Dynamic Load-Aware Routing in Ad hoc Networks

Sung-Ju Lee
Internet & Mobile Systems Labs
Hewlett-Packard Laboratories
Palo Alto, CA 94304-1126
sjlee@hpl.hp.com

Mario Gerla
Computer Science Department
University of California
Los Angeles, CA 90095-1596
gerla@cs.ucla.edu

Abstract* – Ad hoc networks are deployed in situations where no base station is available and a network has to be built impromptu. Since there is no wired backbone, each host is a router and a packet forwarder. Each node may be mobile, and topology changes frequently and unpredictably. Routing protocol development has received much attention because mobility management and efficient bandwidth and power usage are critical in ad hoc networks. No existing protocol however, considers the load as the main route selection criteria. This routing philosophy can lead to network congestion and create bottlenecks. We present Dynamic Load-Aware Routing (DLAR) protocol that considers intermediate node routing loads as the primary route selection metric. The protocol also monitors the congestion status of active routes and reconstructs the path when nodes of the route have their interface queue overloaded. We describe three DLAR algorithms and show their effectiveness by presenting and comparing simulation results with an ad hoc routing protocol that uses the shortest paths.

I. INTRODUCTION

A wireless ad hoc network [6], [9] is composed of mobile hosts that communicate each other with packet radios over a shared wireless medium. Because of limited radio propagation range, routes are mostly multihop. Without any wired infrastructure, ad hoc networks are deployed in applications such as search and rescue, automated battlefields, disaster recovery, crowd control, and sensor networks.

Routing protocols in ad hoc networks [14] must manage frequent topology changes caused by node mobility and need to be bandwidth and power efficient. A new technique designed for ad hoc networks is "on-demand," or reactive routing. Routing tables with full topological views are not maintained and only routes to nodes that a source needs to communicate with are established on demand via source flooding. Existing on-demand routing protocols such as DSR (Dynamic Source Routing) [8], AODV (Ad-hoc On-demand Distance Vector) [12] , and TORA (Temporally-Ordered Routing Algorithm) [11] use the shortest path as their routing criteria. This route selection philosophy can lead to network congestion and long delays (because of congestion). Moreover, most on-demand protocols use caching mechanisms for intermediate nodes to "reply from cache," causing routing load to concentrate on certain nodes. Recent simulation studies have shown that on-demand protocols that use shortest paths suffer from performance degradation as the network traffic increases [4], [7].

In this paper, we present Dynamic Load-Aware Routing (DLAR) protocol. DLAR considers the load of intermediate nodes as the main route selection metric and monitors the congestion status of active routes to reconstruct the path when nodes of the route have their interface queue overloaded.

Routing with load balancing in wired networks has been exploited in various approaches [2], [10], [15], [16]. In ad hoc networks, only Associativity-Based Routing (ABR) [17] considers the load as the metric. ABR, however, uses the routing load as the secondary metric. Furthermore, the load is measured in the number of routes a node is a part of, and hence the protocol does not account for various traffic loads of each data session. DLAR, on the other hand, uses the number of packets buffered in the interface as the primary route selection criteria. Using the least-loaded routes will help distribute and balance the traffic load to the network hosts.

The paper is organized as follows. Section II describes DLAR protocol and its three route selection algorithms. Section III presents simulation results and analysis. We conclude the paper in Section IV.

II. DYNAMIC LOAD-AWARE ROUTING

A. *Overview*

DLAR builds routes on-demand. When a route is required but no information to the destination is known, the source floods the ROUTE REQUEST packet to discover a route. When nodes other than the destination receive a non-duplicate ROUTE REQUEST, they build a route entry for the <source, destination> pair and record the previous hop to that entry (thus, backward learning). This previous node information is needed later to relay the ROUTE REPLY packet back to the source of the route.[1] Nodes then attach their load information (the number of packets buffered in their interface) and broadcast the ROUTE REQUEST packet. After receiving the first ROUTE REQUEST packet, the destination waits for an appropriate amount of time to learn all possible routes. In order

---

[1] If a ROUTE REPLY packet is not received, the entry will timeout and be removed from the route table.

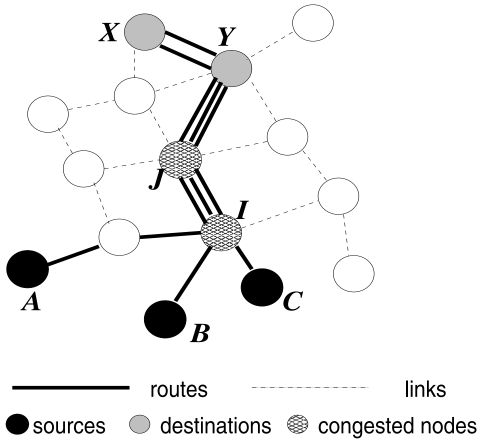

Fig. 1. Congested network.

to learn all the routes and their quality, the destination node accepts duplicate ROUTE REQUESTS received from different previous nodes. The destination then chooses the least loaded route and sends a ROUTE REPLY packet back to the source via the selected route. We propose three different algorithms in determining the best route and they are explained in Section II-B.

In our protocol, intermediate nodes cannot send a ROUTE REPLY back to the source even when they have route information to the destination. To utilize the most up-to-date load information when selecting routes and to minimize the overlapped routes which cause congested bottlenecks, DLAR prohibits intermediate nodes from replying to ROUTE REQUESTS.[2] Figure 1 illustrates a network with congested nodes due to routes built on replies from intermediate nodes. Consider that the route initially acquired from node B to node X is <B-I-J-Y-X>. Later on, node C needs to build a route to node X and sends a ROUTE REQUEST. In protocols such as AODV and DSR, intermediate node I sends a ROUTE REPLY to node C since it has a route to node X. Node C uses this information and builds an overlapped route <C-I-J-Y-X>. The same process occurs when node A constructs a route to node Y. Figure 1 shows the end result where nodes I and J are congested. Intermediate nodes replying to ROUTE REQUESTS has an advantage of reducing the propagation of flooded packets, but causes congestion and a reply storm (i.e., too many nodes send ROUTE REPLIES at the same time resulting in collisions).

During the active data session, intermediate nodes periodically piggyback their load information on data packets. Destination node can thus monitor the load status of the route. If the route is congested, a new and lightly loaded route is selected to replace the overloaded path. Routes are hence reconstructed dynamically in advance of congestion. The process of building new routes is similar to the initial route discovery process except that the destination floods the packet to the source of the route, instead of the source flooding to the destination. The source, upon receiving ROUTE REQUEST packets, selects the best route in the same manner as the destination. The source does not need to send a ROUTE REPLY, and simply sends the next data packet using the newly discovered route.

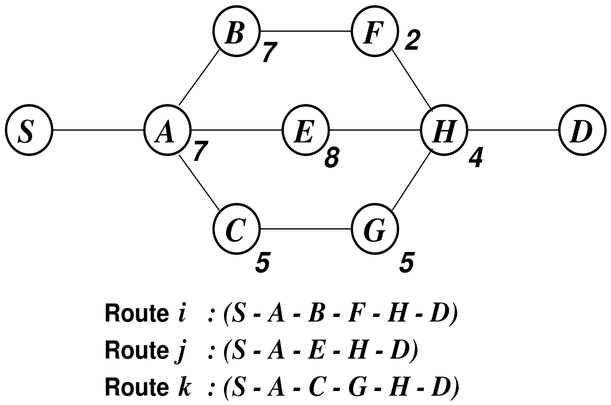

Fig. 2. Example network.

A node can detect a link break by receiving a link layer feedback signal from the MAC protocol,[3] not receiving passive acknowledgments,[4] or not receiving hello packets for a certain period of time. When a route is disconnected, the immediate upstream node of the broken link sends a ROUTE ERROR message to the source of the route to notify the route invalidation. Nodes along the path to the source remove the route entry upon receiving this message and relay it to the source. The source reconstructs a route by flooding a ROUTE REQUEST when informed of a route disconnection.

B. Route Selection Algorithms

We introduce three algorithms in selecting the least loaded route. We use Figure 2 as an example network to describe each scheme.

DLAR *scheme 1* simply adds the routing load of each intermediate node and selects the route with the least sum. If there is a tie, the destination selects the route with the shortest hop distance. When there are still multiple routes that have the least load and hop distance, the path that is taken by the packet which arrived at the destination the earliest between them is chosen. In the example network, route i has the sum of 20 (i.e., 7 + 7 + 2 + 4 = 20), route j has the sum of 19 (i.e., 7 + 8 + 4 = 19), and route k has the sum of 21 (i.e., 7 + 5 + 5 + 4 = 21). Therefore, route j is selected and used as the route.

---

[2] Intermediate nodes can *relay* ROUTE REPLIES from the destination to the source, of course.

[3] MAC protocols such as MACAW [3] and IEEE 802.11 [5] have this capability.

[4] This technique was introduced by Jubin and Tornow in their early work on packet radio networks [9].

TABLE I

ROUTE QUALITIES BASED ON EACH SCHEME.

	Scheme 1	*Scheme 2*	*Scheme 3*
Route i	20	5	2 (A and B)
Route j	19	6.67	2 (A and E)
Route k	21	5.25	1 (A)
Selection	Route j	Route i	Route k

DLAR *scheme 2* is similar to *scheme 1*. However, instead of using the *sum* of number of packets queued at each intermediate node's interface as in *scheme 1*, *scheme 2* uses the *average* number of packets buffered at each intermediate node along the path. We can use the shortest delay as a tie breaker if needed. Considering the example in Figure 2 again, route i has the average value of 5 (i.e., 20 / 4 = 5), route j has the value of 6.67 (i.e., 19 / 3 = 6.67), and route k has the value of 5.25 (i.e., 21 / 4 = 5.25). Route i is thus selected.

DLAR *scheme 3* considers the number of congested intermediate nodes as the route selection metric. Basically, it chooses the route with the least number of intermediate nodes that have their load exceeding the threshold value τ. In our example, if τ is five, route i has two intermediate nodes (i.e., nodes A and B) that have the number of queued packets over the threshold, route j has two (i.e., nodes A and E), and route k has one (i.e., node A). Hence, route k is selected using this algorithm. This scheme applies the same tie breaking rule as in *scheme 1*.

Table II-B summarizes the route qualities in Figure 2 by applying each algorithm.

III. PERFORMANCE EVALUATION

A. Simulation Model

We evaluate three DLAR schemes by comparing the performance with DSR [8], which uses the shortest path. We implemented the simulator within the Global Mobile Simulation (GloMoSim) library [18]. The GloMoSim library is a scalable simulation environment for wireless network systems using the parallel discrete-event simulation capability provided by PARSEC [1].

Our simulation modeled a network of 50 mobile hosts placed randomly within a 1000 meter × 1000 meter area. Each node has a radio propagation range of 250 meters and channel capacity was 2 Mb/s. Each run executed for 300 seconds of simulation time.

A free space propagation model with a threshold cutoff [13] was used in our experiments. In the radio model, we assumed the ability of a radio to lock onto a sufficiently strong signal in the presence of interfering signals, i.e., radio capture. We used the IEEE 802.11 Distributed Coordination Function (DCF) [5] as the medium access control protocol. A traffic generator was developed to simulate constant bit rate sources. The sources and the destinations are randomly selected with uniform probabilities. The size of data payload was 512 bytes. We used random waypoint model [8] as the mobility model. We generated various mobility degree by using different pause times. The minimum and the maximum speeds were set to zero and 10 m/s, respectively.

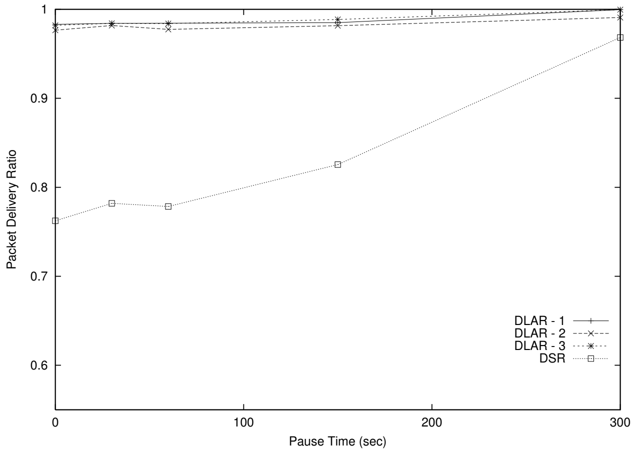

Fig. 3. Packet delivery ratio (20 sources sending 4 pkt/sec).

B. Simulation Results

B.1 Throughput

Figure 3 shows the throughput in packet delivery ratio of each protocol when 20 sources each send 4 data packets per second. Three DLAR schemes perform very well regardless of the mobility degree and outperform DSR. We can observe the performance degradation of DSR when mobility increases (i.e., pause time decreases). In high mobility scenarios, many route reconstruction processes are invoked. When a source floods a new ROUTE REQUEST packet to recover the broken route, many intermediate nodes send ROUTE REPLIES back to the source because they have cached a number of routes by overhearing packets during the initial route construction phase. A good portion of these cached routes overlap existing routes. Nodes that are part of multiple routes become congested and cannot deliver packets along the route. Moreover, DSR does not apply any aging mechanism to cached routes. Intermediate nodes may therefore have stale routes stored in their cache and reply to sources with invalidated routes. Sources propagate data packets to a newly acquired but stale route and more route reconstruction procedures need to be invoked until a fresh and valid route is found. Many data packets are dropped during this process, resulting in poor DSR performance.

We varied the traffic load to investigate its impact on the routing performance. Figure 4 shows the delivery ratio when

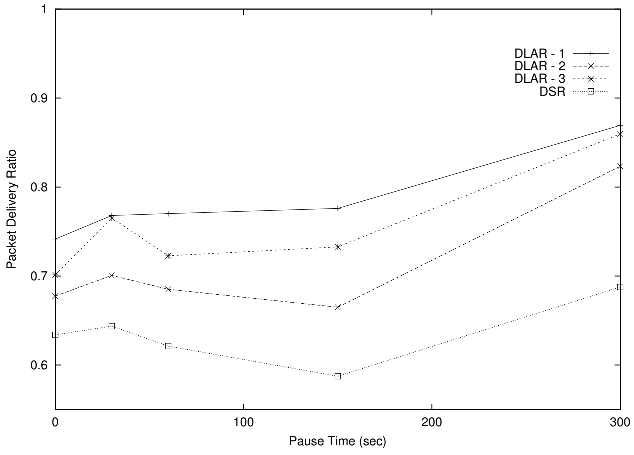

Fig. 4. Packet delivery ratio (20 sources sending 8 pkt/sec).

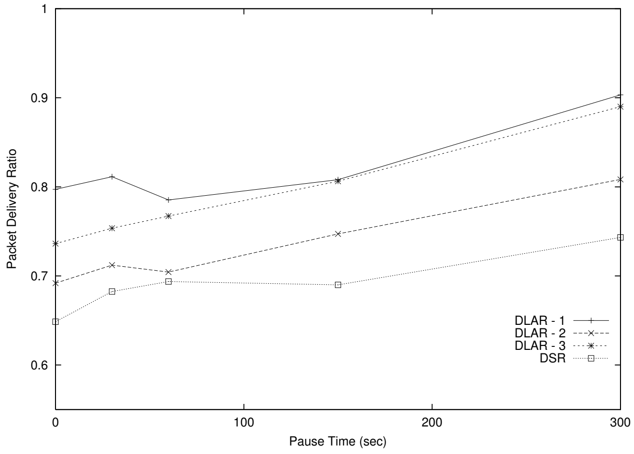

Fig. 5. Packet delivery ratio (40 sources sending 4 pkt/sec).

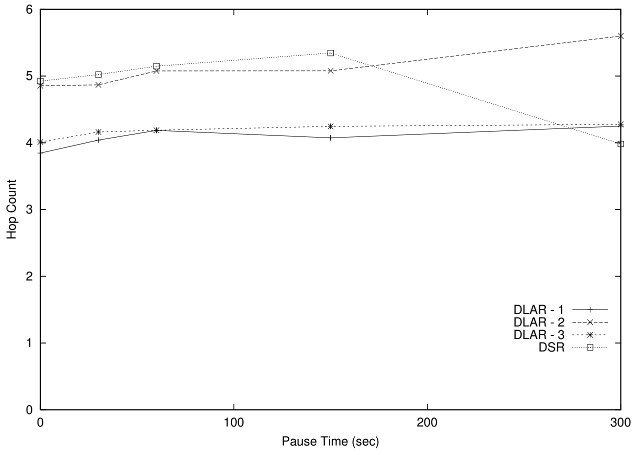

Fig. 6. Hop distance.

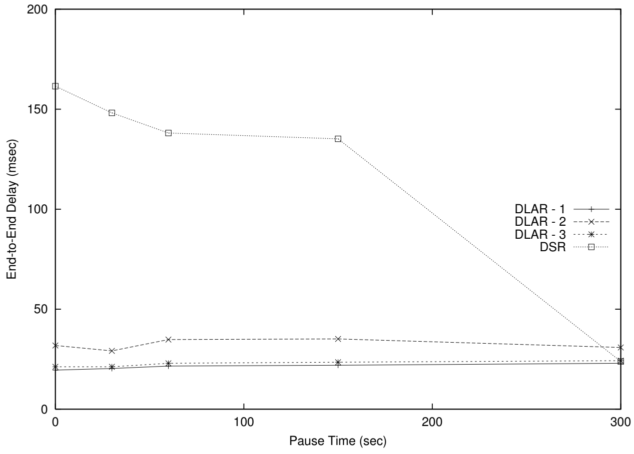

Fig. 7. End-to-end delay.

the traffic load for each source is doubled to 8 packets per second and the number of sources is the same (20), and Figure 5 shows the performance when the number of source is doubled to 40 and the traffic rate for each source is the same (4 packets per second). In both cases, all DLAR schemes perform better than DSR. *Scheme 1* gives the best result and outperforms DSR by 10% to 15%. Between DLAR algorithms, *scheme 2* delivers the least fraction of data packets. Since *scheme 2* considers the average number of load, it does not take hop distance into consideration when selecting routes. Longer paths have a more chance of having route breaks since one link disconnection results in a route invalidation.

Figure 6 reports the average hop distance of each protocol. We can see that *scheme 2* has the longest hop length among DLAR protocols. It is interesting to see the hop counts of DSR. DSR has the shortest hop distance when there is no mobility (the pause time is 300 seconds), but with mobility, the hop distance grows and becomes larger than those of DLAR schemes. If the route is established directly from the destination, it can be shorter in distance since it is built based on the most recent information and accounts for node locations after movements. DSR, however, uses cached routes from intermediate nodes and those routes are not fresh and do not exploit the current network topology.

B.2 End-to-End Delay

Figure 7 presents the end-to-end delay of four protocols. As expected, DSR has the longest delay. In DSR, many parts of the network is congested and data packets traversing through those bottlenecks are buffered at interfaces for a long duration of time. *Scheme 2* has the longest delay among DLAR algorithms because it has the longest hop distance, as shown in Figure 6.

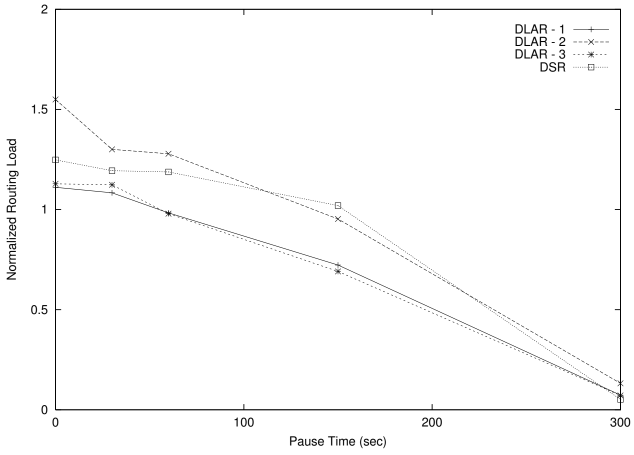

Fig. 8. Normalized routing load.

B.3 Routing Overhead

Figure 8 shows the routing overhead in normalized routing load. Normalized routing load is the ratio of the number of control packets propagated by every node in the network and the number of data packets received by the destination nodes. All protocols give similar results. Compared with DLAR schemes, DSR has fewer number of ROUTE REQUEST propagations during the initial route construction phase since intermediate nodes that have route information to the destination do not broadcast the packet. However, there are more number of ROUTE REPLY transmissions because many intermediate nodes send back ROUTE REPLIES. In addition, route breaks occur more frequently in DSR because it often uses stale routes. Hence, more ROUTE ERROR packets are transmitted, and consequently, more ROUTE REQUESTS are sent to reconstruct routes. These factors accumulate and make DSR's normalized routing load in the same vicinity of those of DLAR protocols.

IV. CONCLUSION

We presented Dynamic Load-Aware Routing (DLAR) protocol that uses the routing load of the intermediate nodes as the main route selection criteria. In the route construction phase, each intermediate node records in the control packet the number of packets queued at the interface and the destination uses that information when selecting the route. Three different route selection algorithms were described. *Scheme 1* uses the total number of packets buffered at the intermediate nodes and *scheme 2* uses the average number of queued packets at each node. *Scheme 3* defines a load threshold and selects the route that has the least number of intermediate nodes that have packets buffered more than the threshold value. To avoid producing bottlenecks and to use the most up-to-date route information when discovering routes, DLAR does not allow intermediate nodes to reply from cache . DLAR periodically monitors the congestion status of active data sessions and dynamically reconfigures the routes that are being congested. Using the least-loaded routes helps balance the load of the network nodes and utilize the network resources efficiently.

Simulation results showed that DLAR schemes outperform DSR which uses the shortest path and does not consider the routing load. DLAR protocols delivered more fraction of data packets, yielded shorter end-to-end delays, and generated nearly equal number of control packets as DSR.

References

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

[2] A. Bestavros and I. Matta, "Load Profiling for Efficient Route Selection in Multi-Class Networks," *Proceedings of IEEE ICNP'97*, Atlanta, GA, October 1997, pp. 183-190.

[3] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LANs," *Proceedings of ACM SIGCOMM'94*, London, UK, September 1994, pp. 212-225.

[4] S.R. Das, C.E. Perkins, and E.M. Royer, "Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks," *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, Tel Aviv, Israel, March 2000, pp. 3-12.

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

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

[7] P. Johanson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, "Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks," *Proceedings of ACM/IEEE MOBICOM'99*, Seattle, WA, August 1999, pp. 195-206.

[8] D.B. Johnson and D.A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," In *Mobile Computing*, edited by Tomasz Imielinski and Hank Korth, Chapter 5, Kluwer Academic Publishers, 1996, pp. 153-181.

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

[10] I. Matta and M. Krunz, "Packing and Least-Loaded Based Routing in Multi-Rate Loss Networks," *Proceedings of IEEE ICC'97*, Montreal, Canada, June 1997, pp. 827-831.

[11] V.D. Park and M.S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," *Proceedings of IEEE INFOCOM'97*, Kobe, Japan, April 1997, pp. 1405-1413.

[12] C.E. Perkins and E.M. Royer, "Ad-Hoc On Demand Distance Vector Routing," *Proceedings of IEEE WMCSA'99*, New Orleans, LA, February 1999, pp. 90-100.

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

[14] E.M. Royer and C.-K. Toh, "A Review of Current Routing Protocols for Ad-Hoc Mobile Networks," *IEEE Personal Communications*, vol. 6, no. 2, April 1999, pp. 46-55.

[15] A. Shaikh, J. Rexford, and K.G. Shin, "Load-Sensitive Routing of Long-Lived IP Flows," *Proceedings of ACM SIGCOMM'99*, Cambridge, MA, September 1999, pp. 215-226.

[16] H. Tode, Y. Sakai, M. Yamamoto, H. Okada, and Y. Tezuka, "Multicast Routing Algorithms for Nodal Load Balancing," *Proceedings of IEEE INFOCOM'92*, Florence, Italy, May 1992, pp. 2086-2095.

[17] C.-K. Toh, "Associativity-Based Routing for Ad-Hoc Mobile Networks," *Wireless Personal Communications Journal*, vol. 4, no. 2, March 1997, pp. 103-139.

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