Split Multipath Routing with Maximally Disjoint Paths 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* – In recent years, routing has been the most focused area in ad hoc networks research. On-demand routing in particular, is widely developed in bandwidth constrained mobile wireless ad hoc networks because of its effectiveness and efficiency. Most proposed on-demand routing protocols however, build and rely on single route for each data session. Whenever there is a link disconnection on the active route, the routing protocol must perform a route recovery process. In QoS routing for wired networks, multiple path routing is popularly used. Multiple routes are however, constructed using link-state or distance vector algorithms which are not well-suited for ad hoc networks. We propose an on-demand routing scheme called Split Multipath Routing (SMR) that establishes and utilizes multiple routes of maximally disjoint paths. Providing multiple routes helps minimizing route recovery process and control message overhead. Our protocol uses a per-packet allocation scheme to distribute data packets into multiple paths of active sessions. This traffic distribution efficiently utilizes available network resources and prevents nodes of the route from being congested in heavily loaded traffic situations. We evaluate the performance of our scheme using extensive simulation.

I. INTRODUCTION

An ad hoc network is an instantly deployable wireless network without any base station or infrastructure support. Because ad hoc networks can be easily deployed, they are developed in applications such as automated battlefields, search and rescue, crowd control, and disaster recovery. Network hosts in ad hoc networks are equipped with packet radios for communications between one another. Because these packet radios have a short propagation range (150 - 250 meters in an open field), the route is multihop when the communication pair is not within each other's proximity. Nodes are operating in power limited batteries, and bandwidth is constrained in wireless networks. Moreover, all nodes can be mobile, and the topology changes frequently. Routing protocols therefore, play an important role in ad hoc network communications.

The most popular routing approach in ad hoc networks is on-demand routing. Instead of periodically exchanging route messages to maintain permanent route table of full topology, on-demand routing protocols build routes only when a node needs to send data packets to a destination. The source floods the packet to search the destination and discover the route. Link failures of only active routes are updated, and since no periodic route table exchange is required, control overhead is minimized and the routing information is utilized efficiently. Most proposed protocols of this type (for example, Dynamic Source Routing (DSR) [6] and Ad hoc On-demand Distance Vector (AODV) [14]) however, use single route for each session.

Multiple paths can be useful in improving the effective bandwidth of communication pairs, responding to congestion and bursty traffic, and increasing delivery reliability. In QoS routing in wired networks, multipath routing has been widely developed [2], [3], [10], [12], [17], [18], [20], [21]. These protocols use table-driven algorithms (link state or distance vector) to compute multiple routes. Studies show however, that proactive protocols perform poorly in mobile networks because of excessive routing overhead [1], [5]. Multipath routing in ad hoc networks has been proposed in [9], [11], [13], [15]. Although these protocols build multiple routes on demand, the traffic is not distributed into multipaths; only one route is primarily used and alternate paths are utilized only when the primary route is broken.

We present Split Multipath Routing (SMR) protocol that builds maximally disjoint paths. Multiple routes, of which one is the shortest delay path, are discovered on demand. Established routes are not necessarily of equal length. Data traffic is split into multiple routes to avoid congestion and to use network resources efficiently. We believe providing multiple routes is beneficial in network communications, particularly in mobile wireless networks where routes are disconnected frequently because of mobility and poor wireless link quality.

The remainder of this paper is organized as follows. Section II describes the protocol mechanism in detail. Performance evaluation by simulation is presented in Section III and concluding remarks are made in Section IV.

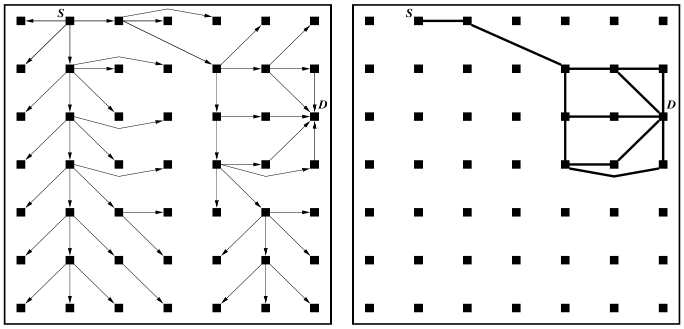
II. SPLIT MULTIPATH ROUTING

A. Route Discovery

Split Multipath Routing (SMR) is an on-demand routing protocol that builds multiple routes using request/reply cycles. When the source needs a route to the destination but no route information is known, it floods the ROUTE REQUEST (RREQ) message to the entire network. Because this packet is flooded, several duplicates that traversed through different routes reach the destination. The destination node selects multiple disjoint routes and sends ROUTE REPLY (RREP) packets back to the source via the chosen routes.

A.1 RREQ Propagation

The main goal of SMR is to build *maximally disjoint multiple paths*. We want to construct maximally disjoint routes to prevent



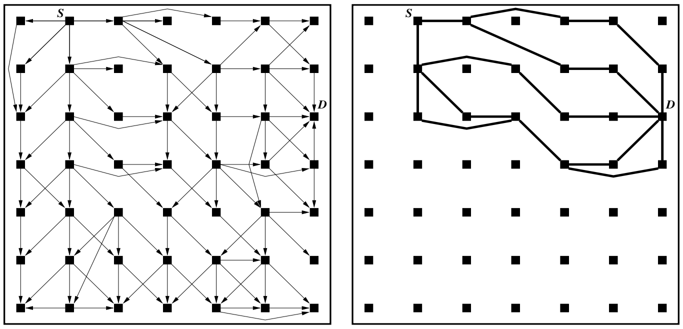
(a) RREQ propagation (b) Available paths

Fig. 1. Overlapped multiple routes.

certain nodes from being congested, and to utilize the available network resources efficiently. To achieve this goal in on-demand routing schemes, the destination must know the entire path of all available routes so that it can select the routes. Therefore, we use the source routing approach where the information of the nodes that consist the route is included in the RREQ packet. Additionally, intermediate nodes are not allowed to send RREPs back to the source even when they have route information to the destination. If nodes reply from cache as in DSR [6] and AODV [14], it is difficult to establish maximally disjoint multiple routes because not enough RREQ packets will reach the destination and the destination node will not know the information of the route that is formed from the cache of intermediate nodes.

When the source has data packets to send but does not have the route information to the destination, it transmits a RREQ packet. The packet contains the source ID and a sequence number that uniquely identify the packet. When a node other than the destination receives a RREQ that is not a duplicate, it appends its ID and re-broadcasts the packet. During simulation experiments however, we found out that dropping all duplicate RREQs only generate multiple paths that are mostly overlapped. Figure 1 (a) shows the paths taken by RREQs from the source node S to the destination node D, and Figure 1 (b) depicts the available routes. We can observe that all five routes share the first two links.

In order to avoid this overlapped route problem, we introduce a different packet forwarding approach. Instead of dropping every duplicate RREQs, intermediate nodes forward the duplicate packets that traversed through a different incoming link than the link from which the first RREQ is received, and whose hop count is not larger than that of the first received RREQ. Figure 2 (a) shows the paths taken by RREQs using this technique. We can select more disjoint paths from routes available in Figure 2 (b) than those in Figure 1 (b). Our approach has a disadvantage of transmiting more RREQ packets, but it enables us to discover maximally disjoint routes.



(a) RREQ propagation (b) Available paths

Fig. 2. Multiple routes with maximally disjoint paths.

A.2 Route Selection Method

In our algorithm, the destination selects two routes that are maximally disjoint. More than two routes can be chosen, but we limit the number of routes to two in this study. One of the two routes is the shortest delay route; the path taken by the first RREQ the destination receives. We use the shortest delay path as one of the two routes to minimize the route acquisition latency required by on-demand routing protocols. When receiving the first RREQ, the destination records the entire path and sends a RREP to the source via this route. The node IDs of the entire path is recorded in the RREP, and hence the intermediate nodes can forward this packet using this information. After this process, the destination waits a certain duration of time to receive more RREQs and learn all possible routes. It then selects the route that is maximally disjoint to the route that is already replied. The maximally disjoint route can be selected because the destination knows the entire path information of the first route and all other candidate routes. If there are more than one route that are maximally disjoint with the first route, the one with the shortest hop distance is chosen. If there still remain multiple routes that meet the condition, the path that delivered the RREQ to the destination the quickest between them is selected. The destination then sends another RREP to the source via the second route selected. Note that two routes of the session are not necessarily of equal length.

Because our protocol uses the source routing and intermediate nodes do not reply from cache, only the source nodes maintain route information to destinations. Each node hence uses less memory, but packet header size is larger because we use source routing.

B. Route Maintenance

A link of a route can be disconnected because of mobility, congestion, and packet collisions. It is important to recover broken routes immediately to do effective routing. In SMR, when a node fails to deliver the data packet to the next hop of the route (by receiving a link layer feedback from IEEE 802.11 [4] or not receiving passive acknowledgments [7]), it considers the link to be disconnected and sends a ROUTE ERROR (RERR) packet to the upstream direction of the route. The RERR message contains the route to the source, and the immediate upstream and downstream nodes of the broken link. Upon receiving this RERR packet, the source removes every entry in its route table that uses the broken link (regardless of the destination). If only one of the two routes of the session is invalidated, the source uses the remaining valid route to deliver data packets.

When the source is informed of a route disconnection and the session is still active, it may use one of the two policies in rediscovering routes:

- initiates the route recovery process when any route of the session is broken, or
- initiates the route recovery process only when both routes of the session are broken.

The first scheme reconstructs the routes more often and produces more control overhead than the second scheme, but the former provides multiple routes most of the time and be robust to route breaks. We evaluate both schemes by simulation in Section III.

C. *Allocation Granularity*

When the source receives a RREP after flooding the RREQ, it uses the first discovered route to send buffered data packets. When the second RREP is received, the source has two routes to the destination, and can split traffic into two routes. We use a simple *per-packet allocation* scheme when there are more than one available route to the destination. One drawback of this scheme is out of order delivery and re-sequencing burden on the destination (and hence not suitable to coexist with TCP). We believe, however, that cost-effective reordering buffers are easily implementable. We decided to use the per-packet allocation approach because it is known to work well in most networks [8], and most of all, it is fairly difficult to obtain the network condition (such as available bandwidth) in ad hoc networks to apply more sophisticated schemes.

III. PERFORMANCE EVALUATION

A. *Simulation Environment*

We evaluate and compare the performance of the following protocols:

- SMR-1: SMR which performs the route recovery when any route to the destination is invalidated
- SMR-2: SMR which performs the route recovery only when both routes to the destination are invalidated
- DSR: Dynamic Source Routing [6] which uses single path.

We implemented the simulator within the Global Mobile Simulation (GloMoSim) library [19]. 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.

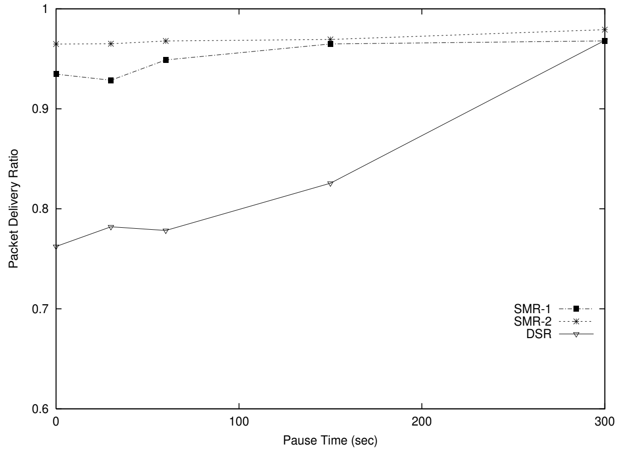


Fig. 3. Packet delivery ratio.

A free space propagation model with a threshold cutoff [16] 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) [4] as the medium access control protocol. A traffic generator was developed to simulate constant bit rate sources. There are twenty data sessions, and the sources and the destinations are randomly selected with uniform probabilities. The size of data payload was 512 bytes. We used random waypoint model [6] as the mobility model. We generated various mobility degree by using different pause times. The minimum and the maximum speed were set constant to zero and 10 m/s, respectively.

B. *Results and Analysis*

Figure 3 shows the throughput of each protocol in packet delivery fraction. Packet delivery ratio is obtained by dividing the number of data packets correctly received by the destinations by the number of data packets originated by the sources. We can observe from the result that both SMR schemes outperform DSR, especially when the mobility increases (i.e., the pause time decreases). In DSR, only one route is used for each session and when that route is invalidated, the source uses the cached route that is learned from overhearing packets. If no such cached route is available, it sends a RREQ to discover a new route. In the latter case, intermediate nodes that have cached routes to the destination provide those route to the source by sending RREPs. DSR however, does not apply any aging mechanism for cached route entries, and hence routes stored in the cache (either by the source or the intermediate nodes) may be stale. After a route break, source nodes will use these newly acquired but obsolete routes only to learn that they are also invalid, and will attempt another route recovery. Many data packets are dropped during this process and more delay is needed to discover correct routes.

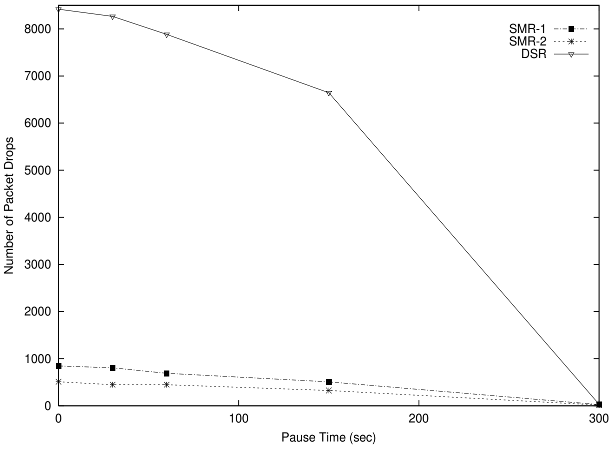


Fig. 4. Number of packet drops.

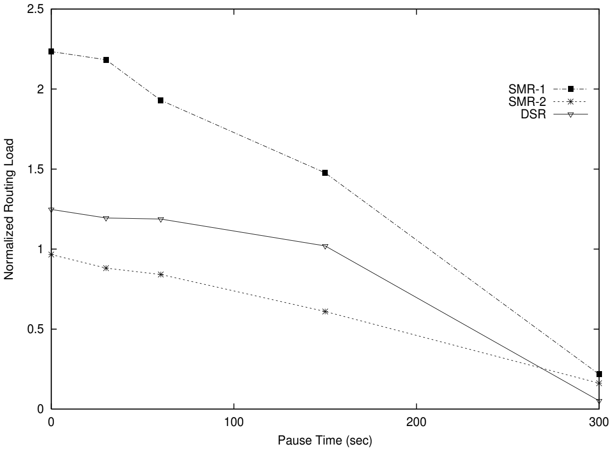


Fig. 5. Normalized routing load.

Between SMR protocols, SMR-2 delivers more packets than SMR-1. We can analyze that the control packets generated by the route rediscovery processes of SMR-1 cause collision and contention with data packets. Even though SMR-2 will have only one available route to the destination after the other route is broken, it can still deliver data packets without producing control traffic as long as the remaining route stays connected, and that leads to a good throughput performance.

Figure 4 illustrates the number of packets dropped by each protocol. Both data and control packets are measured. The reasons for packet drops can be incorrect route information, mobility, collisions, and congestion. DSR cannot maintain precise routes and drops more packets as nodes move more often (i.e., less pause time). The usage of state routes from caches is the major reason of DSR packet drops. Both SMR schemes have considerably fewer packet drops compared with DSR. SMR-2 has fewer packet drops than SMR-1 because it invokes fewer route recovery processes and consequently, transmits less control messages.

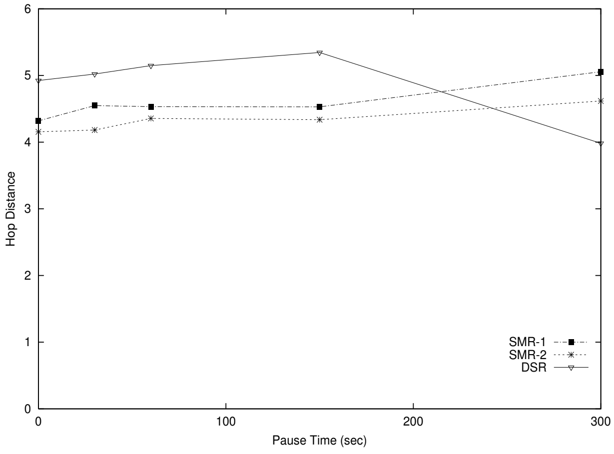


Fig. 6. Hop distance.

Figure 5 presents the control 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. This value hence represents the protocol efficiency. When there is no mobility, DSR has the least value. This result is expected because SMR protocols generate more control packets while building multiple routes. On the other hand, DSR builds single route for each session and minimizes flooding overhead by allowing intermediate nodes of replying from cache. Cached routes are useful in static networks as they remain valid for the entire duration. As mobility is increased, however, SMR-2 shows better efficiency than DSR. DSR yields less overhead in initial route discovery process, but it invokes more route reconstruction procedures than SMR-2 since DSR intermediate nodes often reply with stale routes. Additionally, DSR transmits considerably more RERR packets than SMR schemes because DSR has more route disconnections and route recoveries. Furthermore, DSR sends RERR packets whenever a unicast packet (data, RREP, and RERR) fails to be delivered to the next hop. SMR sends RERR only when the data packet is undeliverable. Therefore, DSR shows higher normalized routing load than SMR-2 when mobility is present. We can also observe that SMR-1 shows less efficiency than other protocols regardless of mobility. Since the source floods the network with RREQs when any route of a session is disconnected, more control packets are transmitted than DSR and SMR-2. We can deduce from this result that excessive flooding makes the protocol inefficient.

Figure 6 reports the average hop distance of each protocol. DSR has the shortest hop distance when there is no mobility because SMR schemes' second routes may have longer distance than the first routes. With mobility however, the hop distance of DSR grows and becomes larger than those of SMR protocols. If the route is established directly from the destination, it can be the shortest route 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. These

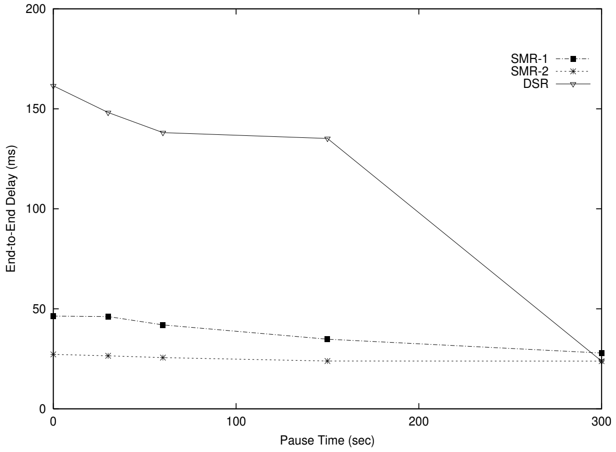


Fig. 7. End-to-end delay.

routes may not be fresh and do not exploit the current network topology. DSR therefore builds longer routes than SMR protocols. Longer paths have a more chance of having route breaks since one link disconnection results in a route invalidation. Results from Figure 3 confirms our observation.

Figure 7 shows the end-to-end delay. DSR has the longest delay in mobile scenarios because it delivers data packets on routes longer than those of SMR. In addition, DSR yields longer delays in reconstructing routes and the period of time the data packets are buffered at the source node during route recovery results in larger end-to-end delays. SMR on the other hand, uses the remaining valid route when one of the multiple route is disconnected, and hence no route acquisition latency is required.

IV. CONCLUSION

We presented the Split Multipath Routing (SMR) protocol for ad hoc networks. SMR is an on-demand protocol that builds maximally disjoint routes. Our scheme uses two routes for each session; the shortest delay route and the one that is maximally disjoint with the shortest delay route. We attempt to build maximally disjoint routes to avoid having certain links from being congested, and to efficiently utilize the available network resources. Providing multiple paths is useful in ad hoc networks because when one of the route is disconnected, the source can simply use other available routes without performing the route recovery process.

We introduced two approaches in SMR route maintenance. The first scheme builds a new pair of routes when any existing route of the session is disconnected. The second scheme performs rerouting only when both routes are broken. We have conducted a simulation performance evaluation of these two SMR schemes and DSR which uses single shortest delay route. Our study indicates that SMR outperforms DSR because multiple routes provide robustness to mobility. The performance difference becomes evident as the mobility degree increases. SMR had considerably fewer packet drops compared with DSR. Splitting the traffic into multiple routes helps distribute the load to the network hosts. SMR also showed shorter end-to-end delay because route acquisition latency is not required for all route disconnections. Between SMR protocols, the second scheme showed better efficiency as it performs fewer route recoveries and hence generates less control overhead.

References

[1] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, and J. Jetcheva, “A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols,” *Proceedings of ACM/IEEE MOBICOM’98*, Dallas, TX, Oct. 1998, pp. 85-97.

[2] J. Chen, P. Druschel, and D. Subramanian, “An Efficient Multipath Forwarding Method,” *Proceedings of IEEE INFOCOM’98*, San Francisco, CA, Mar. 1998, pp. 1418-1425.

[3] I. Cidon, R. Rom, and Y. Shavitt, “Analysis of Multi-Path Routing,” *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, Dec. 1999, pp. 885-896.

[4] 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.

[5] P. Johansson, 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, Aug. 1999, pp. 195-206.

[6] 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.

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

[8] R. Krishnan and J.A. Silvester, “Choice of Allocation Granularity in Multipath Source Routing Schemes,” *Proceedings of IEEE INFOCOM’93*, San Francisco, CA, Mar. 1993, pp. 322-329.

[9] S.-J. Lee and M. Gerla, “AODV-BR: Backup Routing in Ad hoc Networks,” *Proceedings of IEEE WCNC 2000*, Chicago, IL, Sep. 2000.

[10] S. Murthy and J.J. Garcia-Luna-Aceves, “Congestion-Oriented Shortest Multipath Routing,” *Proceedings of IEEE INFOCOM’96*, San Francisco, CA, Mar. 1996, pp. 1028-1036.

[11] A. Nasipuri and S.R. Das, “On-Demand Multipath Routing for Mobile Ad Hoc Networks,” *Proceedings of IEEE ICCCN’99*, Boston, MA, Oct. 1999, pp. 64-70.

[12] R. Ogier, V. Rutenburg, and N. Shacham, “Distributed Algorithms for Computing Shortest Pairs of Disjoint Paths,” *IEEE Transactions on Information Theory*, vol. 39, no. 2, Mar. 1993, pp. 443-455.

[13] V.D. Park and M.S. Corson, “A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks,” *Proceedings of IEEE INFOCOM’97*, Kobe, Japan, Apr. 1997, pp. 1405-1413.

[14] C.E. Perkins and E.M. Royer, “Ad-Hoc On Demand Distance Vector Routing,” *Proceedings of IEEE WMCSA’99*, New Orleans, LA, Feb. 1999, pp. 90-100.

[15] J. Raju and J.J. Garcia-Luna-Aceves, “A New Approach to On-demand Loop-Free Multipath Routing,” *Proceedings of IEEE ICCCN’99*, Boston, MA, Oct. 1999, pp. 522-527.

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

[17] D. Sidhu, R. Nair, and S. Abdallah, “Finding Disjoint Paths in Networks,” *Proceedings of ACM SIGCOMM’91*, Zurich, Switzerland, Sep. 1991, pp. 43-51.

[18] N. Taft-Plotkin, B. Bellur, and R. Ogier, “Quality-of-Service Routing Using Maximally Disjoint Paths,” *Proceedings of IEEE IWQoS’99*, London, UK, Jun. 1999, pp. 119-128.

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

[20] S. Vutukury and J.J. Garcia-Luna-Aceves, “An Algorithm for Multipath Computation Using Distance-Vectors with Predecessor Information,” *Proceedings of IEEE ICCCN’99*, Boston, MA, Oct. 1999, pp. 534-539.

[21] W.T. Zaumen and J.J. Garcia-Luna-Aceves, “Loop-Free Multipath Routing Using Generalized Diffusing Computations,” *Proceedings of IEEE INFOCOM’98*, San Francisco, CA, Mar. 1998, pp. 1408-1417.