises above a limiting value. A limiting value of 12 was used, and all nodes use the same anchors. Using fewer anchors would lower local storage requirements, but we observed that it also tends to produce slightly longer path lengths and larger alias tables and require a larger ring search radius. Further study of the effect of the number of anchors would be useful.

The primary metric with respect to scalability is the maximum amount of memory used for the routing caches at any node. Local memory is measured in memory cells. Each entry in the escape table requires two cells (for intermediate node and next hop identifiers). Each entry in the alias table requires a number of cells equal to one plus the path length to the alias (for the alias's identifier and path). Each entry in the neighbor table takes three cells for RGR (for node identifier, *X* and *Y*) and 1+*A* cells for VGR, where *A* is the number of anchors (for node identifier and virtual coordinates). The cache size given is the size required for each node to be able to route messages to every other node, so it is an upper bound on the amount of memory that would be required in an actual network.

Performance statistics from the runs are summarized in a series of graphs and tables in Appendix A.

Figures 21 and 22 show the dependence of average path length on average node degree, with and without barriers; the graphs have about 1000 nodes. The average path length approaches the average shortest path length as the average node degree increases. When the average node degree reaches about 10, the path lengths for both real and virtual geographic routing are quite close to the shortest path lengths. The approach is a little more rapid in the absence of barriers, and VGR approaches shortest paths somewhat more rapidly than RGR when barriers are present.

Figures 23 through 38 contain graphs showing the maximum storage required at any node during a run, for layouts of about 500, 1000 and 2000 nodes in size and having an average node degree between 4 and 11. The storage usage is broken down into its components on each graph: routing cache (escape and alias tables), neighbor information tables and the sum of the two. Note that the maximum amount of storage used at any node is not simply the sum of the maximum amount of cache at any node and the maximum size of the neighbor table at any node, since nodes with many neighbors tend to see fewer voids. Figures 23 through 30 are for layouts without barriers and Figures 31 through 38 are for layouts with barriers.

Figures 23 through 26 are for layouts with relatively low average node degree and no barriers. For both real and virtual routing, there is a linear increase in routing cache size. VGR uses more routing cache, since it must store more information per neighbor, but its increase has a lower slope. Figures 27 through 30 are for average node degrees around 8 and 10, respectively, and both show nearly flat storage usage above 1000 nodes. The major contributor for both real and virtual routing is the neighbor tables. When nodes have many neighbors, voids are less of a problem. The rather dramatic change in behavior as average node degree rises is an example of the sort of criticality phenomena that are characteristic of random graphs [Bollobás 1998].

The graphs for layouts with barriers, in Figures 31 through 38, show behavior that is generally similar to that without barriers. We will see below, though, that there are important differences between VGR and RGR in the way this performance is achieved.

The fact that the routing cache size tends to stop increasing with the number of nodes as the average node degree increases suggests that in random networks with dense enough interconnections, escape from voids can be accomplished by essentially local means.

Tables 4 through 27 provide information about several other aspects of the routing protocols. The ratio of geographic to non-geographic hops, the ratio of expanding ring search packets to total transmissions and the average flood radius tell the extent to which the protocols have to rely on non-geographic routing. Tables 4 through 15 give the results for networks without barriers. For all network sizes, as the average node degree increases from about 4 to about 10, the fraction of the routing steps performed geographically increases from 65-75% to around 95%. The performance of VGR and RGR is very similar, with VGR enjoying a slight advantage in the ratio of geographic hops and a decided advantage in the amount of control traffic, as measured by search packets, particularly in sparsely connected networks.

Tables 16 through 27 show corresponding results for networks with barriers. Here, the advantage of VGR is more marked. In networks of a thousand nodes or more, the fraction of geographic hops taken by RGR never exceeds 66% and the amount of control traffic produced by RGR ranges from 8 to 38 times as much as VGR. VGR, meanwhile, behaves nearly as well as in networks without barriers. It might be expected that real geographic routing would have trouble when routing paths no longer resemble straight lines in physical space. It appears that even when barriers are present, the VGR paths do a reasonably good job of approximating straight lines in K-dimensional virtual space.

It would be desirable to extend the scalability studies to even greater numbers of nodes. The present effort was limited by the compute time required for large runs. It may be possible to devise a sampling scheme that is more efficient than simply computing all possible routes, as the present software does. Parallelizing the software and running it on a multiprocessor system might also speed it up, as long as the communication costs do not outweigh the gain in aggregate computing power.

Conclusion

A novel routing protocol has been presented based on geographic principles and using coordinates derived only from the connectivity of the network. It was shown to perform at least as well as geographic routing based on physical coordinates and considerably better in the presence of transmission obstacles. Simple means of dealing with the failings of geographic routing were employed, which will work with either virtual or physical coordinates, with three-dimensional physical node layouts, and in the presence of obstacles. The protocol shares the excellent scalability properties of conventional geographic routing protocols.

Future Work

We can extend the investigation to even larger networks by speeding up the analysis code. Ways of achieving a speed-up are to develop an appropriate sampling scheme and to parallelize the code.

The next logical steps are to do stochastic simulations and simulations including the effects of the wireless medium.

We have consideredVGR only in the context of static networks. It remains to extend the protocol to mobile networks. As a node moves, its virtual coordinates will change, even as its real coordinates do. It can learn its current virtual coordinates from its one-hop neighbors (its distance from anchor *A* is the least distance of any of its neighbors from *A* plus one). If the anchor nodes can move, however, it will be necessary to select new anchors from time to time. Interesting questions include how to recognize when it is time to change the anchors and whether it is possible to devise an incremental way of changing them.

It would likely improve the efficiency of the search for non-geographic routing hops to use some adaptation of the techniques for cutting down route request and reply overhead described in the previous section of this report.

Once the distributed algorithms making up the VGR protocol are implemented, they can be used directly in simulation systems such as SWAN [Perrone 2001][SWAN 2003]. Subsequently, they can be transplanted to suitable wireless platforms for field tests.

References

C. L. Barrett, M. Drozda, A. Marathe, V. Madhav, and V. Marathe, 2002, Characterizing the Interaction Between Routing and MAC Protocols in Ad Hoc Networks, in *Proceedings of The Third ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC 2002)*, held at Lausanne, Switzerland, June, 2002.

L. Barrière, P. Fraigniaud, L. Narayanan and J. Opatrny, 2001, Robust Position-based Routing in Wireless Ad Hoc Networks with Irregular Transmission Ranges, in *Proceedings of the 5th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, P. Crescenzi and B. Yener, ed., pages 19-27, ACM Press, New York, New York.

V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, 1994, MACAW: A Media Access Protocol for Wireless LAN's, *SIGCOMM*, pages 212-225.

B. Bollabás, 1998, *Modern Graph Theory.* Springer Verlag, New York, New York.

K. Fall, and K.Varadhan, 1998, *NS-2 Notes and Documentation*, available at http://www mash.cs.berkeley.edu/ns.

S. R. Das, C. E. Perkins and E. M. Royer, 2000, Performance Comparisons of Two On-demand Routing Protocols for Ad Hoc Networks, in *Proceedings of IEEE Infocom 2000: the Conference on Computer Communication*, held at Tel Aviv, Israel, March, 2000. Available at http://www.ieee-infocom.org/2000/papers/673.ps.

S. Giordano, I. Stojmenovic and L. Blazevic, 2003, Position Based Routing Algorithms for Ad Hoc Networks: a Taxonomy, to be published in *Ad Hoc Wireless Networking*, X. Cheng, X. Huang and D.Z. Du, ed. Kluwer.

IEEE Computer Society LAN MAN Standards Committee, 1997, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. New York, New York. IEEE Std. 802.11-1997.

D. Johnson, D. Maltz and Y.-C. Hu, 2003, The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR), April 15, 2003, available at http://www.ieft.org/internet-drafts/draft-ietf-manet-dsr-09.txt.

P. Karn, 1990, MACA – A New Channel Access Method for Packet Radio. *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, September 22, 1990.

R. Karp. Geographic Routing for Wireless Networks, 2000a. Ph.D. dissertation, Harvard University, Cambridge, Massachusetts.

R. Karp and H. T. Kung, 2000b, GPSR: Greedy Perimeter Routing Using Partial Information for Wireless Ad Hoc Networks, in *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking*, R. Pickholtz, S. K. Das, R. Caceres and J. J. Garcia-Luna-Aceves, ed., pages 243-254. ACM Press, New York, ew York.

L. Kleinrock, and F. Tobagi, 1975, Packet Switching in Radio Channels: Part I-Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics, *IEEE Transactions on Communications*, 23(12), pages 1400-1416.

J. Li, J. Janotti, D. S. J. De Couto, D. Karger and R. Morris, 2000. A Scalable Location Service for Geographic Ad-hoc Routing, in *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking,* R. Pickholtz, S. K. Das, R. Caceres and J. J. Garcia-Luna-Aceves, ed., pages 120-130. ACM Press, New York, New York.

J. Li, C. Blake, D. De Couto, H. Lee, and R. Morris, 2001, Capacity of Ad Hoc Wireless Networks, in *Proceedings of the 7th ACM International Conference on Mobile Computing and Networking (MobiCom '01),* held in Rome, Italy, July, 2001, pages 61-69.

J. Liu, L. Perrone, D. Nicol, M. Liljenstam, C. Elliott and D. Pearson, 2001, Simulation Modeling of Large-scale Ad-hoc Sensor Networks, presented at *European Simulation Interoperability Workshop 2001*, Harrow, UK, June, 2001.

J. Lockhart, 2001, *Extensible Message Layer Protocol*, Internal Report, Sandia National Laboratories, California (available upon request.)

N. A. Lynch, 1996, *Distributed Algorithms.* Morgan Kaufman, San Francisco, California, 1996.

R. Morris, J. Janotti, F. Kaashoek, J. Li and D. S. J. DeCouto, 2000, CarNet: A Scalable Ad Hoc Wireless Network System, in *Proceedings of the 9th ACM SIGOPS European workshop: Beyond the PC: New Challenges for the Operating System,* Kolding, Denmark, pages 61-65. ACM Press, New York, New York.

NIST Wireless Communications Technologies Groups, 2003a, Simulation Model for the AODV MANET Routing Protocol, available from http://w3.antd.hist.gov/wctg/manet/prd_aodvfiles.html.

NIST Wireless Communications Technologies Groups, 2003b, Simulation Model for the DSR MANET Routing Protocol, available from http://w3.antd.hist.gov/wctg/prd_dsrfiles.html.

R. de Oliveira and T. Braun, 2002, *TCP in Wireless Mobile Ad-Hoc Networks,* Technical Report. IAM-02-003, July 2002.

C. Perkins, E. Belding-Royer and S. Das, 2003, Ad Hoc On-demand Distance Vector (AODV) Routing. RFC 3561, Internet Engineering Task Force, July, 2003, available at http://www.ieft.org/rfc/rfc3561.txt.

S. Ratnasamy, B. Karp, Y. Li, Y. Fang, D. Estrin, R. Govindan and S. Shenker, 2002, GHT: A Geographic Hash Table for Data-centric Storage, in *Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications,* Atlanta, Georgia, pages 78-87. ACM Press, New, York, New York.

S. Shenker, S. Ratnasamy, B. Karp, R. Govindan and D. Estrin, 2003, Data-centric Storage in Sensor Nets, in *ACM SIGCOMM Computer Communication Review,* 33(1), pages 137-142. ACM Press, New York, New York.

T. J. Shepard, 1995, Decentralized Channel Management in Scalable
Multihop Spread-Spectrum Packet Radio Networks. Ph.D. dissertation,
MIT-LCS-TR-670, Massachusetts Institute of Technology.

SWAN Simulation System, available at http://www.cs.dartmouth.edu/research/SWAN/.

J. Walrand and P. Varaiya, 2000, *High-Performance Computer Networks,* 2nd edition. Morgan Kaufman, San Francisco, California, 2000.

F. Wang and Y. Zhang, 2002, Improving TCP Performance over Mobile Ad-Hoc Networks with Out-of-Order Detection and Response, *3rd ACM Int'l Symposium on Mobile Ad Hoc Networking and Computing.*

Appendix A

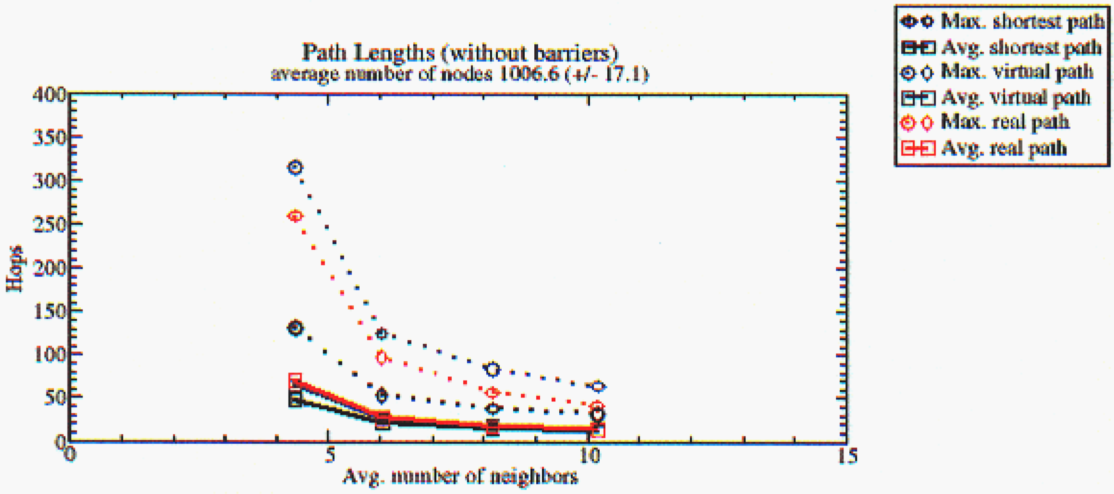

Figure 21. Real geographic routing, path length versus average node degree, without barriers. The data points are averages over 10 runs.

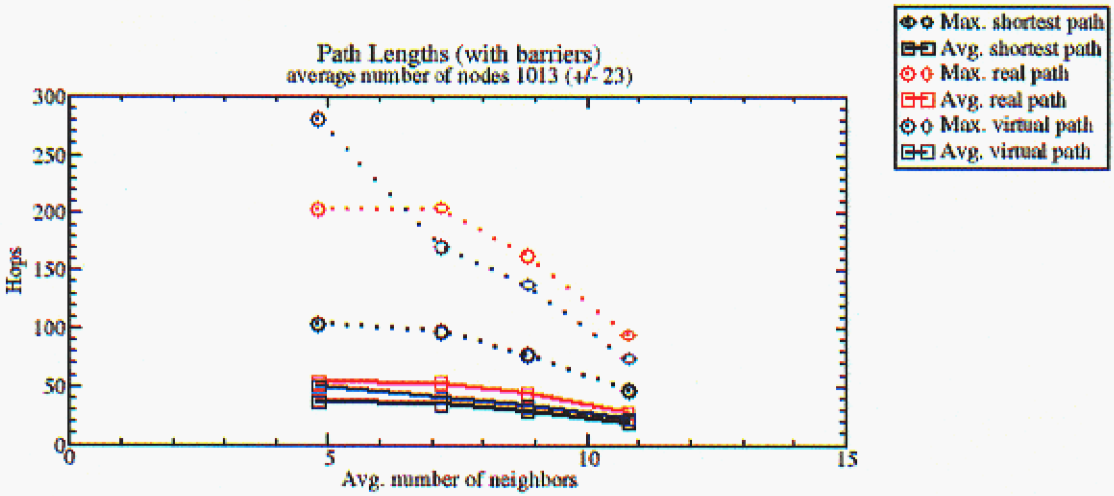

Figure 22. Virtual geographic routing, path length versus average node degree, with barriers. The data points are averages over 10 runs.

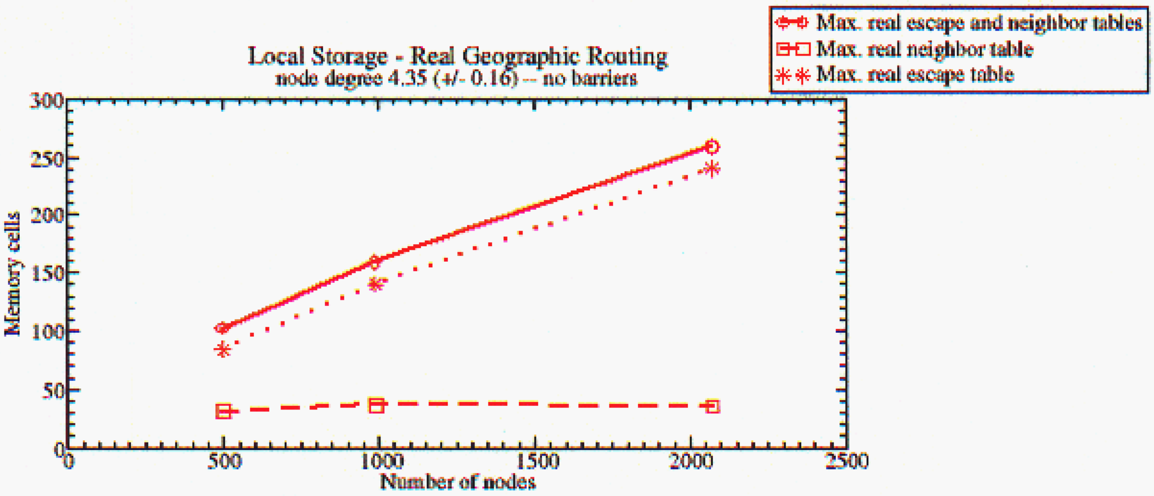

Figure 23. Real geographic routing, degree 4 without barriers. The data points near 500 and 1000 nodes are averages over 10 runs and those near 2000 nodes over 5 runs.

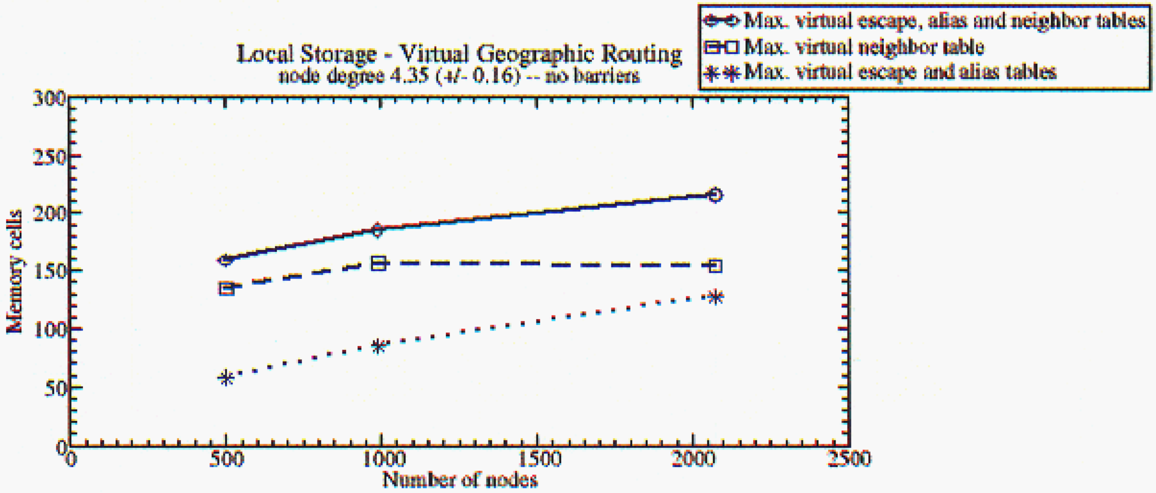

Figure 24. Virtual geographic routing, degree 4 without barriers. The data points near 500 and 1000 nodes are averages over 10 runs and those near 2000 nodes over 5 runs.

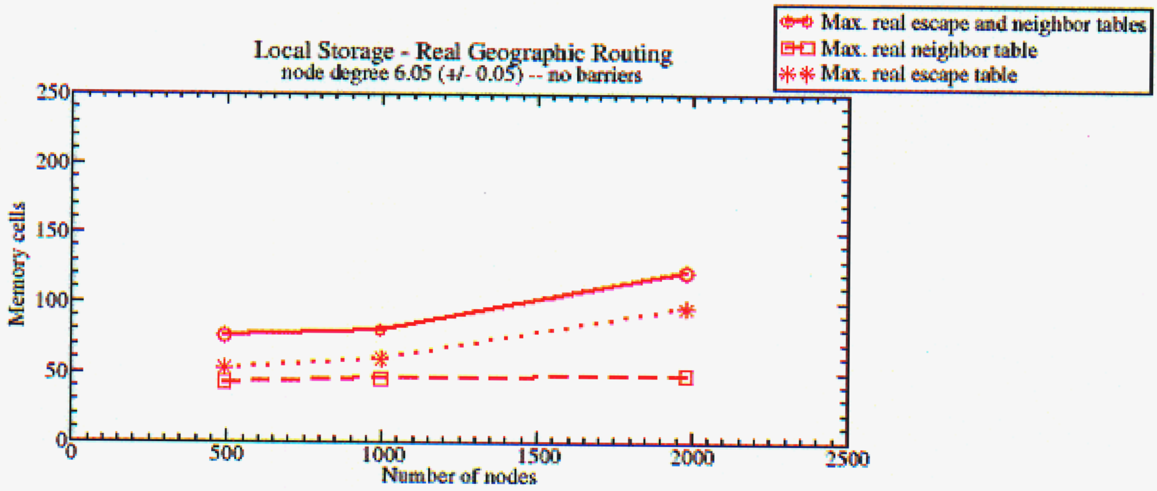

Figure 25. Real geographic routing, degree 6 without barriers. The data points near 500 and 1000 nodes are averages over 10 runs and those near 2000 nodes over 5 runs.

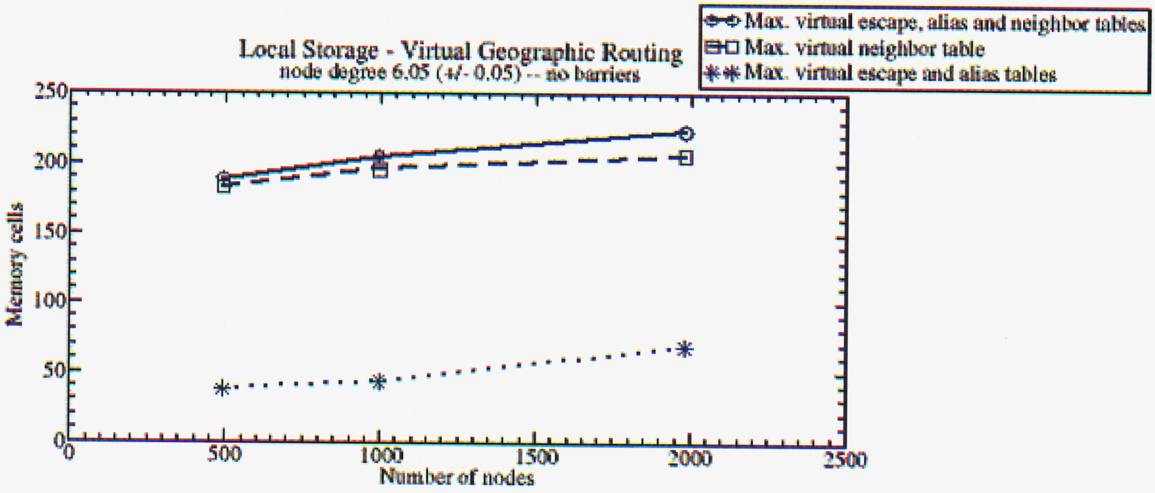

Figure 26. Virtual geographic routing, degree 6 without barriers. The data points near 500 and 1000 nodes are averages over 10 runs and those near 2000 nodes over 5 runs.

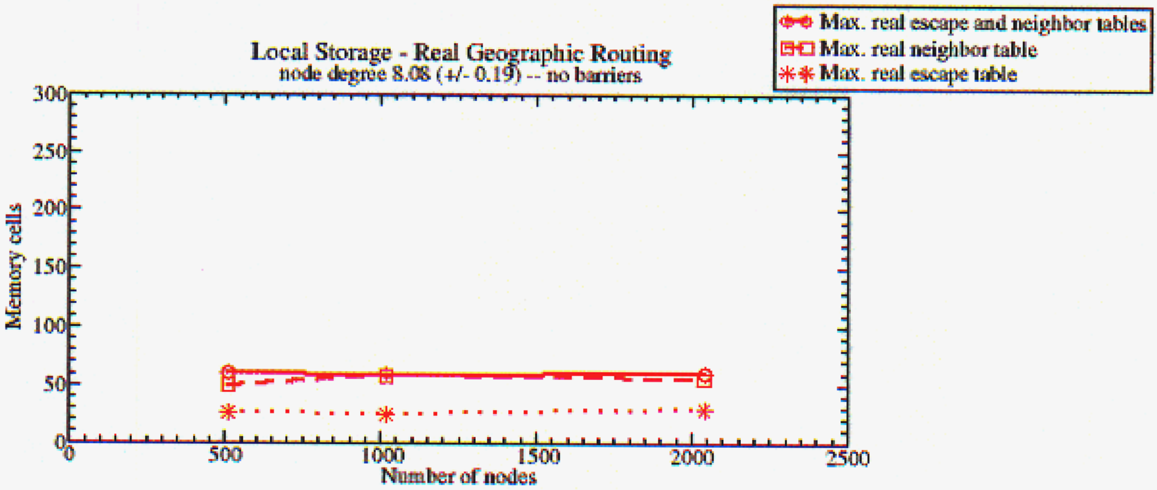

Figure 27. Real geographic routing, degree 8 without barriers. The data points near 500 and 1000 nodes are averages over 10 runs and those near 2000 nodes over 5 runs.

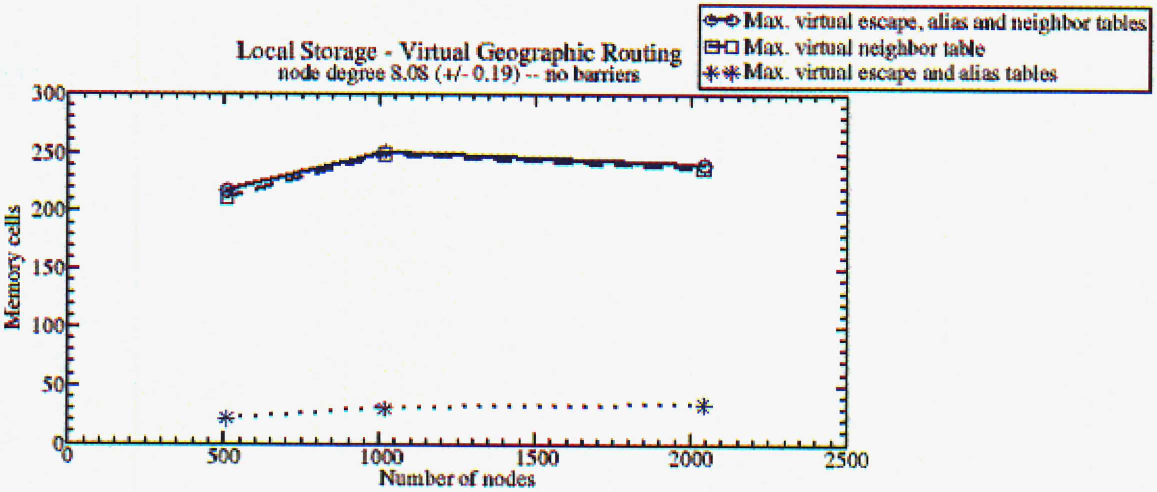

Figure 28. Virtual geographic routing, degree 8 without barriers. The data points near 500 and 1000 nodes are averages over 10 runs and those near 2000 nodes over 5 runs.

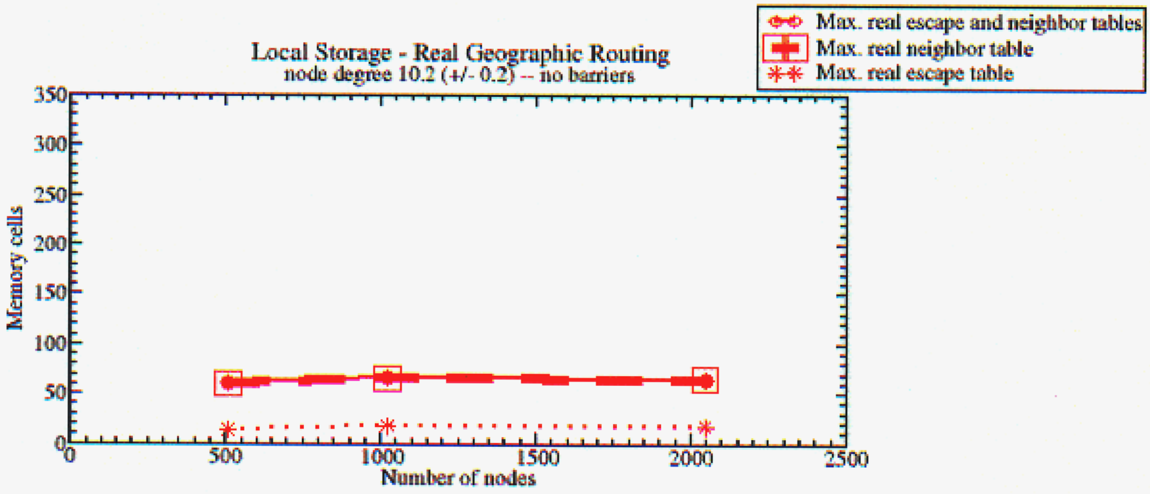

Figure 29. Real geographic routing, degree 10 without barriers. The data points near 500 and 1000 nodes are averages over 10 runs and those near 2000 nodes over 5 runs.

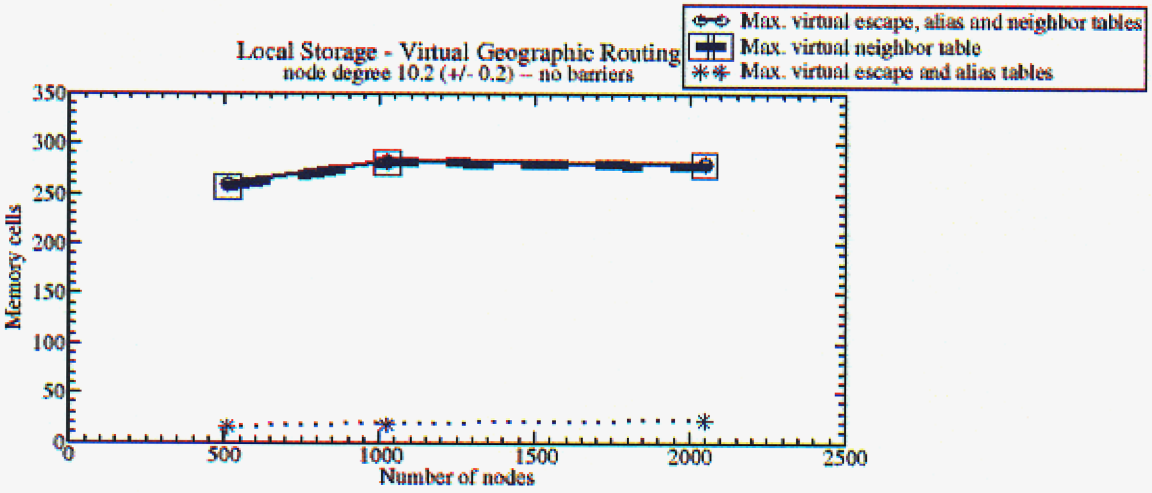

Figure 30. Virtual geographic routing, degree 10 without barriers. The data points near 500 and 1000 nodes are averages over 10 runs and those near 2000 nodes over 5 runs.

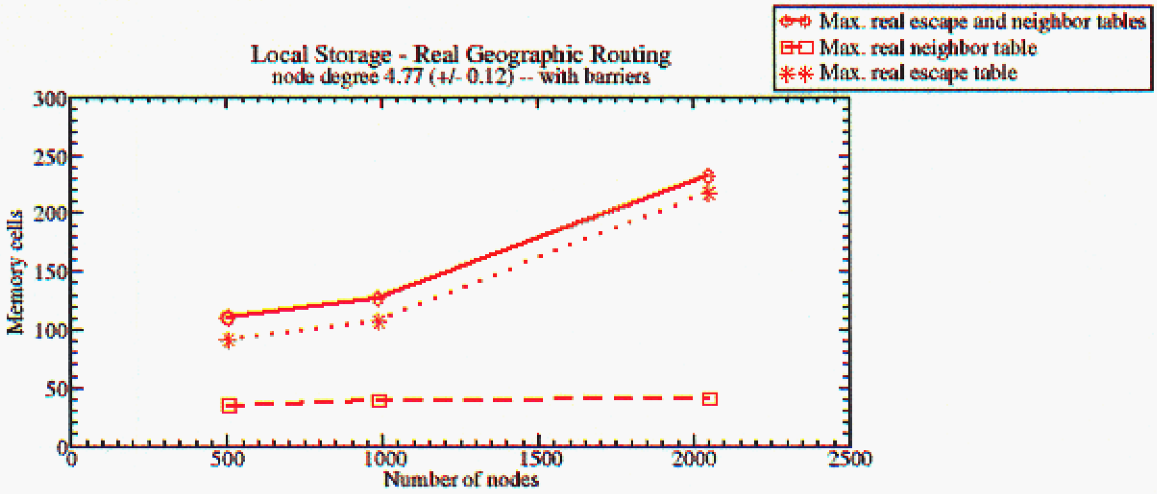

Figure 31. Real geographic routing, degree 5 with barriers. The data points near 500 and 1000 nodes are averages over 10 runs and those near 2000 nodes over 5 runs.

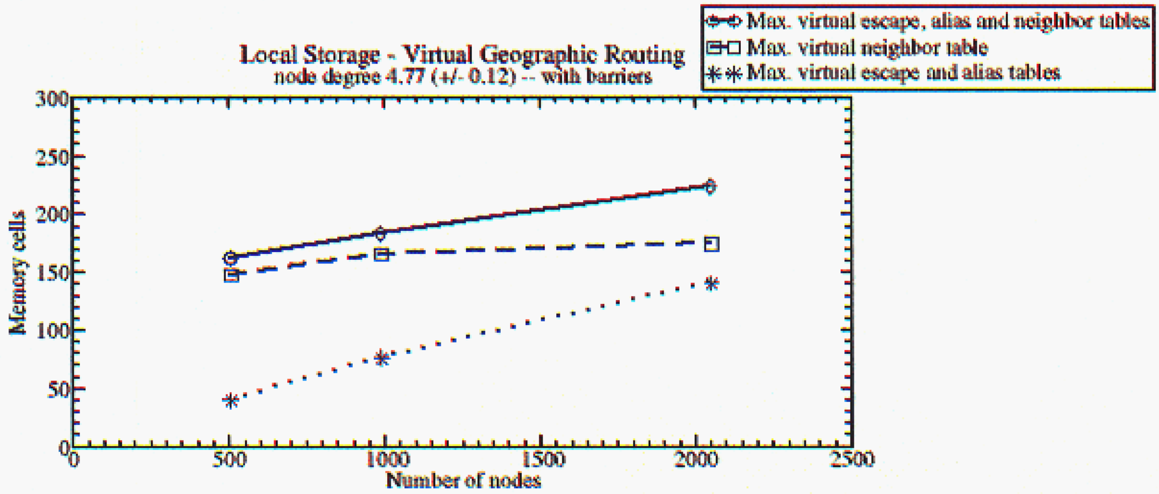

Figure 32. Virtual geographic routing, degree 5 with barriers. The data points near 500 and 1000 nodes are averages over 10 runs and those near 2000 nodes over 5 runs.

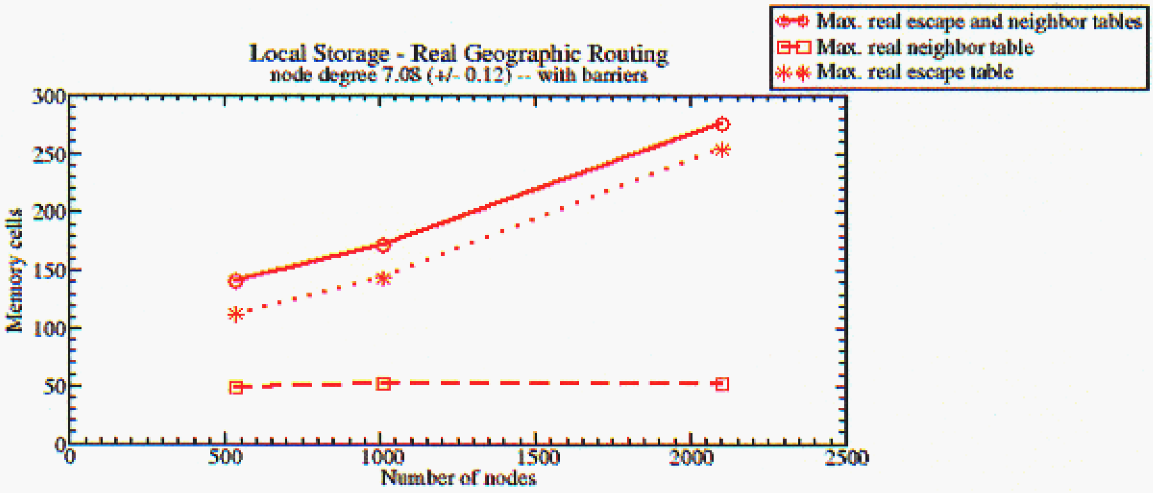

Figure 33. Real geographic routing, degree 7 with barriers. The data points near 500 and 1000 nodes are averages over 10 runs and those near 2000 nodes over 5 runs.

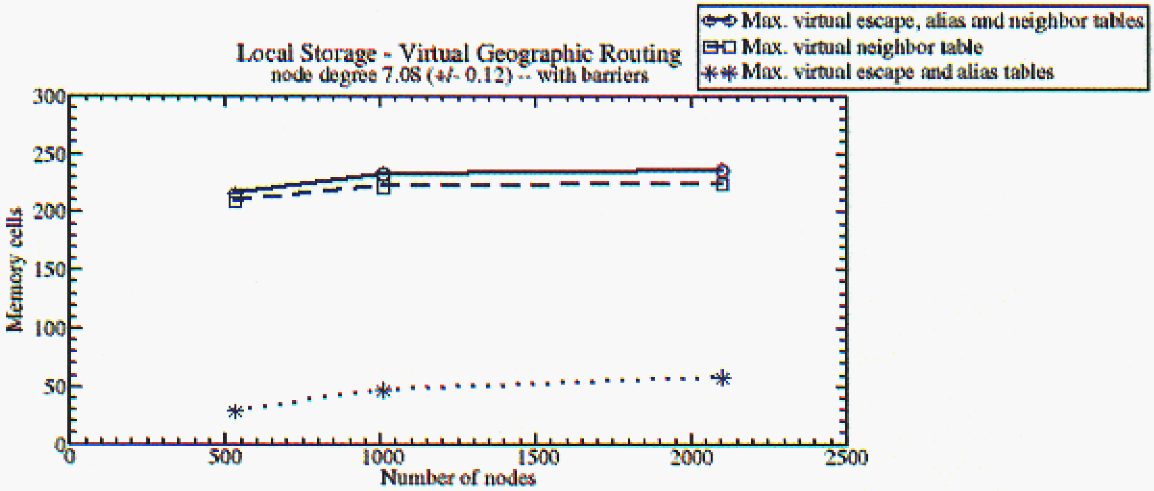

Figure 34. Virtual geographic routing, degree 7 with barriers. The data points near 500 and 1000 nodes are averages over 10 runs and those near 2000 nodes over 5 runs.

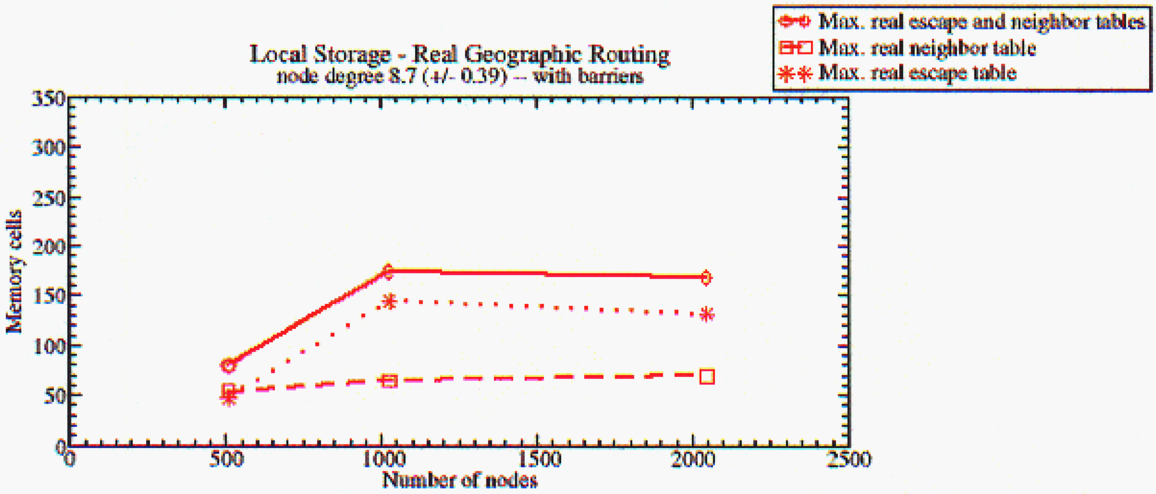

Figure 35. Real geographic routing, degree 9 with barriers. The data points near 500 and 1000 nodes are averages over 10 runs and those near 2000 nodes over 5 runs.

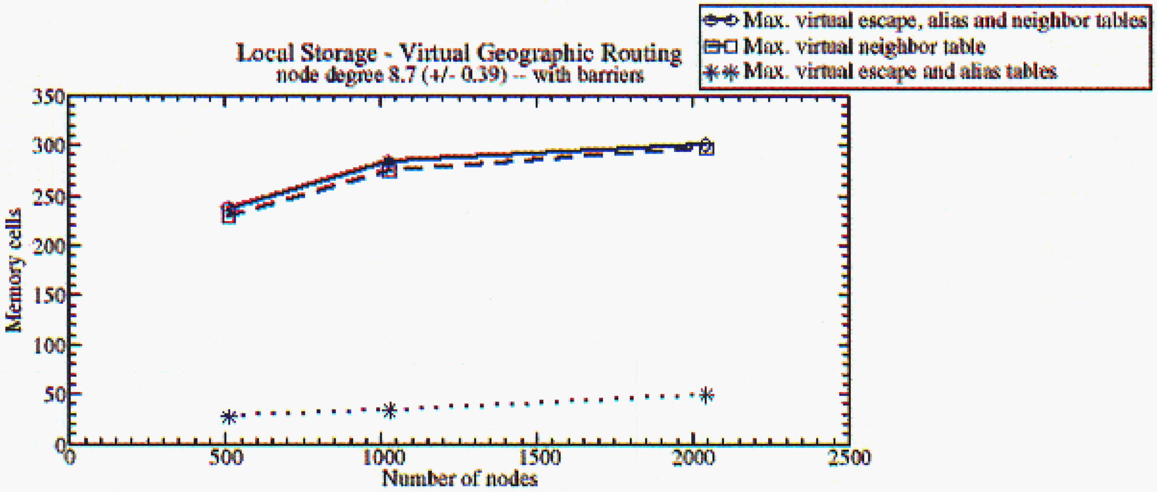

Figure 36. Virtual geographic routing, degree 9 with barriers. The data points near 500 and 1000 nodes are averages over 10 runs and those near 2000 nodes over 5 runs.

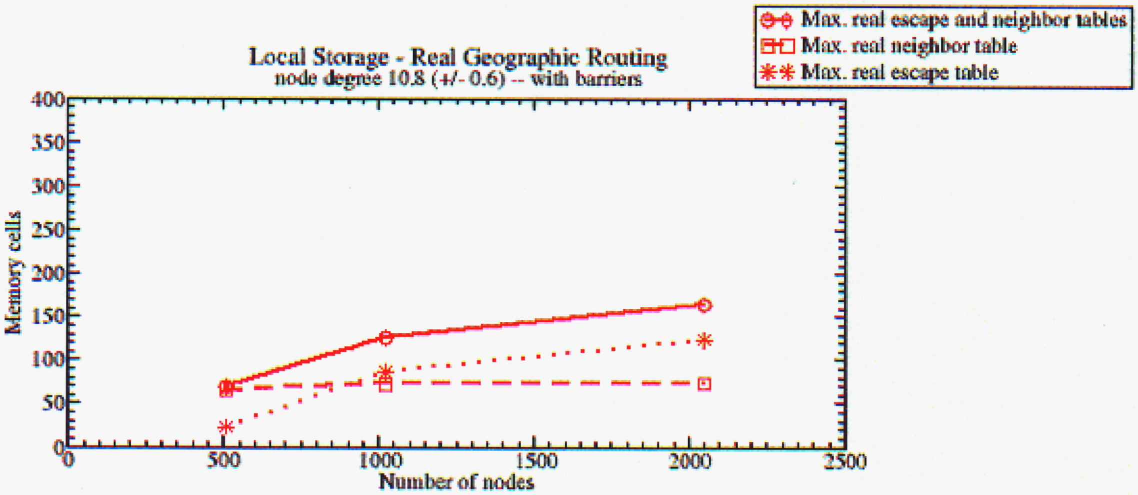

Figure 37. Real geographic routing, degree 11 with barriers. The data points near 500 and 1000 nodes are averages over 10 runs and those near 2000 nodes over 5 runs.

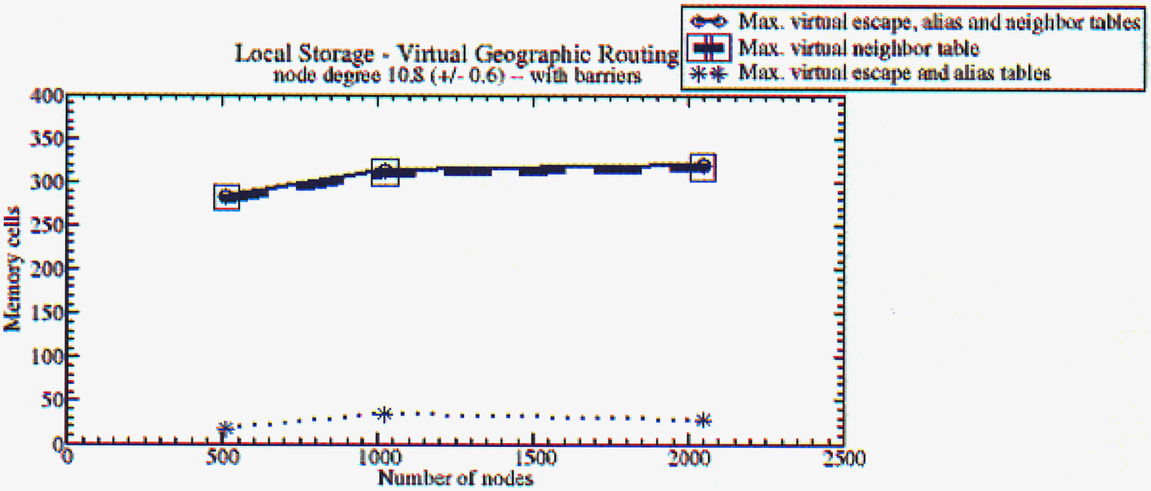

Figure 38. Virtual geographic routing, degree 11 with barriers. The data points near 500 and 1000 nodes are averages over 10 runs and those near 2000 nodes over 5 runs.

Table 4. Order 500 degree 4 without barriers. Averaged over 10 runs with average number of nodes / neighbors = 500.8 / 4.19 and average network diameter / path length = 80.1 / 30.2.

	Real	Virtual	Ratio real / virtual
Max/average path length	140.6 / 41.5	195.2 / 39.7	0.72 / 1.05
Ratio of geographic routing hops to total routing hops	0.65	0.75	0.86
Ratio of cached hops to non-geographic hops [1]	0.989	0.995	0.994
Ratio of search packet transmissions to total hops [2]	0.32	0.047	6.8
Max/average flood radius [3]	44.2 / 3.91	27.9 /2.94	1.58 / 1.33
Max/average steps to alias [4]	------	21.6 / 4.15	------

Table 5. Order 500 degree 6 without barriers. Averaged over 10 runs with average number of nodes / neighbors = 493.1 / 6.00 and average network diameter / path length = 38.3 / 14.4.

	Real	Virtual	Ratio real / virtual
Max/average path length	67.5 / 18.1	90.7 / 18.1	0.74 / 1.0
Ratio of geographic routing hops to total routing hops	0.75	0.80	0.94
Ratio of cached hops to non-geographic hops [1]	0.995	0.994	1.001
Ratio of search packet transmissions to total hops [2]	0.118	0.051	2.3
Max/average flood radius [3]	25.3 / 3.64	16.7 / 2.71	1.51 / 1.34
Max/average steps to alias [4]	------	7.8 / 2.28	------

[1] Escape cache for real routing, escape or alias cache for virtual routing

[2] Request or reply packets, for escape search (real and virtual) or alias search (virtual)

[3] Does not include alias search packets for virtual routing

[4] Average taken only over nodes that have aliases

Table 6. Order 500 degree 8 without barriers. Averaged over 10 runs with average number of nodes / neighbors = 510.0 / 7.89 and average network diameter / path length = 28.1 / 10.9.

	Real	**Virtual**	**Ratio real / virtual**
Max/average path length	44.3 / 12.4	60.3 / 12.6	0.73 / 0.98
Ratio of geographic routing hops to total routing hops	0.86	0.88	0.98
Ratio of cached hops to non-geographic hops [1]	0.997	0.998	0.999
Ratio of search packet transmissions to total hops [2]	0.039	0.013	3.0
Max/average flood radius [3]	14.7 / 3.23	10.5 / 2.80	1.40 / 1.15
Max/average steps to alias [4]	------	2.4 / 1.3	------

Table 7. Order 500 degree 10 without barriers. Averaged over 10 runs with average number of nodes / neighbors = 510.0 / 10.0 and average network diameter / path length = 22.2 / 8.90.

	Real	**Virtual**	**Ratio real / virtual**
Max/average path length	29.3 / 9.57	41.1 / 9.88	0.71 / 0.97
Ratio of geographic routing hops to total routing hops	0.94	0.94	1.00
Ratio of cached hops to non-geographic hops [1]	0.999	0.998	1.001
Ratio of search packet transmissions to total hops [2]	0.0024	0.0064	0.38
Max/average flood radius [3]	7.1 / 2.93	7.6 / 2.40	0.93 / 1.22
Max/average steps to alias [4]	------	0.80 / 0.51	------

[1] Escape cache for real routing, escape or alias cache for virtual routing

[2] Request or reply packets, for escape search (real and virtual) or alias search (virtual)

[3] Does not include alias search packets for virtual routing

[4] Average taken only over nodes that have aliases

Table 8. Order 1000 degree 4 without barriers. Averaged over 10 runs with average number of nodes / neighbors = 989.5 / 4.36 and average network diameter / path length = 130.0 / 47.5.

	Real	Virtual	Ratio real / virtual
Max/average path length	257.7 / 69.0	315.2 / 64.0	0.81 / 1.08
Ratio of geographic routing hops to total routing hops	0.62	0.75	0.82
Ratio of cached hops to non-geographic hops [1]	0.987	0.994	0.99
Ratio of search packet transmissions to total hops [2]	0.439	0.0460	9.5
Max/average flood radius [3]	79.8 / 4.22	40.7 / 3.01	1.96 / 1.4
Max/average steps to alias [4]	------	49.6 / 6.42	------

Table 9. Order 1000 degree 6 without barriers. Averaged over 10 runs with average number of nodes / neighbors = 996.6 / 6.05 and average network diameter / path length = 53.2 / 20.3.

	Real	Virtual	Ratio real / virtual
Max/average path length	96.7 / 26.2	123.9 / 25.3	0.78 / 1.04
Ratio of geographic routing hops to total routing hops	0.74	0.81	0.92
Ratio of cached hops to non-geographic hops [1]	0.992	0.995	0.997
Ratio of search packet transmissions to total hops [2]	0.149	0.0266	5.6
Max/average flood radius [3]	30.0 / 3.44	19.3 / 2.51	1.55 / 1.37
Max/average steps to alias [4]	------	10.9 / 2.58	------

[1] Escape cache for real routing, escape or alias cache for virtual routing
[2] Request or reply packets, for escape search (real and virtual) or alias search (virtual)
[3] Does not include alias search packets for virtual routing
[4] Average taken only over nodes that have aliases

Table 10. Order 1000 degree 8 without barriers. Averaged over 10 runs with average number of nodes / neighbors = 1017.0 / 8.16 and average network diameter / path length = 37.3 / 14.7.

	Real	Virtual	Ratio real / virtual
Max/average path length	54.6 / 16.6	84.0 / 17.4	0.65 / 0.96
Ratio of geographic routing hops to total routing hops	0.886	0.87	1.02
Ratio of cached hops to non-geographic hops [1]	0.998	0.996	1.002
Ratio of search packet transmissions to total hops [2]	0.0135	0.0201	0.67
Max/average flood radius [3]	14.9 / 3.15	13.7 / 2.46	1.09 / 1.28
Max/average steps to alias [4]	------	5.2 / 1.85	------

Table 11. Order 1000 degree 10 without barriers. Averaged over 10 runs with average number of nodes / neighbors = 1023.5 / 10.2 and average network diameter / path length = 31.6 / 12.3.

	Real	Virtual	Ratio real / virtual
Max/average path length	39.2 / 13.2	62.8 / 13.8	0.62 / 0.96
Ratio of geographic routing hops to total routing hops	0.949	0.933	1.02
Ratio of cached hops to non-geographic hops [1]	0.998	0.998	1.00
Ratio of search packet transmissions to total hops [2]	0.0034	0.0030	1.13
Max/average flood radius [3]	8.4 / 2.66	8.5 / 2.28	0.99 / 1.16
Max/average steps to alias [4]	------	2.3 / 0.99	------

[1] Escape cache for real routing, escape or alias cache for virtual routing

[2] Request or reply packets, for escape search (real and virtual) or alias search (virtual)

[3] Does not include alias search packets for virtual routing

[4] Average taken only over nodes that have aliases

Table 12. Order 2000 degree 4 without barriers. Averaged over 5 runs with average number of nodes / neighbors = 2072.8 / 4.49 and average network diameter / path length = 174.6 / 67.6.

	Real	Virtual	Ratio real / virtual
Max/average path length	369.4 / 103.0	548.4 / 105.3	0.67 / 0.98
Ratio of geographic routing hops to total routing hops	0.61	0.69	0.88
Ratio of cached hops to non-geographic hops [1]	0.985	0.992	0.993
Ratio of search packet transmissions to total hops [2]	0.634	0.115	5.5
Max/average flood radius [3]	124.6 / 4.28	73.2 / 3.13	1.70 / 1.37
Max/average steps to alias [4]	------	87.0 / 9.31	------

Table 13. Order 2000 degree 6 without barriers. Averaged over 5 runs with average number of nodes / neighbors = 1981.8 / 6.09 and average network diameter / path length = 79.2 / 29.2.

	Real	Virtual	Ratio real / virtual
Max/average path length	151.9 / 39.2	209.2 / 39.6	0.72 / 0.99
Ratio of geographic routing hops to total routing hops	0.72	0.76	0.96
Ratio of cached hops to non-geographic hops [1]	0.993	0.991	1.002
Ratio of search packet transmissions to total hops [2]	0.197	0.109	1.8
Max/average flood radius [3]	42.8 / 3.49	37.8 / 2.76	1.13 / 1.26
Max/average steps to alias [4]	------	32.0 / 4.56	------

[1] Escape cache for real routing, escape or alias cache for virtual routing

[2] Request or reply packets, for escape search (real and virtual) or alias search (virtual)

[3] Does not include alias search packets for virtual routing

[4] Average taken only over nodes that have aliases

Table 14. Order 2000 degree 8 without barriers. Averaged over 5 runs with average number of nodes / neighbors = 2041.2 / 8.19 and average network diameter / path length = 53.4 / 20.7.

	Real	**Virtual**	**Ratio real / virtual**
Max/average path length	70.8 / 23.7	101 / 24.0	0.70 / 0.99
Ratio of geographic routing hops to total routing hops	0.88	0.90	0.98
Ratio of cached hops to non-geographic hops [1]	0.998	0.998	1.0
Ratio of search packet transmissions to total hops [2]	0.0129	0.00557	2.3
Max/average flood radius [3]	14.8 / 3.04	14.4 / 2.34	1.02 / 1.30
Max/average steps to alias [4]	------	8.4 / 2.25	------

Table 15. Order 2000 degree 10 without barriers. Averaged over 5 runs with average number of nodes / neighbors = 2047.0 / 10.3 and average network diameter / path length = 44.4 / 17.2.

	Real	**Virtual**	**Ratio real / virtual**
Max/average path length	54.4 / 18.6	86.6 / 19.3	0.63 / 0.96
Ratio of geographic routing hops to total routing hops	0.95	0.94	1.01
Ratio of cached hops to non-geographic hops [1]	0.999	0.999	1.0
Ratio of search packet transmissions to total hops [2]	0.00296	0.00114	2.6
Max/average flood radius [3]	9.0 / 2.85	10.2 / 2.22	0.88 / 1.28
Max/average steps to alias [4]	------	1.8 / 0.96	------

[1] Escape cache for real routing, escape or alias cache for virtual routing
[2] Request or reply packets, for escape search (real and virtual) or alias search (virtual)
[3] Does not include alias search packets for virtual routing
[4] Average taken only over nodes that have aliases

Table 16. Order 500 degree 5 with barriers. Averaged over 10 runs with average number of nodes / neighbors = 506.6 / 4.65 and average network diameter / path length = 80.3 / 28.8.

	Real	Virtual	Ratio real / virtual
Max/average path length	141.2 / 40.7	166.7 / 34.7	0.85 / 1.17
Ratio of geographic routing hops to total routing hops	0.64	0.81	0.79
Ratio of cached hops to non-geographic hops [1]	0.990	0.996	0.994
Ratio of search packet transmissions to total hops [2]	0.384	0.031	12.4
Max/average flood radius [3]	46.5 / 4.42	20.2 / 3.11	2.3 / 1.4
Max/average steps to alias [4]	------	12.5 / 2.72	------

Table 17. Order 500 degree 7 with barriers. Averaged over 10 runs with average number of nodes / neighbors = 534.4 / 7.06 and average network diameter / path length = 64.5 / 22.8.

	Real	Virtual	Ratio real / virtual
Max/average path length	123.8 / 32.5	118.8 / 26.0	1.04 / 1.25
Ratio of geographic routing hops to total routing hops	0.62	0.86	0.72
Ratio of cached hops to non-geographic hops [1]	0.992	0.997	0.995
Ratio of search packet transmissions to total hops [2]	0.392	.0.020	20
Max/average flood radius [3]	54.8 / 4.11	14.4 / 3.04	3.80 / 1.35
Max/average steps to alias [4]	------	11.0 / 2.78	------

[1] Escape cache for real routing, escape or alias cache for virtual routing

[2] Request or reply packets, for escape search (real and virtual) or alias search (virtual)

[3] Does not include alias search packets for virtual routing

[4] Average taken only over nodes that have aliases

Table 18. Order 500 degree 8 with barriers. Averaged over 10 runs with average number of nodes / neighbors = 511.1 / 8.15 and average network diameter / path length = 34.7 / 13.7.

	Real	Virtual	Ratio real / virtual
Max/average path length	66.5 / 17.8	65.4 / 15.9	1.02 / 1.12
Ratio of geographic routing hops to total routing hops	0.71	0.86	0.83
Ratio of cached hops to non-geographic hops [1]	0.992	0.994	0.998
Ratio of search packet transmissions to total hops [2]	0.217	0.039	5.6
Max/average flood radius [3]	25.4 / 3.41	12.7 / 2.70	2.00 / 1.26
Max/average steps to alias [4]	------	9.0 / 1.98	------

Table 19. Order 500 degree 10 with barriers. Averaged over 10 runs with average number of nodes / neighbors = 511.7 / 10.3 and average network diameter / path length = 24.3 / 10.2.

	Real	Virtual	Ratio real / virtual
Max/average path length	37.6 / 12.0	40.2 / 11.4	0.94 / 1.06
Ratio of geographic routing hops to total routing hops	0.82	0.92	0.90
Ratio of cached hops to non-geographic hops [1]	0.9968	0.9984	0.998
Ratio of search packet transmissions to total hops [2]	0.099	0.0047	21
Max/average flood radius [3]	11.9 / 3.13	6.4 / 2.37	1.9 / 1.3
Max/average steps to alias [4]	------	3.4 / 1.2	------

[1] Escape cache for real routing, escape or alias cache for virtual routing

[2] Request or reply packets, for escape search (real and virtual) or alias search (virtual)

[3] Does not include alias search packets for virtual routing

[4] Average taken only over nodes that have aliases

Table 20. Order 1000 degree 5 with barriers. Averaged over 10 runs with average number of nodes / neighbors = 990.0 / 4.81 and average network diameter / path length = 103.4 / 36.7.

	Real	Virtual	Ratio real / virtual
Max/average path length	202.7 / 52.3	280.3 / 50.0	0.72 / 1.05
Ratio of geographic routing hops to total routing hops	0.65	0.75	0.87
Ratio of cached hops to non-geographic hops [1]	0.990	0.995	0.995
Ratio of search packet transmissions to total hops [2]	0.301	0.038	7.9
Max/average flood radius [3]	57.0 / 3.72	35.3 / 2.70	1.61 / 1.38
Max/average steps to alias [4]	------	47.4 / 7.18	------

Table 21. Order 1000 degree 7 with barriers. Averaged over 10 runs with average number of nodes / neighbors = 1011.6 / 7.20 and average network diameter / path length = 96.3 / 34.5.

	Real	Virtual	Ratio real / virtual
Max/average path length	202.9 / 52.0	170.1 / 39.8	1.19 / 1.31
Ratio of geographic routing hops to total routing hops	0.60	0.86	0.70
Ratio of cached hops to non-geographic hops [1]	0.993	0.997	0.996
Ratio of search packet transmissions to total hops [2]	0.597	0.040	15
Max/average flood radius [3]	86.2 / 4.39	22.3 / 3.25	3.86 / 1.35
Max/average steps to alias [4]	------	25.5 / 4.32	------

[1] Escape cache for real routing, escape or alias cache for virtual routing

[2] Request or reply packets, for escape search (real and virtual) or alias search (virtual)

[3] Does not include alias search packets for virtual routing

[4] Average taken only over nodes that have aliases

Table 22. Order 1000 degree 9 with barriers. Averaged over 10 runs with average number of nodes / neighbors = 1026.8 / 8.85 and average network diameter / path length = 76.9 / 28.4.

	Real	**Virtual**	**Ratio real / virtual**
Max/average path length	162 / 42.5	136 / 32.8	1.19 / 1.29
Ratio of geographic routing hops to total routing hops	0.61	0.86	0.70
Ratio of cached hops to non-geographic hops [1]	0.994	0.997	0.997
Ratio of search packet transmissions to total hops [2]	0.570	0.015	38
Max/average flood radius [3]	71.6 / 4.66	18.3 / 2.69	3.91 / 1.73
Max/average steps to alias [4]	------	14.9 / 2.76	------

Table 23. Order 1000 degree 11 with barriers. Averaged over 10 runs with average number of nodes / neighbors = 1023.8 / 10.8 and average network diameter / path length = 46.3 / 19.3.

	Real	**Virtual**	**Ratio real / virtual**
Max/average path length	92.7 / 27.2	73.2 / 22.8	1.27 / 1.25
Ratio of geographic routing hops to total routing hops	0.66	0.90	0.73
Ratio of cached hops to non-geographic hops [1]	0.9934	0.9947	0.999
Ratio of search packet transmissions to total hops [2]	0.778	0.046	17
Max/average flood radius [3]	40.5 / 4.7	14.3 / 2.76	2.83 / 1.70
Max/average steps to alias [4]	------	16.6 / 3.56	------

[1] Escape cache for real routing, escape or alias cache for virtual routing
[2] Request or reply packets, for escape search (real and virtual) or alias search (virtual)
[3] Does not include alias search packets for virtual routing
[4] Average taken only over nodes that have aliases

Table 24. Order 2000 degree 5 with barriers. Averaged over 5 runs with average number of nodes / neighbors = 2051.6 / 4.86 and average network diameter / path length = 182.8 / 60.2.

	Real	Virtual	Ratio real / virtual
Max/average path length	338 / 91.2	511 / 90.1	0.66 / 1.01
Ratio of geographic routing hops to total routing hops	0.62	0.71	0.88
Ratio of cached hops to non-geographic hops[1]	0.988	0.991	0.997
Ratio of search packet transmissions to total hops[2]	0.878	0.102	8.6
Max/average flood radius[3]	94.8 / 4.67	62.0 / 2.93	1 53 / 1.60
Max/average steps to alias[4]	------	104 / 9.43	------

Table 25. Order 2000 degree 7 with barriers. Averaged over 5 runs with average number of nodes / neighbors = 2102.2 / 6.99 and average network diameter / path length = 131.2 / 50.0.

	Real	Virtual	Ratio real / virtual
Max/average path length	291 / 77.6	279 / 59.4	1.04 / 1.31
Ratio of geographic routing hops to total routing hops	0.60	0.84	0.71
Ratio of cached hops to non-geographic hops[1]	0.9925	0.9953	0.997
Ratio of search packet transmissions to total hops[2]	0.774	0.039	20
Max/average flood radius[3]	120 / 4.91	31.6 / 2.93	3.79 / 1.67
Max/average steps to alias[4]	------	41.8 / 5.28	------

[1] Escape cache for real routing, escape or alias cache for virtual routing
[2] Request or reply packets, for escape search (real and virtual) or alias search (virtual)
[3] Does not include alias search packets for virtual routing
[4] Average taken only over nodes that have aliases

Table 26. Order 2000 degree 9 with barriers. Averaged over 5 runs with average number of nodes / neighbors = 2045.2 / 9.09 and average network diameter / path length = 80.8 / 33.8.

	Real	Virtual	Ratio real / virtual
Max/average path length	177 / 51.9	144 / 38.9	1.23 / 1.33
Ratio of geographic routing hops to total routing hops	0.63	0.88	0.71
Ratio of cached hops to non-geographic hops [1]	0.993	0.994	0.999
Ratio of search packet transmissions to total hops [2]	0.554	0.037	15
Max/average flood radius [3]	69.8 / 4.25	22.6 / 2.65	3.09 / 1.60
Max/average steps to alias [4]	------	28.0 / 3.23	------

Table 27. Order 2000 degree 11 with barriers. Averaged over 5 runs with average number of nodes / neighbors = 2047.6 / 11.4 and average network diameter / path length = 182.8 / 60.2.

	Real	Virtual	Ratio real / virtual
Max/average path length	140 / 43.6	110 / 32.2	1.27 / 1.35
Ratio of geographic routing hops to total routing hops	0.63	0.91	0.70
Ratio of cached hops to non-geographic hops [1]	0.995	0.997	0.998
Ratio of search packet transmissions to total hops [2]	0.5738	0.01822	31
Max/average flood radius [3]	59.8 / 4.58	16.6 / 2.65	3.60 / 1.73
Max/average steps to alias [4]	------	15.4 2.67	------

[1] Escape cache for real routing, escape or alias cache for virtual routing
[2] Request or reply packets, for escape search (real and virtual) or alias search (virtual)
[3] Does not include alias search packets for virtual routing
[4] Average taken only over nodes that have aliases

Distribution

1		University of Chicago at Urbana-Champaign Attn: D. M. Nicol 457 Coordinated Science Laboratory 1308 West Main Street Urbana, IL 61801
1	MS 0785	D. Kilman, 5516
1	MS 0785	B. P. Van Leeuwen, 5516
1	MS 0785	R. L. Hutchinson, 5516
1	MS 9003	J. L. Handrock, 8960
1	MS 9003	K. E. Washington, 8900
1	MS 9007	M. F. Hardwick, 8964
3	MS 9201	H. R. Ammerlahn, 8112
1	MS 9201	L. D. Brandt, 8112
1	MS 9201	P. K. Falcone, 8110
3	MS 9201	M. E. Goldsby, 8114
1	MS 9201	M. M. Johnson, 8114
1	MS 9201	R. P. Tsang, 8114
1	MS 9915	N. R. Bierbaum, 8961
1	MS 9915	H. Y. Chen, 8961
1	MS 9915	M. L. Koszykowski, 8961
1	MS 0323	D. Chavez, LDRD Office, 1011
1	MS 0899	Technical Library, 9616
3	MS 9018	Central Technical Files, 8945-1
1	MS 9021	Classification Office, 8511 for Technical Library, MS 0899, 9616
		DOE/OSTI via URL