University of California
Los Angeles

Routing and Multicasting Strategies in Wireless Mobile Ad hoc Networks

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Sung-Ju Lee

2000

The dissertation of Sung-Ju Lee is approved.

Babak Daneshrad

Jack W. Carlyle

Rajive L. Bagrodia

Mario Gerla, Committee Chair

University of California, Los Angeles

2000

to my loving parents, grandparents, and my brother Daniel

Table of Contents

List of Figures

List of Tables

Acknowledgments

I thank each one of my committee members for their valuable comments and feedbacks. I owe a special gratitude to my advisor; Professor Mario Gerla. He took a chance by accepting me as his student, and patiently guided me throughout my research career at UCLA. He always found time whenever I needed an intelligent discussion. I thank Professor Rajive L. Bagrodia for letting me participate in the DOMAINS project. Being part of it enabled me to collaborate with fellow students and work as a team. Professor Jack W. Carlyle gave me helpful academic advices since my first year at UCLA. His knowledge and experience in the field is insurmountable. Professor Babak Daneshrad shared his expertise in electrical engineering aspect of the wireless mobile networking field. I was fortunate to have him as one of my committee member.

I also thank my former advisors in Hanyang University; Professor Sung Han Park, Professor Jong Kyu Lee, Professor Young Shik Moon, and Professor Heekuck Oh, for their continuous support.

I was fortunate to work with unbelievably talented individuals. Collaborating and writing papers with Professor C.K. Toh, Charles E. Perkins, Dr. William W. Su, Elizabeth M. Royer, Sang Ho Bae, Julian Hsu, Russell Hayashida, Dr. Ching-Chuan Chiang, and Dr. Guangyu Pei was a precious experience.

I want to thank the following simulation gurus; Ken Tang, Lokesh Bajaj, Mineo Takai, and Jay Martin helped me a great deal in learning simulations and most of all, catching and fixing programming bugs.

My gratitude also goes out to Dr. Ronn Ritke and James Stepanek. They always had the kindness to proofread my papers and give helpful comments.

I am also grateful of the department staff members. My special thanks to

Verra Morgan for making sure I stay out of trouble. I also thank four different assistants (Alex, Teri, Brenda, and Cynthia) who served one after another at BH3731H during my four years of stay. They kindly helped my advisor and his students (including myself).

Kudos to my friends back home in Korea and those whom I met in the United States. The friendship I have with them is what kept me going on during the tough times. I will not even attempt to name all of them here. There are too many, and I do not want to make an awful mistake of leaving out someone. Thank you, guys! You know who you are.

Lastly, I thank DARPA for their financial support.

Vita

1974	Born, Seoul, Korea.
1996	B. S., Computer Science and Engineering Hanyang University Korea
1997-1998	Teaching Assistant Computer Science Department University of California Los Angeles, California
1998	M. S., Computer Science University of California Los Angeles, California
1998-present	Research Assistant Computer Science Department University of California Los Angeles, California

Publications and Presentations

Sang Ho Bae, Sung-Ju Lee, and Mario Gerla, "Unicast Performance Analysis of the ODMRP in a Mobile Ad hoc Network Testbed," To appear in *Proceedings*

of IEEE International Conference on Computer Communications and Networks (ICCCN), Las Vegas, NV, October 2000.

Sang Ho Bae, Sung-Ju Lee, William Su, and Mario Gerla, "Implementation of a Multicast Routing Protocol in a Wireless Ad hoc Network Testbed," *Technical Report*, Computer Science Department, University of California, Los Angeles, 990049, November 1999.

Sang Ho Bae, Sung-Ju Lee, William Su, and Mario Gerla, "The Design, Implementation, and Performance Evaluation of the On-Demand Multicast Routing Protocol in Multihop Wireless Networks," *IEEE Network*, special issue on Multicasting Empowering the Next Generation Internet, vol. 14, no. 1, January/February 2000, pp. 70-77.

Mario Gerla, Guangyu Pei, and Sung-Ju Lee, "Wireless, Mobile Ad-Hoc Routing," *Presented at ACM/IEEE WINLAB/Berkeley FOCUS*, New Brunswick, NJ, May 1999.

Sung-Ju Lee, "ODMRP and GloMoSim Simulation Environment," *Presented at USC/Information Science Institute*, Marina Del Ray, CA, February 1999.

Sung-Ju Lee, "Wireless Unicast and Multicast Protocols," *Presented at PARSEC Workshop*, UCLA, Los Angeles, CA, November 1999.

Sung-Ju Lee and Mario Gerla, "Dynamic Load-Aware Routing in Ad hoc Networks," *Technical Report*, Computer Science Department, University of Califor-

nia, Los Angeles, August 2000.

Sung-Ju Lee and Mario Gerla, "SMR: Split Multipath Routing with Maximally Disjoint Paths in Ad hoc Networks," *Technical Report*, Computer Science Department, University of California, Los Angeles, August 2000.

Sung-Ju Lee and Mario Gerla, "AODV-BR: Backup Routing in Ad hoc Networks," To appear in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Chicago, IL, September 2000.

Sung-Ju Lee, Mario Gerla, and Ching-Chuan Chiang, "On-Demand Multicast Routing Protocol," *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, New Orleans, LA, September 1999, pp. 1298-1302.

Sung-Ju Lee, Mario Gerla, and Chai-Keong Toh, "A Simulation Study of Table-Driven and On-Demand Routing Protocols for Ad Hoc Networks," *IEEE Network*, vol. 13, no. 4, July/August 1999, pp. 48-54.

Sung-Ju Lee, Julian Hsu, Russell Hayashida, Mario Gerla, and Rajive Bagrodia, "Selecting Routing Strategies for Your Ad Hoc Networks," *Technical Report*, Computer Science Department, University of California, Los Angeles, 990045, October 1999.

Sung-Ju Lee, Elizabeth M. Royer, and Charles E. Perkins, "Ad hoc Routing Protocol Scalability," *Technical Report*, Computer Science Department, University of California, Los Angeles, July 2000.

Sung-Ju Lee, William Su, and Mario Gerla, "Ad hoc Wireless Multicast with Mobility Prediction," *Proceedings of the IEEE International Conference on Computer Communications and Networks (ICCCN)*, Boston, MA, October 1999, pp. 4-9.

Sung-Ju Lee, William Su, and Mario Gerla, "On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks," *IETF Internet Draft*, draft-ietf-manet-odmrp-02.txt, January 2000 (Work in Progress).

Sung-Ju Lee, William Su, and Mario Gerla, "Exploiting the UnicastFunctionality of the On-Demand Multicast Routing Protocol," To appear in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Chicago, IL, September 2000.

Sung-Ju Lee, William Su, and Mario Gerla, "Wireless Ad hoc Multicast Routing with Mobility Prediction," To appear in *ACM/Baltzer Mobile Networks and Applications*, special issue on Routing and Multicasting in Wireless Networks, 2000.

Sung-Ju Lee, William Su, and Mario Gerla, "On-Demand Multicast Routing Protocol in Multihop Wireless Mobile Networks," To appear in *ACM/Baltzer Mobile Networks and Applications*, special issue on Multipoint Communication in Wireless Mobile Networks, 2000.

Sung-Ju Lee, William Su, Julian Hsu, Mario Gerla, and Rajive Bagrodia, "A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols," *Pro-*

ceedings of the IEEE Conference on Computer Communications (INFOCOM), Tel Aviv, Israel, March 2000, pp. 565-574.

Sung-Ju Lee, Chai-Keong Toh, and Mario Gerla, "Performance Evaluation of Table-Driven and On-Demand Ad Hoc Routing Protocols," *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communication (PIMRC)*, Osaka, Japan, September 1999, pp. 297-301.

Elizabeth M. Royer, Sung-Ju Lee, and Charles E. Perkins, "The Effects of MAC Protocols on Ad hoc Network Communication," To appear in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Chicago, IL, September 2000.

William Su, Sung-Ju Lee, and Mario Gerla, "Mobility Prediction inWireless Networks," To appear in *Proceedings of the IEEE Military Communications International Symposium (MILCOM)*, Los Angeles, CA, October 2000.

William Su, Sung-Ju Lee, and Mario Gerla, "Mobility Prediction and Routing in Ad Hoc Wireless Networks," To appear in *International Journal of Network Management*, Wiley & Sons, 2000.

Abstract of the Dissertation

Routing and Multicasting Strategies in Wireless Mobile Ad hoc Networks

by

Sung-Ju Lee

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 2000

Professor Mario Gerla, Chair

Ad hoc networks are gaining increasing popularity in recent years because of their ease of deployment. No wired base station or infrastructure is supported, and each host communicasts one another via packet radios. In ad hoc networks, routing protocols are challenged with establishing and maintaining multihop routes in the face of mobility, bandwidth limitation and power constraints. In this dissertation, we study the routing strategies for ad hoc networks. On-demand routing protocols and table-driven algorithms are analyzed and compared against each other. Our study shows that on-demand protocols are better suited for mobile networks because they generate less control overhead and manage the mobility in a more efficient manner. Simulation experiments also indicate that providing multiple routes is beneficial in increasing the robustness against mobility.

We investigate the scalability characteristics of on-demand routing protocols and propose schemes to enhance the performance. We also study the interaction between MAC (Medium Access Control) and routing protocols by simulation.

Based on the lessons learned from the performance evaluation studies, we de-

sign new on-demand protocols. We introduce three unicast routing algorithms with different approaches. AODV-BR (Ad hoc On-demand Distance Vector with Backup Routes) is a scheme applied to the existing AODV protocol for establishing backup routes while the primary route is constructed, without transmitting additional control messages. Backup routes are utilized when the primary path is disconnected. The Split Multipath Routing (SMR) protocol builds maximally disjoint routes. Providing multiple routes helps minimizing route recovery process and reducing control overhead. Distributing traffic into multipaths prevents nodes from being congested. Dynamic Load Aware Routing (DLAR) is a protocol that uses the load of the intermediate nodes instead of the shortest distance, as the main route selection metric. The protocol attempts to avoid building routes with congested links.

We then present the On-Demand Multicast Routing Protocol (ODMRP), a novel multicasting scheme that utilizes a mesh structure. Multiple routes created by the mesh make the protocol robust to mobility. Multicast routes and group membership are obtained on demand to use the network resources efficiently and effectively. Simulation study shows that ODMRP outperforms other popular multicast protocols.

CHAPTER 1

Introduction

1.1 Backgroud and Motivation

1.1.1 Ad hoc Networks

With the advance of wireless communication technology, portable computers with radios are being increasingly deployed in common activities. Applications such as conferences, meetings, lectures, crowd control, search and rescue, disaster recovery, and automated battlefields typically do not have central administration or infrastructure available. In these situations, *ad hoc networks*, or *packet radio networks* [59, 61, 71] consisting of hosts equipped with portable radios must be deployed impromptu without any wired base stations. In ad hoc networks, each host must act as a router since routes are mostly *multihop*. Nodes in such a network move arbitrarily, thus network topology changes frequently, unpredictably, and may consist of unidirectional links as well as bidirectional links. Moreover, wireless channel bandwidth is limited. The scarce bandwidth decreases even further due to the effects of multiple access, signal interference, and channel fading. Network hosts of ad hoc networks operate on constrained battery power which will eventually be exhausted. Ad hoc networks are also more prone to security threats. All these limitations and constraints make multihop network research more challenging.

1.1.2 Challenges in Routing and Multicasting

Routes in ad hoc networks are multihop because of the limited propagation range (250 meters in an open field) of wireless radios. Since nodes in the network move freely and randomly, routes often get disconnected. Routing protocols are thus responsible for maintaining and reconstructing the routes in a timely manner as well as establishing the durable routes. In addition, routing protocols are required to perform all the above tasks without generating excessive control message overhead. Control packets must be utilized efficiently to deliver data packets, and be generated only when necessary. Reducing the control overhead can make the routing protocol efficient in bandwidth and energy consumption.

Multipoint communications [42] have emerged as one of the most researched areas in the field of networking. As the technology and popularity of Internet grow, applications, such as video conferencing, that require multicast support are becoming more widespread. In a typical ad hoc environment, network hosts work in groups to carry out a given task. Therefore, multicast plays an important role in ad hoc networks. Multicast protocols used in static networks (e.g., Distance Vector Multicast Routing Protocol (DVMRP) [40], Multicast Open Shortest Path First (MOSPF) [112], Core Based Trees (CBT) [11], and Protocol Independent Multicast (PIM) [41]) do not perform well in wireless ad hoc networks because multicast tree structures are fragile and must be readjusted as connectivity changes. Furthermore, multicast trees usually require a global routing substructure such as link state [111] or distance vector [104]. The frequent exchange of routing vectors or link state tables, triggered by continuous topology changes, yields excessive channel and processing overhead. Hence, the tree structures used in static networks must be modified, or a different topology between group members (i.e., mesh) need to be deployed for efficient multicasting

in wireless mobile ad hoc networks.

1.2 Accomplishments and Contributions

Our accomplishments, which are elaborated throughout this dissertation, can be broadly listed as follows:

- Evaluated the routing performance of a traditional table-driven algorithm (Bellman-Ford) in ad hoc networks, and compared it with on-demand protocols with different routing algorithms [87, 96].

- Studied and compared the simulation performance of various routing protocols in ad hoc networks [88]. Our work is the first to perform a simulation study of various routing styles in a common realistic environment.

- Performed simulations of up to 10,000 network nodes and evaluated ad hoc routing protocol scalability [89]. We also introduced several schemes to improve the protocol performance in large networks. This work is the first to conduct a simulation study of such size.

- Studied the interaction between Medium Access Control (MAC) and routing protocols [138]. Four different MAC protocols and three styles of routing protocols were simulated and their interactions were analyzed.

- Designed on-demand unicast protocols that build multiple routes. Ad hoc On-demand Distance Vector with Backup Routes (AODV-BR) [85] is a scheme applied to the existing AODV protocol to construct multiple backup routes without generating additional control overhead. Backup routes are utilized when the primary route is disconnected. On the other hand, Split

Multipath Routing (SMR) [83] builds maximally disjoint multiple routes and distributes the traffic into multipaths.

- Introduced a novel route selection method with the Dynamic Load Aware Routing (DLAR) protocol [84]. DLAR uses the routing load as the primary route selection metric instead of the conventional shortest route. The protocol monitors the status of active routes continuously to avoid creating bottlenecks.

- Proposed the On-Demand Multicast Routing Protocol (ODMRP) [86, 91, 94]. ODMRP builds the mesh structure on demand to provide multiple paths among multicast members. The mesh makes the protocol robust to mobility. ODMRP can function as both multicast and unicast [90]. We implemented the protocol in simulation platform using GloMoSim [160], and in a real ad hoc network testbed [7, 8, 9]. The protocol is a standard candidate at the IETF (Internet Engineering Task Force) MANET (Mobile Ad hoc Networks) Working Group [61].

- Applied various techniques to enhance the performance of ODMRP [92, 93]. These enhancements include mobility prediction [152, 153], reliable packet delivery, and elimination of the route acquisition latency.

- Studied and compared the simulation performance of various multicast schemes in ad hoc networks [95]. Our work is the first to perform a simulation study of various multicast routing protocols in a common realistic environment.

1.3 Related Work

1.3.1 Routing Protocols

Routing protocols proposed for mobile ad hoc wireless networks can be generally categorized by the routing strategy. First, there are protocols that are *distance vector* typed. Pure distance vector algorithms (e.g., Distributed Bellman Ford [15, 46], Routing Internet Protocol (RIP) [103], etc.) do not perform well in mobile networks because of slow convergence and *count-to-infinity* problem [155]. Thus, newly proposed protocols modify and enhance the distance vector algorithm. Protocols of this type include Wireless Routing Protocol (WRP) [114], Destination Sequence Distance Vector (DSDV) routing protocol [127], Least Resistance Routing (LRR) [131], and the protocol by Lin and Liu [100].

Second, there are protocols that are based on link state [108, 110] algorithms. Protocols such as Global State Routing (GSR) [28], Fisheye State Routing (FSR) [123], Adaptive Link-State Protocol (ALP) [53], Source Tree Adaptive Routing (STAR) [52], , Optimized Link State Routing (OLSR) protocol [63], and Landmark Ad Hoc Routing (LANMAR) [124] fall into this category.

Third, there are *on-demand* routing protocols [105] that are proposed for ad hoc networks only. On-demand routing protocols do not maintain route to each destination of the network on a continual basis. Instead, routes are established on demand by the source. When a route is needed by the source, it floods a route request packet to construct a route. Upon receiving route requests, the destination selects the best route based on route selection algorithm. Route reply packet is then sent back to the source via the newly chosen route. In on-demand routing protocols, control traffic overhead is greatly reduced since no periodic exchanges of route tables are required. Numerous protocols of this type have

been proposed. Lightweight Mobile Routing (LMR) [36], Dynamic Source Routing (DSR) [69], Temporarily Ordered Routing Algorithm (TORA) [121], Ad-Hoc On Demand Distance Vector (AODV) routing [128], Associativity-Based Routing (ABR) [159], Signal Stability-Based Adaptive (SSA) routing [43], Routing On-demand Acyclic Multipath (ROAM) algorithm [132], Multipath Dynamic Source Routing (MDSR) [118], Relative Distance Micro-discovery Ad Hoc Routing (RDMAR) protocol [3], and Route-Lifetime Assessment Based Routing (RABR) protocol [2] are typical on-demand routing protocols.

Fourth, with the advent of GPS (Global Positioning System) [72], protocols making use of node location information while building routes have been proposed recently. With the knowledge of node position, routing can be more effective at the cost of overhead required to exchange location information. Routing protocols that require GPS are Distance Routing Effect Algorithm for Mobility (DREAM) [14], Location-Aided Routing (LAR) [78], Zone-Based Hierarchical Link State (ZHLS) [67], Flow Oriented Routing Protocol (FORP) [151], Grid Location Service (GLS) [97], and Greedy Perimeter Stateless Routing (GPSR) [74].

In addition to the above mentioned routing disciplines, a few other schemes are being proposed [5, 27, 62, 79, 125, 133, 146, 148]. Zone Routing Protocol (ZRP) [122] uses proactive approach to nodes within the *zone* and reactive approach to nodes outside the zone. Core-Extraction Distributed Ad hoc Routing (CEDAR) algorithm [148] selects a minimum set of nodes (core) to perform QoS route computations. The routing schemes proposed in [25, 146] introduce power aware metrics when selecting routes. Cluster Based Routing Protocol (CBRP) [66] forms a group of nodes into clusters in order to improve scalability. Readers are referred to [56, 134, 140] for surveys of routing protocols.

1.3.2 Multicast Protocols

Many different protocols for multicasting in mobile wireless networks have been proposed in recent years. Acharya and Badrinath [1] were the first to address the issue of wireless multicast. Their protocol uses Mobile Support Stations (MSSs) to interconnect static networks with mobile hosts via wireless links. MSSs execute the protocol instead of mobile hosts to lessen the computation, memory, and power load on mobile hosts and wireless links. However, the protocol assumes that mobile hosts can only receive the multicast packets and senders are on the wired network. A similar protocol that is built on top of a user location strategy has been proposed for Personal Communication Service (PCS) networks in [6]. The protocol in [164] structures MSSs as a de Bruijn network. It guarantees exactly one delivery without broadcasting. Mobile hosts can act both as a multicast receiver and sender. Mobile Multicast (MoM) protocol [166] uses home agent functionality of Mobile IP to extend IP multicast to mobile hosts. It improves scalability by using Designated Multicast Service Providers (DMSP), but it suffers from routing latency. In addition, in order for MoM protocol to work properly, home agents and foreign agents need to be static. A group-based multicasting in wireless networks with incomplete spatial coverage (the union of all cells may not cover the location where mobile hosts reside) is illustrated in [12]. All of the protocols introduced above are designed to extend multicast from wired to wireless networks using stationary base stations or mobile support stations.

A few multicasting protocols have been recently proposed for ad hoc networks [20, 21, 29, 30, 31, 35, 50, 55, 58, 65, 77, 82, 99, 120, 139, 147, 165, 167, 169]. The Reservation-Based Multicast (RBM) routing protocol [35] builds a core (or a Rendezvous Point) based tree for each multicast group. RBM is a combination of

multicast, resource reservation, and admission control protocol where users specify requirements and constraints. The Lightweight Adaptive Multicast (LAM) algorithm [65] is a group shared tree protocol that does not require timer-based messaging. Similar to other core-based protocols, it suffers from disadvantages of traffic concentration and vulnerability of the core. The Adhoc Multicast Routing Protocol (AMRoute) [20] is also a shared-tree protocol which allows dynamic core migration based on group membership and network configuration. The Ad hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS) [167] builds a shared-tree to deliver multicast data. Each node in the multicast session is assigned an ID number and it adapts to connectivity changes by utilizing the ID numbers. A multicast extension of Ad Hoc On Demand Distance Vector (AODV) routing protocol has been newly proposed in [128]. Its uniqueness stems from the use of a destination sequence number for each multicast entry. The sequence number is generated by the multicast grouphead to prevent loops and to discard stale routes. Similar to ODMRP, the Core-Assisted Mesh Protocol (CAMP) [50] uses a mesh. However, a conventional routing infrastructure based on enhanced distance vector algorithm (e.g., Wireless Routing Protocol (WRP) [114]) or link state algorithm (e.g., Adaptive Link-State Protocol (ALP) [53]) is required for CAMP to operate. Core nodes are used to limit the traffic required when a node joins a multicast group.

1.4 Organization of the Dissertation

The rest of this dissertation is organized as follows. The next two chapters perform simulation studies of unicast routing algorithms. Chapter 2 evaluates a traditional routing algorithm (Distributed Bellman-Ford) in ad hoc networks, and compares it with on-demand protocols with different route selection metrics.

We extend this work in Chapter 3 by simulating five protocols, each from various routing approaches. Chapter 4 studies the ad hoc routing protocol scalability, and Chapter 5 investigates the interaction between MAC protocols and routing algorithms. Backup routing is illustrated in Chapter 6 and SMR is explained in Chapter 7. Chapter 8 describes the DLAR protocol. Chapter 9 introduces the ODMRP and Chapter 10 presents the improvement schemes applied to ODMRP. Chapter 11 conducts a simulation performance evaluation of various ad hoc multicast routing protocols, including ODMRP. Chapter 12 focuses on the unicast functionlity of ODMRP, and Chapter 13 reports the ODMRP implementation and experiments in a real ad hoc network testbed. Chapter 14 concludes the dissertation.

CHAPTER 2

A Review of Early Routing Protocols

Bandwidth and power constraints are the main concerns in current wireless networks because multihop, ad hoc mobile wireless networks rely on each node in the network to act as a router and packet forwarder. This dependency places bandwidth, power, and computation demands on mobile hosts which must be taken into account when choosing the best routing protocol. In recent years, protocols that build routes based "on demand" have been proposed. The major goal of on-demand routing protocols is to minimize control traffic overhead. In this chapter, we perform a simulation and performance study on some routing protocols for ad hoc networks. Distributed Bellman-Ford, a traditional table-driven routing algorithm, is simulated to evaluate its performance in multihop wireless networks. In addition, two on-demand routing protocols (Dynamic Source Routing (DSR) [69] and Associativity-Based Routing (ABR) [159]) with distinctive route selection algorithms are simulated in a common environment to quantitatively measure and contrast their performance. We have chosen these three protocols for the following reasons: (i) to evaluate the performance of a conventional table-driven routing scheme (DBF) in multihop wireless networks, and (ii) to study the performance of different routing metrics in dynamic ad hoc networks. The final selection of an appropriate protocol will depend on a variety of factors, which are discussed in this chapter.

2.1 Routing Protocols Review

2.1.1 Distributed Bellman-Ford

Distributed Bellman-Ford (DBF) algorithm was developed originally to support routing in the ARPANET. A version of it is known as RIP (Routing Internet Protocol) [103] and is still being used today to support routing in some Internet domains. It is a table-driven routing protocol, i.e., each router constantly maintains an up-to-date routing table with information on how to reach all possible destinations in the network. For each entry, the next router to reach the destination and a metric to the destination are recorded. The metric can be hop distance, total delay, or cost of sending the message. Each node in the network begins by informing its neighbors about its distance to all other nodes. The receiving nodes extract this information and modify their routing table if any route measure has changed. For instance, a different route may have been chosen as the best route or the metric to the destination may have been altered. The node uses the following formula to calculate the best route:

$$D(i,j) = \min_k [d(i,k) + D(k,j)]$$

where $D(i,j)$ is the metric on the "shortest" path from node i to node j, $d(i,k)$ is the cost of traversing directly from node i to node k, and k is one of the neighbors of node i. After recomputing the metrics, nodes pass their own distance information to their neighbor nodes again. After a while, all nodes/routers in the network have a consistent routing table to all other nodes.

This protocol does not scale well to large networks due to a number of reasons. One problem is the so called "count-to-infinity" problem. In unfavorable circumstances, it takes up to N iterations to detect the fact that a node is disconnected, where N is the number of nodes in the network [155]. Another problem is the in-

crease of route update overhead with mobility. RIP uses time-triggered (periodic, about 30sec interval) and event-triggered (link changes or router failures) routing updates. Mobility can be expressed as rate of link changes and/or router failures. In a mobile network environment, event-triggered routing updates tend to outnumber the time-triggered updates, leading to excessive overhead and inefficient usage of the limited wireless bandwidth.

2.1.2 Dynamic Source Routing

Dynamic Source Routing (DSR) [69] was developed at Carnegie Mellon University. It is a direct descendant of the source routing scheme used in bridged LANs [155]. It uses source routing instead of hop-by-hop packet routing. Each data packet carries the list of routers in the path. The main benefit of source routing is that intermediate nodes need not keep route information because the path is explicitly specified in the data packet. DSR does not require any kind of periodic message to be sent, supports uni-directional and asymmetric links, and sets up routes based on demand by the source. DSR consists of two phases: (a) route discovery and (b) route maintenance, which are explained in the following sections.

2.1.2.1 Route Discovery

When a source has a data packet to send but does not have any routing information to the destination, the source initiates a route discovery. To establish a route, the source floods a ROUTE REQUEST message with a unique request ID. When this request message reaches the destination or a node that has route information to the destination, it sends a ROUTE REPLY message containing path information back to the source. The "route cache" maintained at each node

records routes the node has learned and overheard over time to reduce overhead generated by a route discovery phase.

When a node receives a ROUTE REQUEST packet, this message is forwarded only if all of the following conditions are met: (a) the node is not the target (destination) of the ROUTE REQUEST packet, (b) the node is not listed in source route, (c) the packet is not a duplicate, and (d) no route information to the target node is available in its route cache. If all are satisfied, it appends its identification to the source route and broadcasts the packet to its neighbors. If condition (b) or (c) is not met, it simply discards the packet. If a node is the destination of the packet or has route information to the destination, it builds and sends a ROUTE REPLY to the source, as described above.

2.1.2.2 Route Maintenance

The main innovation of DSR with respect to bridged LAN routing is in route monitoring and maintenance in the presence of mobility. DSR monitors the validity of existing routes based on the acknowledgments of data packets transmitted to neighboring nodes. This monitoring is achieved by passively listening for the transmission of the neighbor to the next hop or by setting a bit in a packet to request an explicit acknowledgment. When a node fails to receive an acknowledgment, a ROUTE ERROR packet is sent to the original sender to invoke a new route discovery phase. Nodes that receive a ROUTE ERROR message delete any route entry (from their route cache) which uses the broken link. Note that a ROUTE ERROR message is propagated only when a node has a problem sending packets through that link. Although this selective propagation reduces control overhead (if no packets traverse a link), it yields a long delay when a packet needs to go through a new link.

2.1.2.3 Information Stored in Each Node

- **Route Cache**: Each node stores routing information it has learned and overheard in its route cache. Routing information can be obtained while processing ROUTE REPLY messages and the source route list of a data packet header. More than one route for each destination can be stored in the cache. When a ROUTE ERROR message is received or overheard, routes that use the broken link specified in the ROUTE ERROR are removed from the route cache.

- **Route Request Table**: Nodes producing a ROUTE REQUEST packet store information in the route request table. Recorded information includes the destination node of a ROUTE REQUEST, the time when the node last sent a ROUTE REQUEST to the destination, and the time the node has to wait until it can send a next ROUTE REQUEST to the destination. The purpose of maintaining this table is to restrict frequent ROUTE REQUEST transmissions to the same destination.

2.1.2.4 Optimizations

To improve the performance and reduce overhead, a few optimizations can be achieved in DSR. Some of the optimizations are:

- **Nonpropagating Route Requests**: When originating a ROUTE REQUEST, senders set the Time-To-Limit (TTL) to zero hop, thus allowing only the neighbors to receive packet. If a neighbor is the destination or has route information to the destination in its cache, it sends a reply to the originator. If no reply is received within a timeout period, an ordinary (propagating) ROUTE REQUEST is flooded by the sender.

- **Piggybacking on Route Discoveries**: To eliminate the route acquisition latency, data can be piggybacked on ROUTE REQUEST packets. If, however, a route is replied by an intermediate node which has route information to the destination in its cache, that node needs to construct a data packet and forward it to the destination node in order not to lose any data.

- **Gratuitous Route Replies**: When receiving a packet not addressed to itself, a node refers the listed source route that has not been traversed yet. If the unprocessed part contains the identification of the node, it realizes that a shorter route can be achieved by not visiting the preceding hops in the source route. This node sends a gratuitous ROUTE REPLY to the sender to inform a shorter route.

- **Gratuitous Route Errors**: When a source of the broken route receives a ROUTE ERROR, it piggybacks the received ROUTE ERROR on the next ROUTE REQUEST packet for route rediscovery. This piggybacking prevents nodes from replying with stale routes.

- **Salvaging**: If an intermediate node of a route detects that the next hop node cannot be reached, it searches its route cache for an alternate route. If such a route is found, it substitutes this available route for the stale route in the data header and forwards it. The intermediate node is still required to send a ROUTE ERROR back to the sender.

- **Snooping**: When processing data, a node examines the unvisited nodes in the source route and inserts those routes into its route cache. This snooping enables nodes to have multiple alternate routes for each destination.

2.1.3 Associativity-Based Routing

Developed at Cambridge University, Associativity-Based Routing (ABR) [158, 159] is a protocol that is designed for an ad hoc mobile network environment. Routes are established based on demand. The uniqueness of this scheme is the route selection criteria. By exploiting the spatial and temporal relationship of mobile hosts, ABR introduces the following new routing metrics:

- Longevity of a route based on associativity,
- Route relaying load of intermediate nodes supporting existing routes, and
- Link capacities of the selected route.

By '*associativity*' or '*affinity*' we mean the spatial, temporal, and connection relationship of a mobile host with its neighbors. Associativity is measured by recording the number of control beacons received by a node from its neighbors. For example, assume each mobile host has a transmission/reception range of ten meters in diameter and there are two mobile hosts A and B. Initially, A and B are not in radio connectivity with each other but each sends a control beacon to signify its presence once every two seconds. If A is migrating at 1 m/s and it starts to enter B's radio range and move through it diagonally, then both A and B record at most five beacons each. Hence, this is the associativity threshold. Namely, if only five or less beacons are recorded, then one can assume that the other mobile host is migrating past it, and this situation is viewed as being *associatively unstable*. Otherwise, if the mobile host is moving but is constantly within the radio coverage of its neighbors, then more than five beacons will be recorded and hence the node is regarded as being *associatively stable*. Note that associativity has an inter-locking characteristic since a node's associativity

stability with its neighbors depends on the mobility profile of the neighbors. By selecting nodes with high associativity counts/ticks, the route is expected to have a long-lived characteristic. This stability could result in a route with non-shortest path, but the route can be maintained with less chance of having to perform route recovery. The detailed algorithm for route selection in ABR can be found in [159].

The following sections shall elaborate further on: (a) route discovery and (b) route reconstruction.

2.1.3.1 Route Discovery Phase

The route discovery process consists of Broadcast Query (BQ) and BQ-REPLY cycle. When a source demands a route, it floods a BQ message. Any Intermediate Node (IN) that receives the BQ packet checks if the message has already been processed by looking up the seen table, which will be explained in Section 2.3.5. If the BQ packet has not been seen before, it appends the following to the BQ packet: (a) its identifier, (b) associativity ticks with its neighbors, (c) route relaying load, (d) link propagation delay, and (e) hop count information. The IN then broadcasts the packet to its neighbors.

When the destination node receives BQ packets, it knows all the possible routes and their qualities. The destination node then selects the best route based on longevity and other qualities (route load, minimum hop, etc.) and sends a BQ-REPLY control packet (which contains a list of INs' addresses/IDs and a summary of selected route QoS) back to the source node via the selected route. When INs of the selected route receive the BQ-REPLY packet, they update their routing tables with this new route.

2.1.3.2 Route Reconstruction (RRC) Phase

In circumstance where nodes' mobility invalidate the selected route, the Route Reconstruction (RRC) process is invoked to discover alternate partial routes quickly. The migration of neighbor nodes can be detected when no beacon message is received within the timeout interval. When an IN of an existing route moves away from radio range of its immediate upstream or downstream, the route is invalidated. The immediate downstream node sends a Route Notification (RN) packet towards the destination to inform the invalidity of that route. Nodes that subsequently receive such a message delete their route entry. The immediate upstream of the moved node, however, performs a Localized Query (LQ) to discover a new partial route. Unlike BQ, a LQ process performs a limited scope broadcast (i.e., the flood radius is controlled by a hop count field). However, similar to BQ, information about route metrics is appended into LQ packets as they make their way to the destination. After the destination node receives several LQ messages, it selects the best partial route (again based on associativity stability) and sends back a LQ-REPLY message to the node that invoked the LQ process. As a result, all nodes in this partial path have their routing entry updated, allowing subsequent data packets to be forwarded via this new partial path.

In the case when the node that sent the LQ message does not receive the LQ-REPLY message within the timeout period (i.e., when partial paths could not be located), it sends a RN packet to the immediate upstream node (i.e., backtrack). When a node receives a RN packet from an immediate downstream node, it recognizes the backtrack and invokes a LQ process again. The fundamental strategy here is to *localize* the route discovery process to a bounded region so that other parts of the route are not affected. This localization also helps in avoiding the use of full broadcast unnecessarily. For a displacement of a node along the route,

LQ processes can be performed at most half the route hop distance. Thereafter, if no partial path can be located, a RN message is sent back to the source node of the route to invoke a BQ process. This quick abort mechanism is to shorten route recovery time (avoiding the possibility of backtracking all the way to the source) by limiting the number of LQ processes.

2.1.3.3 Data Transmission

To utilize the channel efficiently, ABR uses a simple and short packet header. Each data packet header contains only the neighboring node information rather than all the nodes in the route.

Similar to DSR, flow control is achieved by monitoring passive acknowledgments. When node A receives a packet and forwards it to the next-hop node B, A hears B's transmission when B relays the packet to another node. This is known as *passive* acknowledgment and is a technique used in packet radio [71]. Active acknowledgment is used by the destination node (since it has no more neighbors to relay the packet to) where an explicit message is sent to the upstream node. If a node does not receive a passive acknowledgment within the timeout period after forwarding a packet, it retransmits the data packet for an appropriate number of times. If an acknowledgment is not received after a few attempts, a mobile host is considered to have moved out of radio range or has powered down and a RRC phase is therefore invoked.

2.1.3.4 Information Stored in Each Node

- **Routing Table**: If a node is part of an active route in the network, it stores the route information in its routing table. Not only are the source and the destination IDs of the route recorded, but also the incoming and

the outgoing node IDs are kept so that incoming packets can be forwarded accordingly. Information on the hop count to the destination and the total number of active routes that the node is currently supporting are maintained in the routing table as well. Unlike distance vector based routing protocols, ABR routing table contains only routing information for routes that are actually required by the source, not every possible destination in the network.

- **Neighbor Table**: Each node maintains a neighbor table that records its associativity relationship with surrounding neighbors. An associativity counter is incremented when a beacon message transmitted by a neighboring node is received. If no beacon message is received from a neighboring node within the timeout interval, the corresponding associativity counter field is reset to zero (to reflect the associativity instability).

- **Seen Table**: A seen table is used to prevent a mobile host from processing and forwarding the same BQ or LQ message multiple times. When receiving a BQ or LQ message, a node looks up its seen table and checks if the received message has been processed before. If an entry matches the type (BQ or LQ), source ID, destination ID, and sequence number, the received packet is discarded. Note that entries in the seen table need not be maintained permanently. Schemes such as LRU (Least Recently Used) [149] can be employed to expire and remove old entries and prevent the size of seen table to be extensive.

Table 2.1: Summary of DBF, DSR, and ABR.

Protocols	**DBF**	**DSR**	**ABR**
Route Establishment	Proactive	On-Demand	On-Demand
Routing Metric	Shortest Path	Shortest Path	Associativity, load, delay, etc.
Periodic Messages	Route Tables	None	Beacons
Loop-Free	No	Yes	Yes

2.1.4 Summary of Protocols

Key characteristics and properties of DBF, DSR, and ABR are summarized in Table 2.1.

2.2 Simulation Model

The simulator for evaluating three routing protocols is implemented within the Global Mobile Simulation (GloMoSim) library [160]. The GloMoSim library is a scalable simulation environment for wireless network systems using the parallel discrete-event simulation capability provided by PARSEC [10]. The simulation models the network of 30 mobile hosts migrating within a 20 meter $\times$ 20 meter space with a transmission radius of five meters. Every node in the network moves in a random fashion, with a static time of five seconds before migrating again. The channel capacity is 2Mb/s. The IEEE 802.11 Distributed Coordination Function (DCF) [60] is used as the medium access control protocol. A free space propagation model [135] with a threshold cutoff has been used in our experiments. In the free space model, the power of a signal attenuates as $1/d^2$ where d is the distance between radios. In addition to the free space channel model, we have also im-

plemented the SIRCIM (Simulation of Indoor Radio Channel Impulse-response Models) [136] which considers fading, barriers, foliages, multipath interference, etc. The SIRCIM is more accurate than the free space model, but we have decided against using SIRCIM in our study because: (a) the complexity of the SIRCIM increases simulation time by two orders of magnitude; (b) the accuracy of the channel model does not affect the relative ranking of the routing protocols evaluated in this study; and (c) SIRCIM must be "tuned" to the characteristics of the physical environment (e.g., indoor, outdoor etc.), thus requiring a much more specific scenario than we are assuming in our experiments. In the radio model, capture effects are taken into account. If the capture ratio (the minimum ratio of an arriving packet's signal strength relative to those of other colliding packets) [135] is greater than the predefined threshold value, the arriving packet is received while other interfering packets are dropped. A traffic generator was developed to simulate constant bit rate sources. Source nodes and destination nodes were chosen randomly with uniform probabilities. A packet is dropped when no acknowledgment is received after retransmitting it a certain number of times. Simulation runs of 200,000,000,000 simulation ticks (which is 200 seconds of simulation time) were performed multiple times.

2.3 Simulation Results

DBF, a traditional table-driven routing scheme used in wired networks, is compared with on-demand ad hoc routing schemes (ABR and DSR) in a common multihop mobile wireless network simulation platform.

Parameters of interest are: (a) control overhead, (b) data throughput, and (c) end-to-end packet propagation delay. Specifications stated in [103], [22], and [159] are employed to implement DBF, DSR, and ABR, respectively. The results

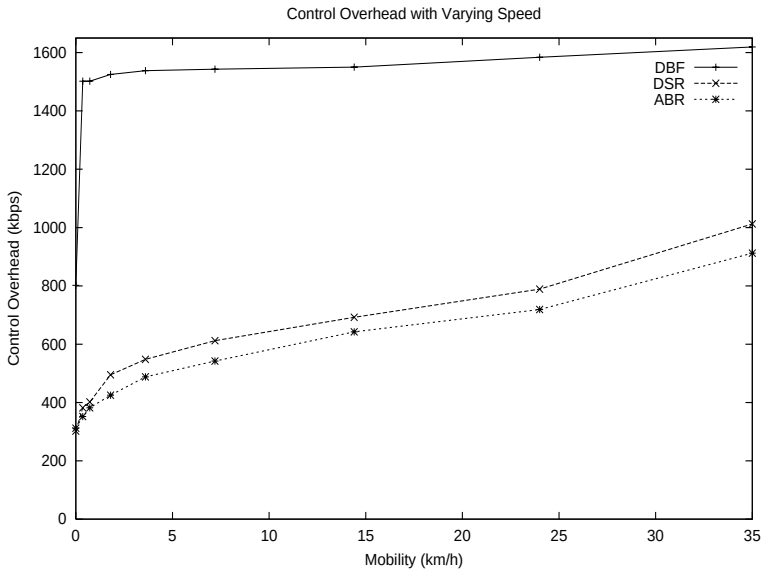


Figure 2.1: Control message overhead for different mobility speed.

obtained are discussed below.

2.3.1 Control Message Overhead

Figure 2.1 shows the control overhead incurred by DBF, DSR, and ABR. Both ABR and DSR on-demand routing schemes have considerably less overhead (as high as 76.56%) than DBF. Sending route updates periodically and triggering updates when the topology changes in order to maintain an up-to-date routing table result in excessive control message overhead, which is unacceptable in a wireless environment with limited bandwidth. We can see that DSR has less overhead than ABR when the network is static. If nodes are not mobile, there is no route breakage and control messages for route reconstruction are not required. ABR sends beacon messages to maintain the list of neighbors, thus resulting in more overhead when there is no mobility. One might expect ABR to have considerably more control overhead when nodes are stable. However, the result

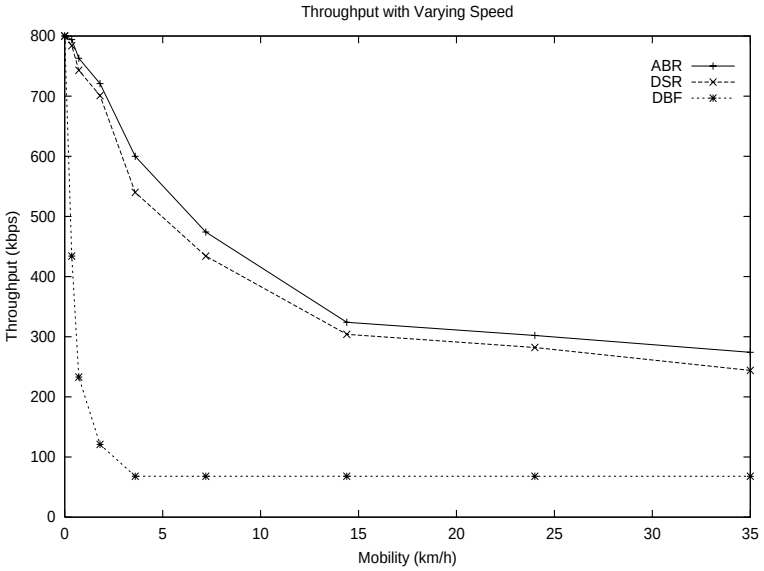


Figure 2.2: Data throughput for different mobility speed.

shows only a small difference since the size of beacon messages is very small.

We can observe from the result that increasing the mobility speed makes ABR more efficient than DSR. This efficiency is attributed to ABR's local route recovery feature. In DSR, if a node in the path becomes unreachable, a control message specifying a route error is propagated all the way back to the source to invoke a new route discovery. In contrast, in ABR the immediate upstream of a migrated node starts the LQ process to find a new partial route without intervention from the source, hence minimizing the transmission of control messages.

2.3.2 Data Throughput

Figure 2.2 shows the throughput comparison of DBF, DSR, and ABR. DBF's poor performance can be attributed to excessive channel usage by route update control messages. Also, as mobility speed increases, more event-triggered updates

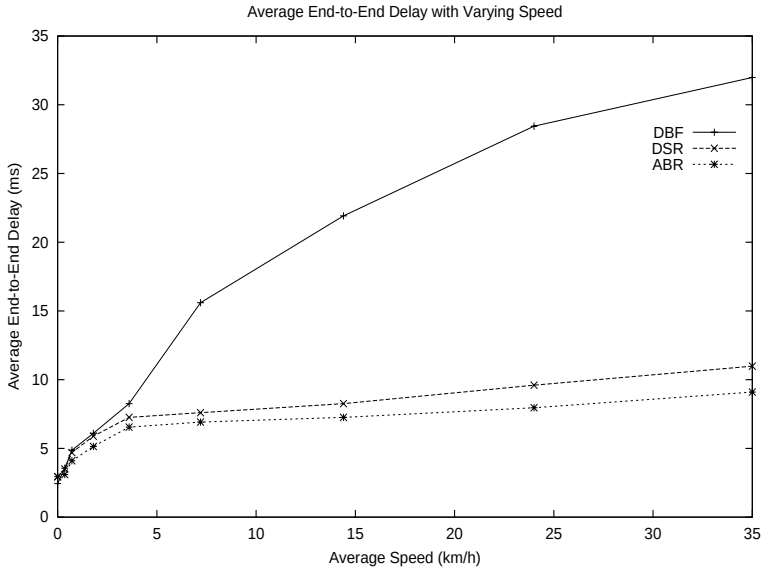


Figure 2.3: Average end-to-end delay for different mobility speed.

are generated. However, this is not present in on-demand routing protocols. The graph also reveals that the ABR has a higher throughput than DSR, resulting from the use of a different route selection process. In DSR, a route is chosen based on the shortest delay at the instance of route establishment. Although this path may be the best route at that instant, it may be a route that lacks routing stability or may have unacceptably high load. In contrast, ABR distinctively selects a route where nodes in the path are associatively stable (spatial, temporal, and connection wise) and have light load. This route selection criteria enhances the longevity of the selected route, avoids bottleneck and congestion at INs, and eventually improves throughput.

2.3.3 End-to-End Delay

Figure 2.3 shows the end-to-end delay of data packets. DBF has a larger delay than on-demand schemes due to high control overhead and thus large queueing

delay. For on-demand protocols, ABR has shorter delays than DSR, and this difference becomes more obvious as mobility speed increases. The better performance of ABR can be traced to the following reasons. First, balancing the route load shortens the delay as the chance of congestion is reduced. Second, adjusting to network mobility via receiving beacon messages from neighbors yields faster convergence. In DSR, a neighbor displacement is noticed only after a packet is sent explicitly to that node. The network reacts if an acknowledgment is not received. Consequently, this increases packet delay since the packet must wait until a new route is established.

2.3.4 Other Considerations

In the previous sections, we have compared routing algorithms based on the performance criteria typically measured in a simulation experiment, namely, throughput, delay, and control traffic overhead. There are other criteria, however, which must be taken into account when selecting the routing scheme for a specific application. Often, these criteria are not easily assessed via simulation. In this section, we examine three such criteria: table storage overhead, probability of detection/interception, and power consumption.

2.3.4.1 Table Storage Overhead

For each route discovered by DSR, a route cache table is kept at the source as well as at each node along the route. Let R be the average number of active routes a node supports and N the total number of nodes in the network. Assuming a grid-like radio connection topology (consistent with optimal radio power range), the average path length is $\sqrt{N}$. So, the total number of route cache entries for each node is on average $R\sqrt{N}$. The source node of route request packets

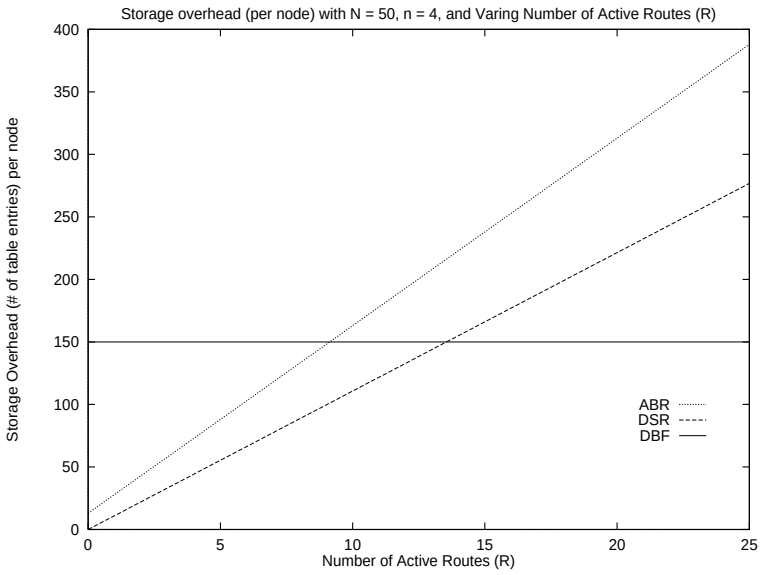


Figure 2.4: Storage overhead for different number of active routes.

maintains a node information cache.[1] Having four fields for each destination, the average number of node information cache entries per node is $4R$. Hence, the total storage overhead for DSR is $R\sqrt{N} + 4R$. Note that if there is no active traffic, i.e., R is zero, the storage overhead is zero.

ABR requires a routing table, a neighbor table, and a seen table by each node in the network[2]. The average number of routing table entries is $5R + 1$ per each node. Moreover, $4R\sqrt{N} + 2RN$ entries are needed for a seen table in the worst case where each route becomes invalid and every LQ process fails. Note that this amount of storage is needed only if entries for every possible BQs and LQs are stored forever. In practice, probability of a node receiving a duplicate packet that has traversed h hops decreases rapidly with h since duplications are automatically filtered by neighbors after the first hop. Thus, an entry needs to

[1]For the details on node information cache, see [22].

[2]See [159] for the structure of a routing table, a neighbor table, and a seen table of ABR.

be kept in the seen table only for a relatively short time and can be removed after a timeout. Therefore, maintaining a fixed number of entries for the seen table is sufficient to detect duplicates (in our comparison which we present in Figure 2.4, we use a conservative value of ten entries per active route). In addition to the storage overhead of the routing table and seen table, neighbor table overhead of $3n$ is required for every node, where n is the average number of neighbors. Note that this is a constant overhead which is incurred even if there is no traffic in the network. In DSR, storage overhead is zero if there is zero traffic.

In Distributed Bellman-Ford, the table overhead of each network node is $3N$, independent of traffic.[3] This overhead is higher than on-demand routing (ABR or DSR) in light traffic but lower in heavy traffic. In Figure 2.4, we show the storage overhead required by each node for varying number of active routes in a network with 50 hosts. We can see that the storage overhead of ABR is higher than DSR, especially if the number of active routes increases. We can also see that DBF requires more storage overhead than on-demand protocols in light traffic.

2.3.4.2 Low Detection/Interception Probability

In some battlefield applications, if no packet needs to be transmitted, nodes should preferably remain silent (sleep mode) to reduce detection/interception probability. ABR sends beacon messages periodically, and this beacon may be received by an unintended receiver (e.g., an enemy).[4] Similarly, in DBF nodes continuously emit update packets, which can be detected or intercepted. DSR on the other hand does not transmit anything if there is no user data to send. Thus

[3]DBF stores destination, distance, and next hop node for each route, thus making it $3N$ for each node.

[4]Using advanced radio modulation techniques, beacons may appear as noise for other radio detection systems.

DSR has a better LDP/LIP property.

2.3.4.3 Low Power Operation

In situations where there is no data traffic, ABR demands more power in order to process beacon messages, and so does DBF to transmit/process updates. Thus, DSR is more attractive when power resource conservation is of paramount concern. However, this deficiency in ABR can be offset by current power conservation techniques in devices, protocols, and operating systems.

2.3.4.4 Problem with Shortest Path on Power Consumption

In routing protocols that use shortest hop or delay as route selection metric, some nodes need to support many routes (i.e., have high route relaying load). These nodes continuously consume energy and their energy will eventually be exhausted, resulting in node failures. Route selection should also consider energy reserves as one of the factor [146]. ABR uses 'route relaying load' as one of its metric and prevents node failures of this kind. However, this is not the case for DSR.

2.4 Conclusion

Many routing protocols for ad hoc mobile wireless networks have been proposed in recent years. In this chapter, we have reviewed and studied key properties of three distinctive routing protocols. Performance evaluation of these protocols have been conducted via simulation in a common network environment. We have compared the performance of Associativity-Based Routing with Distributed Bellman-Ford and Dynamic Source Routing. Simulation results reveal that the

DBF incurs extensive bandwidth and computation overhead in the presence of mobility, yielding inferior performance when compared to on-demand routing protocols (ABR and DSR) in ad hoc networks. We also report that ABR has a better throughput, smaller delay, and lower control overhead than DSR. Chiefly, this is due to the use of innovative associativity criterion, multiple route selection metrics, and local route recovery. On the negative side, ABR exhibits a slightly higher storage overhead than DSR. It is also more prone to detection and interception (by the enemy).

In summary, ABR is a strong candidate for the multihop mobile wireless environment along with DSR. The final selection of the on-demand routing scheme should take into account other considerations in addition to the measures provided by simulation.

CHAPTER 3

Performance Evaluation of Advanced Routing Strategies

In this chapter we investigate the performance of routing strategies in ad hoc networks. Routing protocols for ad hoc networks have adopted a variety of approaches. These protocols can be generally classified as: (a) distance vector based; (b) link state based; (c) on-demand; and (d) location based. The first two categories modify a traditional table-driven scheme to adapt to ad hoc networks. On-demand, or reactive, routing protocols are proposed specifically for ad hoc networks. These protocols do not maintain permanent route tables. Instead, routes are built by the source on demand. With the advent of GPS (Global Positioning System) [72], protocols that utilize location information to establish routes have been proposed. In this chapter, we conduct a performance study of routing protocols that represent each routing category. The distance vector based protocol WRP [114], the link state based protocol FSR [123], the on-demand routing protocol DSR [69], the location based reactive protocol LAR [78], and the location based proactive protocol DREAM [14] are simulated in a common wireless network simulation platform. In addition to routing protocols, we implemented a detailed and realistic model of the physical layer and medium access control protocols.

Related works [23, 38, 68, 87] that also performed comparative evaluation of

ad hoc routing protocols can be found in the literature. However, these articles compared/ranked protocols that are similar in style (e.g., on-demand) and used only a single mobility model. These papers evaluate the single class of protocols using performance metrics such as throughput and pure control overhead that only show the *effectiveness* of the protocol. In this section, we investigate performance of protocols from different categories under various network scenarios (e.g., different mobility patterns, mobility rates, traffic patterns, etc.). We also apply metrics that show the *efficiency* in addition to the *effectiveness* of the protocols. Understanding the protocol's efficiency gives us the ability to study and discuss relative strengths, weaknesses, and applications to various situations of each routing protocol. The ultimate purpose is not to rank the protocols, but to find which routing strategy is best for which environment.

3.1 Protocols Review

3.1.1 Wireless Routing Protocol

Wireless Routing Protocol (WRP) [114] is a distance vector based protocol designed for ad hoc networks. WRP modifies and enhances distance vector routing in the following three ways. First, when there are no link changes, WRP periodically exchanges a simple HELLO packet rather than exchanging the whole route table. If topology changes are perceived, only the 'path-vector tuples' that reflect the updates are sent. These path-vector tuples contain the destination, distance, and the predecessor (second-to-last-hop) node ID. Second, to improve reliability in delivering update messages, every neighbor is required to send acknowledgments for update packets received. Retransmissions are sent if no positive acknowledgments are received within the timeout period. Third, the

Table 3.1: Parameter values for WRP.

HELLO interval	1 sec
Max allowed HELLO miss	4
Update acknowledgment timeout interval	1 sec
Retransmission counter	4
Retransmission timer	1 sec

predecessor node ID information allows the protocol to recursively calculate the entire path from source to destination. With this information, WRP substantially reduces looping situations, speeds up the convergence, and is less prone to the "count-to-infinity" problem. Still, temporary loops do exist and update messages are triggered frequently in networks with highly mobile hosts.

Table 3.1 shows the WRP parameter values used in our experiments. Values suggested by the designers of WRP and specified in [114] were used for the most part. Only a couple of values were modified to maximize WRP performance in our simulation environment. We set the timer values so as to send more frequent connectivity updates, but less frequent retransmissions than suggested. The former modification was required by the high mobility speed on our experiments, and the latter is due to the fact that under the MAC protocol we implemented (to be described in detail in Section 3.2.2), retransmitting at twice the round trip time would flood the MAC buffer as well as cause unnecessary collisions with cross traffic in the channel.

3.1.2 Fisheye State Routing

Fisheye State Routing (FSR) [123] is a link state type protocol which maintains a topology map at each node. To reduce the overhead incurred by control packets,

Table 3.2: Parameter values for FSR.

Scope		1 hop
HELLO interval	speed $\leq$ 3.5 km/hr	5 secs
	speed $>$ 3.5 km/hr	1 sec
Max allowed HELLO miss		3
INTRASCOPE UPDATE interval	speed $\leq$ 3.5 km/hr	5 secs
	speed $>$ 3.5 km/hr	1 sec
INTERSCOPE UPDATE interval	speed $\leq$ 3.5 km/hr	15 secs
	speed $>$ 3.5 km/hr	3 secs

FSR modifies the link state algorithm in the following three ways. First, link state packets are not flooded. Instead, only neighboring nodes exchange the link state information. Second, the link state exchange is only time-triggered, not event-triggered. Third, instead of transmitting the entire link state information at each iteration, FSR uses different exchange intervals for different entries in the table. To be precise, entries corresponding to nodes that are nearby (within a predefined *scope*) are propagated to the neighbors more frequently than entries of nodes that are far away. These modifications reduce the control packet size and the frequency of transmissions. As a result, FSR scales well to large network size since link state exchange overhead is kept low. As mobility increases, however, routes to remote destinations may become less accurate.

Simulation parameter values for FSR are shown in Table 3.2.

3.1.3 Dynamic Source Routing

Dynamic Source Routing (DSR) [69] is an on-demand routing protocol that builds routes only when necessary. A source floods a ROUTE REQUEST if data to send

Table 3.3: Parameter values for DSR.

Time between retransmitted ROUTE REQUESTS	500 msecs
Max time where the same requests can be sent	10 secs
Nonpropagating ROUTE REQUEST timeout	30 msecs

exist but no route to its destination is known. The ROUTE REQUEST packet records in its header the IDs of the nodes it traverses. When the ROUTE REQUEST is received by the destination or a node that knows a route to the destination, a ROUTE REPLY is sent to the source via the recorded route. Each node in the network maintains a route cache storing routes it has learned over time. Aggressive caching helps minimizing the cost incurred by the route discovery process. DSR uses source routing instead of hop-by-hop routing; the source node appends the list of node IDs that comprise the route in the data header. When a node learns the route is obsolete due to topology changes, it builds and sends a ROUTE ERROR to the source. The source then invokes a route discovery process to construct a new route. No periodic message of any kind are required in DSR.

Table 3.3 shows the DSR parameter values used in our implementation. We implemented some optimization features of DSR (explanations and details of DSR optimization can be found in [105]): nonpropagating route requests, replying from cache, salvaging, tapping, and updating shorter routes.

3.1.4 Location-Aided Routing

Location-Aided Routing (LAR) [78] is an on-demand routing protocol which exploits location information. In fact, LAR operates very similarly to DSR. The major difference between the two protocols is that LAR uses location informa-

Table 3.4: A parameter value for LAR.

Timeout to send ordinary flooding request when no reply is received	2 secs

tion obtained from GPS to restrict the flooded area of ROUTE REQUEST packets. There are two schemes to determine which nodes propagate ROUTE REQUESTS. In scheme 1, the source defines a circular area in which the destination may be located. The position and size of the circle is decided with the following information: (a) the destination location known to the source; (b) the time instant when the destination was located at that position; and (c) the average moving speed of the destination. The smallest rectangular area that includes this circle and the source is the *request zone*. This information is attached to a ROUTE REQUEST by the source and only nodes inside the request zone propagate the packet. In scheme 2, the source calculates the distance between the destination and itself. This distance, along with the destination location known to the source, is included in a ROUTE REQUEST and sent to neighbors. When nodes receive this packet, they compute their distance to the destination, and continue to relay the packet only if their distance to destination is less than or equal to the distance indicated by the packet. When forwarding the packet, the node updates the distance field with its distance to the destination. In both schemes, if no ROUTE REPLY is received within the timeout period, the source retransmits a ROUTE REQUEST via pure flooding.

A parameter setting for LAR is shown in Table 3.4. We implemented LAR as specified in [78] and no DSR optimization features were included in LAR. The results shown for LAR in this chapter are those of scheme 1. Both schemes were implemented and scheme 1 gave a slight better performance in our simulations.

3.1.5 Distance Routing Effect Algorithm for Mobility

Distance Routing Effect Algorithm for Mobility (DREAM) [14] is another location based routing protocol. In contrast to LAR, DREAM is a proactive scheme (i.e., it maintains permanent routing tables). The scheme partially floods data to nodes in the direction of the destination. In the route table, *coordinates* of each node are recorded instead of route vectors. Each node in the network periodically exchanges control messages to inform all other nodes in the network of its location. *Distance effect* is achieved by assigning "TTL (Time-To-Live)" value to location control messages. Location updates with low TTL value (*short-lived* updates) are sent more frequently to packets with high TTL value (*long-lived* updates). In addition, DREAM adjusts to network dynamics by controlling update frequency based on movement speed. When sending data, if the source has "fresh enough" location information of the destination, it selects a set of one hop neighbors that are located in the direction from source to destination. If no such nodes are found, the data is flooded to the entire network. If such nodes exist, the list is enclosed in the data header and transmitted. Only nodes specified in the header are qualified to receive and process the packet. These nodes in turn select their own list of possible next hops and forward the packet with such updated list. If no neighbors are located in the direction of the destination, the packet is simply dropped. When the destination receives data, it sends ACKs back to the source in a similar fashion. However, ACKs are not transmitted when data was received via flooding. When the source sends data with designated next hops, (i.e., not by pure flooding), it starts a timer. If no ACK is received before the timer expires, the data is retransmitted by ordinary flooding.

Table 3.5 shows the parameter values for DREAM used in our experiments. After a few experiments, we decided to remove the ACK procedure of DREAM.

Table 3.5: Parameter values for DREAM.

Short-lived update interval	speed $<$ 10km/hr	45 secs
	10km/hr $\leq$ speed $<$ 30km/hr	35 secs
	speed $\geq$ 30km/hr	25 secs
TTL of short-lived updates		200 meters
Ratio of short-lived and long-lived updates sent		10 : 1
Min flooding angle towards the direction of destination		40 degrees

There were situations where data packets reached destinations but ACKs for those packets failed to get back to sources, thus invoking unnecessary flooding. In addition, transmission of ACKs congested the network to a great degree, yielding poor performance.

3.1.6 Routing Protocols Summary

Table 3.6 summarizes key characteristics and properties of the protocols we simulated.

3.2 Simulation Model and Methodology

The simulator for evaluating routing protocols was implemented within the GloMoSim library [160]. The GloMoSim library is a scalable simulation environment for wireless network systems using the parallel discrete-event simulation capability provided by PARSEC [10]. Our simulation modeled a network of 50 mobile hosts placed randomly within a 750 meter $\times$ 750 meter area. Radio propagation range for each node was 200 meters and channel capacity was 2 Mb/s. There were no network partitions throughout the simulation. Each simulation executed for

Table 3.6: Summary of protocols.

Protocols	**WRP**	**FSR**	**DSR**	**LAR**	**DREAM**
Routing Strategy	Distance Vector	Link State	On-Demand	Location Based (reactive)	Location Based (proactive)
Selection Metric	Shortest Path	Shortest Path	Shortest Path	Shortest Path, Location	Shortest Path, Location
Loop-Free	No	Yes	Yes	Yes	Yes
Periodic Messages	HELLOS	HELLOS, Route Entries	None	None	Location Packets
Updates Triggered by	Event, Time	Time	Event	Event	Time
Flooding Packets	None	None	RREQs	RREQs	Location Packets, Data
Routes in Data	No	No	Source Route	Source Route	Next Hop Nodes
Promiscuous Mode	No	No	Yes	No	No
Need for GPS	No	No	No	Yes	Yes

600 seconds of simulation time. Multiple runs with different seed numbers were conducted for each scenario and collected data was averaged over those runs.

3.2.1 Channel and Radio Model

A free space propagation model [135] with a threshold cutoff was used in our experiments. In the free space model, the power of a signal attenuates as $1/d^2$ where d is the distance between radios. In addition to the free space channel model, we also implemented SIRCIM (Simulation of Indoor Radio Channel Impulse-response Models) [136] which considers multipath fading, shadowing, barriers, foliages, etc. SIRCIM is more accurate than the free space model, but we decided against using SIRCIM in our study because: (a) the complexity of SIRCIM increases simulation time by three orders of magnitude; (b) the accuracy of the channel model does not affect the relative ranking of the routing protocols evaluated in this study; and (c) SIRCIM must be *tuned* to the characteristics of the physical environment (e.g., furniture, partitions, etc.), thus requiring a much more specific scenario than we are assuming in our experiments.

In the radio model, we assume the ability of a radio to lock onto a sufficiently strong signal in the presence of interfering signals, i.e., radio capture. If the capture ratio (the minimum ratio of an arriving packet's signal strength relative to those of other colliding packets) [135] is greater than the predefined threshold value, the arriving packet is received while other interfering packets are dropped.

3.2.2 Medium Access Control Protocol

The IEEE 802.11 MAC protocol with Distributed Coordination Function (DCF) [60] is used as the MAC layer in our experiments. DCF is the basic access method used by mobiles to share the wireless channel under independent ad

hoc configuration. The access scheme is Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) with acknowledgments. Optionally, the nodes can make use of Request To Send/Clear To Send (RTS/CTS) channel reservation control frames for unicast, virtual carrier sense, and fragmentation of packets larger than a given threshold. By setting timers based upon the reservations in RTS/CTS packets, the virtual carrier sense augments the physical carrier sense in determining when mobile nodes perceive that the medium is busy. Fragmentation is useful in the presence of high bit error and loss rates, as it reduces the size of the data units that need to be retransmitted.

In our experiments, we employed RTS/CTS and virtual carrier sense. We chose this configuration to minimize the frequency and deleterious effects of collisions over the wireless medium. We did not employ fragmentation because our data packets were small enough that the additional overhead would reduce overall network throughput.

3.2.3 Traffic Pattern

A traffic generator was developed to simulate constant bit rate sources. The size of data payload is 512 bytes. We have chosen this value because smaller payload sizes penalize protocols that append source routes to each data packet. Ten data sessions with randomly selected sources and destinations were simulated. Each source transmits data packets at a rate between 0.5 packet/sec, up to 4 packet/sec. In Section 3.3.6, we vary the traffic load by changing the number of data sessions and examine its effect on routing protocols.

3.2.4 Mobility Pattern

We implemented two different mobility patterns. The random waypoint model [23, 69] was used in the results shown from Sections 3.3.1 to 3.3.6. In this model, a node selects a destination randomly within the terrain range and moves towards that destination at a predefined speed. Once the node arrives at the destination, it stays at its current position for a pause time of 10 seconds. After being stationary for the pause time, it selects another destination randomly and migrates towards it, staying there for 10 seconds, and so forth. Mobility speed varies from 0 km/hr to 72 km/hr across the range of experiments. Note that the stationary period is not considered in computing node speed. The results presented in Section 3.3.7 are obtained by using a group mobility model [133]. The details of the model will be described in the corresponding section.

3.2.5 Metrics

We have used the following metrics in comparing protocol performance. Some of these metrics were suggested by the MANET working group for routing protocol evaluation [37]. The metrics are chosen to evaluate the efficiency in addition to the effectiveness of the protocols.

- **Packet delivery ratio**: The ratio of data packets delivered to the destinations and data packets originated by the sources. This number presents the routing effectiveness of a protocol.

- **Hop distance**: Average number of hops traveled by data packets that reached their destinations. One might argue that a low hop count indicates effectiveness of route selection. This argument is true when different routing protocols have the same packet delivery ratio. However, if routing protocols

give different ratios (especially in networks with high mobility rates and link changes), hop count is closely related to the packet delivery ratio. Namely, the higher the delivery rate, the higher the hop count. Since only data packets that survive all the way to destinations are reflected, low hop count means that most of the data packets delivered are destined for nearby nodes, and packets sent to remote hosts are likely dropped. Thus, the hop count measure provides us with information about the survivability of the protocols.

- **Number of data packets transmitted per data packet delivered**: One should not confuse this measure with average hop count. 'Data packets transmitted' is the count of every transmission of data by each node. This count includes transmissions of packets that are eventually dropped and retransmitted by intermediate nodes. Since we divide this figure by the number of packets delivered to the destinations, this measure can be viewed as the efficiency of delivering data [37].

- **Number of control bytes transmitted per data byte delivered**: In place of using a pure control overhead, we chose to use a ratio of control bytes transmitted to data byte delivered to investigate how efficiently control packets are utilized in delivering data. Note that not only bytes of control packets (i.e., route tables, route update vectors, hellos, location updates, etc.), but also bytes of data packet headers (including source routes) are included in the number of control bytes transmitted. Accordingly, only bytes of the data payload contribute to the data bytes delivered.

- **Number of control and data packets transmitted per data packet delivered**: This measure shows the efficiency in terms of channel access [37]. This efficiency is very important in ad hoc networks since link

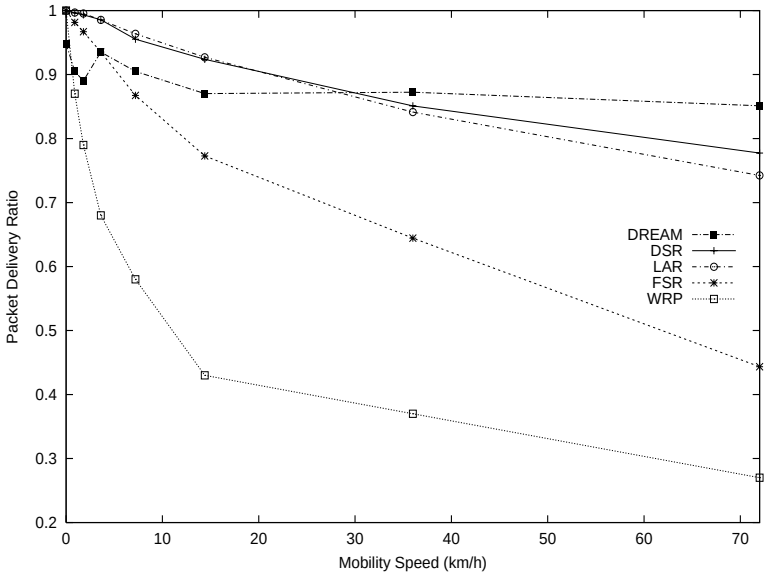


Figure 3.1: Packet delivery ratio as a function of mobility speed.

layer protocols are typically contention-based.

3.3 Simulation Results

3.3.1 Packet Delivery Ratio

Figure 3.1 highlights the packet delivery ratio of five protocols. All protocols perform well under low mobility rates, but they become less effective as the mobility speed increases. DREAM is the most robust to mobility. This robustness is due to the partial flooding of data. With this flooding, multiple packets can reach the destination via different paths. Utilizing location information, this flooded area is confined to reduce network congestion. However, flooding did induce increased congestion, contention, and collisions, causing DREAM to be the only protocol that did not successfully deliver all packets in the absence of mobility.

On-demand routing protocols (DSR and LAR) have very high packet delivery ratios overall, especially when subject to relatively low mobility. We observe only a slight performance degradation with mobility. In highly mobile situations, routes taken by ROUTE REQUESTS may already be broken when the source sends data or even when ROUTE REPLIES are being returned back to the source. Thus, we find that the delay resulting from discovering routes plays an important role in the performance degradation at high mobility speed. Since LAR is an improvement of basic DSR, one might wonder why LAR does not perform much better than DSR . However, remember that DSR has several optimization features that are not implemented in LAR. In addition, the location information used by LAR may be out-of-date when nodes move at high speeds.

FSR was sensitive to mobility. Update messages in FSR are time-triggered only, i.e., there are no event-triggered updates. Additionally, routes to remote destinations become less accurate as mobility increases. As a result, some of the link state information maintained in route tables is imprecise. Shortening the periodic update interval may resolve this problem, but at the cost of excessive routing overhead.

WRP showed less effectiveness when compared to other protocols, especially at high mobility rates. As nodes move faster, link connectivity changes more often and more update messages are triggered. For each triggered update, neighboring nodes are required to send back an acknowledgment, and this adds to control overhead. Moreover, temporary loops were being formed because the network view converged slowly, with many changes needing to be absorbed and propagated. Loops, triggered updates, and ACKs created an enormous amount of packets, contributing further to collisions, congestion, contention, and packet drops.

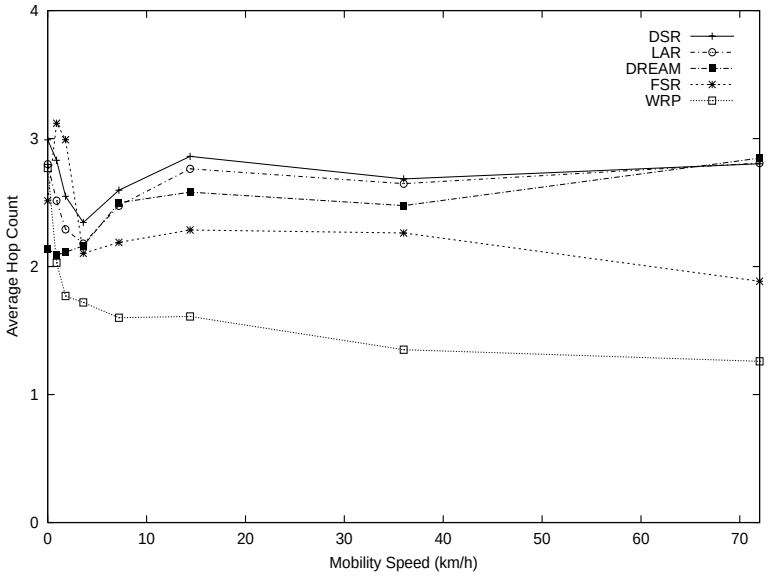


Figure 3.2: Hop count as a function of mobility speed.

3.3.2 Hop Distance

As mentioned in Section 3.2.5, average hop count only accounts for data packets that *survive* to destinations. As expected, Figure 3.2 reveals that protocols that delivered more data packets (as was shown in Figure 3.1) have higher average hop count. If the distance between source and destination is greater, the number of intermediate nodes that data packets need to visit increases. The likelihood of a packet being dropped becomes greater as packets are required to traverse many links, particularly if network topology changes often. Thus, if a routing protocol cannot handle connectivity changes rapidly, more data packets get dropped.

3.3.3 Number of Data Packets Transmitted per Data Packet Delivered

The average number of data transmissions per data packet delivery for each protocol is shown in Figure 3.3. As expected, DREAM has the highest measure since

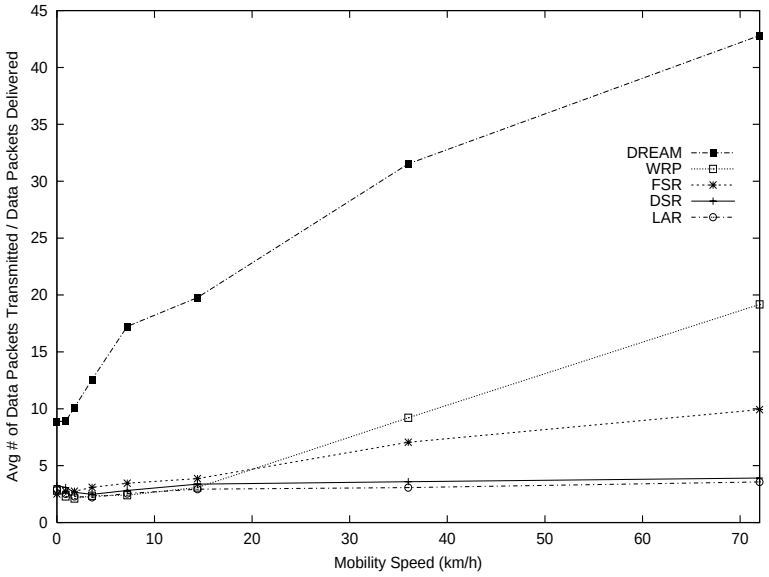


Figure 3.3: Number of data packets transmitted per data packet delivered as a function of mobility speed.

it partially floods data while other protocols unicast data. The value increases with mobility because sources are more likely to send data by pure flooding. The values of WRP and FSR increase with mobility as well and these increases stem from packet drops by intermediate nodes.

It is interesting to compare Figure 3.2 and Figure 3.3. With the exception of DREAM, the difference between the two measures indicates the number of packet drops and retransmissions per single data delivery. We can observe that there are only minor differences between the two measures for on-demand protocols. On-demand protocols were able to deliver data packets without much wasted data transmissions. DSR, in particular, has an optimization feature called *salvaging* where the node detecting a route break salvages the data by sending it through another route to the destination, via a path it already knows (i.e., stored in route cache). Hence, data packets are dropped much less frequently when compared to proactive schemes. Proactive schemes (WRP and FSR) suffer from a large

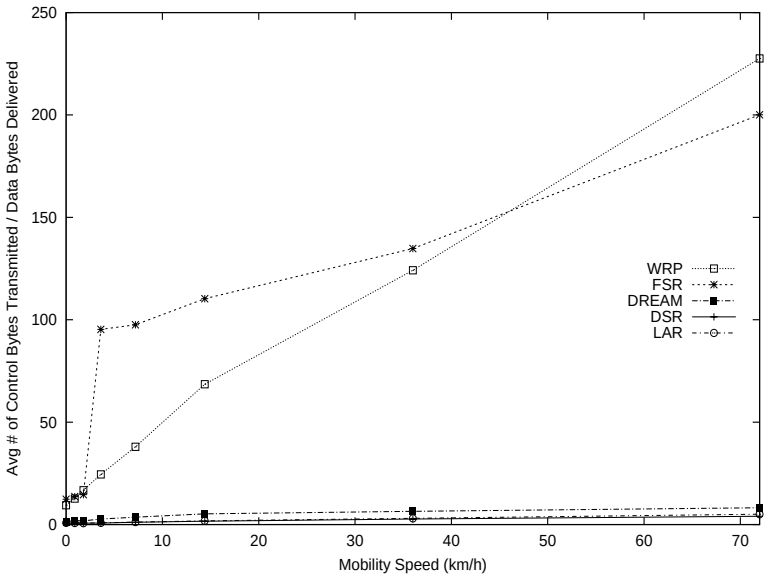


Figure 3.4: Number of control bytes transmitted per data byte delivered as a function of mobility speed.

difference that grows with mobility speed. This observation confirms that WRP and FSR have numerous packet drops in highly dynamic networks.

3.3.4 Number of Control Bytes Transmitted per Data Byte Delivered

Figure 3.4 shows the efficiency of control overhead utilized in data delivery. The graph demonstrates that proactive protocols with periodic messages (e.g., HELLOS, route entries) have high comparative overhead. In WRP, each node sends acknowledgments for each HELLO it receives. Additionally, route update entries are produced more frequently in high mobility where there are many link changes. As the WRP path vector has an extra field (next-to-last-hop node), control byte overhead actually becomes larger than that of a basic distance vector algorithm when the mobility rate is high. In FSR, route update messages are sent periodically only, thus the pure control overhead value does not increase. However,

recall that in our measure, control byte overhead is divided by data bytes delivered. FSR delivered less data in high mobility cases. We can also observe a sudden jump in the FSR plot. The point of sharp increase represents the point when the update interval is adjusted to node movement speed. DREAM shows a very low control overhead in the figure because the size of location information packets is small. If our implementation used ACK procedure, where ACKs are partially flooded in a similar manner to data, the value would be much higher. DSR and LAR have the least control traffic because they have no periodic messages and send control packets only when necessary. Link changes that are not part of existing data session routes are not updated in DSR and LAR while proactive protocols still send this information. In other words, control packets in on-demand protocols are used efficiently. The two on-demand protocols have almost equal overhead. Although LAR sends Route Requests to a limited area, extra overhead is produced by attaching location information in Route Requests and Route Replies and that evens out the difference.

3.3.5 Number of Total Packets Transmitted per Data Packet Delivered

The average number of data and control packets transmitted per data packet delivered is shown in Figure 3.5. We believe this measure is particularly significant in ad hoc networks since most link layer protocols are contention-based. The graph is nearly identical to Figure 3.4 except for the vertical scale and higher values of DREAM. Remember that Figure 3.4 accounts for transmitted *bytes* of control packets only while Figure 3.5 accounts for the *number* of all packets transmitted. As mentioned above, data flooding accounts for higher values of DREAM. As expected, on-demand routing protocols show much lower values

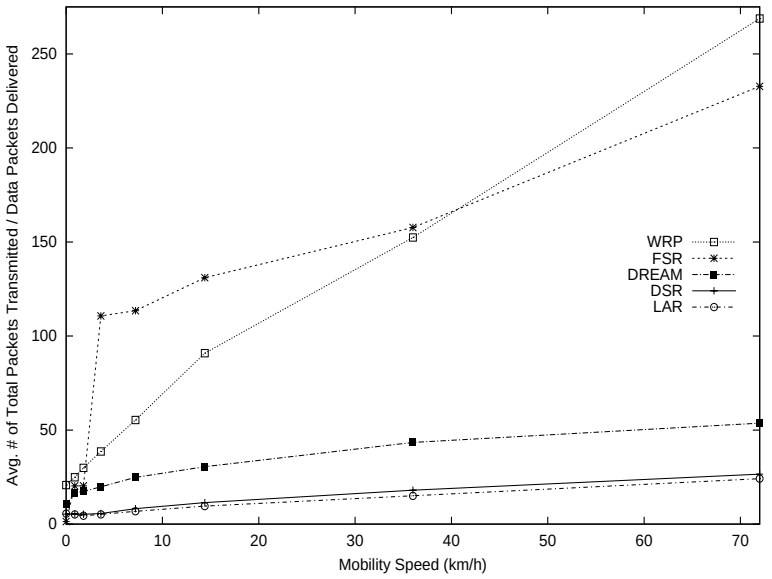


Figure 3.5: Number of total packets transmitted per data packet delivered as a function of mobility speed.

compared to those of other protocols. Although the difference is very small, LAR has less packets transmitted than DSR. Restricting the propagation of ROUTE REQUESTS using location information accounts for the difference.

3.3.6 Effect of Traffic Load

In this section, we vary the number of data sessions while keeping the packet rate for each session constant. The mobility rate was set constant at 1 m/s. Figure 3.6 and Figure 3.7 reveal the packet delivery ratio and the average number of total packets transmitted per data packet delivered, respectively. Only DREAM and WRP suffer a packet delivery ratio drop with increase in the number of data sessions. Since data packets of DREAM are partially flooded, having many sessions increases the amount of flooded packets resulting in contention, collisions, and congestion. As for WRP, due to the random waypoint mobility, the routing algo-

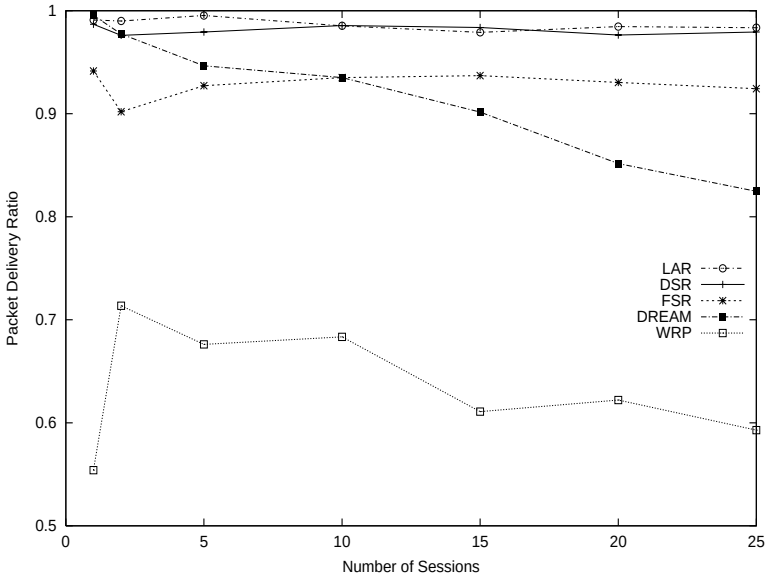


Figure 3.6: Packet delivery ratio as a function of number of sessions.

rithm is in a constant state of reconciling its tables to the perceived link changes, and propagating those changes across the network. Because of the method by which WRP reduces loops and invalid paths, there is a significant percentage of destinations that are temporarily unreachable from a given node while these link updates are being propagated. The effect of these temporarily unreachable destinations becomes increasingly noticeable with a larger number of sessions, as packets are dropped by the source or intermediate nodes with invalid routing table entries to a given destination.

When increasing the number of sessions, the number of total packets transmitted per data packet delivered decreases for proactive schemes while they remain nearly constant for on-demand schemes. FSR and DREAM send periodic updates and the number of update transmissions remain the same regardless of number of data sessions. WRP sends event triggered updates, but since the mobility rate is constant, having a different number of sessions does not affect the number of update transmissions. Meanwhile, the number of data packets received by desti-

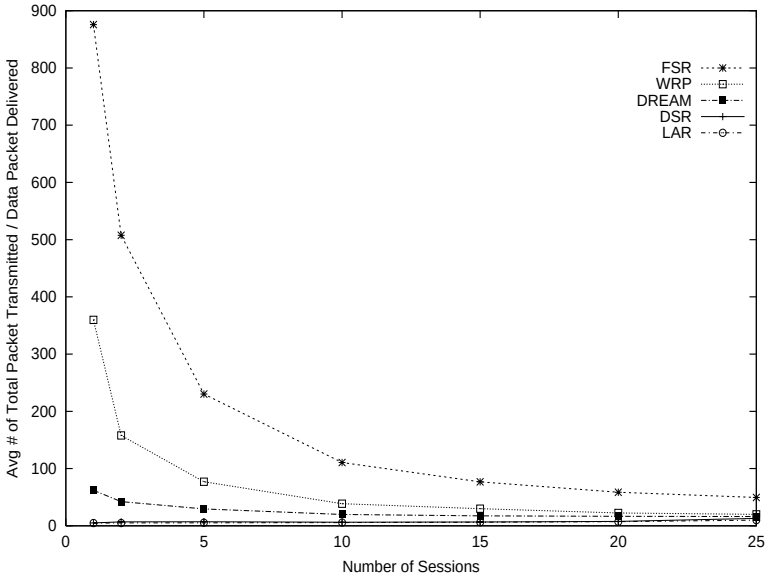


Figure 3.7: Number of total packets transmitted per data packet delivered as a function of number of sessions.

nations increases linearly with number of data sessions, resulting in the decrease of values. On-demand protocols, however, send more control packets when there are more data sessions. As the number of sessions increase, more route discovery and route maintenance procedures are executed. The increase of these control packets are in the same rate of that of data packets, and the measure remains almost constant.

3.3.7 Group Mobility Model

In certain ad hoc networking situations (e.g., troops moving in military situations, a number of students moving to the seminar room, etc.), network hosts form groups and nodes within the group move in a similar fashion. In this section, we model this *group mobility* in evaluating routing performances.

3.3.7.1 Mobility Description

In the model we implemented, nodes in a network form groups, and nodes in the same group are placed close to one another. Nodes within a group move in a similar direction and speed while each group may move differently from the others. Movement of each group and each node in a group can be characterized as Exponentially Correlated Random Mobility (ECRM) [133]. The model can be best described by the following equation:

$$b(t+1) = b(t)e^{1/\tau} + s\sigma\sqrt{1-e^{2/\tau}}r,$$

where

$b(t)$ is the position (r, θ) of a group or a node at time t,

τ is a time constant that regulates the rate of change,

σ is the variance that regulates the variance of change,

s is the speed of the node, and

r is a Gaussian random variable.

Variables τ and σ control the movement. We chose to use the same values for nodes within the group but different value for each group. There are 5 groups in our simulation, each with 10 nodes. One group is stationary and other four groups move in different directions. If nodes hit the boundary of our simulation terrain range, they are bounced back in the reverse direction (i.e., west to east, northeast to southwest, etc.). Mobile nodes move constantly; there is no pause time. The average node degree in the group mobility model was 10.52 while it was 10.24 in the random waypoint model.

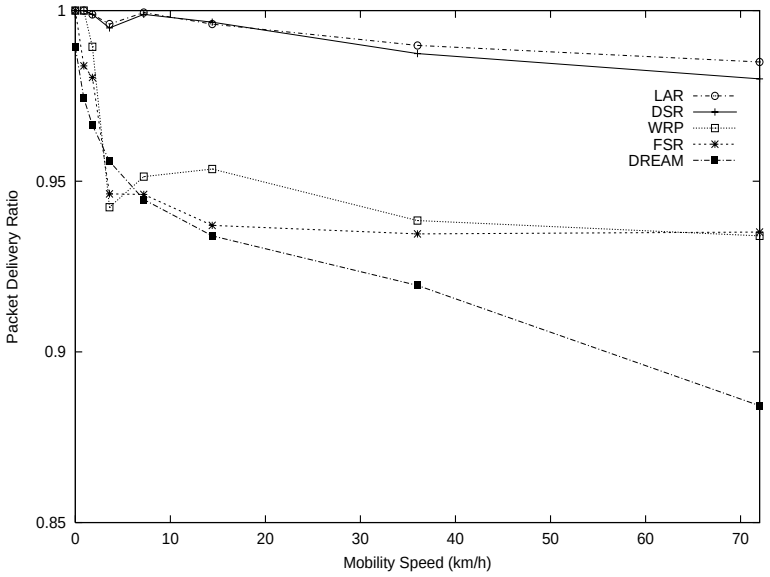


Figure 3.8: Packet delivery ratio as a function of group mobility speed.

3.3.7.2 Results

Figure 3.8 shows the packet delivery ratio of each protocol in the group ECRM model. All protocols are able to deliver more data packets successfully than in the random waypoint model. Notice the difference in the vertical scales between Figure 3.8 and Figure 3.1. WRP is the most improved protocol under the group mobility model. In the group ECRM model, nodes in the same group (i.e., immediate neighbors) move similarly and there are relatively few link changes. Even in highly mobile situations, route breaks occur much less frequently than in the random waypoint model. Few update packets are sent and the network view converges more quickly, thus improving WRP performance dramatically.

Although the packet delivery ratio improved, DREAM is the protocol which benefited the least from this model. The number of link changes and route breaks does not affect the number of control packet transmissions in DREAM and it has no performance influence in delivering partially flooded data. In other words,

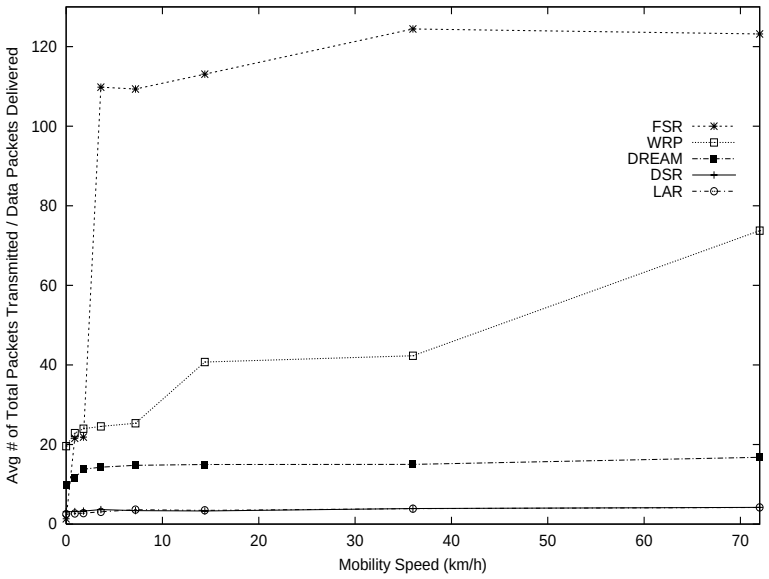


Figure 3.9: Number of total packets transmitted per data packet delivered as a function of group mobility speed.

DREAM is not only robust to mobility speed, it is also robust to movement pattern.

The average number of total packets transmitted per data packet delivered as a function of group mobility speed is shown in Figure 3.9. The measures also improved (i.e., decreased) when compared with those in the random waypoint model. Because protocols delivered more data, the efficiencies are enhanced accordingly.

3.4 Lessons Learned

Distance vector protocols work well in static networks. Since they maintain the full topology view all the time, distance vector type protocols are good choices when delivering real-time and heavy traffic. However, they do not scale well to large and highly mobile networks because they suffer from the *count-to-infinity*

problem, slow convergence, and excessive control overhead.

Link state algorithms are best suited for networks that require QoS (Quality of Service) guarantees because they provide link costs and capacities. Similar to distance vector protocols, however, link state protocols do not scale well to large networks and suffer from enormous amount of control overhead, especially in highly dynamic situations.

On-demand routing protocols produce less control traffic overhead than the above mentioned proactive schemes since no route tables are periodically exchanged. Control packets are generated only as needed, i.e., there are no control messages which are not utilized. Due to less overhead, they performed well in most of our simulation scenarios, even in highly mobile situations. However, extra delay (route acquisition latency) is required to obtain a route and this delay does not favor on-demand protocols when traffic needs to be delivered quickly (e.g., real-time traffic). Additionally, if the network has a large number of data sessions, the amount of control overhead grows to be comparable to those of proactive schemes. DSR, a typical on-demand scheme, has even less control overhead than other on-demand schemes (e.g., ABR [159], SSA [43]) since it does not exchange any 'hello' or 'beacon' messages. However, the drawback is that route breaks and link changes are detected only after data packets fail to go through the broken link, thus yielding longer delays. Intermediate nodes of a route in DSR need not maintain up-to-date route information since source routing is used, but additional overhead is introduced by listing the route in the data header.

With the appearance of GPS, protocols that utilize node location information in building routes have been recently proposed. With the knowledge of node position, routing can be more effective at the cost of overhead incurred by exchanging coordinates. In addition, location information recorded can be out-of-date and

these protocols cannot be applied to networks where nodes are not equipped with GPS. We studied two distinctive location-based protocols. LAR, a reactive approach, further reduces control traffic of DSR by restricting the propagation of flood packets. Even though LAR can diminish the number of control packet transmissions, more byte overhead is generated to exchange additional location information. DREAM is another location-based protocol, but in contrast to LAR, it is a proactive scheme. The key characteristic of DREAM is its partial flooding of data packets to nodes that are in the direction of the destination. Because of this partial flooding, multiple packets travel to destinations via different paths. The performance of DREAM was not greatly affected by the speed or movement pattern of network hosts. However, we saw a performance degradation when the number of sessions in the network increased. Even though flooding is resilient to mobility, it creates a lot of (duplicate) packets and increases the number of packets in the network as the number of sessions become larger. Congestion, collisions, and channel contention occur more frequently in those situations.

3.5 Conclusion

We have conducted a performance study of five protocols that represent various routing categories. Simulations were run under many diverse scenarios and each protocol showed competence in different situations. Overall, all protocols performed much better with the group mobility model than with the random waypoint model. WRP and FSR, especially, were the main beneficiaries of the group movement model. Each protocol's performance degraded as mobility rates increased, but DREAM was the most robust to the speed of network hosts. However, because of the data flooding, DREAM became less effective under heavy traffic scenarios. On-demand protocols were highly effective and efficient in most

of our scenarios. Extra delay in acquiring routes, though, make them less attractive in delivering real-time traffic. LAR further improved an on-demand protocol by using location information, but produced more overhead to exchange location information.

In summary, there is no single routing strategy that is best for all network situations. Every protocol has its advantages and disadvantages in different scenarios. The choice of a routing protocol should be made carefully after considering every aspect we provided in this chapter (and possibly more).

CHAPTER 4

Ad hoc Routing Protocol Scalability

As mobile networking continues to experience increasing popularity, the need to connect large numbers of wireless devices will become more prevalent. Many recent proposals for ad hoc routing have certain characteristics which may limit their scalability to large networks. This chapter proposes five different combinations of enhancements which may be incorporated into virtually any on-demand protocol in order to improve its scalability. The scalability of current on-demand routing protocols is evaluated through the selection of a representative from this class of protocols. The performance of the un-modified on-demand protocol is compared against that of it combined with each of the scalability modifications. Each scheme's behavior is analyzed in networks as large as 10,000 nodes through detailed simulation. Based on the observations, conclusions are drawn as to the expected scalability improvement which can be achieved by each enhancement.

4.1 Background and Motivation

Recent advances in the portability, power, and capabilities of wireless devices and applications have resulted in the proliferation and increased popularity of these devices. As the number of users continues to grow, wireless routing protocols will be required to scale to increasingly larger populations of nodes. Conference networking scenarios can require the formation of networks on the order of tens

to hundreds of nodes, while many military applications can involve thousands to tens of thousands of nodes. Furthermore, as the deployment of wireless networks becomes more widespread, new applications may encourage the formation of large ad hoc networks. For instance, sensor networks may include thousands of sensors which must be able to self-configure and establish routes. Similarly, military battlefield operations often require the formation of ad hoc networks containing hundreds to thousands of soldiers and personnel.

There have been many recent proposals of unicast routing protocols for ad hoc mobile networks [3, 69, 121, 122, 128, 148, 159]. Many of these papers include simulations of the protocols they describe, illustrating the performance of their protocol. To determine the relative merits and strengths of the various protocols, studies have been performed which simulate the protocols under various input conditions [23, 39, 68, 95]. While these simulations and studies are informative in evaluating the performance of the protocols for relatively small numbers of nodes (i.e., 50 nodes), they do not show how any of the protocols scales to larger node populations. The simulations presented in [128] and [122] evaluate the Ad hoc On-Demand Distance Vector (AODV) Routing protocol and the Zone Routing Protocol (ZRP), respectively, for networks as large as 1,000 nodes, and are simulations of some of the largest network sizes to date. Because ad hoc routing protocols could well be used in networks containing a large number of nodes, it is important to be able to know how these protocols will scale and perform in these scenarios.

Many of the proposed protocols for ad hoc networks [3, 69, 121, 128, 159] use a broadcast route discovery mechanism whereby a route request is flooded across the entire network. While the impact of such route discovery floods may be limited in small networks, the impact will be significantly larger for larger

networks. When a link break in an active route occurs, many of these protocols [69, 121, 128] require that an error notification be sent to nodes that were using that link. Again, for small networks with limited network diameters, this route error message can be propagated back to a source node relatively quickly, and some repair action can be taken. However, as the network diameter and average path length increases, the error message may have to propagate across tens of hops to reach the source node. For such large networks, or even smaller networks with rapidly moving nodes, it is likely that the source node will be unable to make a repair before another link in the route breaks. Hence, this mechanism may prove ineffective for more stressful scenarios.

There are other approaches to unicast routing in ad hoc mobile networks than those previously described which may prove more scalable; however, each of these methods also has its limitations. Clustering and hierarchical addressing methods have long been known for attempting to increase protocol scalability [13, 32, 57, 66, 81, 98, 133, 142]. Clustering protocols group nodes into *clusters* based on their proximity to each other. Each cluster generally has a cluster leader, which is the representative of the nodes in its cluster. The cluster leader typically participates in the network routing protocol, freeing the other network nodes of this burden. Routes in clustered networks may often be recorded hierarchically between clusters, as in [32]. These logical hierarchical paths may be longer lived than routes which utilize flat addressing, because any gateway between two clusters can be used to route between the clusters. This may result in fewer route reconstructions, and hence also reduce the number of on-demand control messages required to maintain the routes. Cluster-based protocols, however, have their drawbacks. They require periodic messaging from each network node in order to maintain the clusters. This periodic messaging results in higher processing and control packet overhead, as well as increased bandwidth utiliza-

tion and longer delays. Moreover, if the protocol constrains routes to traverse cluster-heads, longer path lengths will be required, which is another contributor to increased bandwidth utilization. Finally, there may be complications when the cluster leaders fail or give up their cluster leader status.

Instead of performing routing on-demand, other protocols have instead been based on modified versions of either the distance vector [104] or link state [111] routing algorithms. Because both distance vector and link state algorithms not only use periodic updates, but also triggered updates in the event of a change in link status, they are not well-suited for mobile networks. A network composed of moderately moving nodes will result in a high number of triggered updates, consuming bandwidth and making route convergence difficult, if not impossible. The protocols described in [18, 52, 70, 114, 123, 127, 150] each present a modified version of one of these protocols. For instance, [70], [123] and [150] each utilize a prioritized connectivity information exchange algorithm, whereby information about parts of the network more distant from the sending node is sent less frequently than information about neighboring nodes. [70] and [150] apply this technique to the distance vector algorithm, while [123] modifies the link state protocol. A similar approach [18] sends update information to only those nodes that actually need the information. These protocols have the benefit of a reduction in routing update overhead as compared with the basic link state and distance vector algorithms. However, they still have the drawback that they require updates based on node movement, which can result in a large amount of control overhead and bandwidth consumption in a mobile network.

A different approach to route finding is taken by the Core Extraction Distributed Ad Hoc Routing (CEDAR) algorithm [148]. CEDAR is an algorithm that builds a set of nodes (i.e., a *core*) to perform route computation. Using the

local state information, a minimum dominating set of the network is approximated to form the core. CEDAR establishes QoS routes that satisfy bandwidth requirements using the directionality of the core path. Link state and bandwidth availability is exchanged to maintain important information for computing QoS routes. Although CEDAR builds a core infrastructure that yields low overhead, the protocol is fairly complex and difficult to implement. The problem of calculating the minimum dominating set and the core is known to be NP-hard.

Finally, [3] and [122] are variations of on-demand routing protocols which attempt to increase scalability through other methods. The Relative Distance Micro-discovery Ad Hoc Routing (RDMAR) protocol allows for local repair of link breaks in active routes [3], and the ZRP protocol maintains route information to *all* nodes within a "zone" [122]. ZRP is a hybrid protocol which maintains the route information for the zone via a link state or distance vector protocol and then applies the on-demand technique communication for nodes outside the zone. These protocols may reduce the number of route discovery floods required by a source node by either repairing link breaks locally where they occur (RDMAR) or by maintaining routes to some destinations before they are actually needed (ZRP). Nevertheless, the protocols still suffer from the same disadvantage as the class of on-demand protocols whereby efficiency drops as the number of source-destination pairs increases, due to the likely requirement of a route discovery flood.

This chapter evaluates the scaling potential of on-demand ad hoc routing protocols by comparing a base routing protocol with the performance of it combined with various enhancements. The Ad hoc On-Demand Distance Vector (AODV) Routing protocol [128, 129] is used as a representative of on-demand routing protocols. AODV was chosen because it is currently one of the leading protocols

for routing in ad hoc mobile networks. The scalability of AODV is investigated by evaluating its performance in networks as large as 10,000 nodes. Then, three methods for improving the scalability of ad hoc routing protocols are described and integrated into the AODV protocol for their evaluation. The enhancements include an expanding ring search for route discoveries initiated by a source node, a query localization protocol (proposed in [24]) which also attempts to prevent the flooding of route requests, and the local repair of link breaks in active routes. Further, the methods for preventing discovery floods are each in turn combined with the local repair mechanism, to yield a total of five possible improvement algorithms. Each of these enhancement combinations is evaluated, through detailed simulation, in networks of up to 10,000 nodes, and compared with the results achieved by the un-modified AODV protocol.

4.2 Overview of the Routing Protocol

The routing protocol utilized for the scalability study is the Ad hoc On-Demand Distance Vector (AODV) protocol [128, 139]. AODV is an on-demand protocol which is capable of providing unicast, multicast, and broadcast communication. For the purposes of this study, its unicast operation is focused upon. Route discovery is based on a route request/route reply query cycle. Once discovered, a route is maintained as long as needed by the source. To guarantee loop freedom, AODV utilizes per node sequence numbers. A node increments the value of its sequence number whenever there is a change in its local connectivity information.

4.2.1 Route Discovery

Route discovery begins when a source node needs a route to some destination. It places the destination IP address and last known sequence number for that destination, as well as its own IP address and current sequence number, into a Route Request (RREQ). It then broadcasts the RREQ and sets a timer to wait for a reply.

When a node receives the RREQ, it first creates a *reverse route entry* for the source node in its route table. It then checks whether it has an unexpired route to the destination node. In order to respond to the RREQ, the node must either be the destination itself, or it must have an unexpired route to the destination whose corresponding sequence number is at least as great as that contained in the RREQ. If neither of these conditions are met, the node rebroadcasts the RREQ.

On the other hand, if it does meet either of these conditions, the node then creates a Route Reply (RREP) message. It places the current sequence number of the destination, as well as its distance in hops to the destination, into the RREP, and then unicasts this message back to the source. The node from which it received the RREQ is used as the next hop. When an intermediate node receives the RREP, it creates a *forward route entry* for the destination node in its route table, and then forwards the RREP to the source node. Once the source node receives the RREP, it can begin using the route to transmit data packets to the destination. If it later receives a RREP with a greater destination sequence number or equivalent sequence number and smaller hop count, it updates its route table entry and begins using the new route.

If the source node does not receive a RREP by the time its discovery timer expires, it rebroadcasts the RREQ. It attempts discovery up to some maximum number of times. If no route is discovered after the maximum number of attempts,

the session is aborted.

4.2.2 Route Maintenance

An active route is defined as a route which has recently been used to transmit data packets. Link breaks in non-active links do not trigger any protocol action. However, when a link break in an active route occurs, the node *upstream* of the break determines whether any of its neighbors use that link to reach the destination. If so, it creates a Route Error (RERR) packet. The RERR contains the IP address of each destination which is now unreachable, due to the link break. The RERR also contains the sequence number of each such destination, incremented by one. The node then broadcasts the packet and invalidates those routes in its route table.

When a neighboring node receives the RERR, it in turn invalidates each of the routes listed in the packet, *if* that route used the source of the RERR as a next hop. If one or more routes is deleted, it then goes through the same process, whereby it checks whether any of its neighbors route through it to reach the destinations. If so, it creates and broadcasts its own RERR message.

Once a source node receives the RERR, it invalidates the listed routes as described. If it determines it still needs any of the expired routes, it then re-initiates route discovery for that route.

4.3 Enhancements

The scalability of many on-demand routing protocols may be limited due to a couple of important factors. The first is the need for flooding each RREQ. In small networks, flooding the RREQ across the network has a limited impact due

to the small number of nodes in the network. However, as networks grow to thousands and tens of thousands of nodes, flooding the entire network each time a route needs to be discovered consumes significant processing power at each network node, as well as excessive bandwidth during the floods.

As path lengths increase and as node mobility speeds rise, breaks in active routes occur with increasing frequency. Requiring an error message to be sent to the source node for each link break may result in an overwhelming number of route repairs by the source node. Particularly for high mobility and/or long path lengths, it may be true that the source node barely has time to rediscover a route before that route suffers from another link break.

Because of these characteristics, on-demand routing protocols may not be able to scale well to networks of large numbers of nodes and high mobility. To improve their scalability, the following modifications are offered. The expanding ring search and query localization can be used to reduce the area searched during a route discovery, and hence prevent flooding of the network. The current Internet draft specification of AODV [129] recommends such an expanding ring search be used for route discoveries. Local repair can also be used to provide immediate patching of breaks in active routes. Finally, the expanding ring search and query localization can be combined with local repair to provide increased scalability in both of these domains.

4.3.1 Expanding Ring Search

An expanding ring search works by searching successively larger areas, centered around the source node, until a node with a route to the destination is located. The basic premise behind the expanding ring search is to find some local node with a route to the destination, and thereby avoid flooding the entire network in

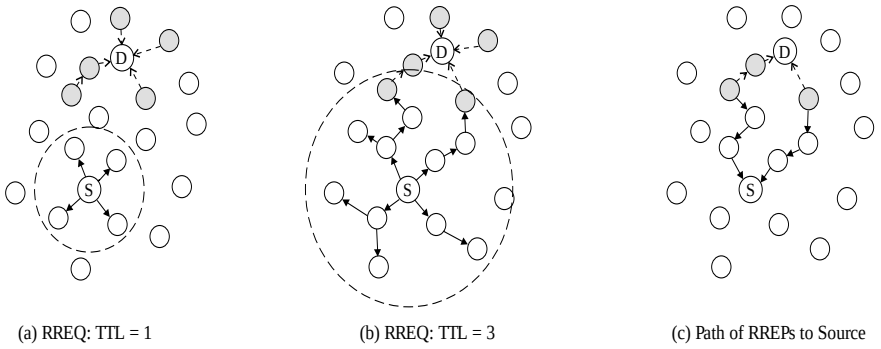


Figure 4.1: Example of an expanding ring search.

search of such a route. Using an expanding ring search, the initial RREQ has a small time to live (TTL), i.e., two hops. Each time the RREQ is rebroadcast, the sending node decrements the TTL. Once the TTL reaches zero, the RREQ is no longer forwarded. The source node waits the discovery period for a RREP to be returned. If it has not received a RREP by the end of the discovery time, it initiates a new RREQ with the TTL increased by an increment. This process continues until a threshold TTL value is reached. After this point, if no route has been located, the RREQ is flooded across the network. Figure 4.1 illustrates an example of an expanding ring search. In the figure, the shaded nodes indicate nodes which have a route to the destination. In a larger network with more nodes than that illustrated, the number of nodes undisturbed by the route query would likewise be greater.

To optimize the expanding ring search, the discovery time can be calculated so that it is proportional to the size of the area being searched. For instance,

$$\texttt{rte\_disc\_tmo} = 2 \times \text{TTL} \times \texttt{node\_traversal\_time}$$

results in the route discovery timeout being directly proportional to the TTL used for that discovery. Here the node traversal time is an approximation of the time required by the node to process and transmit a packet.

When re-discovering a route after a link break, the source places the last known hop count to the destination in the TTL field of the RREQ. If no route is found in this attempt, the TTL is increased by the TTL increment value. The TTL is increased on each subsequent route discovery attempt until the TTL threshold is reached. After this point, the RREQ is simply flooded to the entire network.

Utilizing the expanding ring search, a tradeoff exists between both the latency in finding the route (if it is not located on the first attempt) and the number of times local nodes receive the RREQ, and the drawback of flooding the entire network.

4.3.2 Query Localization

The query localization technique was developed by Castaneda and Das and described in [24]. Query localization is a method by which the flooding of the RREQ is restricted to some area that is based on the previously known route to the destination node. Hence the RREQ is not actually flooded at all, but instead is limited to a specific region of the network. [24] presents two different techniques for performing query localization. For the purposes of this chapter, method 2 (*Exploiting node locality*) is selected. This method assumes that the destination has traveled a bounded distance from its previous location, and hence can be found within some small number of hops from the most recently used route to it. To enable query localization, a counter is placed in the RREQ packet. Whenever a node that was not on the previous route to the destination receives the RREQ, it increments the counter. Conversely, when a node that was previously on the route to the destination receives the RREQ, it resets the counter to zero. Once the counter exceeds the threshold value κ, the RREQ is dropped.

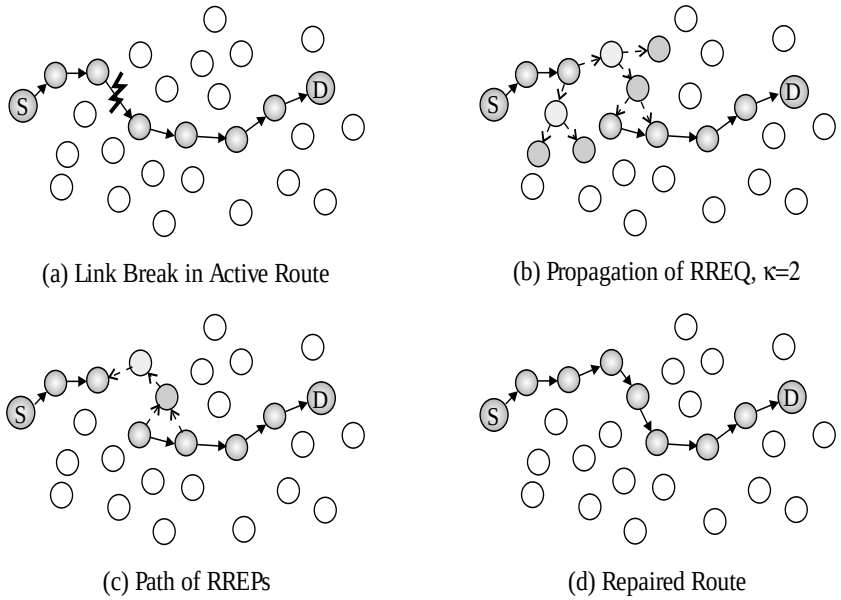


Figure 4.2: Example of query localization.

On the initial route discovery for a destination, κ is set to the network diameter, so that the RREQ traverses the entire network. For a route repair, however, κ is initialized to a small value, i.e., two. If a route to the destination is not located on the first attempt, the value of κ may be increased until some maximum value is reached. Figure 4.2 illustrates an example of query localization. In the figure the last known route between the source and destination is highlighted. On the repair route discovery, κ is set to two. The shading of the nodes indicates their distance off of the previously known route to the destination. As is evident from the figure, many of the network nodes do not need to receive the RREQ, and the query is able to be contained.

As with the expanding ring search, the drawback of the query localization protocol occurs when a route to the destination is not located on the first attempt. This results in certain nodes being repeatedly queried, as well as an increased route acquisition latency.

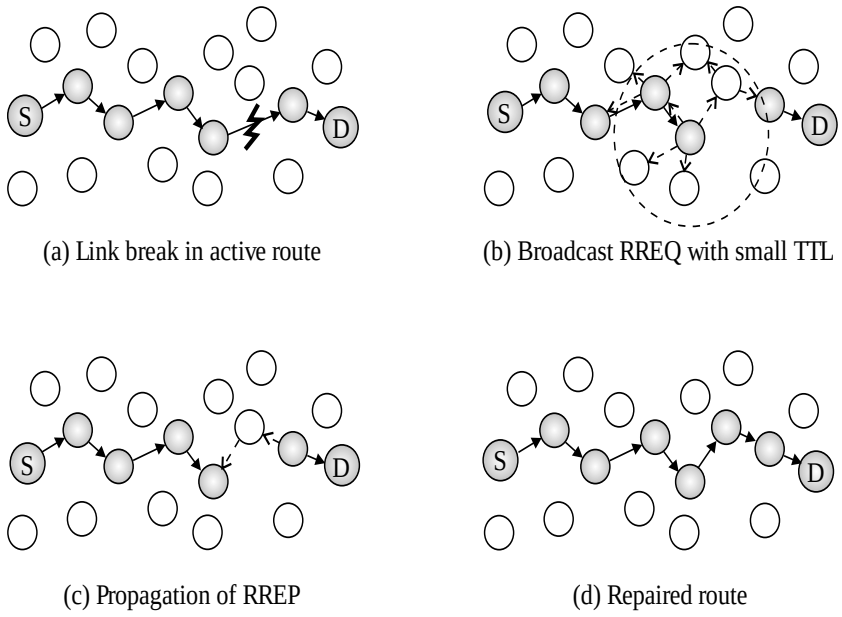


Figure 4.3: Example of local repair.

4.3.3 Local Repair

Local repair of link breaks in active routes is another approach to increasing scalability. In the current AODV specification, when a link break in an active route occurs, the node *upstream* of the break creates a Route Error (RERR) message listing all the destinations which have become unreachable due to the break. It then sends this message to its upstream neighbors, as described in Section 4.2.2. If, instead of sending an error message to the source node, the upstream node attempts to repair the broken link itself, fewer data packets may be lost and the link can be repaired without the source node (and other upstream nodes) being disturbed. For short routes, local repair may not have any significant performance advantages. But for large networks with increasingly long routes (i.e., 10+ hops), it is likely that link breaks will occur so frequently that it will be nearly impossible for the source node to keep up with all the necessary repairs.

A node upstream of a link break that attempts to repair the route does so by broadcasting a RREQ with a TTL set to the last known distance to the destination, plus an increment value. This TTL value is used so that only the most recent whereabouts of the destination will be searched, which prevents flooding the entire network. The upstream node places the sequence number of the destination, incremented by one, into the RREQ. This prevents nodes further upstream on the route from replying to the RREQ, which would form a loop. Figure 4.3 illustrates an example of a local repair.

If a route to the destination is not located on the first attempt, a RERR message is sent back to the source node, and route re-discovery continues as described in Section 4.2.2.

4.3.4 Combining the Enhancements

The above enhancements can be combined in various ways for increased protocol scalability. Specifically, the expanding ring search and local repair can work together, as can query localization and local repair. The expanding ring search and query localization are used to optimize route discoveries initiated by a source node, while local repair is used to decrease the number of route discoveries which a source node must initiate. The local repair, when combined with those two enhancements, operates in the same fashion as previously described. One attempt at the repair is made locally. If this attempt is unsuccessful, a RERR message is sent back to the source.

Table 4.1: Summary of room sizes.

# of Nodes	Room Size (m^2)	Avg. # of Neighbors
50	1000 * 1000	7.32
100	1500 * 1500	7.46
500	3500 * 3500	7.33
1000	5000 * 5000	7.69
5000	11500 * 11500	7.22
10000	16000 * 16000	7.50

4.4 Simulation Model and Methodology

4.4.1 Simulation Environment

The simulations used to evaluate the scalability of AODV and its enhancements were implemented within the GloMoSim library [160]. The simulations model networks between 50 and 10,000 mobile hosts placed randomly within the simulation area. The simulation boundary and average connectivity for each simulated number of nodes are shown in Table 4.1. The room size for each simulation was chosen so as to keep the node density approximately constant in the different size networks.

The radio propagation range for each node is 250 meters and channel capacity is 2 Mb/s. Each simulation is executed for 300 seconds of simulation time. Multiple runs with different seed numbers were conducted for each scenario and collected data were averaged over those runs.

A free space propagation model [135] with a threshold cutoff was used in the experiments. In the radio model radio capture is assumed, whereby a radio has the ability to lock onto a sufficiently strong signal in the presence of interfering

signals. The IEEE 802.11 MAC protocol with Distributed Coordination Function (DCF) [60] is used as the MAC layer in the experiments. A traffic generator was developed to simulate constant bit rate sources. The size of data payload is 512 bytes. Twenty data sessions with randomly selected sources and destinations are simulated. Each source transmits data packets at a rate of four packets/sec. The random waypoint model [69] is utilized as the mobility model. In this model, a node selects a random destination within the terrain range and moves towards that destination at a speed between the pre-defined minimum and maximum speed. Once the node arrives at the destination, it stays at its current position for a pause time. After being stationary for the pause time, it randomly selects another destination and speed and then resumes movement. The minimum speed for the simulations is 0 m/s while the maximum speed is 10 m/s. The selected pause time is 30 seconds.

4.4.2 Parameter Values

Table 4.2 gives a summary of the chosen parameter values. The network diameter (`net_diameter`) for the simulations represents the approximate diameter of the network, and is used for setting the TTL value of broadcast control packets. It is also a factor in the calculation of how long a node should wait to receive a RREP after sending another RREQ. If the RREQ is broadcast across the network, the reception of the RREP may take longer for large networks than for small. The setting of the `net_diameter` variable to 35 for small networks (50, 100, 500, and 1,000 nodes) and 70 for the larger networks (5,000 and 10,000 nodes) provides an upper bound of the actual network diameter for these networks.

The `node_traversal_time` represents an estimation of the processing time of a packet at a given node. It is also used for estimating the period of time a source

Table 4.2: Parameter values.

	Parameter	Value
General	`net_diameter`	35, 70
	`node_traversal_time`	40 ms
Expanding Ring Search	`ttl_start`	1
	`ttl_increment`	2
	`ttl_threshold`	7
Query Localization	κ	2
	κ_2	$\kappa \times 2$
	τ	10 sec
Local Repair	`local_add_ttl`	2

node should wait to receive a RREP after broadcasting a RREQ.

The values selected for the enhancement parameters represent a tradeoff between minimizing the number of searches required to locate a given destination, and reducing the number of nodes that must receive and process the RREQ packet. For the expanding ring search, the initial TTL `ttl_start` of the RREQ is set to one. Each time a reply is not received, the TTL is incremented by `ttl_increment`, until the threshold value (`ttl_threshold`) is reached. After that point, the RREQ is broadcast across the network. When rediscovering routes, the initial TTL is the last known hop count to the destination by the source.

The value of κ for the query localization enhancement represents the number of hops the RREQ is allowed to travel off the previously known path to the destination. The initial value of κ is set to two. If no reply is received, the value of κ is doubled for the second attempt. The value of τ is ten seconds. If a node has been part of the most recent route for the past τ time units and receives the

Table 4.3: Protocol abbreviations.

Protocol Combination	Abbreviation
AODV	AODV
AODV and Expanding Ring Search	AODV-ERS
AODV and Query Localization	AODV-QL
AODV and Local Recovery	AODV-LR
AODV, Expanding Ring Search and Local Recovery	AODV-ERS-LR
AODV, Query Localization and Local Recovery	AODV-QL-LR

RREQ, it resets the κ to zero.

Finally, the `local_add_ttl` parameter is used for a local repair. It represents the value added to the previously known distance to the destination. This sum is used as the TTL of the RREQ for the local repair.

4.5 Simulation Results and Analysis

The following sections present the results achieved by the different protocol combinations. Table 4.3 indicates the abbreviation associated with each protocol enhancement in the following figures.

4.5.1 Throughput

Figure 4.4 shows each scheme's throughput performance, where throughput is calculated to be the number of data bytes delivered to destination hosts. The figure shows that the ability of the protocols to deliver packets to their destina-

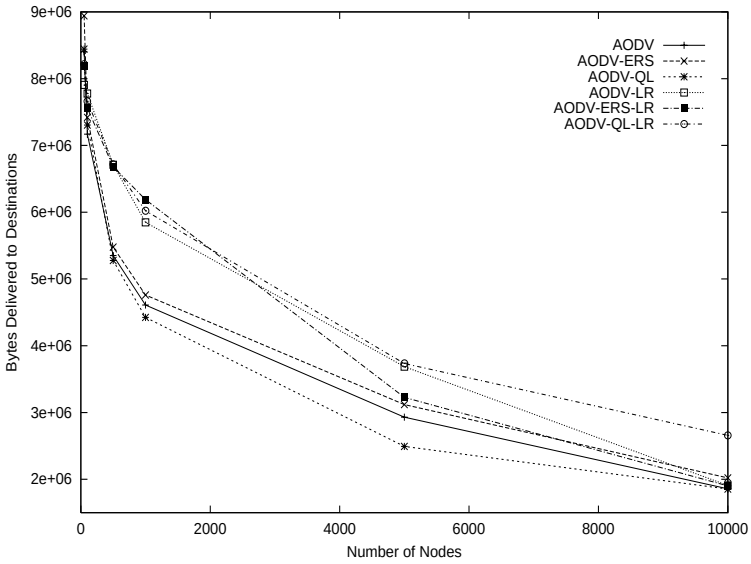


Figure 4.4: Throughput.

tion degrades as the network size becomes larger. The path length is greater in larger networks because the simulation terrain range and the number of nodes increases while the average number of neighbors is kept relatively constant (see Table 4.1). Routes are more prone to disconnections in mobile networks when path lengths are longer. Because a single link failure results in the inability of the source to reach the destination, longer routes have a greater probability of route disconnection than shorter hop routes. An increased route length in larger multihop networks is a characteristic not only of AODV (or on-demand routing protocols), but any routing protocols such as table driven (i.e., distance vector and link state) algorithms and hierarchical clustered routing protocols. It is observed that performing route recovery locally improves throughput. Since nodes closer to the destination than the source initiate route rediscovery, new routes are repaired more quickly and less data packets are dropped.

It is interesting to note that AODV-QL has the poorest throughput. The main purpose of query localization is to exploit node locality and reduce the

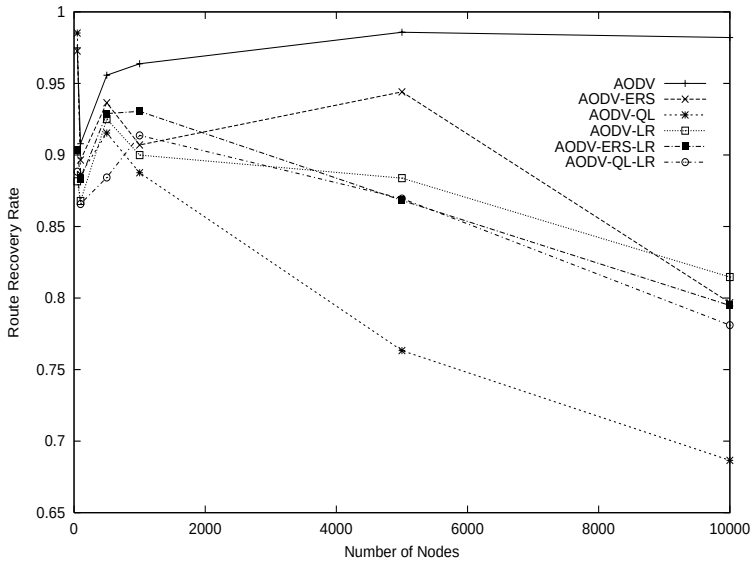


Figure 4.5: Route recovery rate.

number of routing message transmissions. Localizing the query, however, has the risk of not being able to establish the route. Figure 4.5 depicts the route recovery rate. AODV-QL is the least successful is reconstructing routes. As the number of nodes in the network increases, the recovery rate drops accordingly. If a more conservative and larger κ value had been used in query localization, route recovery rate would have been better at the expense of more control message overhead. Performing route recovery by simple flooding is not affected by the network size. Compared with all other enhanced schemes, AODV has the best recovery rate.

The path length of each scheme is presented in Figure 4.6. The route length is measured by calculating the distance between the source and destination when the route is constructed. The measure includes the first discovered route for both the construction of new routes, and the repair of broken routes. It is observed that schemes that utilize the local recovery technique yield longer paths. For protocols that do not use local recovery, only the source node can recon-

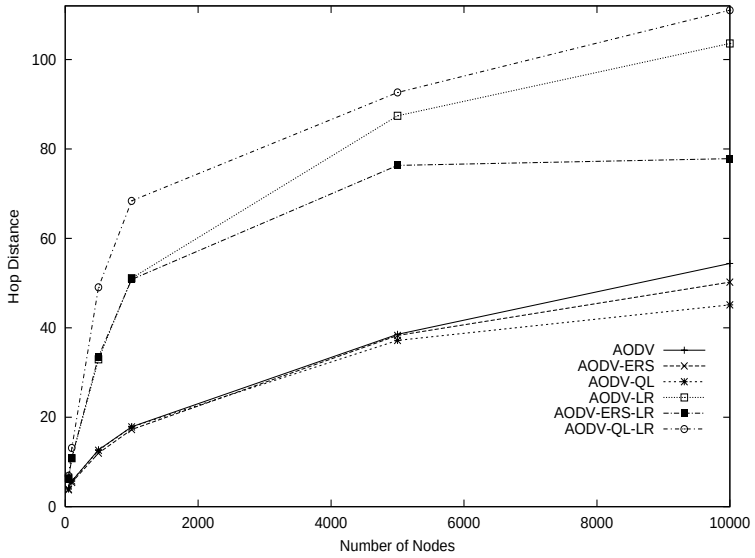


Figure 4.6: Path length.

struct routes. When a source rediscovers routes with a request/reply cycle, a new route is obtained based on current network information such as hop count, route freshness, node location, network topology, etc. On the other hand, in local recovery schemes, the node immediately upstream of the disconnected link initiates a route reconstruction. Because of the possibility that the destination has actually moved closer to the source node, but the distance between the node reconstructing the route and destination has increased, path lengths may tend to grow as intermediate nodes repair routes.

Longer path lengths naturally result in more route breaks and more route recoveries, as shown in Figure 4.7. Local recovery schemes have more route reconstruction attempts for the following two reasons. First, longer routes can fail more easily than shorter routes. Second, no RERR message is sent upstream to the source in local recovery. If a link upstream of the previously broken link becomes disconnected while a new route is being discovered, another local recovery procedure is initiated. This results in more route reconstruction attempts by

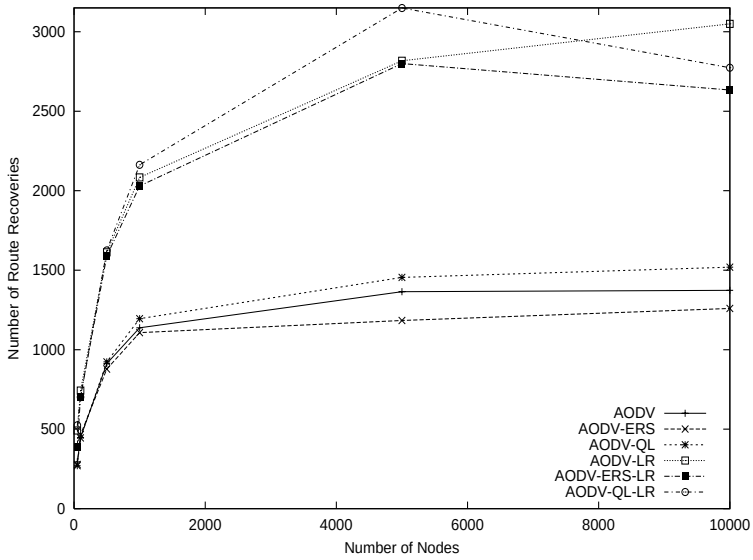


Figure 4.7: Number of route recovery attempts.

local recovery schemes. It is also interesting to note that the number of recovery attempts does not significantly increase between 5,000 and 10,000 nodes. This is due to the difficultly of all the protocols in maintaining routes with such a large path length. In these scenarios, many sessions are forced to abort due to the inability to maintain a route. Hence, with fewer sessions being maintained, fewer route discoveries are necessary.

4.5.2 Control Message Overhead

The routing message overhead is presented as the number of control message transmissions in Figure 4.8. Each hop-wise transmission of a control message by a node is counted as one transmission. As expected, AODV without enhancements has the most control packet transmissions. Local recovery schemes have less control overhead compared with schemes that perform route recovery by sources. AODV-QL-LR has the least control overhead among all protocols. Local recovery schemes reduce the number of RREQ transmissions. As shown

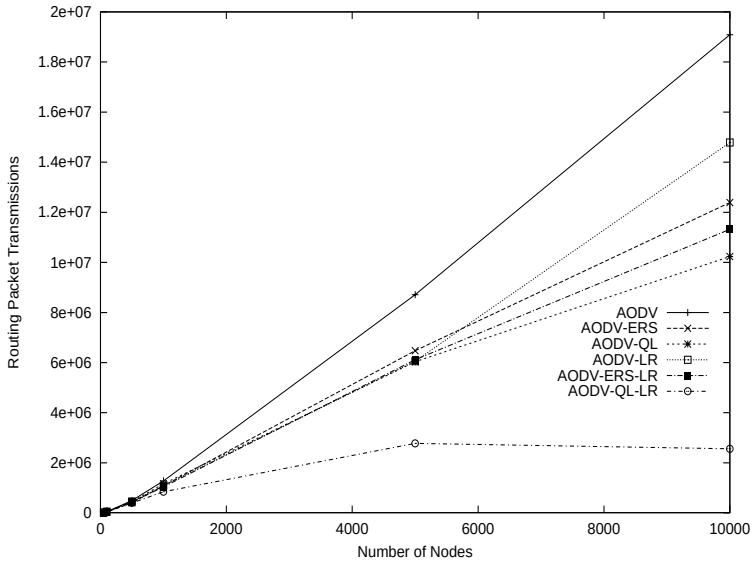


Figure 4.8: Routing message overhead.

in Figure 4.9, the percentage of RREQ transmissions among all routing packet transmissions (RREQ, RREP, and RERR) by local recovery schemes is lower than source recovery schemes.

In order to evaluate protocol efficiency, the number of all packet (i.e., data, RREQ, RREP, and RERR) transmissions per data delivery is investigated. Because link layer protocols for ad hoc networks are contention-based, this measure is very important for protocol analysis. The measure is presented in Figure 4.10. The scope and the ranking of protocols are similar to those of Figure 4.8. Note the huge number of packet transmissions per successful delivery at high node populations shown in Figure 4.10. This ratio, which can grow as large as 5,000, indicates the drastic need for work in a crucial area affecting the scalability of AODV, and probably all known ad hoc routing protocols, to large network populations. The ratio should be brought down by three orders of magnitude; such a reduction will probably also be accompanied by a proportional increase in the packet delivery fraction, which is sometimes as low as 15%. Work towards de-

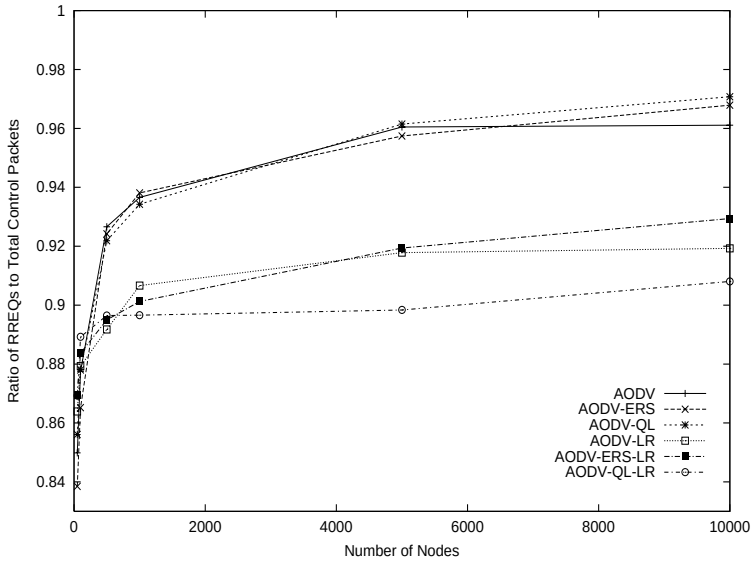


Figure 4.9: Percentage of RREQ transmissions among control packet transmissions.

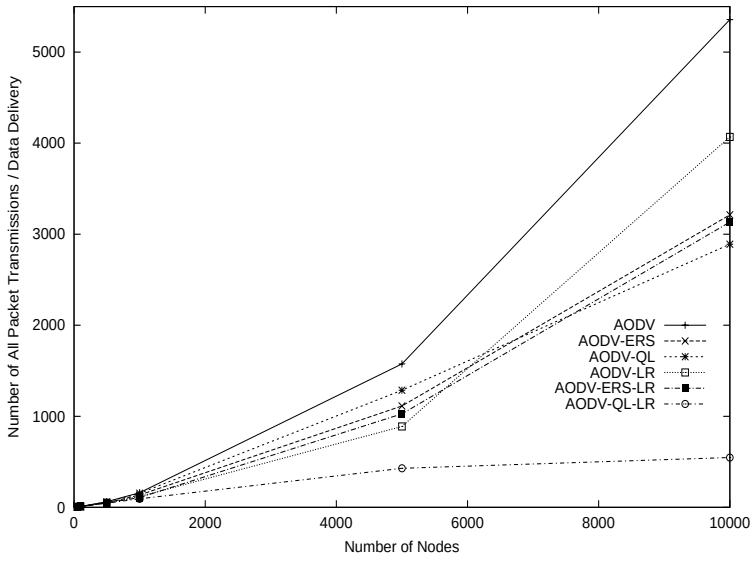


Figure 4.10: Number of all packet transmissions per data delivery.

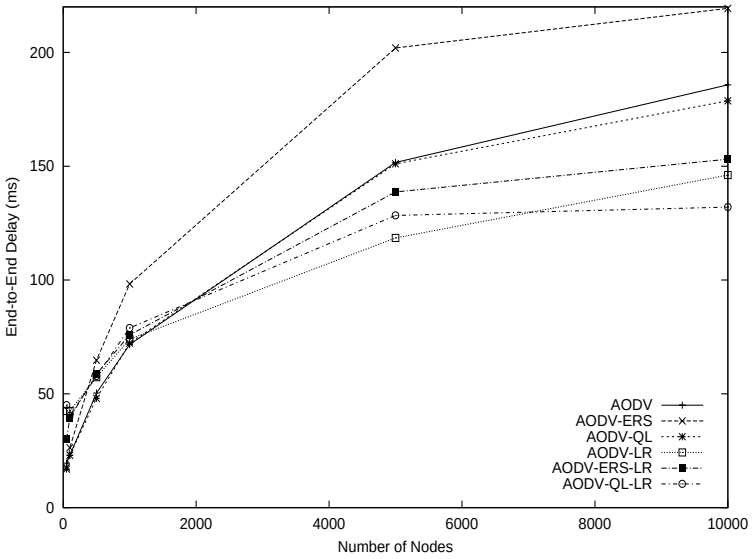


Figure 4.11: End-to-end delay.

veloping techniques for quickly re-establishing valid routes is likely to be of the highest importance for improving the scalability of ad hoc networks.

4.5.3 Delay

The end-to-end delay of each protocol is reported in Figure 4.11. Schemes that utilize the local recovery technique have shorter delays. Protocols in which sources initiate route recovery have longer end-to-end delays because of longer route re-establishment latency. To recover a broken route, a RERR packet must first be delivered from the node upstream of the broken link to the source of the route. The RREQ must then be broadcast from the source to the destination, and a RREP consequently has to be transmitted back to the source. Data packets are buffered at the source node during this process and this duration of time adds to the end-to-end delay. In local recovery schemes, on the other hand, the node upstream of the disconnected link initiates an immediate route reconstruction. Since route rediscovery is done locally, less time is needed to search for and obtain

a new route. Local recovery schemes can, therefore, yield shorter delays. Note that AODV-ERS has the longest delay because a route may not be built in the initial attempt (i.e., `ttl` = 1 or last known hop count of the route). Among local recovery schemes, AODV-ERS-LR has the longest delay for the same reason.

4.6 Observations

In the previous sections, we have studied the scalability characteristics of on-demand routing protocols, which are known to generally perform best in mobile multihop networks. We learned and verified that routing in ad hoc networks of tens of thousands of nodes is extremely difficult. In large networks, path lengths are longer compared with those in small networks (i.e., 50 or 100 nodes). Because network hosts are capable of mobility, longer routes are more prone to disconnection since a single link failure results in a route break. Each route invalidation invokes a route recovery process and clogs the network with control messages. Worse, because there are generally multiple hops between a source and destination, and because nodes are mobile, many route discoveries are unsuccessful. Although the flooded RREQ packet reaches the destination or intermediate nodes with routing information to the destination, the unicast RREP packet may not reach the source due to link breaks during route discovery. Even when the RREP packet survives to reach the source, the route may break shortly after and the source will need to initiate another route discovery. Therefore, maintaining routes with many hops in mobile ad hoc networks is a difficult challenge.

This chapter introduces three techniques and applies five different modification combinations to improve AODV scalability. The expanding ring search reduces the routing message overhead, but yields longer delays because of initial route discovery failures. Query localization also decreases control overhead, but it has

poor throughput performance due to low route recovery rate. This is especially true when routes have long distances. Local repair proves to be effective in enhancing AODV's performance in large networks. Because route recovery is localized, new routes are found more quickly than source initiated route discoveries. Consequently, packet drops are minimized. Local repair works efficiently with expanding ring search and query localization to reduce control message overhead. The drawback of local repair is however, that multiple repairs for the same route can be present at the same time.

Local repair may benefit from some mechanism to reduce the growth in path lengths which result from this method. One possible solution is to combine local repair with a RERR unicast back to the source. If a link breaks in an active route, the node upstream of that break could repair the route using local repair, and then send a RERR message back to the source. In this way, as the upstream node continues to receive data packets while the RERR is traveling to the source node, the data packets can still be forwarded to the destination. When the source receives the RERR, it can decide whether to reinitiate route discovery to look for a better route in order to reduce the length of the route if it has increased significantly. This method will results in fewer dropped packets than not using local repair while also reducing the increase in path lengths which result from local repair.

4.7 Conclusion

This chapter has evaluated the scalability of on-demand ad hoc routing protocols by selecting a representative from this set of protocols and simulating it in networks of up to 10,000 nodes. To improve the performance of on-demand protocols in large networks, five modification combinations have been separately incorpo-

rated into an on-demand protocol, and their respective performance has been studied. It has been shown that the use of local repair is beneficial in increasing the number of data packets that reach their destinations. Expanding ring search and query localization techniques seem to further reduce the amount of control overhead generated by the protocol, by limiting the number of nodes affected by route discoveries.

The results of this work are not specific to the AODV protocol. The expanding ring search, query localization and local repair modifications are protocol independent and can be incorporated into virtually any on-demand protocol to improve that protocol's scaling potential. Scalability in ad hoc mobile networks is inherently difficult due to the mobility of the nodes and the transience of network links. Work on large-scale ad hoc networks is likely to uncover techniques that would be valuable for stabilizing routing protocols in the Internet at large, leading to faster route convergence and reduced route flaps. Creating ad hoc routing protocols which experience minimal performance degradation when used in increasingly large networks is a challenge, and there remains a significant amount of work to reach this goal.

CHAPTER 5

The Effects of MAC Protocols on Ad hoc Network Communication

The number and variety of wireless devices and applications has dramatically increased within the past few years. As these products begin to permeate the marketplace, the need to provide communication between them is becoming increasingly important. In an effort to establish and maintain routing paths in these ad hoc mobile networks, numerous unicast and multicast routing protocols have been designed. To determine the relative merits of the protocols, there have recently been investigations comparing the performance of these protocols under various conditions and constraints [23, 39, 68, 95]. One question that arises is whether the choice of MAC protocol affects the relative performance of the routing protocols being studied.

There has been some discussion as to the correct Medium Access Control (MAC), or link layer (level-2 of the OSI reference model), protocol to use for channel access when performing these simulations. Many early protocol simulations utilized the Carrier Sense Multiple Access (CSMA) protocol [76]. Since the advent of the IEEE 802.11 protocol [60], however, most protocol evaluations have elected to run over this channel access protocol, since it provides both prevention and detection of the hidden terminal problem [156].

It is the intent of this chapter to compare the performance of different ad hoc

routing protocols to determine whether the selection of the MAC layer affects the relative performance of ad hoc routing protocols. It is likely that the performance of the protocols will be best when run over IEEE 802.11, due to its channel acquisition characteristics. However, the question is whether protocols degrade proportionately to each other when run over the other MAC layer protocols. To determine whether the selection of MAC protocol is a factor when comparing routing protocols, this chapter explores the behavior of different unicast routing protocols when run over varying MAC protocols.

5.1 Routing Protocols

To analyze the effects of MAC protocols, three ad hoc routing protocols are selected for study. The first is the Wireless Routing Protocol [114], which is a distance vector table-driven protocol. Table-driven protocols periodically exchange routing table information in an attempt to maintain an up-to-date route from each node to every other node in the network at all times.

The second protocol studied is the Fisheye State Routing protocol [123]. This protocol is a variation on the basic link state table-driven algorithm, whereby update message entries are exchanged between nodes at different frequencies, depending on their distance from each other. Routing information for a node's immediate neighborhood is kept the most up-to-date, while that for nodes further away is less accurate. This method helps reduce the table size in routing table exchanges while still maintaining routes to each network node.

Finally, the Ad hoc On-Demand Distance Vector Routing protocol [128, 129] is included as an example of an on-demand protocol. On-demand protocols only establish routes when they are needed by a source node, and only maintain these

routes as long as the source node requires them.

The following sections provide overviews of the protocols.

5.1.1 Wireless Routing Protocol

The Wireless Routing Protocol (WRP) [114] maintains routing information through the exchange of triggered and periodic updates. When a node notices a link break with one of its neighbors, it broadcasts an update message containing the distance and second-to-last hop information for each destination for which the routing information has changed. The second-to-last hop information is used to reduce routing loops. A neighboring node receiving an update message modifies its distance table entries and checks for new paths through other nodes. Any new paths are relayed back to the original node so that routing consistency is maintained throughout the network. Furthermore, a node successfully receiving an update message transmits an acknowledgment back to the sender, indicating the link is still viable.

In the event that a node has not transmitted anything within a specified period of time, it must transmit a *Hello* message (instead of exchanging the entire route table) to ensure connectivity. Otherwise, the lack of messages from a node indicates the failure of that link. When a node receives a Hello message from a new node, it sends that neighbor a copy of its routing table information.

5.1.2 Fisheye State Routing

Fisheye State Routing (FSR) [123] is a variation of link state table-driven routing which maintains a topology map at each node. To reduce the overhead incurred by control packets, FSR modifies the link state algorithm in three ways. First,

link state packets are not flooded; only neighboring nodes exchange link state information. Second, the link state exchange is time-triggered, not event-triggered. Finally, instead of transmitting all routing table information at each iteration, FSR uses different exchange intervals for different entries in the table. More precisely, entries corresponding to nodes that are nearby (within a predefined *scope*) are propagated to neighbors more frequently than entries of nodes that are far away. These modifications reduce the control packet size and the frequency of transmissions. As a result, FSR scales well to large networks since link state exchange overhead is kept low. As mobility increases, routing information for remote destinations may become less accurate; however, as a packet travels nearer to its destination, it is forwarded by nodes with increasingly more accurate routing information.

5.1.3 Ad hoc On-Demand Distance Vector Routing

The Ad hoc On-Demand Distance Vector (AODV) Routing protocol [128, 129] is an on-demand routing protocol which utilizes a route discovery cycle for the establishment of routes. A node desiring a route to some destination broadcasts a *Route Request* (RREQ) packet across the network. When either the destination or an intermediate node with a current route to the destination receives the RREQ, it responds by unicasting to the source node a *Route Reply* (RREP). Once the source node receives the RREP, it can begin using the route for data packet transmissions.

Route maintenance in AODV takes the form of *Route Error* (RERR) messages. When a link break in an *active* route occurs, the node upstream of the break sends a RERR to any upstream neighbors which were using that link to reach the destination. The RERR message lists each destination which is now

Table 5.1: Summary of MAC protocols.

Protocol	Mechanism
CSMA	CSMA
MACA	PSMA/RTS/CTS
FAMA	CSMA/RTS/CTS
IEEE 802.11 DCF	CSMA/CA/RTS/CTS/ACK

unreachable due to the loss of the link. When a source node receives a RERR message, it may re-initiate route discovery if it still requires the route.

5.2 MAC Protocols

The MAC protocols selected for this study represent a progression in protocol development. Each one builds upon the previous one through the addition of either control overhead or carrier sensing in order to mitigate the effects of the hidden terminal problem and achieve better network throughput.

Table 5.1 summarizes the mechanism of each MAC protocol included in the study. Packet sensing (PSMA) implies that carrier sensing is not performed before packet transmissions. The following sections describe each of the MAC protocols utilized in this evaluation.

5.2.1 Carrier Sense Multiple Access

The Carrier Sense Multiple Access (CSMA) [76] protocol is the most primitive of the MAC protocols utilized in this study. The CSMA version used is non-persistent CSMA. In this protocol, a node senses the channel for ongoing trans-

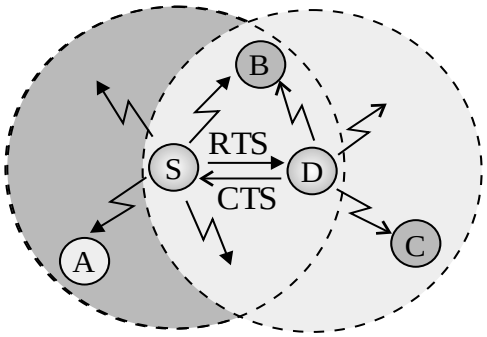


Figure 5.1: Effect of RTS/CTS control messages.

missions before sending a packet. If the channel is already in use, the node sets a random timer and then waits this period of time before re-attempting the transmission. On the other hand, if the channel is not currently in use, the node begins transmission.

5.2.2 Multiple Access with Collision Avoidance

The Multiple Access with Collision Avoidance (MACA) [73] protocol improves upon CSMA by taking steps towards the avoidance of the hidden terminal problem. The protocol defines Request-To-Send (RTS) and Clear-To-Send (CTS) control packets to announce an upcoming transmission. A node wishing to send a data packet broadcasts a RTS message containing the length of the data frame that will follow. Upon receiving the RTS, the receiver responds by broadcasting a CTS packet which also contains the length of the upcoming data frame. Any node hearing either of these two control packets must be silent long enough for the data packet to be transmitted. In this way, neighboring nodes will not transmit during the data transmission, and the number of collisions is reduced. Figure 5.1 illustrates the basic idea behind the RTS/CTS control messages. When S broadcasts the RTS message, both nodes A and B receive it and delay their

transmission attempts. Similarly, when node D responds with a CTS, nodes B and C also receive the CTS and are silent during the data transmission.

In the event that two nodes send simultaneous RTS frames to the same node, the RTS transmissions collide and are lost. If this occurs, the nodes which sent the unsuccessful RTS packets set a random timer utilizing the binary exponential backoff algorithm for the next transmission attempt.

5.2.3 Floor Acquisition Multiple Access

The Floor Acquisition Multiple Access (FAMA) variant utilized in this study is FAMA-NTR (Non-persistent Transmit Request) [47]. FAMA-NTR builds upon the MACA protocol by adding non-persistent carrier sensing to the RTS-CTS exchange. Before transmitting a RTS frame, a node first listens to the channel to determine if it is already in use. If the channel is busy, the node calculates a random backoff period to wait before sensing the channel again. The addition of this carrier sense to the control packet exchange aids in the prevention of control packet collisions.

5.2.4 IEEE 802.11 Distributed Coordination Function

The IEEE 802.11 MAC protocol specifies a Distributed Coordination Function (DCF) [60] which is based on the same RTS/CTS message exchange for unicast data transmissions as the previous MAC protocols. Where 802.11 differs, however, is in its use of collision avoidance before RTS transmission, and its requirement of an acknowledgment (ACK) transmission by the receiver after the successful reception of the data packet. The inclusion of the ACK allows immediate retransmission if necessary by verifying that the data packet was successfully received. In the case of node mobility, the ACK may also aid in the detection of

hidden-terminal interference that was not detectable when the CTS message was sent.

5.3 Simulation Environment

The simulations were performed using the GloMoSim Network Simulator developed at UCLA [160]. This simulator models the OSI seven layer network architecture and includes models of IP and UDP routing. The simulator also allows for network node mobility, thereby providing for simulation of mobile ad hoc networks.

Node movement is modeled by the random waypoint mobility model [23]. Nodes move at a speed between 0 and 10m/s. When the node arrives at its randomly chosen destination, it rests for some pause time. It then chooses a new destination and begins moving once again. The pause times are varied between 0 and 300 seconds. Each MAC protocol/routing protocol/pause time combination is run for five different initial network configurations.

Each run is executed for 300 seconds of simulation time and models a network of 100 nodes in a 1500 meter $\times$1500 meter area. Each node has a transmission radius of 250 meter. The propagation model is the free space model [135] with threshold cutoff. This model has a power signal attenuation of $1/d^2$, where d is the distance between nodes. The radio model also has capture capability, whereby a node may successfully receive a packet even in the presence of noise. There are 20 data sessions between randomly selected sources and destinations. The bandwidth is 2 Mb/s, the data packet size is 512 bytes, and packets are sent at a rate of four per second by each source.

Table 5.2 shows the parameter values used for the routing protocols in the

Table 5.2: Parameter values.

	Parameter	Value
WRP	HELLO Interval	1 sec
	Max Allowed Missed HELLOS	4
	Update ACK Timeout Interval	1 sec
	Retransmission Timer	1 sec
	Retransmission Counter	4
FSR	Scope	2 hops
	HELLO interval	5 sec
	Max Allowed Missed HELLOS	3
	INTRASCOPE UPDATE interval	5 sec
	INTERSCOPE UPDATE interval	15 sec
AODV	HELLO Interval	1 sec
	Max Allowed Missed HELLOS	3
	RETRANSMIT TIME	750 msec

experiments. The majority of the parameter values for WRP were taken from those suggested by the designers of the protocol and specified in [114]; however, a few of the values were modified to maximize WRP's performance in the simulation environment. The timer values were set so as to send more frequent connectivity updates but less frequent retransmissions than suggested. The former modification is needed because of the high mobility speed in the experiments, and the latter is due to the fact that with the MAC protocols selected, retransmitting at twice the round trip time would flood the MAC buffer, in addition to causing unnecessary collisions with cross traffic in the channel.

Using the FSR protocol, a node includes in its route update message entries

for nodes outside its scope every Interscope update interval. Entries for nodes inside the scope are included in every update message transmission. Note that the Interscope update interval is much larger than that of Intrascope update.

When AODV is run over IEEE 802.11, Hello messages do not need to be used due to the MAC layer feedback of unreachable next hops. When combined with the other MAC protocols, however, Hello messages are needed since such feedback is not available. When Hello messages are used, a node transmits a Hello once each second as long as the node has not broadcast any other control messages during the previous second. Additionally, promiscuous listening mode is enabled for AODV whenever Hello messages are utilized. This allows AODV to determine more quickly when link breaks have occurred. The Retransmit Time value in Table 4.2 is the maximum allowable time between promiscuous receptions of data packets from neighbors on active paths.

5.4 Simulation Results

5.4.1 Throughput

To determine whether the selection of MAC protocols affects the relative performance of the protocols, three results are examined: the number of data packets received by their destinations, the control packet overhead, and the normalized routing load. The control packet overhead is computed by counting the number of hop-wise control packet transmissions. The normalized routing load is calculated by taking the total number of per-hop control packet transmissions, and dividing this by the number of data packets successfully delivered to their destinations.

Figures 5.2, 5.3, 5.4, and 5.5 illustrate the number of data packets delivered to destinations in each of the networks. The relative performances of WRP and

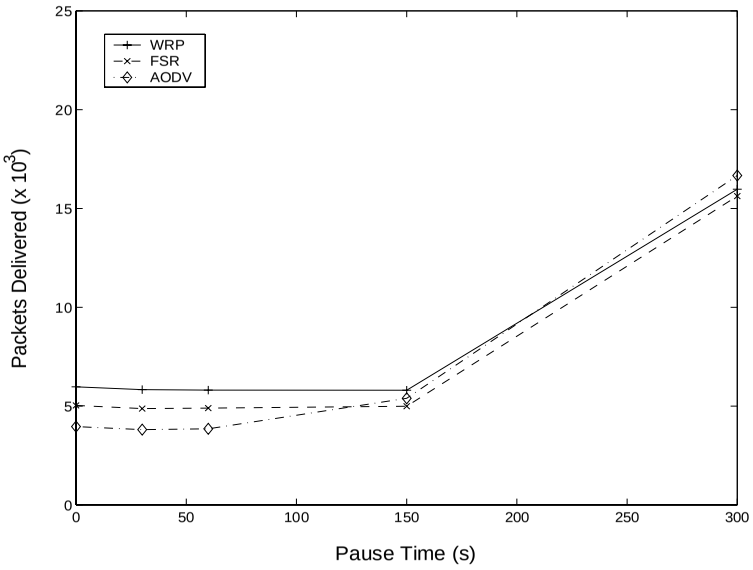


Figure 5.2: Data packets delivered on CSMA.

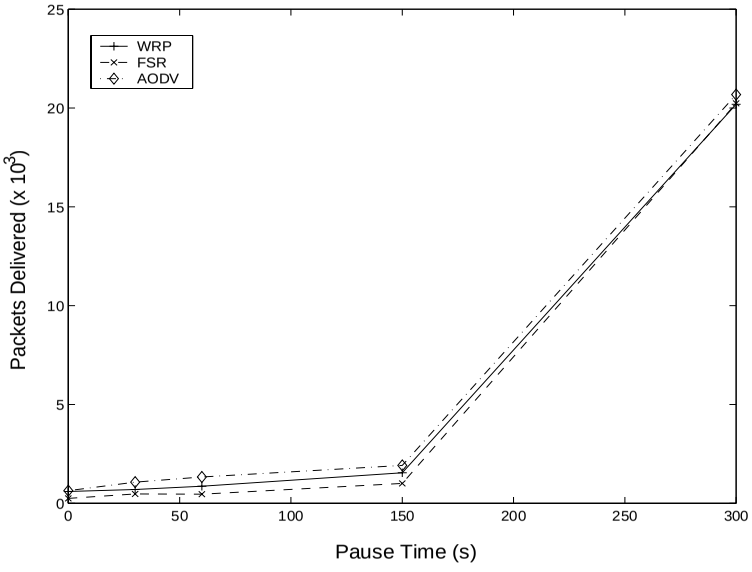


Figure 5.3: Data packets delivered on MACA.

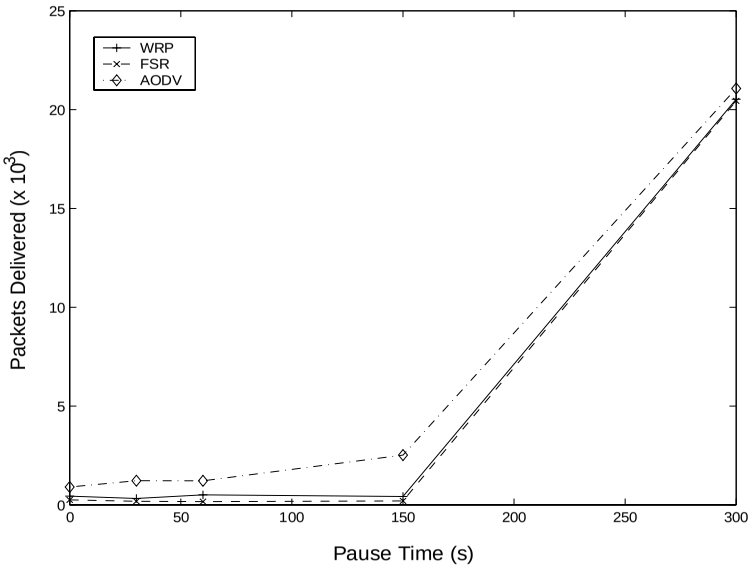


Figure 5.4: Data packets delivered on FAMA.

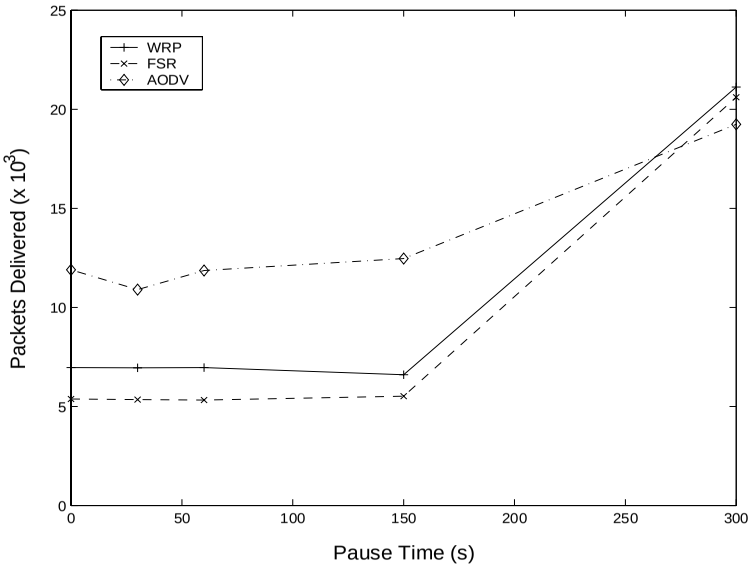


Figure 5.5: Data packets delivered on IEEE 802.11 DCF.

FSR remains fairly constant while that of AODV tends to vary by the MAC protocol used. When run over CSMA, WRP performs best for the higher mobility scenarios; however, while using IEEE 802.11, AODV outperforms the other protocols. The protocols achieve nearly the same number of delivered data packets when combined with the MACA and FAMA protocols, with AODV performing slightly better using the FAMA MAC protocol. The protocols have better overall performance using CSMA than using MACA or FAMA because of the RTS/CTS messages. MACA sources transmit RTS packets whenever they have a data packet to send without first sensing the channel. This results in an increase in packet collisions and hence decreased throughput. The collision avoidance mechanism incorporated into IEEE 802.11 for the transmission of RTS packets aids in the reduction of the number of collisions. Consequently, more data packets reach their destinations. Further analysis of the MAC protocols under UDP can be found in [34].

5.4.2 Control Overhead

The number of hop-wise control packet transmissions during each simulation is shown from Figure 5.6 to Figure 5.9. Because FSR uses periodic messaging regardless of the underlying MAC protocol, the amount of control overhead generated by this protocol remains relatively constant over the different simulations. WRP has both triggered and periodic updates, and hence the amount of control overhead increases as mobility increases (i.e., as the pause time becomes shorter). AODV is the only protocol significantly affected by the MAC layer. When run over CSMA, MACA and FAMA, AODV must utilize Hello messages in order to maintain connectivity. Hence it is expected that the number of control messages in these simulations is greater than in the IEEE 802.11 simulation. Additionally,

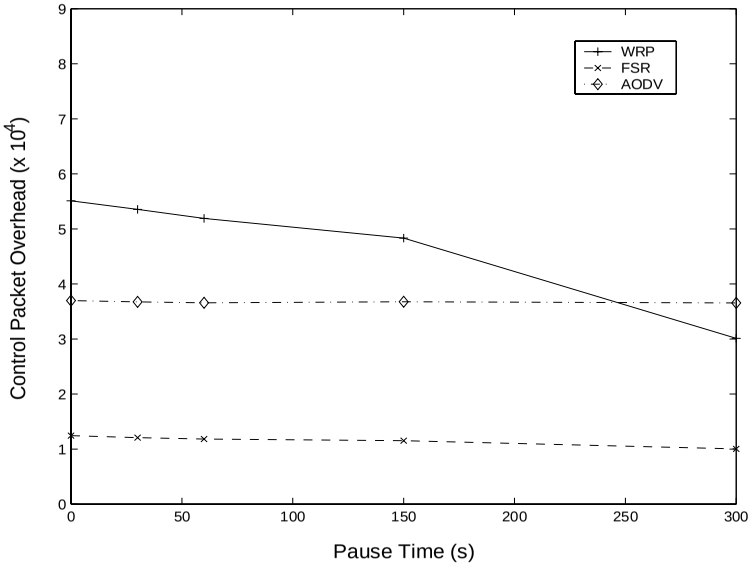


Figure 5.6: Control packet overhead on CSMA.

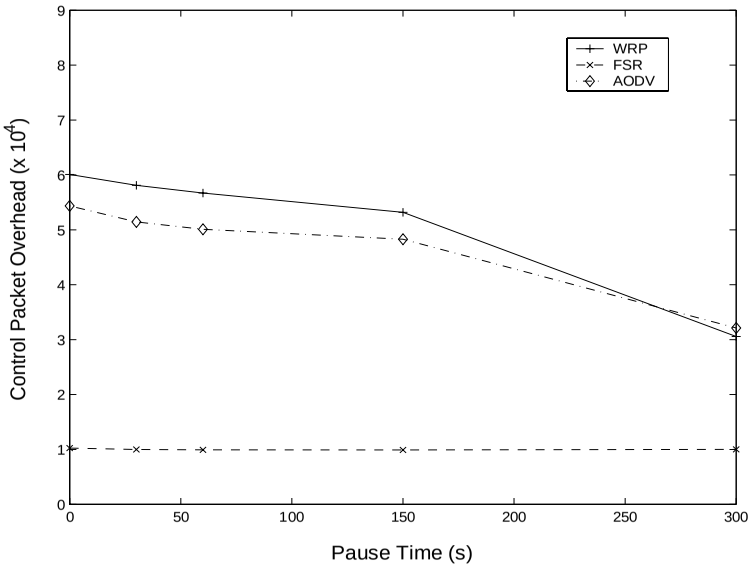


Figure 5.7: Control packet overhead on MACA.

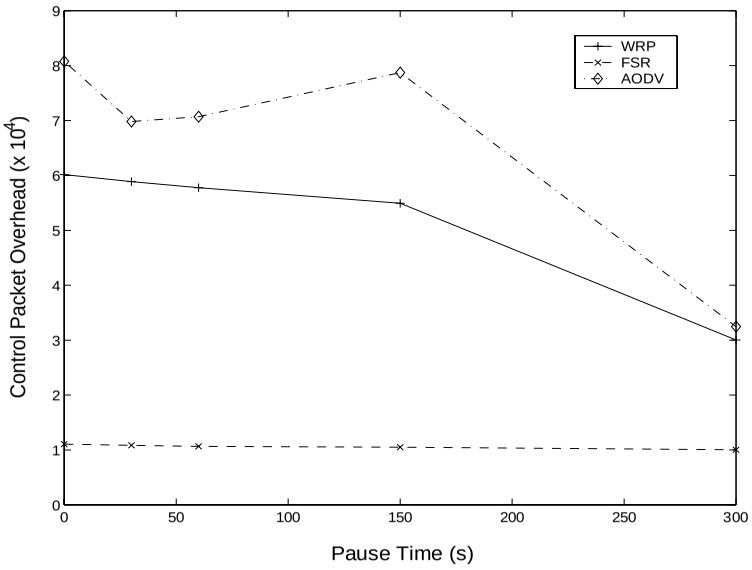


Figure 5.8: Control packet overhead on FAMA.

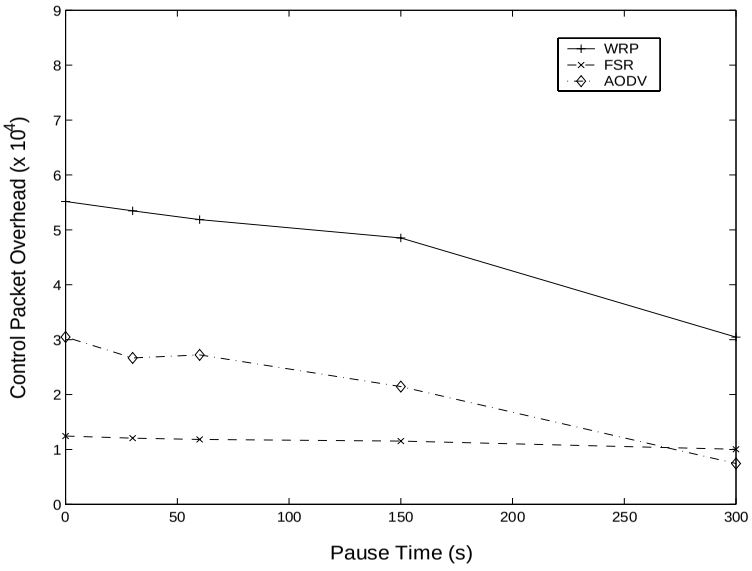


Figure 5.9: Control packet overhead on IEEE 802.11 DCF.

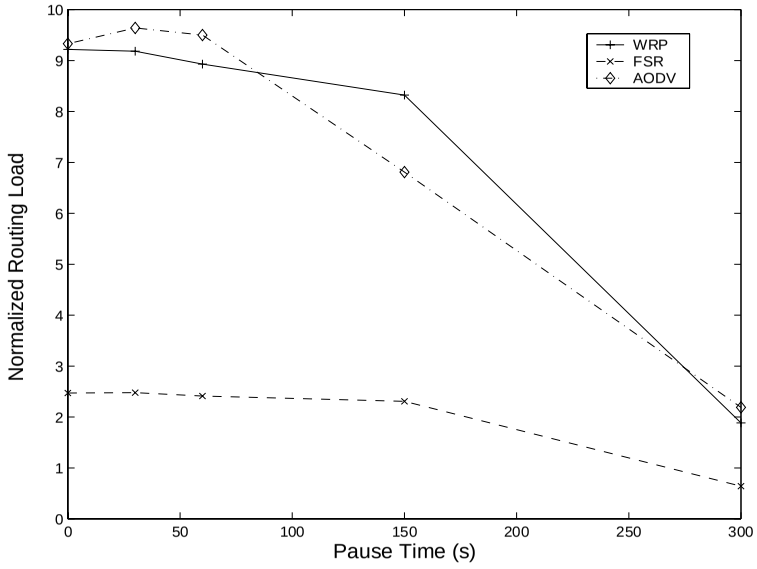


Figure 5.10: Normalized routing load on CSMA.

the amount of control overhead generated by AODV is directly related to the number of routes it is maintaining. Because there are so many packet collisions when utilizing the CSMA MAC layer protocol, AODV is not able to maintain as many routes. Hence the control overhead is lower for this simulation. As the number of routes AODV attempts to maintain increases, however, the amount of control traffic generated similarly increases.

5.4.3 Normalized Routing Load

The normalized routing load (NRL) is a measure of a protocol's efficiency. This measure is important because link layer protocols in ad hoc networks are contention-based. This result is shown from Figure 5.10 to Figure 5.13. WRP consistently has a greater NRL than FSR, and has greater NRL than AODV in all but a few cases of CSMA. The ratio of control messages generated by WRP and FSR remains approximately constant regardless of the underlying MAC protocol. Note

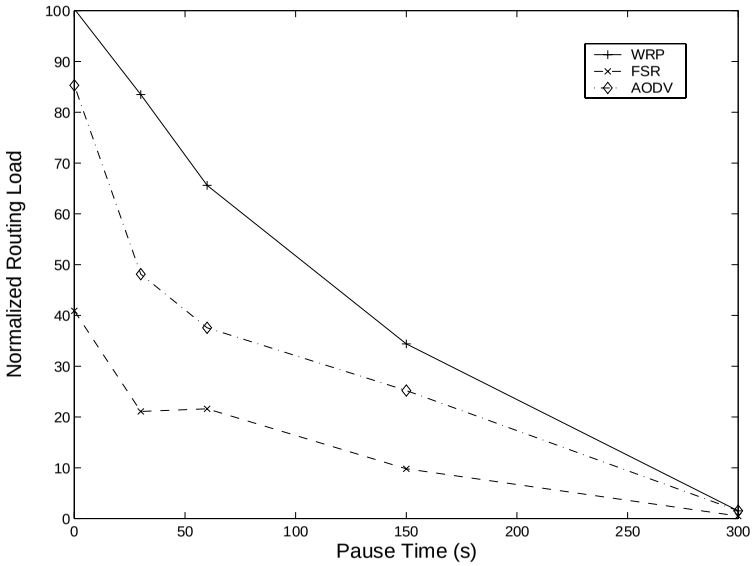


Figure 5.11: Normalized routing load on MACA.

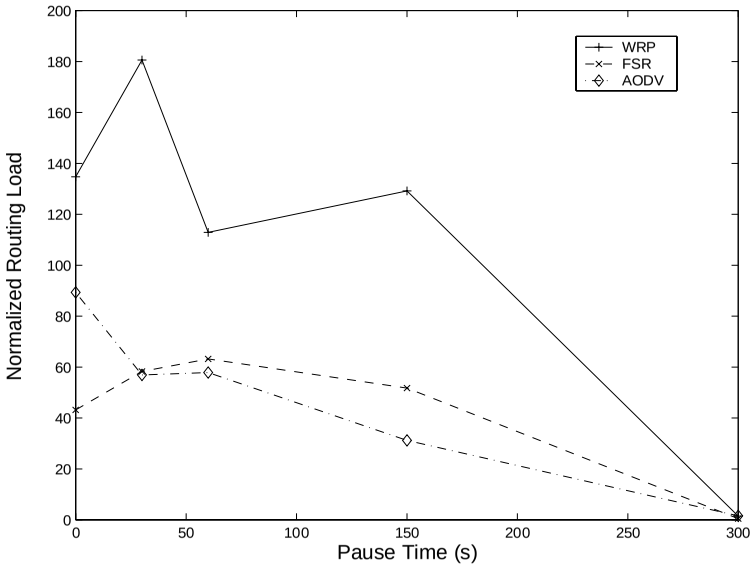


Figure 5.12: Normalized routing load on FAMA.

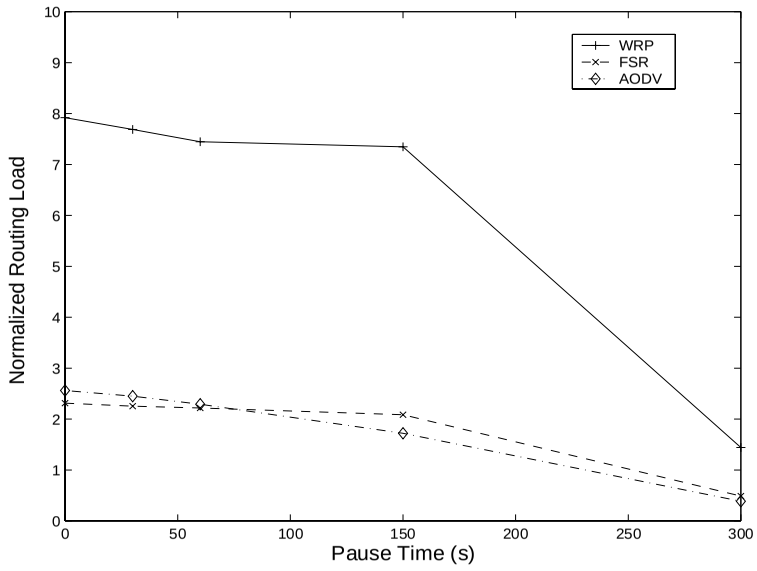


Figure 5.13: Normalized routing load on IEEE 802.11 DCF.

the variation in y-axis scaling. The NRL quantitative measure varies because the throughput of WRP and FSR is dependent upon the MAC protocols used. Hence, this metric aids in the analysis of how efficiently the routing protocols utilize routing packets to deliver data packets. AODV is most efficient when used with IEEE 802.11. This result is expected since AODV does not need Hello packet transmissions when combined with IEEE 802.11.

5.5 Conclusion

This chapter has presented a performance comparison of the WRP, FSR, and AODV routing protocols when combined with varying MAC protocols. The relative performance of the WRP and FSR protocols does not show notable variation when run over the different MAC protocols. Neither routing protocol requires operational changes dependent upon the underlying MAC protocol, and the results show that their relative performance remains approximately constant. This leads

to the conclusion that table-driven protocols act similarly with different MAC protocols, although further study of additional table-driven protocols is needed to validate this conclusion.

Because AODV requires periodic Hello messaging when run over link layer protocols that do not provide feedback when the next hop is unreachable, the amount of control traffic generated with these MAC protocols is considerably greater than when it is run over IEEE 802.11 DCF. AODV proves to be sensitive to the functionality of the MAC protocol, and hence its relative performance varies depending upon which MAC layer is used.

Table-driven and on-demand protocols may react differently depending upon the MAC protocol used; however, the question of whether two different on-demand ad hoc routing protocols would exhibit the same variation due to MAC layer effects remains open. The results show that the MAC protocol selected for simulation study is a key component of the performance of a routing protocol, and this aspect must be taken into consideration when doing comparative studies of the performances of routing protocols.

CHAPTER 6

Backup Routing in Ad hoc Networks

A recent trend in ad hoc network routing is the reactive on-demand philosophy where routes are established only when required. Most of the protocols in this category (tThe Ad-Hoc On-Demand Distance Vector (AODV) [128] protocol, for instance), however, use single route and do not utilize multiple alternate paths. Consequently, when route disconnects, nodes of the broken route simply drop data packets because no alternate path to the destination is available until a new route is established. When the network traffic requires real time delivery (voice, for instance), dropping data packets at the intermediate nodes can be costly. Likewise, if the session is a best effort, TCP connection, packet drops may lead to slow start, timeout, and throughput degradation. In this chapter, we propose an algorithm that utilizes a mesh structure to provide multiple alternate paths to existing on-demand routing protocols without producing additional control messages. Having multiple alternate paths in ad hoc networks is beneficial because wireless networks are prone to route breaks resulting from node mobility, fading environment, signal interference, high error rate, and packet collisions. It is also important to generate multiple routes without propagating more control messages than when building only single route. Minimizing the number of packet transmissions is critical in ad hoc networks with limited bandwidth and shared wireless medium.

Our scheme is inspired by the duct routing scheme [143] proposed in the early

1980s. Duct routing, however, suffers from some limitations; data packets are propagated in duplicates through multiple routes at all instances, thus creating excessive redundancy that causes congestion and collision. In our algorithm, on the contrary, multiple alternate paths are utilized only when the primary route is disconnected. Another difference between the two algorithms is that our protocol builds routes on demand. Wang and Crowcroft [163] also proposed a protocol that uses an alternate path only when data packets are not deliverable through the primary route. That scheme however, is based on Shortest Path First (SPF) algorithm for wire-line networks. There are some related work on protocols using multiple routes in ad hoc networks; the scheme by Nasipuri and Das [117, 118], Temporally-Ordered Routing Algorithm (TORA) [121], and Routing On-demand Acyclic Multipath (ROAM) [132], but these algorithms require additional control message to construct and maintain alternate routes.

We apply our scheme to the Ad-hoc On-Demand Distance Vector (AODV) protocol and evaluate the performance improvements by simulation. Since the purpose of our study is to improve the performance of existing on-demand protocols (specifically AODV in this paper), our protocol description is based on AODV. Our modifications to AODV for applying our scheme is also introduced.

6.1 Route Construction

Our scheme can be incorporated with reactive routing protocols that build routes on demand via a query and reply procedure. Our algorithm does not require any modification to the AODV's RREQ (route request) propagation process. When a source needs to initiate a data session to a destination but does not have any route information, it searches a route by flooding a Route Request (RREQ) packet. Each RREQ packet has a unique identifier so that nodes can detect and

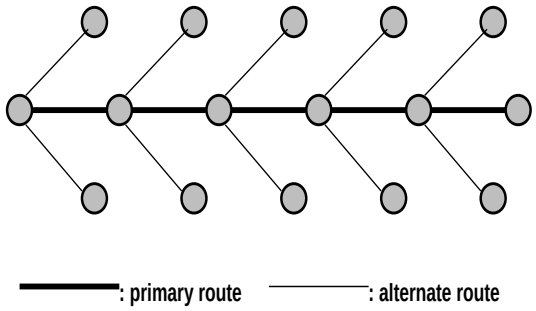


Figure 6.1: Multiple routes forming a fish bone structure.

drop duplicate packets. An intermediate node, upon receiving a non-duplicate RREQ, records the previous hop and the source node information in its route table (i.e., backward learning). It then broadcasts the packet or sends back a Route Reply (RREP) packet to the source if it has a route to the destination. The destination node sends a RREP via the selected route when it receives the first RREQ or subsequent RREQs that traversed a better route (in AODV for instance, fresher or shorter route) than the previously replied route.

The mesh structure and alternate paths are established during the route reply phase. We slightly modify the AODV protocol in this procedure. Taking advantage of the broadcast nature of wireless communications, a node promiscuously "overhears" packets that are transmitted by their neighboring nodes. From these packets, a node obtains alternate path information and becomes part of the mesh as follows. When a node that is not part of the route overhears a RREP packet not directed to itself transmitted by a neighbor (on the primary route), it records that neighbor as the next hop to the destination in its *alternate route table*. A node may receive numerous RREPs for the same route if the node is within the radio propagation range of more than one intermediate node of the primary route. In this situation, the node chooses the best route among them and inserts it to the alternate route table. When the RREP packet reaches the

source of the route, the primary route between the source and the destination is established and ready for use. Nodes that have an entry to the destination in their alternate route table are part of the mesh. The primary route and alternate routes together establish a mesh structure that looks similar to a fish bone (see Figure 6.1).

6.2 Route Maintenance and Mesh Routes

Data packets are delivered through the primary route unless there is a route disconnection. When a node detects a link break (for example, receives a link layer feedback signal from the MAC protocol,[1] does not receive passive acknowledgments,[2] does not receive hello packets for a certain period of time, etc.), it performs a one hop data broadcast to its immediate neighbors. The node specifies in the data header that the link is disconnected and thus the packet is candidate for "alternate routing." Upon receiving this packet, neighbor nodes that have an entry for the destination in their alternate route table, unicast the packet to their next hop node. Data packets therefore can be delivered through one or more alternate routes and are not dropped when route breaks occur. To prevent packets from tracing a loop, these mesh nodes forward the data packet only if the packet is not received from their next hop to the destination and is not a duplicate. When a node of the primary route receives the data packet from alternate routes, it operates normally and forwards the packet to its next hop when the packet is not a duplicate. The node that detected the link break also sends a Route Error (RERR) packet to the source to initiate a route rediscovery. The reason for reconstructing a new route instead of continuously using the alter-

[1]MAC protocols such as MACAW [19] and IEEE 802.11 [60] have this capability.

[2]This technique was introduced by Jubin and Tornow in their early work on packet radio networks [71]

nate paths is to build a fresh and optimal route that reflects the current network situation and topology.

Our alternate route utilization mechanism is similar to that of DSR (Dynamic Source Routing) [69], but has the following differences. Our scheme uses the mesh link only to "go around" the broken part of the route. In DSR, on the other hand, the node that detects a route disconnection can salvage the data by replacing in the source header the entire remaining route to the destination with an alternate route stored in its route cache. The DSR backup scheme requires considerable cache storage overhead. Another difference is that the node of DSR sends a RERR packet to the source only when it has no alternate route and cannot salvage the data. Therefore, routes in DSR are refreshed less often compared with our scheme.

In AODV, a route is timed out when it is not used and updated for a certain duration of time. We use the same technique for timing out alternate routes. Nodes that provide alternate paths overhear data packets and if the packet was transmitted by the next hop to the destination as indicated in their alternate route table, they update the path. If an alternate route is not updated during the timeout interval, the node removes the path from the table.

6.3 Example

Figure 6.2 is an example showing how the mesh and alternate routes are constructed and used in data delivery. When the RREQ reaches the destination node D, the primary route $<S\text{-}A\text{-}B\text{-}C\text{-}D>$ is selected. The destination D sends a RREP to node C. Nodes Y and Z, who are within the propagation range of D, overhear the packet and insert an entry into their alternate route table. This

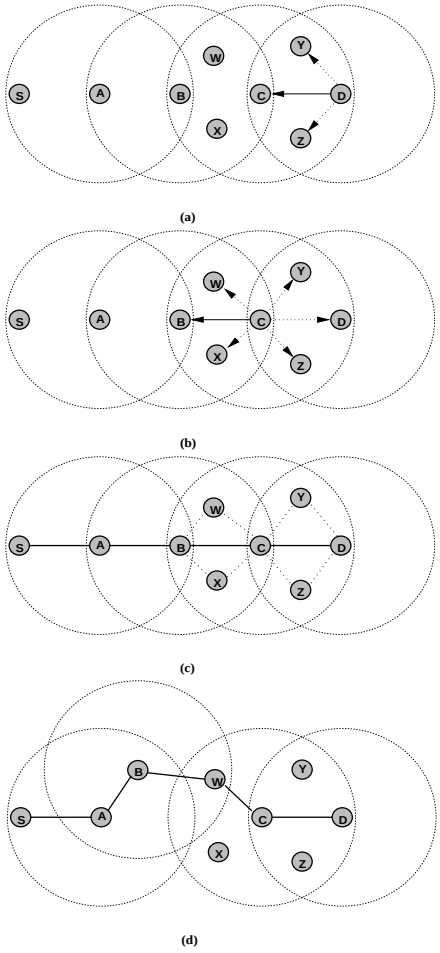


Figure 6.2: Multiple route construction and their usage: (a) node D sends a RREP, (b) node C forwards the RREP, (c) the primary route and alternate routes are established, (d) data packet is delivered via an alternate route when the primary route is disconnected.

process is shown in Figure 6.2 (a). After receiving this RREP, only node C relays the packet to node B since it is part of the route. Again, one hop neighboring nodes can overhear the packet. Nodes W and X record node C as the next hop to the destination D in their alternate route table. Node Y and Z, on the contrary, do not update their table since they already have a path to D. Likewise, node D does not react to the RREP transmission by node C since it is the destination (and part of the route). Figure 6.2 (c) shows the state when the RREP reaches the source node S and builds the primary and multiple alternate routes. Figure 6.2 (d) illustrates the usage of an alternate path when the primary route gets disconnected. Node B moved out the radio range of its next hop node C. After receiving the data packet from node A, node B forwards it to node C. The packet will fail to be delivered since node C is not reachable. Node B then broadcasts the packet to its neighbors for alternate paths to salvage the data. Nodes A and W receive the packet, but node A drops it upon duplicate detection. Node W, on the other hand, recognizes the primary route disconnection by reading the packet header. It looks up in its alternate route table and finds C as its next hop to the destination. It unicasts the packet to node C, and eventually the packet reaches the destination.

In the above example, the destination of the route receives the data packet via an alterate route that is longer in hop distance than the primary route. There can be instances where alternate routes have the same path length as the primary route. In Figure 6.3, for example, when the link between nodes B and C fails, node Z of the mesh forwards the packet from node B directly to the destination node D without sending it through node C. Therefore, the packet is delivered through the path $<S$-A-B-Z-$D>$ that has the same hop length as the primary route $<S$-A-B-C-$D>$.

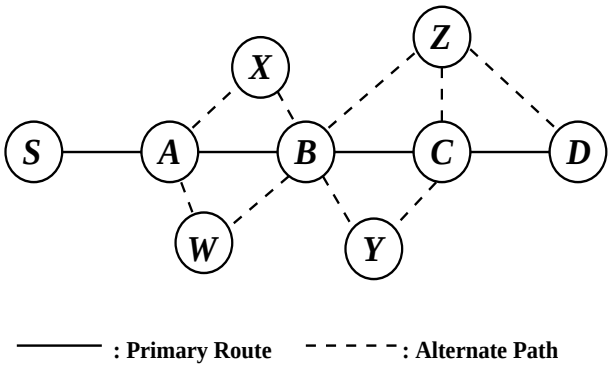


Figure 6.3: An alternate path with the same path length as the primary route.

6.4 A Variant

To improve the efficiency of the protocol, mesh nodes optionally may not relay the data packet when they overhear other salvaged transmissions. Let us use Figure 6.3 as an example again. Consider that node A has failed to verify the packet delivery to node B. When node A seeks help from neighboring nodes in the mesh, nodes W and X are available. Assume that node W receives the packet first and sends it to node B. Node X hears the transmission from node W to node B if it is within the radio propagation range of node W. Node X can choose not to relay the data packet from node A, since node W already attempted to salvage the data. In our current implementation however, node X still sends the data packet to node B for added redundancy since node B might have moved out of the radio range of node W.

6.5 Simulation Environment

To evaluate the performance improvements made by our backup routing, we compare the simulation results of the AODV protocol with and without applying

our scheme. In this section, we termed the AODV protocol that applied our algorithm as AODV-BR (AODV with Backup Routes).

The simulator was implemented within the Global Mobile Simulation (GloMoSim) library [160]. Our simulation modeled a network of 50 mobile hosts placed randomly within a 1500 meter $\times$ 300 meter area. Radio propagation range for each node was 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 [135] was used in our experiments. In the radio model, we assumed the ability of a radio to lock on to a sufficiently strong signal in the presence of interfering signals, i.e., radio capture. We used the IEEE 802.11 Distributed Coordination Function (DCF) [60] 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. There were ten data sessions, each with the traffic rate of four packets per second. The size of data payload was 512 bytes. The random waypoint mobility model [69] was used. Each node randomly selects a position, and moves toward that location with a speed between the minimum and the maximum speed. Once it reaches that position, it becomes stationary for a predefined pause time. After that pause time, it selects another position and repeats the process. We varied the pause time to simulate different mobility degrees. Longer pause time implies less mobility. The minimum and the maximum speed were zero and 20 m/s, respectively.

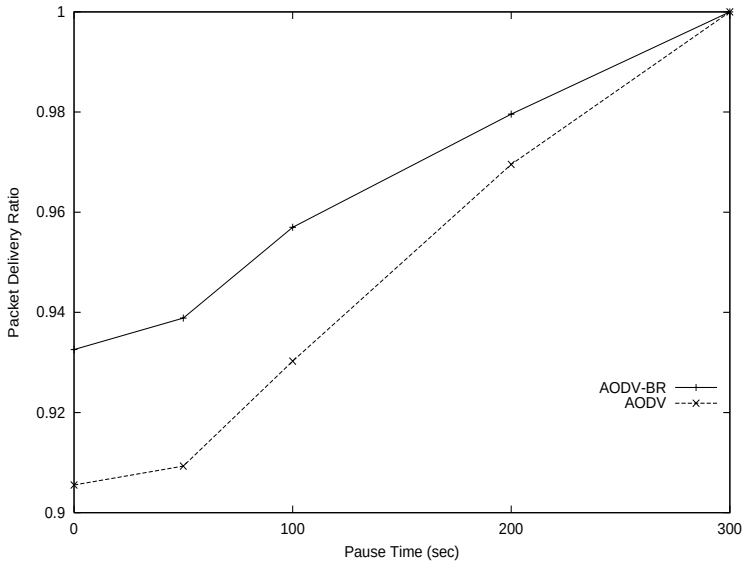


Figure 6.4: Packet delivery ratio.

6.6 Simulation Results and Analysis

6.6.1 Throughput

Figure 6.4 shows the throughput in packet delivery ratio. We can see that our scheme improves the throughput performance of AODV. As the mobility increases (i.e., pause time gets shorter), the performance gain by alternate routes becomes more significant. Because AMR attempts to use multiple alternate paths for data delivery in the presence of route breaks, the protocol is able to deliver more packets to the destination than AODV. AODV simply drops data packets when routes are disconnected. AODV-BR also has some packet losses. Alternate paths may be broken as well as the primary route because of mobility, or be unavailable and not discovered during the route reply phase. Moreover, packets can be lost because of collisions and contention problems.

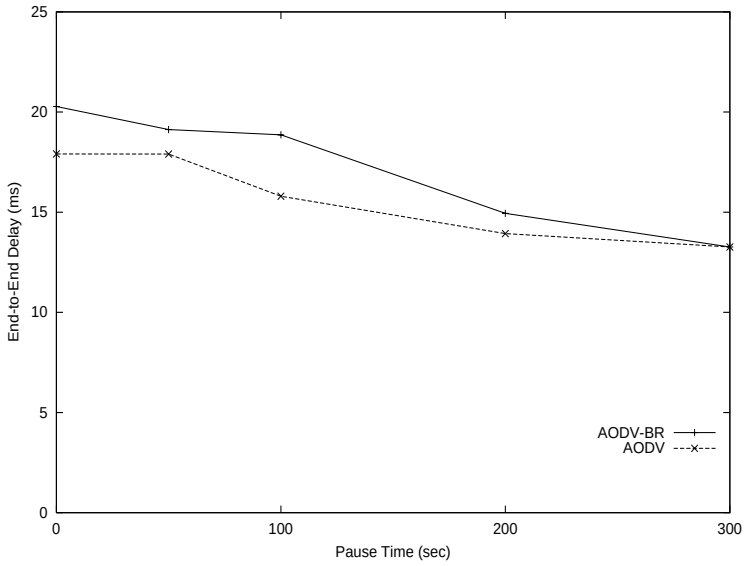


Figure 6.5: End-to-end delay.

6.6.2 Latency

End-to-end delay is presented in Figure 6.5. As expected, AODV-BR has longer delays than AODV. We can only measure delays for data packets that survived to reach their destination. AODV-BR delivers more packets, and those packets that are delivered in AODV-BR but not in AODV, take alternate and possibly longer hop routes. AODV-BR having longer delays than AODV does not represent its ineffectiveness since these protocols use the same primary route.

6.6.3 Efficiency

Because AODV-BR and AODV both have the same amount of control message overhead, we used a different metric for efficiency evaluation. We present the number of hop-wise data transmission per data delivery to the destination in Figure 6.6. We can observe that AODV-BR transmits slightly more data packets than AODV. There are two reasons for this result. First, when route break occurs,

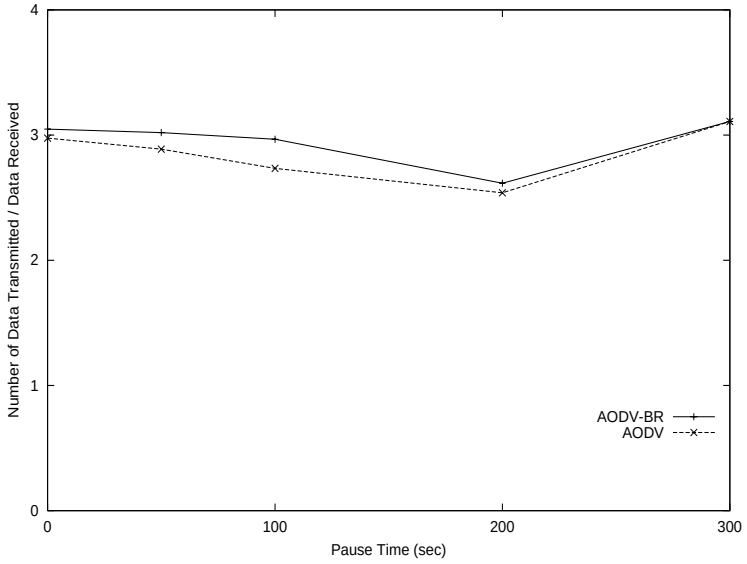


Figure 6.6: Number of data transmitted per data delivery.

AODV-BR uses longer alternate paths to deliver packets that are dropped in AODV, as explained above. Second, when there are multiple alternate paths, redundancy is created and hence increases the number of data transmission. We can learn from this result that we need to sacrifice efficiency in order to improve throughput and protocol effectiveness.

6.6.4 Throughput under Heavy Traffic

To investigate whether our scheme is still effective in heavy traffic situations, we increased the traffic load. In one experiment, we increased the number of data sessions with each session having the same traffic rate of four packets per second. In another experiment, we kept the number of sessions constant to ten and varied the traffic rate. Figure 6.7 shows the packet delivery ratio for ten sessions and twenty sessions. Ten sessions results are from Figure 7.3. We can see that the effectiveness of both protocols decreases because of the increase in packet

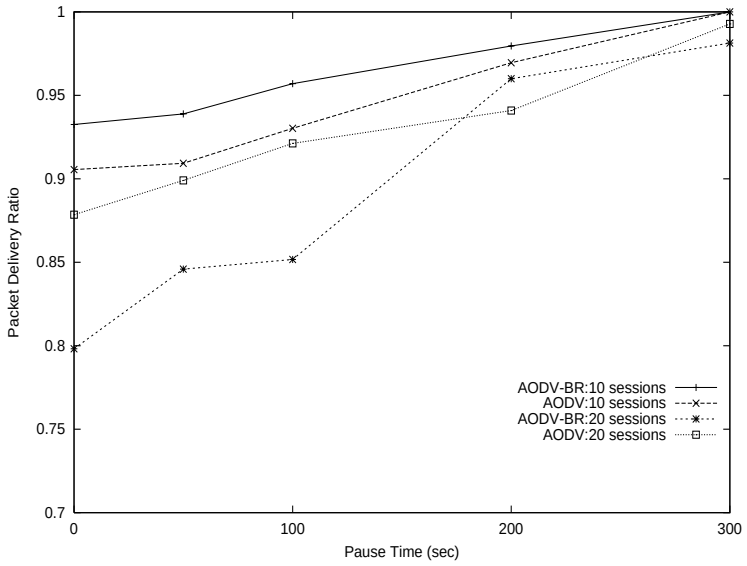


Figure 6.7: Packet delivery ratio with increased number of data sessions.

collisions when there are more data sessions. Even though AODV-BR improved the performance of AODV in a ten sessions network, it actually performs worse than AODV when we doubled the number of data sessions. Since there are more communication routes, AODV-BR generates more alternate routes accordingly. When the mobility rate is high, many route disconnects occur and a number of nodes that are part of the mesh transmit data packets. These transmissions cause collision and make the scheme lose its effectiveness. In fact, data packets traversing through alternate paths collide with packets using primary routes and degrade the overall throughput.

Figure 6.8 illustrates the packet delivery ratio with various data session traffic rate. Similar to Figure 6.7, throughput of both schemes degrades as the network traffic load increases. AODV-BR still performs better than AODV when each source sends six packets per second. As we further increase the data traffic however, AODV-BR cannot deliver more data packets than AODV. We can explain this behavior in the same way as we analyzed results in Figure 6.7. Basically,

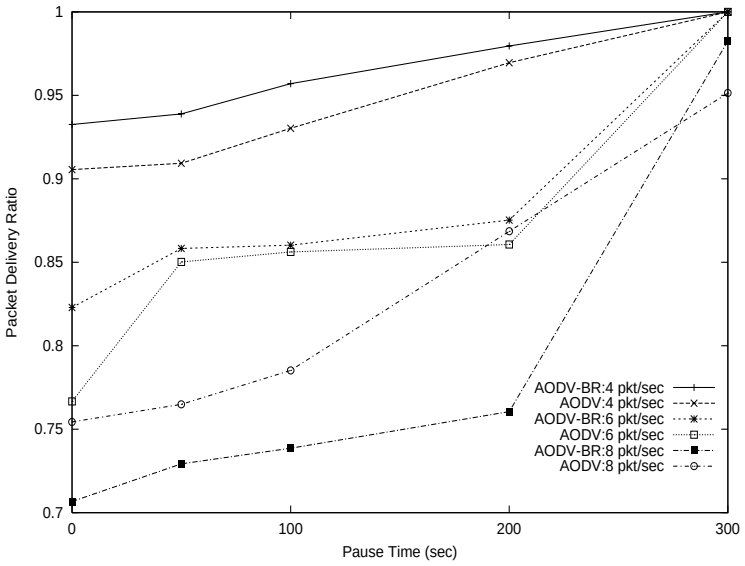


Figure 6.8: Packet delivery ratio with increased number of data rate.

AODV-BR is not as effective and efficient in heavily loaded network as in lightly loaded network because of increased packet collisions and channel contention.

6.7 Conclusion

We presented a scheme that utilizes a mesh structure and alternate paths. Our scheme can be incorporated into any ad hoc on-demand unicast routing protocol to improve reliable packet delivery in the face of node movements and route breaks. The mesh configuration provides multiple alternate routes and is constructed without yielding any extra overhead. Alternate routes are utilized only when data packets cannot be delivered through the primary route. As a case study, we applied our algorithm to AODV and measured performance improvements. Simulation results indicated that our technique provides robustness to mobility and enhances protocol performance. We also learned that however, our scheme does not perform well under heavy traffic networks. We are currently

investigating ways to make our protocol robust to traffic load. Additionally, we plan to further evaluate our scheme by using more detailed and realistic channel models with fading and obstacles in the simulation. We believe the advantage of providing backup routes will be significant in those environments

CHAPTER 7

Split Multipath Routing with Maximally Disjoint Paths

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. 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 [26, 33, 115, 119, 145, 154, 162, 168]. These protocols use table-driven algorithms (link state [108] or distance vector [104]) to compute multiple routes. Studies show however, that proactive protocols perform poorly because of excessive routing overhead [23, 68, 87]. Multipath routing in ad hoc networks has been proposed in [85, 118, 121, 132], including the one we introduced in the previous chapter. 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 propose a routing scheme called Split Multipath Routing (SMR) that es-

tablishes and utilizes multiple routes of 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. Providing multiple routes helps minimizing route recovery process and control message overhead. We believe utilizing 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. 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. We evaluate the performance of our scheme by extensive simulation.

7.1 Route Discovery

Split Multipath Routing (SMR) is an on-demand routing protocol that builds multiple routes using request/reply cycle. 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.

7.1.1 RREQ Propagation

The main goal of SMR is to build *maximally disjoint multiple paths*. We want to construct maximally disjoint routes to prevent certain nodes from being congested, and to utilize the available network resources efficiently. To achieve this

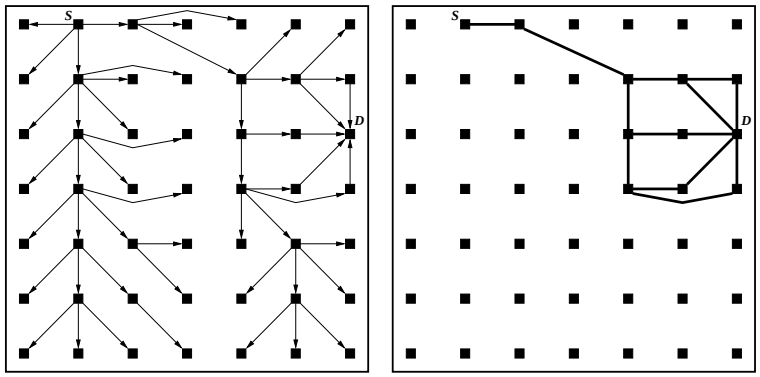


Figure 7.1: Overlapped multiple routes.

goal in on-demand routing schemes, the destination must know the entire path of all available 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 [69] and AODV [128], 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 7.1 (a) shows the paths taken by RREQs from the

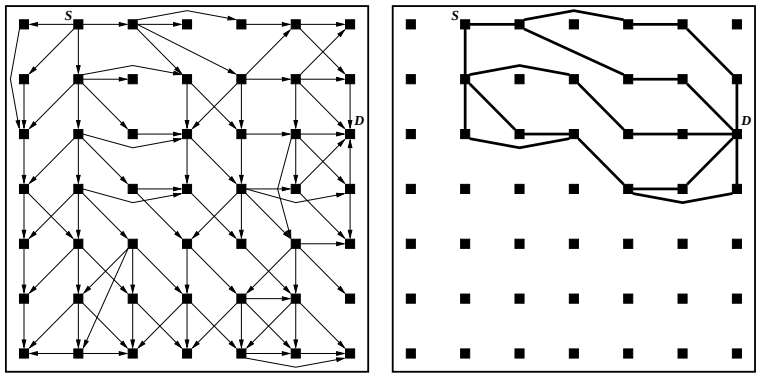


Figure 7.2: Multiple routes with maximally disjoint paths.

source node S to the destination node D, and Figure 7.1 (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 7.2 (a) shows the paths taken by RREQs using this technique. We can select more disjoint paths from routes available in Figure 7.2 (b) than those in Figure 7.1 (a). Our approach has a disadvantage of transmiting more RREQ packets, but it enables us to discover maximally disjoint routes.

7.1.2 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 this 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.

7.2 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 [60] or not

receiving passive acknowledgments [71]), 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 re-discovering 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 7.5.

7.3 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. 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 [80], 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.

7.4 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 [69] which uses single path.

We implemented the simulator within the Global Mobile Simulation (GloMoSim) library [160]. Our simulation modeled a network of 50 mobile hosts placed randomly within a 1000 meter $\times$ 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 [135] 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. If the capture ratio (the ratio of an arriving packet's signal strength over the sum of all colliding packets) [135] was greater than a predefined threshold value, the

packet was received while all other interfering packets were dropped. We used the IEEE 802.11 Distributed Coordination Function (DCF) [60] 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 [69] as the mobility model. Each node randomly selects a position, and moves toward that location with the speed between the minimum and the maximum speed. Once it reaches that position, it becomes stationary for a predefined pause time. After that pause time, it selects another position and repeats the process. 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.

7.5 Simulation Results and Analysis

7.5.1 Packet Delivery Ratio

Figure 7.3 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 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

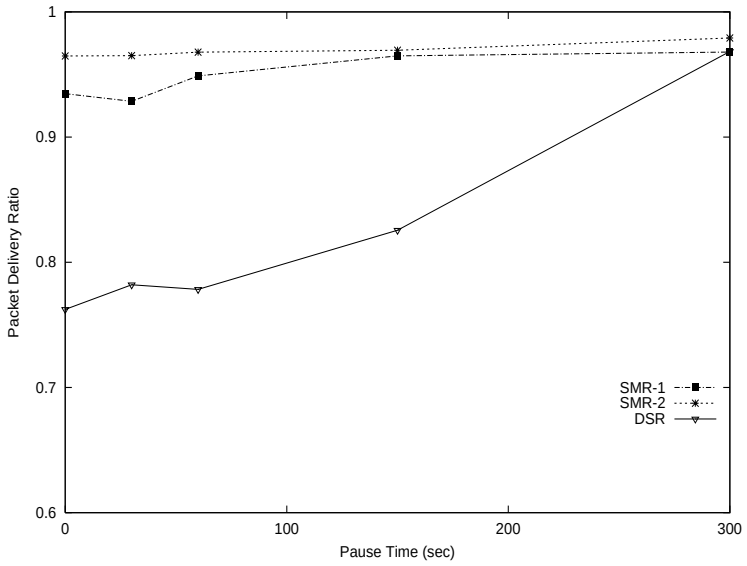


Figure 7.3: Packet delivery ratio.

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.

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 7.4 illustrates the number of packets dropped by each protocol. Both data and control packets are measured. The reasons for packet drops can be

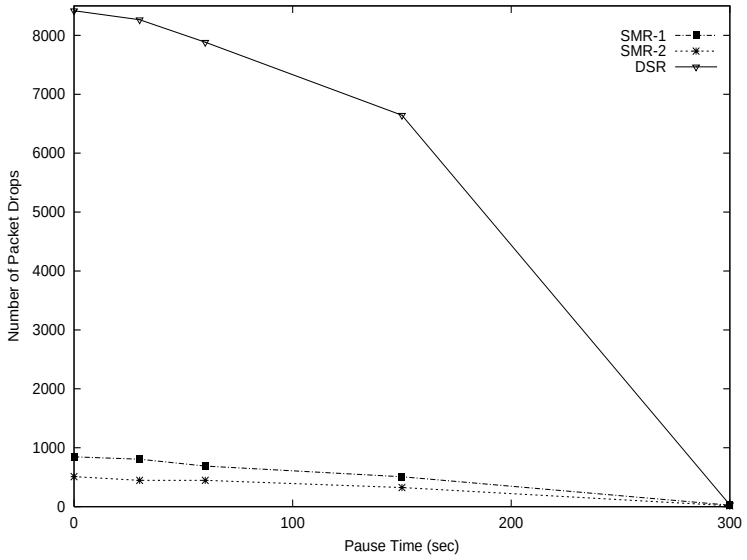


Figure 7.4: Number of packet drops.

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 because it invokes fewer route recovery processes and consequently, transmits less control messages.

7.5.2 Control Overhead

Figure 7.5 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 smallest value. This result is expected because SMR protocols generate more control packets while building multiple routes. On the

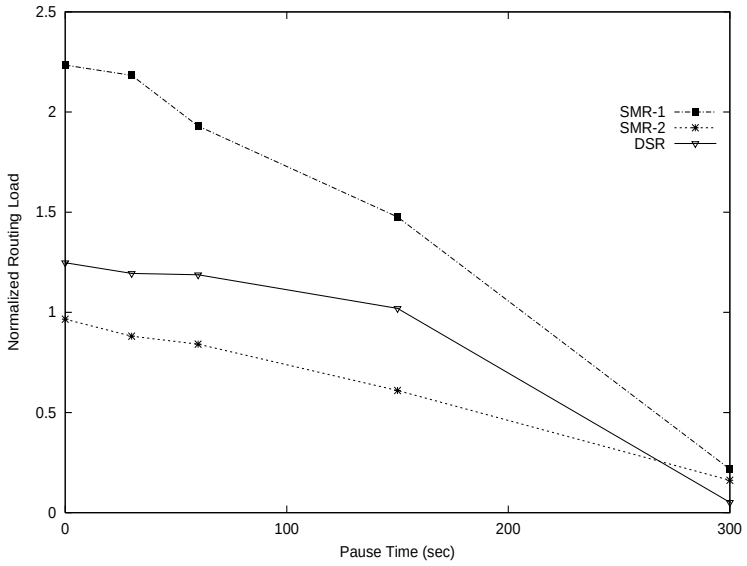


Figure 7.5: Normalized routing load.

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 the former 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

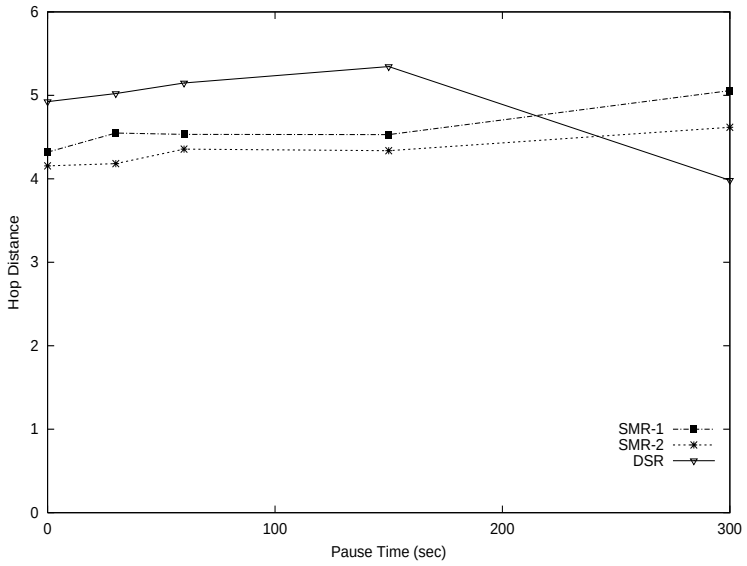


Figure 7.6: Hop distance.

can deduce from this result that excessive flooding makes the protocol inefficient.

7.5.3 Hop Length

Figure 7.6 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 routes may not be fresh enough and do not exploit the current network topology. DSR therefore builds longer routes than SMR protocols. Longer paths have a better chance of having route breaks since one link disconnection results in a route invalidation. Results from Figure 7.3 confirms our observation.

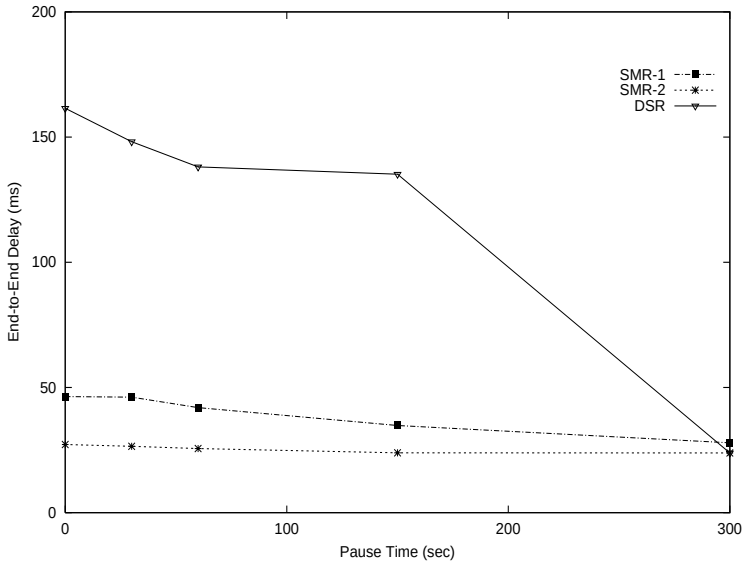


Figure 7.7: End-to-end delay.

7.5.4 Delay

Figure 7.7 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.

7.6 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 max-

imally 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 generate less control overhead.

CHAPTER 8

Dynamic Load-Aware Routing

Numerous routing protocols are proposed for ad hoc networks. No existing protocol however, considers the load as the primary route selection criteria. Using only the shortest delay as the route metric 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 network traffic increases [39, 68]. We present Dynamic Load-Aware Routing (DLAR) protocol that considers intermediate node routing loads for 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.

Routing with load balancing in wired networks has been exploited in various approaches [16, 107, 144, 157]. In ad hoc networks, only Associativity-Based Routing (ABR) [159] 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.

We introduce three routing algorithms that use load as the main route selection metric and show the effectiveness of DLAR protocols by presenting and comparing simulation results with an ad hoc routing protocol that uses the shortest paths.

8.1 Protocol Overview

DLAR builds routes on-demand. When a route is required but no information 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 <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 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 8.2.

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

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

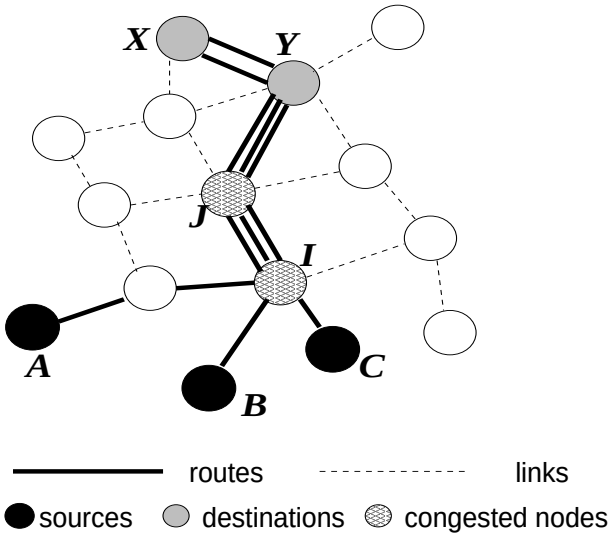


Figure 8.1: Congested network.

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 8.1 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 $<B\text{-}I\text{-}J\text{-}Y\text{-}X>$. 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 $<C\text{-}I\text{-}J\text{-}Y\text{-}X>$. The same process occurs when node A constructs a route to node Y. Figure 8.1 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).

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

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 the 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.

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.

8.2 Route Selection Algorithms

We introduce three algorithms in selecting the least loaded route. We use Figure 8.2 as an example network to describe each scheme.

DLAR *scheme 1* simply adds the routing load of each intermediate node and

[3]MAC protocols such as MACAW [19] and IEEE 802.11 [60] have this capability.

[4]This technique was introduced by Jubin and Tornow in their early work on packet radio networks [71].

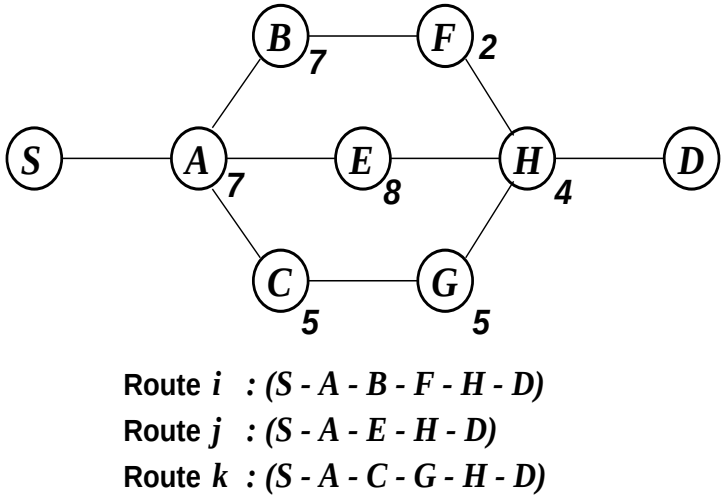


Figure 8.2: Example network.

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 earlier 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.

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 8.2 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

Table 8.1: 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

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 8.2 summarizes the route qualities in Figure 8.2 by applying each algorithm.

8.3 Simulation Model

We evaluate three DLAR schemes by comparing the performance with DSR [69], which uses the shortest path. We implemented the simulator within the Global Mobile Simulation (GloMoSim) library [160]. Our simulation modeled a network of 50 mobile hosts placed randomly within a 1000 meter $\times$ 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 [135] 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) [60] 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 [69] as the mobility model. Each node randomly selects a position, and moves toward that location with the speed between the minimum and the maximum speed. Once it reaches that position, it becomes stationary for a predefined pause time. After that pause time, it selects another position and repeats the process. We generated various mobility degree by using different pause times. The minimum and the maximum speeds were set constant to zero and 10 m/s, respectively.

8.4 Simulation Results

8.4.1 Throughput

Figure 8.3 shows the throughput in packet delivery ratio of each protocol when 20 sources 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 already existing routes. Nodes that are part of multiple routes become congested

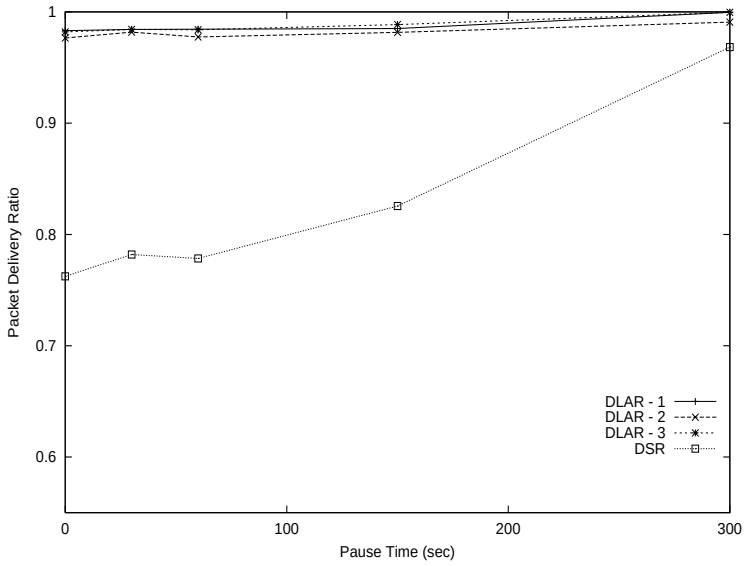


Figure 8.3: Packet delivery ratio (20 sources sending 4 pkt/sec).

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 8.4 shows the delivery ratio when traffic load is doubled to 8 packets per second and the number of sources is the same (20), and Figure 8.5 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

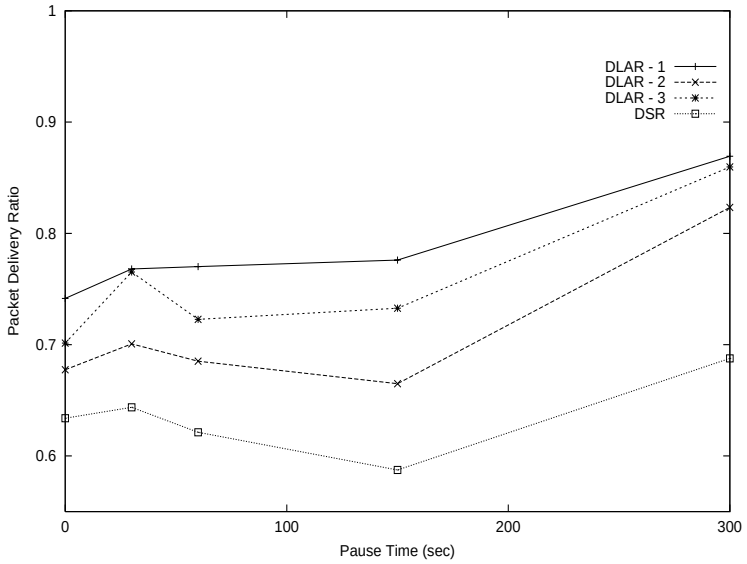


Figure 8.4: Packet delivery ratio (20 sources sending 8 pkt/sec).

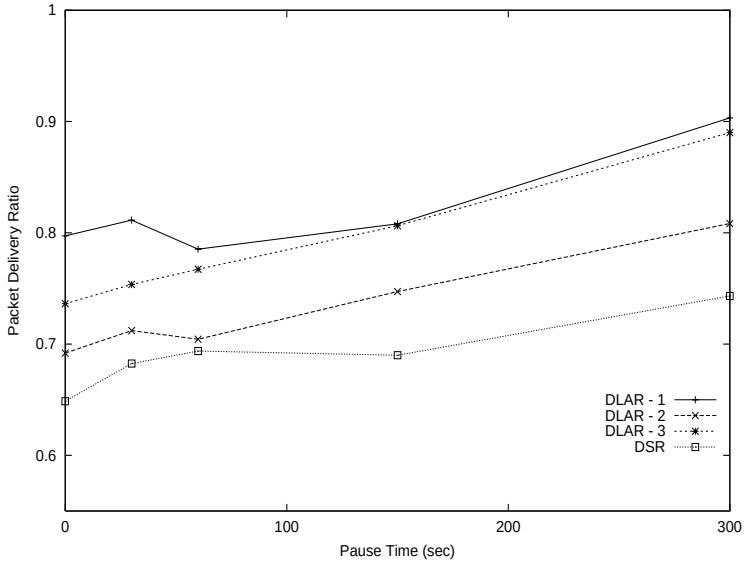


Figure 8.5: Packet delivery ratio (40 sources sending 4 pkt/sec).

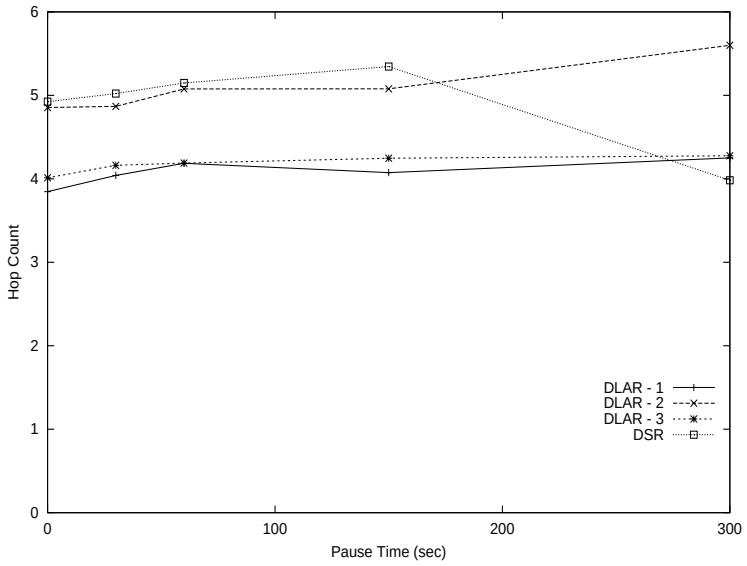


Figure 8.6: Hop distance.

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 better chance of having route breaks since one link disconnection results in a route invalidation.

8.4.2 Hop Count

Figure 8.6 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

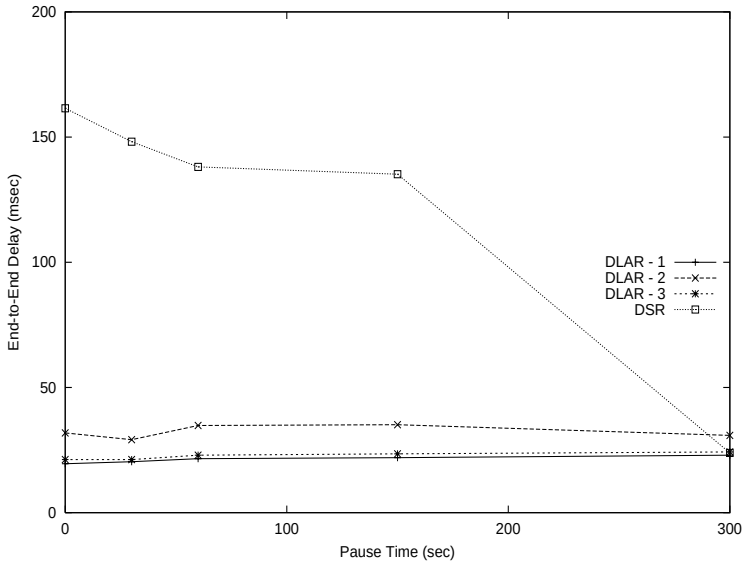


Figure 8.7: End-to-end delay.

not fresh enough and do not exploit the current network topology.

8.4.3 End-to-End Delay

Figure 8.7 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 8.6.

8.4.4 Routing Overhead

Figure 8.8 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,

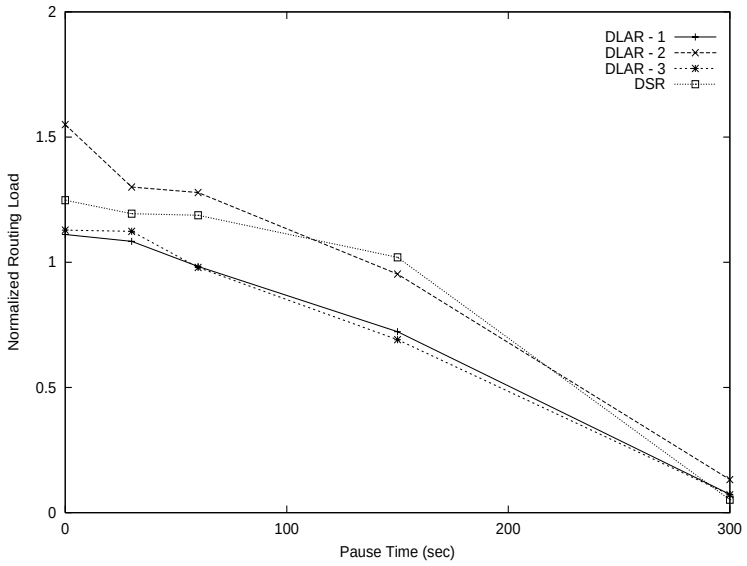


Figure 8.8: Normalized routing load.

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.

8.5 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 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.

CHAPTER 9

On-Demand Multicast Routing Protocol

This chapter presents a novel multicast routing protocol for mobile ad hoc wireless networks. The protocol, termed ODMRP (On-Demand Multicast Routing Protocol), is a *mesh*-based, instead of a tree-based, multicast protocol that provides richer connectivity among multicast members. By building a mesh and supplying multiple routes, multicast packets can be delivered to destinations in the face of node movements and topology changes. In addition, the drawbacks of multicast trees in mobile wireless networks (e.g., intermittent connectivity, traffic concentration, frequent tree reconfiguration, non-shortest path in a shared tree, etc.) are avoided. To establish a mesh for each multicast group, ODMRP uses the concept of *forwarding group* [31]. The forwarding group is a set of nodes responsible for forwarding multicast data on shortest paths between any member pairs. ODMRP also applies *on-demand* routing techniques to avoid channel overhead and improve scalability. A *soft-state* approach is taken to maintain multicast group members. No explicit control message is required to leave the group. We believe the reduction of channel/storage overhead and the richer connectivity make ODMRP more attractive in mobile wireless networks.

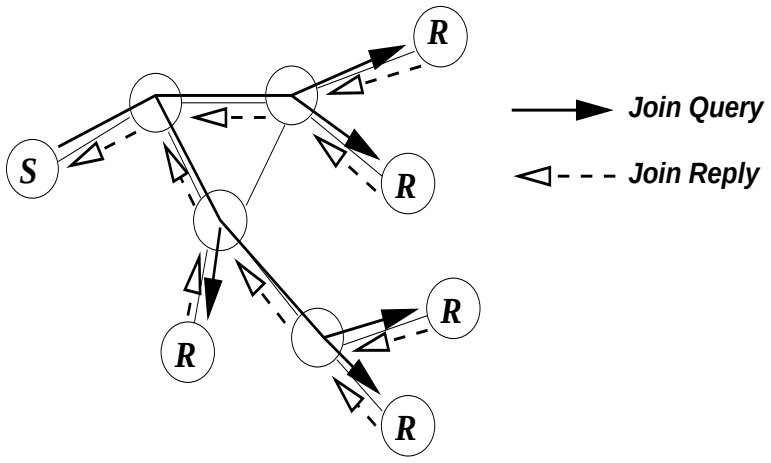


Figure 9.1: On-demand procedure for membership setup and maintenance.

9.1 Multicast Route and Mesh Creation

In ODMRP, group membership and multicast routes are established and updated by the source *on demand*. Similar to on-demand unicast routing protocols, a request phase and a reply phase comprise the protocol (see Figure 9.1). While a multicast source has packets to send, it periodically broadcasts to the entire network a member advertising packet, called a JOIN QUERY. This periodic transmission refreshes the membership information and updates the route as follows. When a node receives a non-duplicate JOIN QUERY, it stores the upstream node ID (i.e., backward learning) and rebroadcasts the packet. When the JOIN QUERY packet reaches a multicast receiver, the receiver creates or updates the source entry in its *Member Table*. While valid entries exist in the *Member Table*, JOIN REPLIES are broadcasted periodically to the neighbors. When a node receives a JOIN QUERY, it checks if the next node ID of one of the entries matches its own ID. If it does, the node realizes that it is on the path to the source and thus is part of the forwarding group. It then sets the `FG_Flag` and broadcasts its own

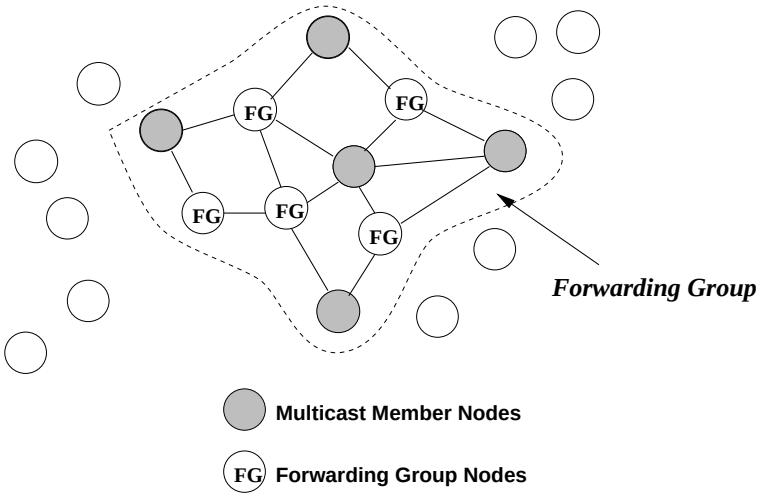


Figure 9.2: The forwarding group concept.

JOIN REPLY built upon matched entries. The JOIN REPLY is thus propagated by each forwarding group member until it reaches the multicast source via the shortest path. This process constructs (or updates) the routes from sources to receivers and builds a mesh of nodes, the *forwarding group*.

We have visualized the forwarding group concept in Figure 9.2. The forwarding group is a set of nodes in charge of forwarding multicast packets. It supports shortest paths between any member pairs. All nodes inside the *bubble* (multicast members and forwarding group nodes) forward multicast data packets. Note that a multicast receiver can also be a forwarding group node if it is on the path between a multicast source and another receiver. The mesh provides richer connectivity among multicast members compared to trees. Flooding redundancy among forwarding group helps overcome node displacements and channel fading. Hence, unlike trees, frequent reconfigurations are not required.

Figure 9.3 is an example to show the robustness of a mesh configuration. Three sources (S_1, S_2, and S_3) send multicast data packets to three receivers

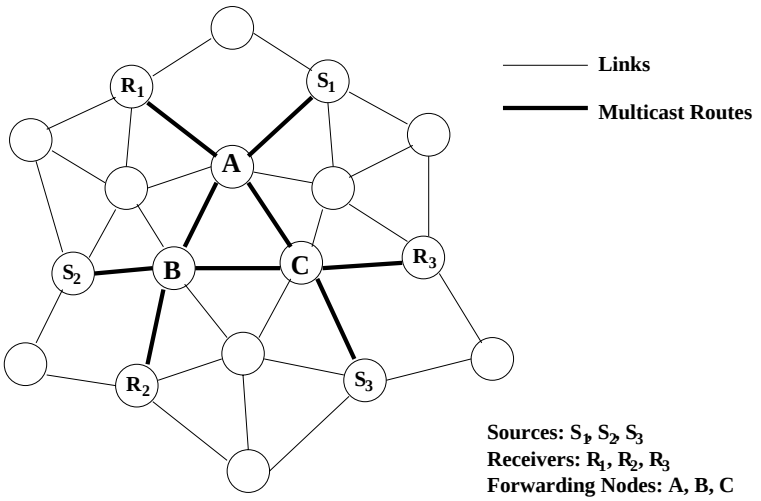


Figure 9.3: Why a mesh?

(R_1, R_2, and R_3) via three forwarding group nodes (A, B, and C). Suppose the route from S_1 to R_2 is <S_1-A-B-R_2>. In a tree configuration, if the link between nodes A and B breaks or fails, R_2 cannot receive any packets from S_1 until the tree is reconfigured. ODMRP, on the other hand, already has a redundant route <S_1-A-C-B-R_2> to deliver packets without going through the broken link between nodes A and B.

9.2 Example

Figure 9.4 is shown as an example of a JOIN REPLY forwarding process. Nodes S_1 and S_2 are multicast sources, and nodes R_1, R_2, and R_3 are multicast receivers. Nodes R_2 and R_3 send their JOIN REPLIES to both S_1 and S_2 via I_2, and R_1 sends its packet to S_1 via I_1 and to S_2 via I_2. When receivers send their JOIN REPLIES to next hop nodes, an intermediate node I_1 sets the `FG_Flag` and builds its own JOIN REPLY since there is a next node ID entry in the JOIN REPLY received from R_1 that matches its ID. Note that the JOIN REPLY built by I_1

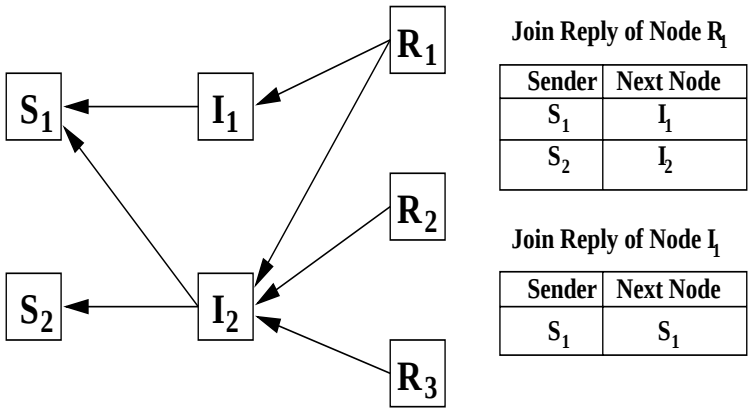


Figure 9.4: An example of a JOIN REPLY forwarding.

has an entry for sender S_1 but not for S_2 because the next node ID for S_2 in the received JOIN REPLY is not I_1. In the meantime, node I_2 sets the `FG_Flag`, constructs its own JOIN REPLY and sends it to its neighbors. Note that even though I_2 receives three JOIN REPLIES from the receivers, it broadcasts the JOIN REPLY only once because the second and third table arrivals carry no new source information. Channel overhead is thus reduced dramatically in cases where numerous multicast receivers share the same links to the source.

9.3 Data Forwarding

After the group establishment and route construction process, a multicast source can transmit packets to receivers via selected routes and forwarding groups. Periodic control packets are sent only when outgoing data packets are still present. When receiving a multicast data packet, a node forwards it only if it is not a duplicate and the setting of the `FG_Flag` for the multicast group has not expired. This procedure minimizes traffic overhead and prevents sending packets through stale routes.

9.4 Soft State

In ODMRP, no explicit control packets need to be sent to join or leave the group. If a multicast source wants to leave the group, it simply stops sending JOIN QUERY packets since it does not have any multicast data to send to the group. If a receiver no longer wants to receive from a particular multicast group, it removes the corresponding entries from its *Member Table* and does not transmit the JOIN REPLY for that group. Nodes in the forwarding group are demoted to non-forwarding nodes if not refreshed (no JOIN REPLIES received) before they timeout.

9.5 Selection of Timer Values

Timer values for route refresh interval and forwarding group timeout interval can have impacts on ODMRP performance. The selection of these soft state timers should be adaptive to network environment (e.g., traffic type, traffic load, mobility pattern, mobility speed, channel capacity, etc.). When small route refresh interval values are used, fresh route and membership information can be obtained frequently at the expense of producing more packets and causing network congestion. On the other hand, when large route refresh values are selected, even though less control traffic will be generated, nodes may not know up-to-date route and multicast membership. Thus in highly mobile networks, using large route refresh interval values can yield poor protocol performance. The forwarding group timeout interval should also be carefully selected. In networks with heavy traffic load, small values should be used so that unnecessary nodes can timeout quickly and not create excessive redundancy. In situations with high mobility, however, large values should be chosen so that more alternative paths can be

provided. It is important to note that the forwarding group timeout value must be larger (e.g., three to five times) than the value of route refresh interval.

9.6 Data Structures

Network hosts running ODMRP are required to maintain the following data structures.

9.6.1 Member Table

Each multicast receiver stores the source information in the *Member Table*. For each multicast group the node is participating in, the source ID and the time when the last JOIN QUERY is received from the source is recorded. If no JOIN QUERY is received from a source within the refresh period, that entry is removed from the *Member Table*.

9.6.2 Route Table

A *Route Table* is created on demand and is maintained by each node. An entry is inserted or updated when a non-duplicate JOIN QUERY is received. The node stores the destination (i.e., the source of the JOIN QUERY) and the next hop to the destination (i.e., the last node that propagated the JOIN QUERY). The *Route Table* provides the next hop information when transmitting JOIN REPLIES.

9.6.3 Forwarding Group Table

When a node is a forwarding group node of the multicast group, it maintains the group information in the *Forwarding Group Table*. The multicast group ID and

the time when the node was last refreshed is recorded.

9.6.4 Message Cache

The *Message Cache* is maintained by each node to detect duplicates. When a node receives a new JOIN QUERY or data, it stores the source ID and the sequence number of the packet. Note that entries in the *Message Cache* need not be maintained permanently. Schemes such as LRU (Least Recently Used) or FIFO (First In First Out) can be employed to expire and remove old entries and prevent the size of the *Message Cache* to be extensive.

9.7 Unicast Capability

One of the major strengths of ODMRP is its unicast routing capability. Not only can ODMRP coexist with any unicast routing protocol, it can also operate efficiently as an unicast routing protocol. Thus, a network equipped with ODMRP does not require a separate unicast protocol. Other ad hoc multicast routing protocols such as AMRoute [20], CAMP [50], RBM [35], and LAM [65] must be run on top of a unicast routing protocol. Moreover, some of the protocols, such as CAMP, RBM, and LAM, only work with certain underlying unicast protocols. In contrast, ODMRP offers the advantage of sharing the same optional software for both unicast and multicast operation. Chapter 12 describes the details of ODMRP unicast operation.

9.8 Summary

We have proposed ODMRP (On-Demand Multicast Routing Protocol) for a mobile ad hoc wireless network. ODMRP is based on mesh (instead of tree) forwarding. It applies on-demand (as opposed to periodic) multicast route construction and membership maintenance. The key advantages of ODMRP are:

- Low channel and storage overhead
- Usage of fresh and shortest routes
- Robustness to host mobility
- Maintenance and exploitation of multiple redundant paths
- Exploitation of the broadcast nature of wireless environments
- Unicast routing capability.

CHAPTER 10

Improving the Performance of ODMRP

As studied in Chapter 9, the major strengths of ODMRP are its simplicity. We can further improve its performance by several enhancements. In this chapter, we propose new techniques to enhance the effectiveness and efficiency of ODMRP. Our primary goals are the following:

- Improve adaptivity to node movement patterns
- Transmit control packets only when necessary
- Reconstruct routes in anticipation of topology changes
- Improve hop-by-hop transmission reliability
- Eliminate route acquisition latency
- Select stable routes.

10.1 Adapting the Refresh Interval via Mobility Prediction

ODMRP requires periodic flooding of JOIN QUERIES to build and refresh routes. Excessive flooding, however, is not desirable in ad hoc networks because of bandwidth constraints. Furthermore, flooding often causes congestion, contention,

and collisions. Finding the optimal flooding interval is critical in ODMRP performance. Here we propose a scheme that adapts the flooding interval to mobility patterns and speeds.[1] By utilizing the location and mobility information provided by GPS (Global Positioning System) [72], we predict the duration of time routes will remain valid.[2] With the predicted time of route disconnection, JOIN QUERIES are only flooded when route breaks of ongoing data sessions are imminent.

In our prediction method, we assume a free space propagation model [135], where the received signal strength solely depends on its distance to the transmitter. We also assume that all nodes in the network have their clock synchronized (e.g., by using the NTP (Network Time Protocol) [109] or the GPS clock itself). Therefore, if the motion parameters of two neighbors (e.g., speed, direction, radio propagation range, etc.) are known, we can determine the duration of time these two nodes will remain connected. Assume two nodes i and j are within the transmission range r of each other. Let (x_i, y_i) be the coordinate of mobile host i and (x_j, y_j) be that of mobile host j. Also let v_i and v_j be the speeds, and θ_i and θ_j $(0 \leq \theta_i, \theta_j < 2\pi)$ be the moving directions of nodes i and j, respectively. Then, the amount of time that they will stay connected, D_t, is predicted by:

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

where

$a = v_i \cos\theta_i - v_j \cos\theta_j$,

$b = x_i - x_j$,

$c = v_i \sin\theta_i - v_j \sin\theta_j$, and

[1] The mobility prediction scheme is proposed by my former colleague Dr. William W. Su. Since the scheme is part of the protocol, it is introduced in this dissertation with the permission of Dr. Su.

[2] Mobility speed and heading information can be obtained from GPS or the node's own instruments and sensors (e.g., campus, odometer, speed sensors, etc.).

$d = y_i - y_j$.

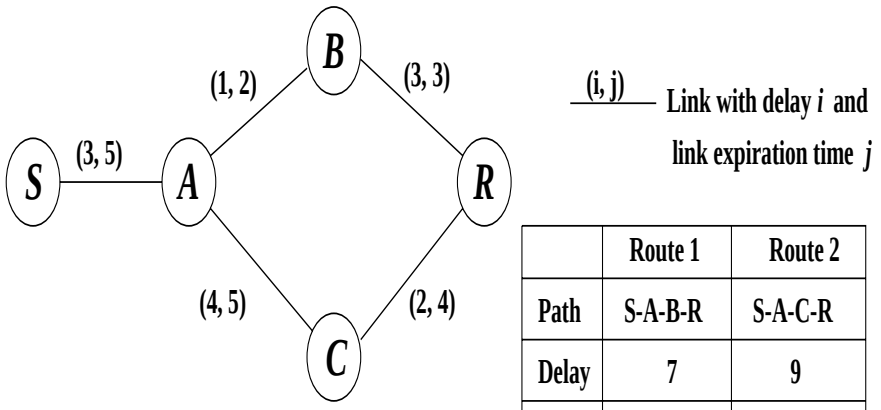
Note that when $v_i = v_j$ and $\theta_i = \theta_j$, D_t is set to ∞ without applying the above equation.

To utilize the information obtained from the prediction, extra fields must be added into JOIN QUERY and JOIN REPLY packets. When a source sends JOIN QUERIES, it appends its location, speed, and direction. It sets the MIN_LET (Minimum Link Expiration Time) field to the `MAX_LET_VALUE` since the source does not have any previous hop node. The next hop neighbor, upon receiving a JOIN QUERY, predicts the link expiration time between itself and the previous hop using the above equation. The minimum between this value and the MIN_LET indicated by the JOIN QUERY is included in the packet. The rationale is that as soon as a single link on a path is disconnected, the entire path is invalidated. The node also overwrites the location and mobility information field written by the previous node with its own information. When a multicast member receives the JOIN QUERY, it calculates the predicted LET of the last link of the path. The minimum between the last link expiration time and the MIN_LET value specified in the JOIN QUERY is the RET (Route Expiration Time). This RET value is enclosed in the JOIN REPLY and broadcasted. If a forwarding group node receives multiple JOIN REPLIES with different RET values (i.e., lies in paths from the same source to multiple receivers), it selects the minimum RET among them and sends its own JOIN REPLY with the chosen RET value attached. When the source receives JOIN REPLIES, it selects the minimum RET among all the JOIN REPLIES received. Then the source can build new routes by flooding a JOIN QUERY before the minimum RET approaches (i.e., route breaks). Note that JOIN REPLY need not be periodically transmitted by multicast receivers. Since sources flood JOIN QUERY only when needed, receivers only send JOIN REPLIES after receiving JOIN QUERIES.

In addition to the estimated RET value, other factors need to be considered when choosing the flooding interval of JOIN QUERIES. If the node mobility rate is high and the topology changes frequently, routes will expire quickly and often. The source may propagate JOIN QUERIES excessively and this excessive flooding can cause collisions and congestion, and clogs the network with control packets. Thus, the `MIN_REFRESH_INTERVAL` should be enforced to avoid control message overflow. On the other hand, if nodes are stationary or move slowly and link connectivity remains unchanged for a long duration of time, routes will hardly expire and the source will rarely send JOIN QUERIES. A few problems arise in this situation. First, if a node in the route suddenly changes its movement direction or speed, the predicted RET value becomes obsolete and routes will not be reconstructed in time. Second, when a non-member node which is located remotely to multicast members wants to join the group, it cannot inform the new membership or receive data until a JOIN QUERY is received. Hence, the `MAX_REFRESH_INTERVAL` should be set. The selection of the `MIN_REFRESH_INTERVAL` and the `MAX_REFRESH_INTERVAL` should be adaptive to network situations (e.g., traffic type, traffic load, mobility pattern, mobility speed, channel capacity, etc.).

10.2 Route Selection Criteria

In the basic ODMRP, a multicast receiver selects routes based on the minimum delay (i.e., routes taken by the first JOIN QUERY received). A different route selection method is applied when we use the mobility prediction. The idea is inspired by the Associativity-Based Routing (ABR) protocol [159] which chooses associatively stable routes. In our new algorithm, instead of using the minimum delay path, we can choose a route that is the most stable (i.e., the one with the



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

Figure 10.1: Route selection example.

largest RET). To select a route, a multicast receiver must wait for an appropriate amount of time after receiving the first JOIN QUERY so that all possible routes and their RETs will be known. The receiver then chooses the most stable route and broadcasts a JOIN REPLY. Route breaks will occur less often and the number of JOIN QUERY propagation will reduce because stable routes are used. An example showing the difference between two route selection algorithms is presented in Figure 10.1. Two routes are available from the source S to the receiver R. Route 1 has a path of $<S$-A-B-$R>$ and route 2 has a path of $<S$-A-C-$R>$. If the minimum delay is used as the route selection metric, the receiver node R selects route 1. Route 1 has a delay of 7 $(3 + 1 + 3 = 7)$ while route 2 has a delay of 9 $(3 + 4 + 2 = 9)$. Since the JOIN QUERY that takes route 1 reaches the receiver first, node R chooses route 1. If the stable route is selected instead, route 2 is chosen by the receiver. The route expiration time of route 1 is 2 $(\min(5, 2, 3) = 2)$ while that of route 2 is 4 $(\min(5, 5, 4) = 4)$. The receiver selects the route with the maximum RET, and hence route 2 is selected. We will evaluate different route selection methods by simulation in Section 10.6.

10.3 Reliability

The reliable transmission of JOIN REPLIES plays an important role in establishing and refreshing multicast routes and forwarding groups. Hence, if JOIN REPLIESS are not properly delivered, effective multicast routing cannot be achieved by ODMRP. The IEEE 802.11 MAC (Medium Access Control) protocol [60], which is the emerging standard in wireless networks, performs reliable transmission by retransmitting the packet if no acknowledgment is received. However, if the packet is broadcasted, no acknowledgments or retransmissions are sent. In ODMRP, the transmission of JOIN REPLIES are often broadcasted to more than one upstream neighbors since we are handling multiple sources (e.g., see the JOIN REPLY from node R_1 in Figure 9.4). In such cases, the hop-by-hop verification of JOIN REPLY delivery and the retransmission cannot be handled by the MAC layer. It must be done indirectly by ODMRP. Another option for reliable delivery is to subdivide the JOIN REPLY into separate sub-tables, one for each distinct next node. In Figure 9.4 for example, the JOIN REPLY at node R_1 is split into two JOIN REPLIES, one for neighbor I_1 and one for neighbor I_2. These JOIN REPLIES are separately unicasted using a reliable MAC protocol such as IEEE 802.11 or MACAW [19]. Since the number of neighbors is generally limited (typically, about six neighbors in the optimum in a multihop network [75]), the scheme still scales well to large number of sources. This option can actually be used as backup to the passive acknowledgment option as discussed below.

We adopt a scheme that was used in [71]. Figure 10.2 is shown to illustrate the mechanism. When node B transmits a packet to node C after receiving a packet from node A, node A can hear the transmission of node B if it is within B's radio propagation range. Hence, the packet transmission by node B to node C is used as a *passive acknowledgment* to node A. We can utilize this passive

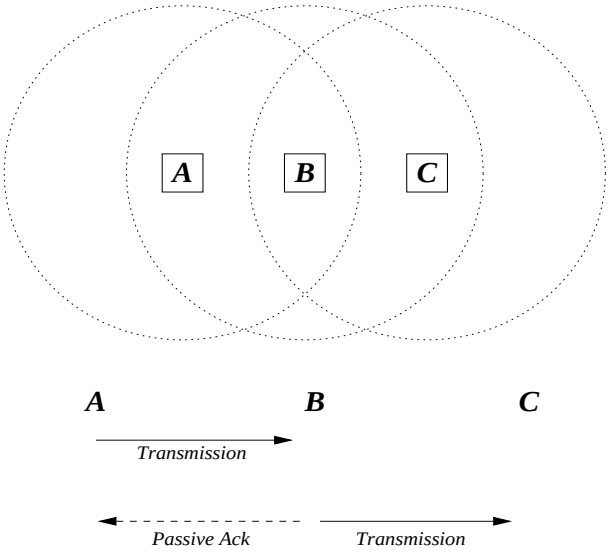


Figure 10.2: Passive acknowledgments.

acknowledgment to verify the delivery of a JOIN TABLE. Note that the source itself must send an active acknowledgment to the previous hop since it does not have any next hop to send a JOIN REPLY to unless it is also a forwarding group node for other sources.

Considering the case in Figure 9.4 again, we note that once the nodes I_1 and I_2 receive the JOIN TABLE from node R_1, they will construct and forward their own JOIN TABLES to next hops (in this case, sources S_1 and S_2). In transmitting their JOIN TABLES, nodes I_1 and I_2 may overlap with each other. If I_1 and I_2 are within receiving range, they will recover because of the carrier sense feature in CSMA (Carrier Sense Multiple Access) [76]. However, if they are out of range, they will be unaware of the *hidden terminal* condition [156] of node R_1, which cannot hear the (overlapped) passive acknowledgments. Thus, a node may not hear the passive acknowledgments of its upstream neighbor because of conflicts due to the hidden terminal problem. It will also not hear the passive

acknowledgment if the upstream neighbor has moved away. In either case, when no acknowledgment is received within the timeout interval, the node retransmits the message. Note that the node may get acknowledgments from some, but not all upstream neighbors. As an option, the retransmission could be carried out in unicast mode, to selected neighbors, with reduced sub-tables. If packet delivery cannot be verified after an appropriate number of retransmissions, the node considers the route to be invalidated. At this point, the most likely cause of route failure is the fact that a node on the route has failed or has moved out of range. An alternate route must be found *on the spot*. The node thus broadcasts a message to its neighbors specifying that the next hop to a set of sources cannot be reached. Upon receiving this packet, each neighbor builds and unicasts the JOIN REPLY to its next hop if it has a route to the multicast sources. If no route is known, it simply broadcasts the packet specifying the next hop is not available. In both cases, the node sets its `FG_FLAG`. In practical implementations, this redundancy is sufficient to establish alternate paths until a more efficient route is established during the next refresh phase. The `FG_FLAG` setting of every neighbor may create excessive redundancy, but most of these settings will expire because only necessary forwarding group nodes will be refreshed in the next JOIN REPLY propagation phase.

10.4 Elimination of Route Acquisition Latency

The major drawback of on-demand routing protocols is the delay required to obtain a route. This route acquisition latency makes on-demand protocols less attractive in networks where real-time traffic is exchanged. In the basic ODMRP, when no multicast route information is known by the source, data transmission is delayed for a certain period of time. In contrast to unicast routing, the selection

of the waiting time is not straightforward. In unicast, the source can send data as soon as a ROUTE REPLY is received. In ODMRP, however, the data transmission cannot be made immediately after receiving the first JOIN REPLY since routes to receivers that are farther away may not yet have been established.

To eliminate these problems, when a source has data to send but no multicast route is known, it floods the data instead of the JOIN QUERY. The periodic transmission of JOIN QUERY is also replaced by data.[3] Basically, JOIN DATA becomes a JOIN QUERY with data payload attached. Thus, the flooding of JOIN DATA achieves data delivery in addition to constructing and refreshing the routes. Although the size of the flooded packet is larger compared to JOIN QUERIES, route acquisition latency is eliminated.

10.5 Simulation Model and Methodology

10.5.1 Simulation Environment

The simulator was implemented within the Global Mobile Simulation (GloMoSim) library [160]. Our simulation modeled a network of 50 mobile hosts placed randomly within a 1000 meter $\times$ 1000 meter area. Radio propagation range for each node was 250 meters and channel capacity was 2 Mb/s. Each simulation executed for 600 seconds of simulation time. Multiple runs with different seed numbers were conducted for each scenario and collected data were averaged over those runs. A free space propagation model [135] with a threshold cutoff was used in our experiments. In the radio model, we assumed the ability of a radio to lock on to a sufficiently strong signal in the presence of interfering signals, i.e., radio capture. If the capture ratio (the minimum ratio of an arriving packet's signal

[3]To differentiate between the flooded data that performs the JOIN QUERY role and the ordinary data, we term the flooded data packet as JOIN DATA, only in this chapter.

strength relative to those of other colliding packets) [135] was greater than the predefined threshold value, the arriving packet was received while other interfering packets were dropped. The IEEE 802.11 Distributed Coordination Function (DCF) [60] was used as the medium access control protocol. A traffic generator was developed to simulate constant bit rate sources. The size of data payload was 512 bytes. Each node moved constantly with the predefined speed. Moving direction was selected randomly, and when nodes reached the simulation terrain boundary, they bounced back and continued to move. One multicast group with one source was simulated. The multicast members and the source were chosen randomly with uniform probabilities. Members joined the group at the start of the simulation and remained as members throughout the simulation.

10.5.2 Methodology

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

1. *Scheme A*: the basic ODMRP as specified in Chapter 9
2. *Scheme B*: the enhanced ODMRP that uses the minimum delay as the route selection metric
3. *Scheme C*: the enhanced ODMRP that uses the route expiration time as the route selection metric.

Both enhanced schemes included reliable transmission and route acquisition latency elimination features. The protocols were evaluated as a function of (i) speed, and (ii) multicast group size. In the first set of experiments, the size of the multicast group was set constant to 10 and speed was varied from 0 km/hr to 72 km/hr. In the second set of simulations, node mobility speed was constant

at 18 km/hr and the multicast group size was varied from two (unicast) to 20. The metrics of interest are:

- **Packet delivery ratio**: The number of data packets actually received by multicast members over the number of data packets supposed to be received by multicast members.
- **End-to-end delay**: The time elapsed between the instant when the source has data packet to send and the instant when the destination receives the data. Note that if no route is available, the time spent in building a route (i.e., route acquisition latency) is included in the end-to-end delay.
- **Control overhead**: The total control bytes transmitted. Bytes of data packet and JOIN DATA headers in addition to bytes of control packets (i.e., JOIN QUERIES, JOIN REPLIES, active acknowledgments) are calculated as control overhead.
- **Number of total packets transmitted per data packet delivered**: The number of all packets (i.e., data and control packets) transmitted divided by data packet delivered to destinations. This measure shows the efficiency in terms of channel access and is very important in ad hoc networks since link layer protocols are typically contention-based.

10.6 Simulation Results

10.6.1 Packet Delivery Ratio

The packet delivery ratio as a function of the mobility speed and the multicast group size is shown in Figure 10.3 and Figure 10.4, respectively. We can see from Figure 10.3 that as speed increases, the routing effectiveness of *scheme A* degrades

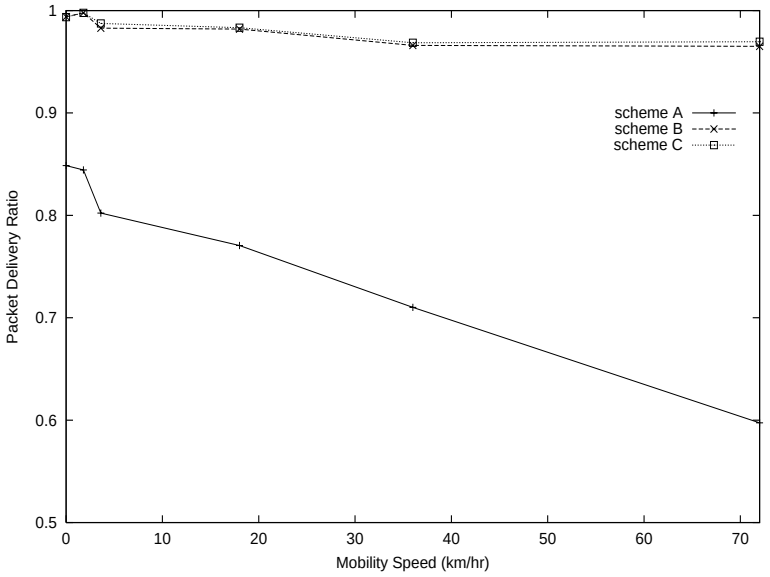


Figure 10.3: Packet delivery ratio as a function of speed.

rapidly compared to *schemes B* and *C*. Both *schemes B* and *C* have very high delivery ratios of over 96% regardless of speed. As the routes are reconstructed in advance of topology changes, most data are delivered to multicast receivers without being dropped. In *scheme A*, however, JOIN QUERIES and JOIN REPLIES are transmitted periodically (every 400 ms and 180 ms, respectively) without adapting to mobility speed and direction. Frequent flooding resulted in collisions and congestion, leading to packet drops even in low mobility rates. At high speed, routes that are taken at the JOIN QUERY phase may already be broken when JOIN REPLIES are propagated. In *scheme A*, nodes do not verify the reception of JOIN REPLIES transmitted. Most JOIN REPLIES failed to reach the source and establish the forwarding group. Thus, when data is sent by the source, the multicast route is not properly built and packets can not be delivered. Both *scheme B* and *scheme C* enforce reliable transmissions of JOIN REPLIE. Routes and forwarding group nodes are established and refreshed appropriately even in high mobility situations and the schemes proved to be robust to the mobility

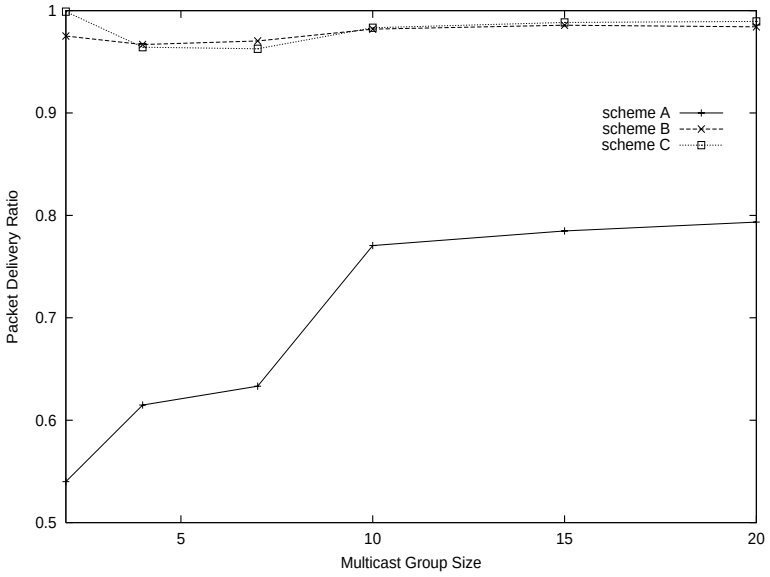


Figure 10.4: Packet delivery ratio as a function of number of multicast members.

speed.

In Figure 10.4, *scheme B* and *scheme C* outperform *scheme A* again. The result shows that our enhanced protocols are robust to multicast group size in addition to mobility speed. *Scheme A*'s performance improves as the size becomes larger. As the number of receivers increases, the number of forwarding group nodes increases accordingly. Hence, the connectivity of multicast group members becomes richer and the redundancy of the paths helps delivering data to destinations.

10.6.2 End-to-End Delay

Figure 10.5 and Figure 10.6 show the end-to-end delay of each scheme. *Scheme B* and *scheme C* have shorter delay compared with *scheme A*. In *scheme A*, sources flood JOIN QUERIES and must wait for a certain amount of time to send data until routes are established among multicast members. In *schemes B* and *C*,

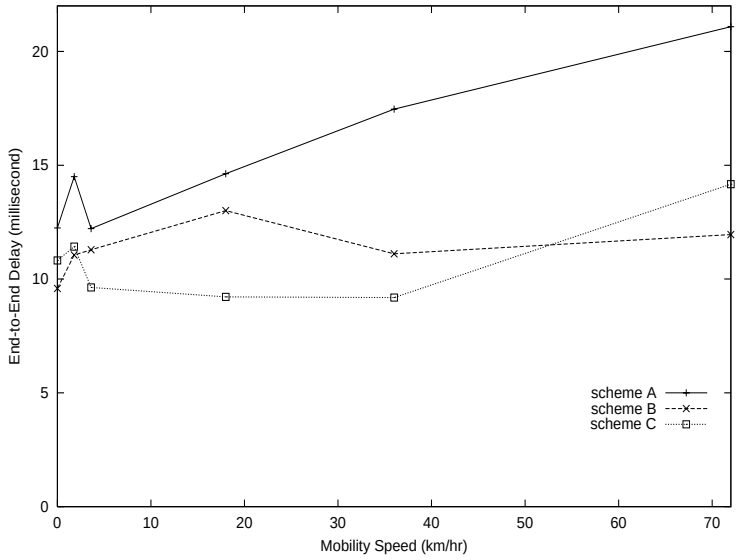


Figure 10.5: End-to-end delay as a function of speed.

on the contrary, sources flood JOIN DATA immediately even before routes and forwarding group are constructed. The route acquisition latency is eliminated and packets are delivered to receivers in shorter delays. One might be surprised to see that the delay of *scheme B* which uses the minimum delay route is larger than that of *scheme C* which uses the stable (and possibly longer delay) route. Even though the route taken by JOIN DATA is the shortest delay route at that instant, it may not be the minimum delay route later on as nodes move. In addition, compared to stable routes, the minimum delay routes break more frequently and data may need to traverse through longer redundant routes formed by forwarding group nodes.

10.6.3 Control Overhead

Figure 10.7 shows the control byte overhead as a function of mobility speed for each protocol. Remember that the transmission of control packets in *scheme A*

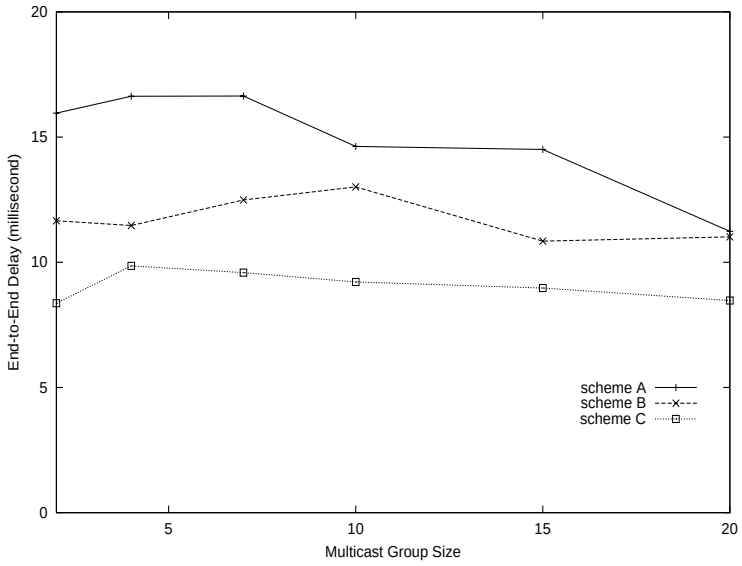


Figure 10.6: End-to-end delay as a function of number of multicast members.

is time triggered only without adapting to mobility speed. Hence, the amount of control overhead does not increase as the mobility speed increases. Actually, control overhead decreases as nodes move faster. As JOIN REPLIES are less likely to reach the target nodes in a highly mobile environment, the JOIN REPLY propagations by the next nodes are triggered less. Furthermore, data packets (whose header is calculated as control overhead), are transmitted less because forwarding group nodes and routes are not established or refreshed appropriately as the speed increases. On the other hand, the overhead of *schemes B* and *C* goes up as mobility speed increases. Since mobility prediction is used to adapt to mobility speed, more JOIN DATA and JOIN REPLIES are sent when mobility is high. In addition, JOIN REPLY retransmission and active acknowledgment propagation also increase with mobility and add to the control overhead. It is important to observe that the overhead of *scheme B* and *scheme C* are both significantly less than that of *scheme A* in low mobility cases because control packets are transmitted only when necessary in *schemes B* and *C*. The enhanced schemes have more

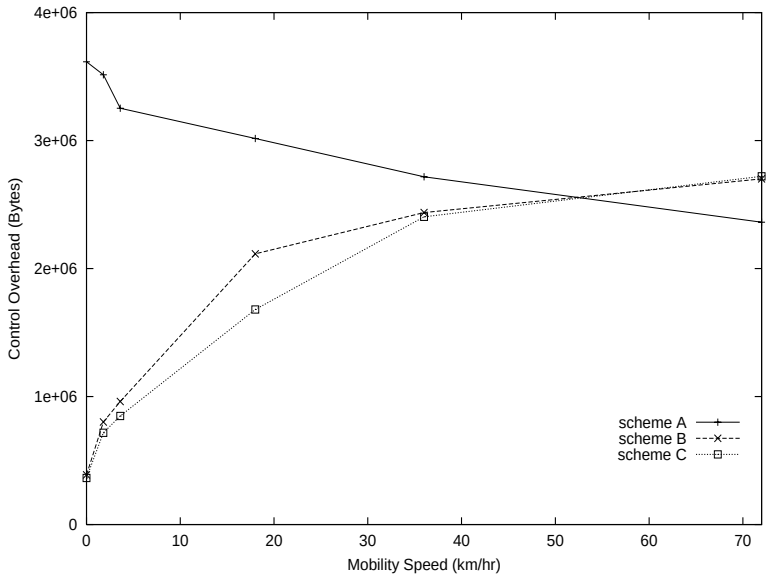


Figure 10.7: Control overhead as a function of speed.

overhead when nodes move fast, but the extra control packets are used efficiently in delivering data (see Figure 10.3). When comparing *scheme B* with *scheme C*, we can see that *scheme B* yields more overhead in low mobility while both schemes produce nearly equal amount of overhead in high mobility. Since *scheme C* chooses a stable route, JOIN DATA are flooded less often. However, when nodes move relatively fast (e.g., 72 km/hr in our simulation), routes are broken often and links will remain connected for a short duration of time. Sources are thus likely to use `MIN_REFRESH_INTERVAL` and the overhead incurred by both *schemes B* and *C* becomes almost identical.

In Figure 10.8, control overhead of all schemes increases when the number of multicast group increases. As there are more multicast receivers, more JOIN REPLIES are built and propagated. *Scheme B* and *scheme C* have much less overhead than that of *scheme A*. JOIN QUERIES and JOIN REPLIES are sent periodically in *scheme A*, while JOIN DATA and JOIN REPLIES are sent only in advance of topology changes in enhanced schemes. As expected, *scheme C*

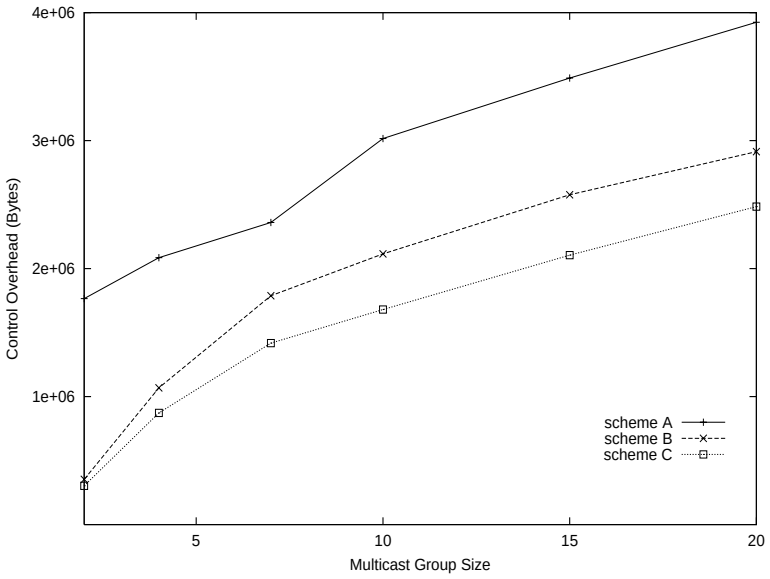


Figure 10.8: Control overhead as a function of number of multicast members.

further improves *scheme B*. The number of control packet transmissions are less as stable routes are used in *scheme C*.

10.6.4 Number of Total Packets Transmitted per Data Packet Delivered

The number of total packets (i.e., JOIN QUERIES, JOIN REPLIES, JOIN DATA, Data, and active acknowledgments) transmitted per data packet delivered is presented in Figure 10.9 and 10.10. We have mentioned previously that this measure indicates the channel access efficiency. We can see the improvements made by enhanced schemes from the results. In Figure 10.9, the number for *scheme A* remains relatively constant to mobility speed. As shown in Figures 10.3 and 10.7, the number of data packets delivered and the amount of control bytes transmitted both decrease as mobility increases. The number for *scheme A* thus remains almost unchanging. The measures for *scheme B* and *scheme C* gradually in-

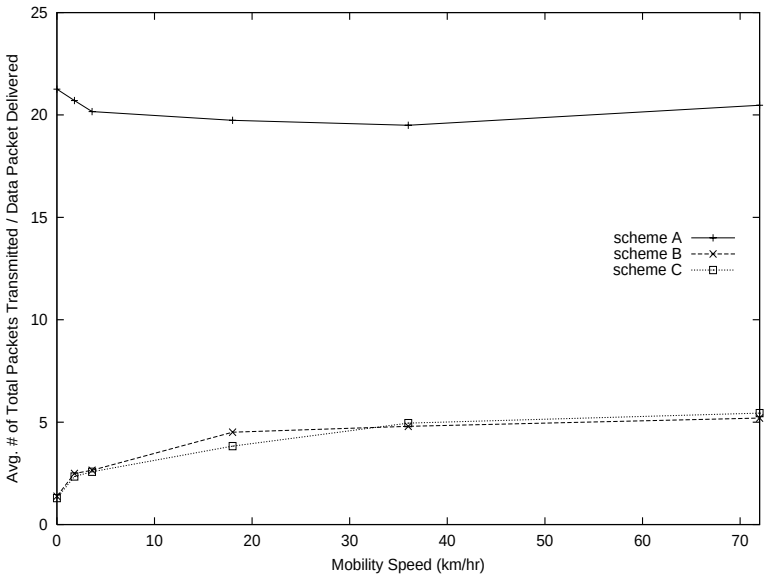


Figure 10.9: Number of total packets transmitted per data packet delivered as a function of speed.

crease with mobility speed. Both schemes deliver a high portion of the data to destinations regardless of speed (see Figure 10.3) and the number of data packets delivered remains similar. However, more control packets must be sent in order to adapt to node mobility speed, and thus the total number of packets transmitted increases with speed.

In Figure 10.10, the number of all packets transmitted per data packet delivered decreases as the group size becomes larger for all schemes. This result is expected as the number of multicast members increases, the number of data packets received by members increases accordingly. Again, *scheme B* and *scheme C* have greatly improved the efficiency of *scheme A*.

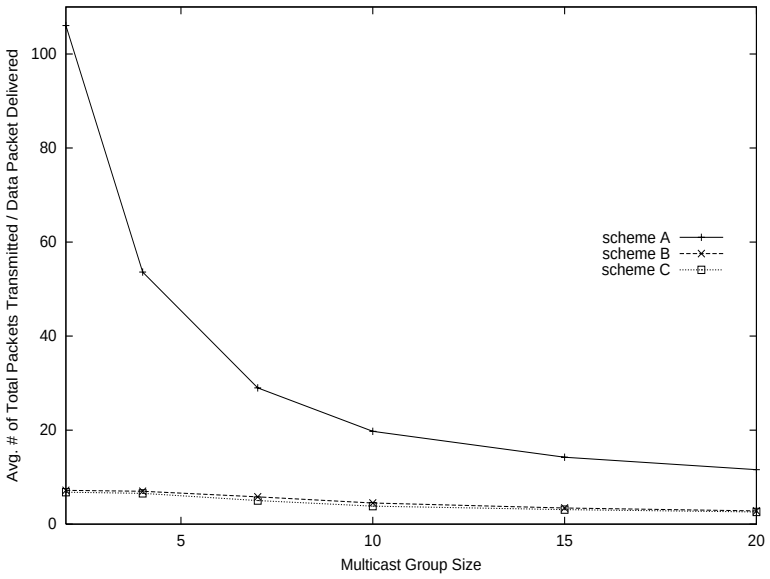


Figure 10.10: Number of total packets transmitted per data packet delivered as a function of number of multicast members.

10.7 Conclusion

We presented new techniques to improve the performance of ODMRP. By using the mobility and link connectivity prediction, routes and forwarding groups are reconstructed in anticipation of topology changes. This adaptive selection of the refresh interval avoids the transmission of unnecessary control packets and the resulting bandwidth wastage. We have applied a new route selection algorithm to choose routes that will stay valid for the longest duration of time. The usage of stable routes further reduces the control overhead. Passive acknowledgments and retransmissions have been used to improve the reliable delivery of JOIN REPLIES. The improved reliability plays a factor in protocol enhancement since the delivery of JOIN REPLIES is critical in establishing the routes and forwarding group nodes. We have also introduced a method to eliminate the route acquisition latency.

Simulation results showed that our new methods improved the basic scheme

significantly. More data packets were delivered to destinations, less control packets were produced in low mobility, control packets were utilized more efficiently in high mobility, and end-to-end delay was shorter. The enhanced ODMRP was scalable, robust to host mobility, and efficient in channel access.

CHAPTER 11

Performance Evaluation of Multicast Routing Protocols

In recent years, a number of new multicast protocols of different styles have been proposed for ad hoc networks. However, systematic performance evaluations and comparative analysis of these protocols in a common realistic environment has not yet been performed. In this chapter, we simulate a set of representative wireless ad hoc multicast protocols and evaluate them in various network scenarios. We provide quantitative performance analysis of five protocols with different characteristics: AMRoute, ODMRP, AMRIS, CAMP, and flooding. The five multicast routing protocols were simulated in diverse network scenarios. We studied the impact of mobility on performance by varying the speed of network hosts. We varied the number of data packet senders to emulate a variety of different multicast applications. One source to many receivers can correspond to battlefield data dissemination. Many sources to many receivers can correspond to search and rescue team communication. Different multicast group member sizes were simulated to investigate the impact on performance. Various traffic loads were also applied to study how traffic patterns influence multicast protocol performance. The relative strengths, weaknesses, and applicability of each multicast protocol to diverse situations are studied and discussed.

11.1 Multicast Protocols Review

In this section, we introduce the ad hoc wireless multicast protocols we have selected. Basic operation procedures and implementation choices are described.

11.1.1 Adhoc Multicast Routing

AMRoute [20] is a tree based protocol. It creates a bidirectional shared multicast tree using unicast tunnels to provide connections between multicast group members. Each group has at least one logical core that is responsible for member and tree maintenance. Initially, each group member declares itself as a core for its own group of size one. Each core periodically floods JOIN-REQS (using an expanding ring search) to discover other disjoint mesh segments for the group. When a member node receives a JOIN-REQ from a core of the same group but a different mesh segment, it replies with a JOIN-ACK and marks that node as a mesh neighbor. The node that receives a JOIN-ACK also marks the sender of the packet as its mesh neighbor. After the mesh creation, each core periodically transmits TREE-CREATE packets to mesh neighbors in order to build a shared tree. When a member node receives a non-duplicate TREE-CREATE from one of its mesh links, it forwards the packet to all other mesh links. If a duplicate TREE-CREATE is received, a TREE-CREATE-NAK is sent back along the incoming link. The node receiving a TREE-CREATE-NAK marks the link as mesh link instead of tree link. The nodes wishing to leave the group send the JOIN-NAK to the neighbors and do not forward any data packets for the group.

The key characteristic of AMRoute is its usage of virtual mesh links to establish the multicast tree. Therefore, as long as routes between tree members exist via mesh links, the tree need not be readjusted when network topology changes.

Table 11.1: Parameter values for AMRoute.

JOIN-REQ interval	60 sec
JOIN-REQ interval when no group members are connected to the core	5 sec
TREE-CREATE interval	20 sec
TREE-CREATE timeout	40 sec
Core resolution algorithm	Highest ID

Non-members do not forward data packets and need not support any multicast protocol. Thus, only the member nodes that form the tree incurs processing and storage overhead. AMRoute relies on an underlying unicast protocol to maintain connectivity among member nodes and any unicast protocol can be used. The major disadvantage of the protocol is that it suffers from temporary loops and creates non-optimal trees when mobility is present.

Table 11.1 shows the AMRoute parameter values used in our experiments. The implementation followed the specification in [20].

11.1.2 On-Demand Multicast Routing Protocol

ODMRP [91, 94] creates a mesh of nodes (the "forwarding group") which forward multicast packets via flooding (within the mesh), thus providing path redundancy. ODMRP is an on-demand protocol, thus it does not maintain route information permanently. It uses a soft state approach in group maintenance. Member nodes are refreshed as needed and do not send explicit leave messages.

In ODMRP, group membership and multicast routes are established and updated by the source on demand. Similar to on-demand unicast routing protocols,

a request phase and a reply phase comprise the protocol. When multicast sources have data to send, but do not have routing or membership information, they flood a JOIN QUERY packet. When a node receives a non-duplicate JOIN QUERY, it stores the upstream node ID (i.e., backward learning) and rebroadcasts the packet. When the JOIN DATA packet reaches a multicast receiver, the receiver creates a JOIN REPLY and broadcasts to the neighbors. When a node receives a JOIN REPLY, it checks if the next node ID of one of the entries matches its own ID. If it does, the node realizes that it is on the path to the source and thus is part of the forwarding group. It then broadcasts its own JOIN REPLY built upon matched entries. The JOIN REPLY is thus propagated by each forwarding group member until it reaches the multicast source via the shortest path. This process constructs (or updates) the routes from sources to receivers and builds a mesh of nodes, the *forwarding group*. Multicast senders refresh the membership information and update the routes by sending JOIN QUERY periodically.

In networks where GPS (Global Positioning System) [72] is available, ODMRP can be made adaptive to node movements by utilizing mobility prediction [153]. By using location and mobility information supported by GPS, route expiration time can be estimated and receivers can select the path that will remain valid for the longest time. With the mobility prediction method, sources can reconstruct routes in anticipation of route breaks. This way, the protocol becomes more resilient to mobility. The price is, of course, the cost and additional weight of GPS. The details of mobility prediction and the procedure are described in [94].

The data transfer phase is identical for both versions. Nodes forward the data if they are forwarding nodes and the packet they receive is not a duplicate. Since all forwarding nodes relay data, redundant paths (when they exist) can help deliver data when the primary path becomes disconnected because of mo-

Table 11.2: Parameter values for ODMRP.

JOIN QUERY refresh interval	3 sec
Acknowledgment timeout for JOIN REPLY	25 msec
Maximum JOIN REPLY retransmission	3

bility. Another unique property of ODMRP is its unicast capability. Not only can ODMRP coexist with any unicast routing protocol, it can also operate very efficiently as unicast routing protocol. Thus, a network equipped with ODMRP does not require a separate unicast protocol.

The specification in [94] was used in our implementation. For consistency with comparison, we used the version *without* mobility prediction. ODMRP parameter values used are shown in Table 11.2.

11.1.3 Ad hoc Multicast Routing protocol utilizing Increasing id-numberS

AMRIS [167] establishes a shared tree for multicast data forwarding. Each node in the network is assigned a multicast session ID number. The ranking order of ID numbers is used to direct the flow of multicast data. Like ODMRP, AMRIS does not require a separate unicast routing protocol.

Initially, a special node called Sid broadcasts a NEW-SESSION packet. The NEW-SESSION includes the Sid's msm-id (multicast session member id). Neighbor nodes, upon receiving the packet, calculate their own msm-ids which are larger than the one specified in the packet. The msm-ids thus increase as they radiate from the Sid. The nodes rebroadcast the NEW-SESSION message with the msm-id replaced by their own msm-ids. Each node is required to broadcast

Table 11.3: Parameter values for AMRIS.

Beacon interval	1 sec
Max allowed beacon losses	3
NEW SESSION lifetime	3 sec
Acknowledgment timeout for JOIN-REQ	2 sec
Random broadcast jitter time	50 msec

beacons to its neighbors. The beacon message contains the node id, msm-id, membership status, registered parent and child's ids and their msm-ids, and partition id. A node can join a multicast session by sending a JOIN-REQ. This JOIN-REQ is unicasted to a potential parent node with a smaller msm-id than the node's msm-id. The node receiving the JOIN-REQ sends back a JOIN-ACK if it already is a member of the multicast session. Otherwise, it sends a JOIN-REQ.PASSIVE to its potential parent. If a node fails to receive a JOIN-ACK or receives a JOIN-NAK after sending a JOIN-REQ, it performs "Branch Reconstruction (BR)." The BR process is executed in an expanding ring search until the node succeeds in joining the multicast session.

AMRIS detects link disconnection by a beaconing mechanism. If no beacons are heard for a predefined interval of time, the node considers the neighbor to have moved out of radio range. If the former neighbor is a parent, the node must rejoin the tree by sending a JOIN-REQ to a new potential parent. If the node fails to join the session or no qualified neighbors exist, it performs the BR process.

Data forwarding in done by the nodes in the tree. Only the packets from the registered parent or registered child are forwarded. Hence, if the tree link breaks, the packets are lost until the tree is reconfigured.

Our AMRIS implementation followed the specification in [167]. The AMRIS

parameter values are shown in Table 11.3.

11.1.4 Core-Assisted Mesh Protocol

CAMP [50, 51, 101, 102] supports multicasting by creating a shared mesh structure. All nodes in the network maintain a set of tables with membership and routing information. Moreover, all member nodes maintain a set of caches that contain previously seen data packet information and unacknowledged membership requests. CAMP classifies nodes in the network as duplex or simplex members, or non-members. Duplex members are full members of the multicast mesh, while simplex members are used to create one-way connections between sender-only nodes and the rest of the multicast mesh. "Cores" are used to limit the flow of JOIN REQUEST packets.

CAMP consists of mesh creation and maintenance procedures. A node wishing to join a multicast mesh first consults a table to determine whether it has neighbors which are already members of the mesh. If so, the node announces its membership via a CAMP UPDATE. Otherwise, the node either propagates a JOIN REQUEST towards one of the multicast group "cores," or attempts to reach a member router by an expanding ring search of broadcast requests. Any duplex member of the node can respond with a JOIN ACK, which is propagated back to the source of the request.

Periodically, a receiver node reviews its packet cache in order to determine whether it is receiving data packets from those neighbors which are on the reverse shortest path to the source. If not, the node sends either a HEARTBEAT or a PUSH JOIN message towards the source along the reverse shortest path. This process ensures that the mesh contains all such reverse shortest paths from all receivers to all senders. The nodes also periodically choose and refresh their selected

Table 11.4: Parameter values for CAMP.

Number of cores in the network	1
Beacon interval	3 sec
Update interval	3 sec
Age out anchor timeout	45 sec
Heartbeat interval	15 sec
Request retransmission interval	9 sec
Max number of JOIN REQUEST retransmission	3

"anchors" to the multicast mesh by broadcasting updates. These anchors are neighbor nodes which are required to re-broadcast any non-duplicate data packets they receive. A node is allowed to discontinue anchoring neighbor nodes which are not refreshing their connections. It can then leave the multicast mesh if it is not interested in the multicast session and is not required as anchor for any neighboring node.

CAMP relies on an underlying unicast routing protocol which guarantees correct distances to all destinations within finite time. Routing protocols that are based on the Bellman-Ford algorithm cannot be used with CAMP, and CAMP needs to be extended in order to work with on-demand routing protocols.

Our implementation of CAMP followed the specification in [50]. Table 11.4 shows the CAMP parameter values used in our simulation. Periodic beacon interval is three seconds, but the beacon is sent only when no packet has been transmitted during the beacon interval.

Table 11.5: Summary of protocols.

Protocols	**AMRoute**	**ODMRP**	**AMRIS**	**CAMP**	**Flooding**
Configuration	Tree	Mesh	Tree	Mesh	Mesh
Loop-Free	No	Yes	Yes	Yes	Yes
Dependency on Unicast Protocol	Yes	No	No	Yes	No
Periodic Messaging	Yes	Yes	Yes	Yes	No
Control Packet Flood	Yes	Yes	Yes	No	No

11.1.5 Protocols Summary

Table 11.5 summarizes key characteristics and properties of the protocols we simulated. Note that ODMRP requires periodic messaging (JOIN QUERY) only when sources have data packets to send.

11.2 Simulation Model and Methodology

The simulator for evaluating routing protocols was implemented within the GloMoSim library [160]. Our simulation modeled a network of 50 mobile hosts placed randomly within a 1000 meter $\times$ 1000 meter area. Radio propagation range for each node was 250 meters and channel capacity was 2 Mb/s. There were no network partitions throughout the simulation and the average number of neighbors for each node was 6.82. Each simulation executed for 600 seconds of simulation time. Multiple runs with different seed numbers were conducted for each scenario and collected data was averaged over those runs.

11.2.1 Channel and Radio Model

A free space propagation model [135] with a threshold cutoff was used in our experiments. In the free space model, the power of a signal attenuates as $1/d^2$ where d is the distance between radios. In the radio model, we assumed the ability of a radio to lock onto a sufficiently strong signal in the presence of interfering signals, i.e., radio capture. If the capture ratio (the ratio of an arriving packet's signal strength over the sum of all colliding packets) [135] was greater than a predefined threshold value, the packet was received while all other interfering packets were dropped.

11.2.2 Medium Access Control Protocol

The IEEE 802.11 MAC with Distributed Coordination Function (DCF) [60] was used as the MAC protocol. DCF is the mode which allows mobiles to share the wireless channel in an ad hoc configuration. The specific access scheme is Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) with acknowledgments. Optionally, the nodes can make use of Request To Send/Clear To Send (RTS/CTS) channel reservation control frames for unicast, virtual carrier sense, and fragmentation of packets larger than a given threshold. By setting timers based upon the reservations in RTS/CTS packets, the virtual carrier sense augments the physical carrier sense in determining when mobile nodes perceive that the medium is busy. Fragmentation is useful in the presence of high bit error and loss rates, as it reduces the size of the data units that need to be retransmitted.

In our experiments, we employed RTS/CTS exclusively for unicast control packets directed to specific neighbors (e.g., replies). All other transmissions use CSMA/CA. We chose this configuration to minimize the frequency and deleterious effects of collisions over the wireless medium. We did not employ fragmen-

tation because our data packets were small enough that the additional overhead would reduce overall network throughput.

11.2.3 Multicast Protocols

When implementing the multicast protocols, we followed the specifications of each protocol as defined in the published literature. We directly queried the protocol designers about details which were not specified in the publications (e.g., various timer values, core selection algorithm, etc.). ODMRP and AMRIS do not require underlying unicast protocol to operate, but AMRoute and CAMP do. While AMRoute can work with any protocol, the designers of CAMP specifically state that it can operate only with certain unicast protocols [50]. We have implemented one of those protocols, WRP [114], a distance-vector based unicast routing protocol developed by the same group which developed CAMP. For a fair comparison, WRP was used as the underlying unicast protocol also for AMRoute.

11.2.4 Traffic Pattern

traffic generator was developed to simulate constant bit rate sources. The size of data payload was 512 bytes. The senders were chosen randomly among multicast members who in turn were chosen with uniform probability among 50 network hosts. The member nodes join the multicast session at the beginning of the simulation and remain as members throughout the simulation. Hence, the simulation experiments do not test/account for the overhead produced in the session leave process.

11.2.5 Metrics

We have used the following metrics in comparing protocol performance.

- **Packet delivery ratio**: The ratio of the number of data packets actually delivered to the destinations versus the number of data packets supposed to be received. This number presents the effectiveness of a protocol.

- **Number of data packets transmitted per data packet delivered**: 'Data packets transmitted' is the count of every individual transmission of data by each node over the entire network. This count includes transmissions of packets that are eventually dropped and retransmitted by intermediate nodes. Note that in unicast protocols, this measure is always equal or greater than one. In multicast, since a single transmission can deliver data to multiple destinations, the measure may be less than one.

- **Number of control bytes transmitted per data bytes delivered**: Instead of using a measure of pure control overhead, we chose to use the ratio of control bytes transmitted to data bytes delivered to investigate how efficiently control packets are utilized in delivering data. Note that not only bytes of control packets (e.g., beacons, route updates, join requests, acknowledgments, etc.), but also bytes of data packet headers are included in the number of control bytes transmitted. Accordingly, only the data payload bytes contribute to the data bytes delivered.

- **Number of control and data packets transmitted per data packet delivered**: This measure shows the efficiency in terms of channel access and is very important in ad hoc networks since link layer protocols are typically contention-based.

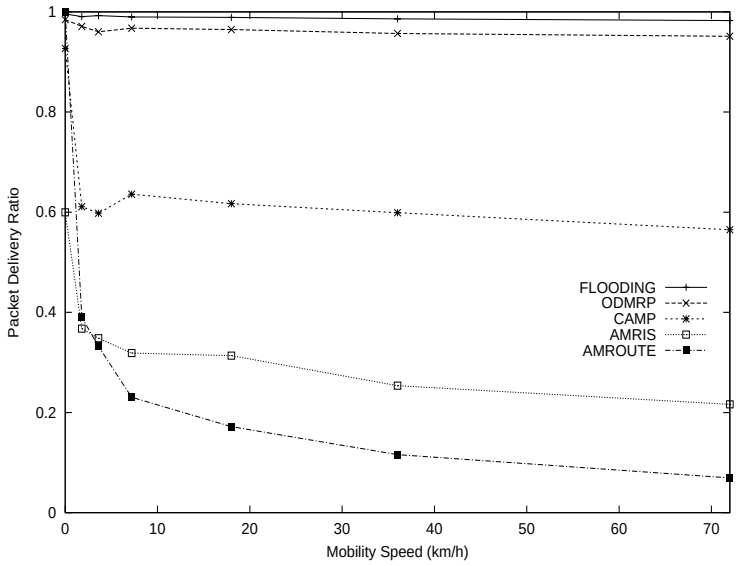


Figure 11.1: Packet delivery ratio as a function of mobility speed.

11.3 Simulation Results

We tried to emulate as many scenarios as possible to investigate the protocol performance under different network situations. We have varied the following four items: mobility speed, number of multicast senders, multicast group size, and network traffic load.

11.3.1 Mobility Speed

Each node moved constantly with the predefined speed. Moving directions of each node were selected randomly, and when nodes reached the simulation terrain boundary, they bounced back and continued to move. The node movement speed was varied from 0 km/hr to 72 km/hr. In the mobility experiment, twenty nodes are multicast members and five sources transmit packets at the rate of 2 pkt/sec each.

Figure 11.1 illustrates the packet delivery ratio of the protocols under different speeds. ODMRP shows good performance even in highly dynamic situations. ODMRP provides redundant routes with a mesh topology and the chances of packet delivery to destinations remain high even when the primary routes are unavailable. The path redundancy enables ODMRP to suffer only minimal data loss and be robust to mobility. In fact, ODMRP was as effective as flooding in this experiment.

CAMP, which also uses a mesh topology, shows a better performance than protocols which use trees. However, CAMP exhibited poorer performance than we had expected, especially under mobility. A major reason CAMP was not as effective as ODMRP was that many packets headed to distant routers in the mesh were not delivered. In CAMP, since the paths to distant destinations have fewer redundant paths than those closer to the center of the mesh, they are more prone to occasional link breaks preventing a vital *anchoring* node from successfully receiving packets. Most of the successful packet transmissions occur in this mesh center, and require fewer data transmissions per delivery than transmissions to the mesh edges. In addition, in the presence of mobility and link breaks, WRP (which is the unicast protocol CAMP prefers to coexist with) can require a period of network re-convergence in regards to a subset of destinations. During this interval, this subset of destinations will be marked as unreachable by the loop-detection facilities. If the group core is a part of this subset of temporarily unreachable nodes, the multicast routing updates regarding mesh maintenance will be postponed, which also contributes to delays in mesh response to mobility.

AMRIS shows a poor delivery ratio compared to protocols that use mesh configuration. Since AMRIS builds a shared tree for data dissemination, there is only one path between member nodes. If a single tree link breaks because of node

movements, packet collision, or congestion, destinations can not receive packets. AMRIS detects node movements and tree breaks by a beaconing mechanism. Nodes send beacons every second, and neighbors are considered to have moved away if 3 consecutive beacons are not received. Thus, in the best case, it takes 3 seconds after the link break for AMRIS to start tree readjustment. A number of packets can be lost during that period. There are possible solutions to this problem, but they all have respective drawbacks. If beacons are sent more often, that could increase packet collisions. If the number of allowed beacon losses is decremented, a node may attempt to find a new route when the link is not broken but beacons are lost due to collisions. Finding the optimal beacon interval and allowed number of beacon losses for AMRIS is beyond the scope of the chapter and we used the values recommended by the AMRIS designers. The result that surprised us was for zero mobility. While other protocols showed data delivery ratio approaching unity, AMRIS delivered only 60% of data packets. Since each node sends beacons every second, there are a number of packets contending for the channel. The beacon size of AMRIS is relatively large compared to other protocols that send beacons (see [167]). Thus, the beacon traffic combined with the data traffic causes a large number of collisions leading to 40% drop. Under very light data traffic, AMRIS shows improved performance as will be shown in Figure 11.8.

AMRoute was the least effective of the protocols with mobility. Although its delivery ratio is near perfect in no mobility, it fails to deliver a significant number of packets even at low mobility speeds. The delivery ratio steadily worsens as the mobility speed is increased. One of the reasons AMRoute performs so poorly is due to the formation of loops and the creation of sub-optimal trees when mobility is present (at 72 km/hr, the average hop count was nearly 8 while other protocols were below 4). Loops occur during the tree reconstruction phase when some

nodes are forwarding data according to the stale tree and others according to the newly built tree. The existence of loops is critical in protocol performance because they cause serious congestion. At some instants, nodes had up to 13.75 packets dropped per second. The loss of packets due to buffer overflow has two consequences. First, if a data packet is dropped in the early stage of its multicast tree traversal, a large portion of tree members will not receive it. Second, if control packets (TREE-CREATE, JOIN-ACK, etc.) are dropped, the tree is not properly built or becomes segmented and data will not be delivered. Another reason for AMRoute ineffectiveness is its dependency on the underlying unicast protocol. AMRoute relies on the unicast protocol to set up bidirectional tunnels between group members for the multicast tree. However, as shown in [130], when mobility speed increases, the bidirectional link assumption in ad hoc networks becomes weak (i.e., a node can reach a neighboring node, but not necessarily vice versa). In our experiments, unidirectional *critical* links existed in AMRoute trees. Critical links are such that packets sent by the one end of the link are mostly received by the other end but not vice versa. A great number of packets are lost at these critical links. Since there are no alternate routes in the AMRoute shared tree (although AMRoute creates the mesh in order to build a tree, data is forwarded only by tree nodes), data delivery ratio is very low.

Figure 11.2 shows the number of data transmissions per data delivery to destinations. AMRoute has the highest number of transmissions because of loops. We can observe that protocols using meshes (i.e., ODMRP and CAMP) transmit more data packets than AMRIS, which uses a tree. In fact, ODMRP transmits nearly as much data as flooding because it exploits multiple redundant routes for data delivery.

The control byte overhead per data byte delivered is shown in Figure 11.3.

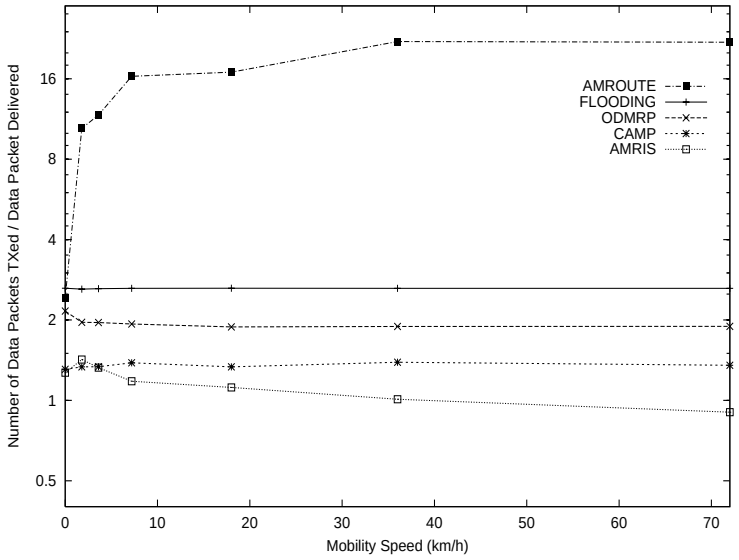


Figure 11.2: Number of data packets transmitted per data packet delivered as a function of mobility speed.

Remember that data packet header is included in control overhead. Flooding has no control packets. Hence, only the data header contributes to control overhead and this overhead does not increase with mobility. Other protocols generate increasing overhead as speed increases. AMRIS shows a low control overhead compared to other multicast schemes. The primary reason is that it transmitted less data packets (as seen in Figure 11.2). CAMP shows a larger control overhead under high mobility than ODMRP because of its reliance on the unicast routing protocol WRP, which sends triggered updates. WRP suffers from exponential growth in control traffic overhead under increasing mobility. Moreover, CAMP piggybacks its own update messages onto WRP updates and those packets play a role in overhead growth. In ODMRP, the control overhead remains relatively constant because no updates are triggered by mobility. JOIN DATA refresh interval was set constant to three seconds and hence no additional overhead is required as mobility increases. AMRoute has the highest ratio because of the data headers

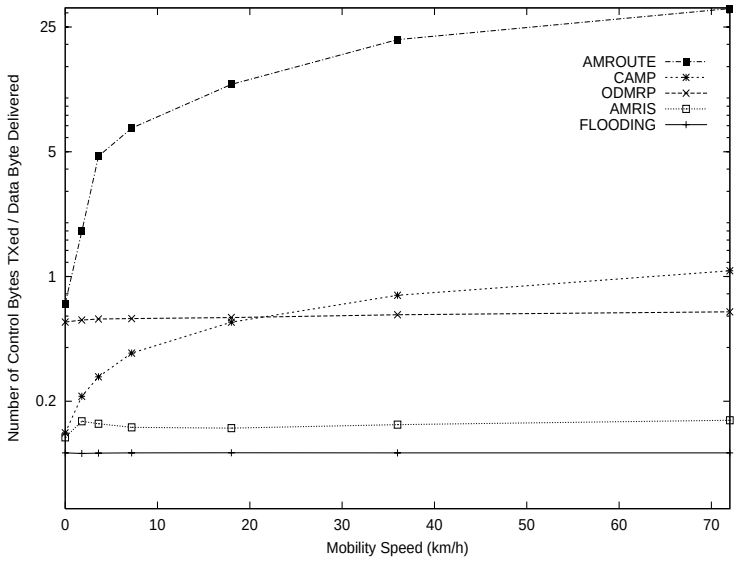


Figure 11.3: Number of control bytes transmitted per data byte delivered as a function of mobility speed.

that caught in the loops. The high ratio is also due to the formation of inefficient trees. During the tree creation phase, an inefficient tree can be formed when the TREE-CREATE packets from distant mesh neighbors arrives earlier than packets from nearby nodes (e.g., due to network congestion, etc.). The non-optimal tree results in having longer hops between member nodes and increasing the number of data transmissions.

The number of all packets transmitted per data packet delivered is presented in Figure 11.4. An interesting result is that CAMP has a smaller number of transmissions than ODMRP. This result stems from two factors. First, ODMRP transmits more data packets on redundant paths than CAMP. Second, although CAMP has more control overhead bytes, the number of control packet transmissions is lower since CAMP updates are piggybacked onto WRP updates. Again, AMRIS has the smallest number of packet transmissions because it uses a tree and AMRoute has the highest value because of loops.

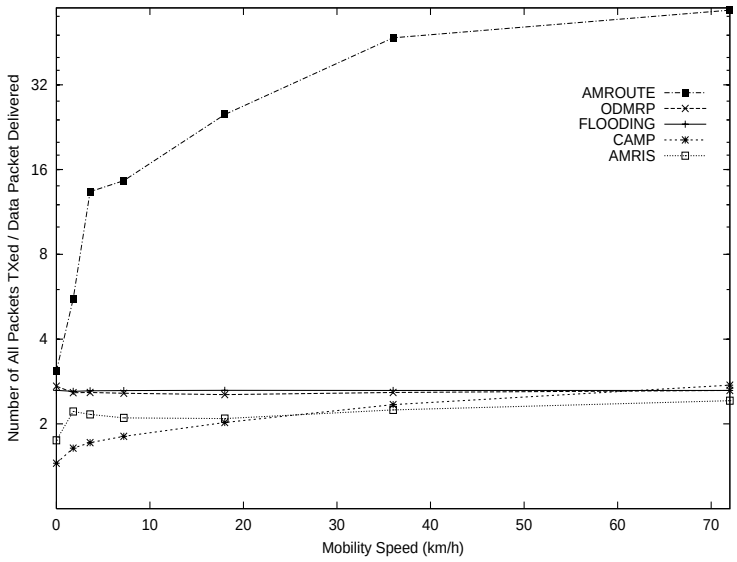


Figure 11.4: Number of total packets transmitted per data packet delivered as a function of mobility speed.

11.3.2 Number of Senders

In this experiment, the multicast group size is set constant at twenty, node mobility speed is slow (1 m/s), and network traffic load is relatively light (10 pkt/sec). The number of multicast senders range in the set {1, 2, 5, 10, 20}. A single sender represents a class lecture scenario, while at the other extreme, 20 senders model a video conference situation.

The packet delivery ratio as a function of the number of multicast senders is shown in Figure 11.5. As the number of sources increases, performance of flooding slightly degrades as more packets are lost by collision, congestion, and channel contention. ODMRP shows robustness to the number of sources. In fact, performance even improves with number of senders because of increasing number of forwarding nodes and thus better path redundancy. ODMRP limits the number of sources that can send JOIN DATA at the same time. Whenever a source needs

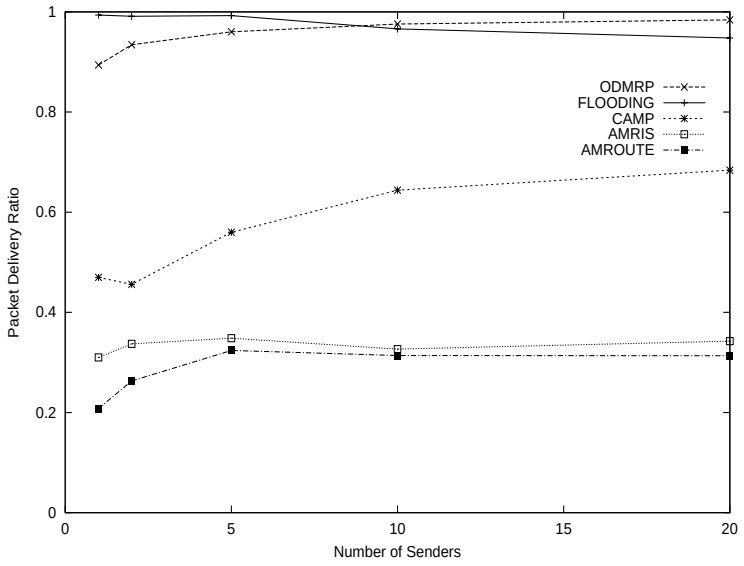


Figure 11.5: Packet delivery ratio as a function of number of senders.

to flood a JOIN DATA, it listens if any other source is flooding the packet. It proceeds to send the JOIN DATA only if no flooded packets are received within a certain period. Thus, the number of collisions decreases and the the protocol remains effective. Like ODMRP, CAMP shows improved performance with a larger number of senders due to the increase in the number of anchors that each node requires. Each member node requests every neighbor which is in the reverse shortest path to some source, to rebroadcast multicast update packets it receives initially. Hence increasing the number of sources increases the redundant paths in the mesh. AMRIS and AMRoute performance was unaffected by the number of senders because they use a shared tree for the multicast session.

Figure 11.6 shows the control overhead per data byte delivered. Every protocol except ODMRP shows a constant value. While the other three multicast protocols form a shared mesh or tree, ODMRP builds per-source meshes. If the number of senders increases, more JOIN DATA packets are propagated and control overhead grows accordingly. We can speculate from this result that ODMRP

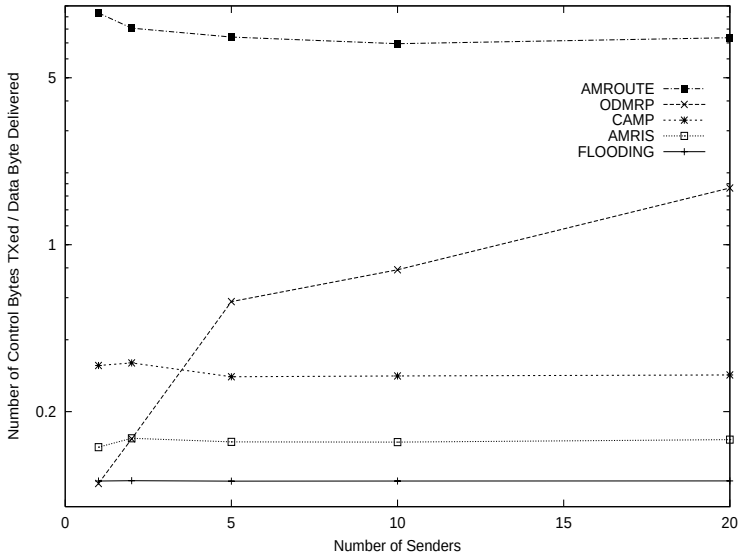


Figure 11.6: Number of control bytes transmitted per data byte delivered as a function of number of senders.

in its present form may not be as efficient in networks where a large number of nodes (e.g., hundreds and thousands) are multicast sources.

11.3.3 Multicast Group Size

We varied the number of multicast members to investigate the scalability of the protocol. While fixing the number of senders at five, mobility speed at 1 m/s, and network traffic rate at 10 pkt/sec, the multicast group size was varied from 5 to 40 members.

The routing effectiveness of protocols as a function of multicast group size is illustrated in Figure 11.7. Flooding and ODMRP performance were not affected by the number of multicast members. CAMP, on the other hand, performs markedly better as the number of receivers increases. Since the mesh becomes massive with the growth of the members, more redundant routes are formed and that improves

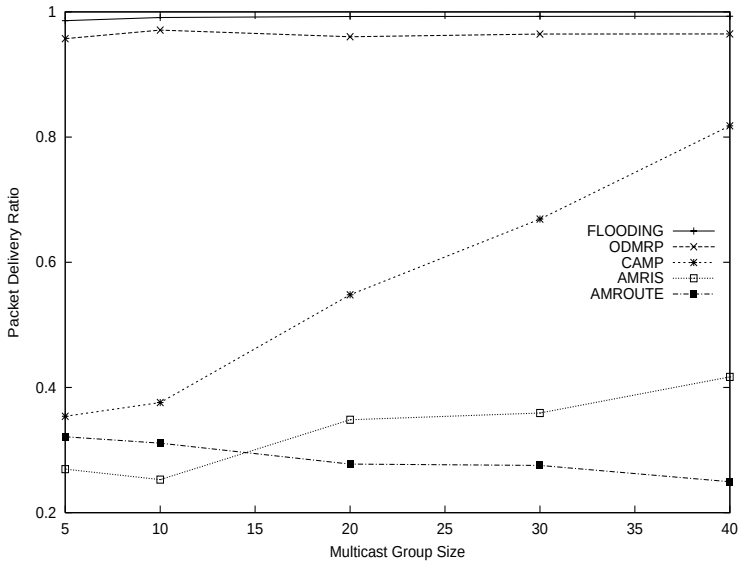


Figure 11.7: Packet delivery ratio as a function of multicast group size.

the performance. If only a small number of nodes join the multicast session, the mesh actually appears closer to a tree for distant nodes, and the performance is reflected in this graph. AMRIS also shows improvements with the member size growth, but they are less dramatic than CAMP because redundant routes are not established in AMRIS. AMRoute shows the complete opposite behavior. As the group size increases, the delivery ratio actually drops. This behavior is due to the *critical* links that exist in the AMRoute multicast tree (critical links were described in section 11.3.1). As the group size increases, the number of tree links increases and the probability of sources being isolated in the tree by critical links increases as well.

11.3.4 Network Traffic Load

To study the impact of data traffic load on multicast protocols, we varied the load on the network. There were five senders and the multicast group size was

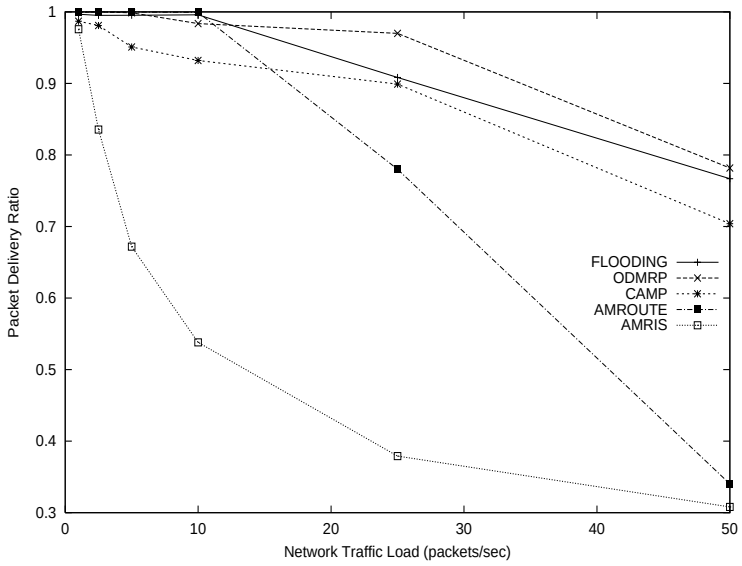


Figure 11.8: Packet delivery ratio as a function of network traffic load with no mobility.

twenty. In this experiment, there was no node mobility. Therefore, the packet drops are only caused by buffer overflow, collision, and congestion. The network traffic loads used were between 1 pkt/sec and 50 pkt/sec.

Packet delivery ratios for various traffic loads are shown in Figure 11.8. AMRIS was the most sensitive to traffic load. AMRIS delivers a high percentage of data packets in extremely light load (i.e., less than 5 pkt/sec). As the load increases however, the ratio drops rapidly. As explained in section 11.3.1, the transmission and the size of beacons resulted in numerous packet collisions. AMRoute performance is nearly perfect when the packet rate is relatively low, but it drops rather quickly when the traffic load is increased. The degradation is caused by buffer overflow at the members in the tree and at the mesh nodes that connect the tree members. CAMP performance is also affected by traffic load. As the load increases, the number of collisions and packet losses increase. When important control packets are dropped, anchor construction can be delayed and

data packets can fail to reach all the anchors. The degradation follows a pattern similar to flooding and ODMRP, indicating a common behavior in mesh based data delivery. Flooding shows worse delivery ratios than ODMRP as load grows. Since every data packet is flooded, the number of collisions and buffer overflows grows with the load. ODMRP is also affected by load, but the packet loss rate is less severe than flooding because the number of data packet transmissions is less than flooding. Although ODMRP shows the same patters of behaviors as CAMP, it gives a better delivery rate because it has less control overhead and suffers less buffer overflows than CAMP.

11.4 Discussion

In previous sections, we have studied the effectiveness and efficiency of several multicast protocols. In this section, we summarize the merits and shortcomings of protocols and derive suggestions for improvements. We also explain why our results differ from previous works by other researchers for some of the protocols. Finally, we share some of the lessons we have learned while conducting the study.

11.4.1 Protocol Analysis

AMRoute showed some promise in its simplicity and scalability in the number of senders. However, the presence of unidirectional "critical" links prevented reliable data delivery. The problem became worse as mobility was increased. Other drawbacks of AMRoute were the existence of loops and inefficient formation of trees. A possible improvement for AMRoute is to take reachability information (i.e., packets sent to a neighbor/packets received from that neighbor) into account when selecting tree links. Using this method, the impact of unidirectional

critical links can be reduced. In addition, introducing adaptivity into the protocol (e.g., periodic TREE-CREATE interval) can build more optimal trees. Most importantly, a loop prevention mechanism must be utilized for AMRoute to be efficient.

ODMRP performed well in most of our experiments. Providing redundant paths by the formation of mesh configuration made the protocol robust to mobility. The protocol did not yield excessive overhead in high mobility scenarios because no control packets are triggered by link breaks. However, when there are a large number of multicast senders, the protocol may suffer from excessive control overhead. Enhancements to make the protocol more scalable to large member groups must be developed.

AMRIS performance was very sensitive to mobility and traffic load. The main reasons for the poor performance were the number of transmissions and the size of beacons. As shown in section 11.3.4, beacons caused a number of packet collisions even when nodes are stationary. In more dense networks, the performance may become worse. We believe AMRIS can be improved by using a beaconing mechanism similar to CAMP. If the beacon is sent only when no packet has been transmitted in given interval, the number of beacon transmissions can be reduced while still delivering node information to neighboring nodes. In addition, the selection of Sid can affect the shape of the tree and possibly its performance. The research into the Sid selection algorithm along with beaconing methods will help improve AMRIS.

CAMP has good control traffic scalability for increasing multicast group size. Since JOIN REQUESTS only propagate until they reach a mesh member, CAMP does not incur exponential growth of multicast updates as the number of nodes and group members increase. However, it is dependent upon the unicast routing

protocol for behaviors regarding network convergence and control traffic growth in the presence of mobility. WRP's response to link breaks is not immediate, and can incorrectly deduce a link break in the presence of high network load. CAMP may perform better if it is modified so as to operate with an on-demand routing protocol. As shown in [23, 38, 68, 87], on-demand protocols performed favorably in terms of control packet overhead and response to mobility. If CAMP were able to leverage these advantages, it should dramatically improve its packet delivery ratio and control overhead.

11.4.2 Related Work

As of October 1999 when we performed this study, only CAMP and ODMRP designers have performed simulation study of their protocols. AMRoute and AMRIS performance evaluation have not been published. In simulation works reported in [50, 101, 102], the results are quite different from the results we have obtained in our experiments. In [50, 101, 102], a simplified simulator was used. A perfect channel was assumed and radio propagation was not considered. FAMA [47] was used as the medium access control protocol, which is different from IEEE 802.11 [60], the emerging standard MAC protocol for wireless LAN, that we used in our simulation. Only a small portion of network hosts had mobility (5 out of 30 or 15 out of 30) in their study. The critical nodes for CAMP performance (e.g., core, senders), however, remained stationary. All the nodes in [50, 102, 101] were multicast session members, which is not realistic in typical multicast applications. The network traffic load was extremely light (4 packets/sec). Information on data size, radio propagation range, or simulation terrain range were not given. Thus, the results in [50, 101, 102] are somewhat limited. In any way, they cannot be directly compared to the results from this chapter.

11.4.3 Lessons Learned

While implementing and evaluating multicast protocols, we have learned a great deal and would like to share our experience with researchers who design and implement ad hoc wireless multicast protocols. In our study, the mesh protocols performed significantly better than the tree protocols in mobile scenarios. In trees, when routes are invalidated due to node movements, the packets must be buffered or dropped until the tree is reconfigured. On the other hand, redundant routes in the mesh provide alternate routes for data delivery in the face of mobility and link breaks. Data packets can still reach the destinations while the primary route is being reconstructed.

Using detailed lower layer (i.e., link layer and physical layer) implementations in the network simulator along with programmable mobility patterns highlighted differences in the protocol tolerance to various wireless network conditions. We strongly recommend fellow researchers to use publicly available simulators which are validated by frequent use and which permit replication of the experiments.

11.5 Conclusion

We have conducted a performance evaluation of five multicast protocols that have been proposed for ad hoc networks. The channel, radio, IEEE 802.11 MAC protocol, and multicast protocols (AMRoute, ODMRP, AMRIS, CAMP, and flooding) have been carefully implemented. The detailed simulator has enabled us to perform fair and accurate comparisons of the multicast protocols under a realistic wireless environment, for a broad range of parameters including mobility, number of senders, multicast group size, and traffic load.

A general conclusion is that, in a mobile scenario, mesh-based protocols out-

performed tree-based protocols. The availability of alternate routes provided robustness to mobility. AMRoute performed well under no mobility, but it suffered from loops and inefficient trees even for low mobility. AMRIS was effective in a light traffic environment with no mobility, but its performance was susceptible to traffic load and mobility. CAMP showed better performance when compared to tree protocols, but with mobility, excessive control overhead caused congestion and collisions that resulted in performance degradation. ODMRP was very effective and efficient in most of our simulation scenarios. However, the protocol showed a trend of rapidly increasing overhead as the number of senders increased.

We experimented with scenarios which we thought were the most representation of ad hoc wireless network applications. However, we did not cover every possible situation. While the results of this chapter can provide guidelines, the final selection of a multicast protocol should take into account other considerations which cannot be evaluated via simulation alone.

CHAPTER 12

Exploiting the Unicast Functionality of the ODMRP

The On-Demand Multicast Routing Protocol (ODMRP) is an effective and efficient protocol designed for mobile wireless ad hoc networks with multicast purposes. One of the major strengths of ODMRP is its capability to operate both as a unicast and a multicast routing protocol. This versatility of ODMRP can increase network efficiency as the network can handle both unicast and multicast traffic with one protocol. ODMRP's another strength is its option to use mobility prediction in networks equipped with Global Positioning System (GPS) [72]. The primary goal of mobility prediction is to perform route reconstruction prior to topology changes. The use of mobility prediction helps minimize packet losses and efficiently utilize control packets. We believe mobility prediction will benefit more in unicast situations than in multicast environment. In this chapter, we describe ODMRP unicast routing functionality and assess the mobility prediction effectiveness and effficiency. We evaluate the ODMRP performance via detailed simulation and compare it with other ad hoc routing schemes.

12.1 Unicast Operation of ODMRP

12.1.1 Basic Mechanism

ODMRP builds and maintains routes on demand by the source. A query phase and a reply phase comprise the protocol. When a source has to communicate with a node but no route information to that destination is known, it floods a control packet called JOIN QUERY with a piggybacked data payload. When a node receives a non-duplicate JOIN QUERY, it stores the last hop node information in its route table (i.e., backward learning) and rebroadcasts the packet. When the JOIN QUERY packet reaches the destination, the destination replies back to the source via the selected route with a JOIN REPLY packet.[1] Intermediate nodes of the route forward the JOIN REPLY to the next hop towards the source of the route. The next hop node information is obtained from the route table where the entry was recorded when JOIN QUERY was received. The JOIN REPLY packet is propagated until it reaches the source of the route. This process constructs the route from the source to the destination. Figure 12.1 depicts the route $<S$-i-j-k-$D>$ establishment procedure.

One drawback of on-demand routing protocols is the route acquisition latency. Since routes are only built when needed, the source must wait until a route is established before transmitting data. To eliminate this delay, JOIN QUERY packets carry user data traffic in our protocol. Since the destination will receive the packet unless the network is partitioned, no route acquisition delay is needed. The size of flooded packet however, becomes larger. There is a tradeoff between

[1]Packet types JOIN QUERY and JOIN REPLY have the term "Join" because ODMRP is originally a multicast protocol. These packets are exchanged to collect multicast group membership information as well as to build routes in multicast sessions, hence the term "Join." We keep the packet names the same in unicast mode even though group membership information is not obtained.

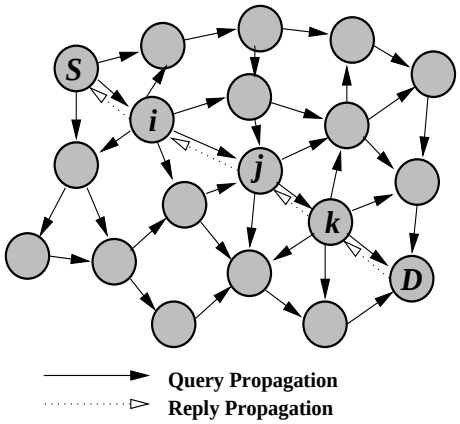


Figure 12.1: On-demand procedure for route setup.

delay and efficiency. When data payload size is very large, we should avoid data piggybacking on JOIN QUERY.

To use the most recent route information, our protocol enforces two policies that are different from other well-known on-demand routing protocols such as Ad-hoc On-Demand Distance Vector (AODV) [128] and Dynamic Source Routing (DSR) [69]. First, intermediate nodes cannot send a JOIN REPLY in response to a JOIN QUERY even when they have route information to the destination node.[2] One reason is to deliver the JOIN QUERY data payload to the destination. If intermediate nodes send replies to the source and drop the JOIN QUERY packet, the destination cannot receive the data portion of the packet. The second reason is to utilize the most up-to-date topology information to build the shortest-distance route. Routes obtained from intermediate nodes yield longer hop distances since they do not account for node locations and network topology during and after node movements.

Second, as long as the source still need to communicate with the destination, JOIN QUERIES are periodically broadcasted to the entire network to refresh the

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

route. Therefore, fresh routes are continuously built and utilized. We should adaptively select periodic route refresh interval based on network environment (for example, traffic type, traffic load, mobility pattern, mobility speed, channel capacity). When we use small route refresh intervals, we can frequently obtain fresh route information at the expense of producing more packets and causing network congestion. On the other hand, when we select large route refresh intervals, even though less control traffic will be generated, routes may not use fresh topology information. Thus in highly mobile networks, using large route refresh intervals will yield poor protocol performance.

Although the periodic route refresh reconstructs the routes, a node of the route sends a ROUTE ERROR message back to the source to invoke a fast route recovery process when it detects a route break during data propagation. Nodes detect a link disconnection either by MAC layer feedbacks using reliable MAC protocols such as IEEE 802.11 [60] and MACAW [19], or by passive acknowledgments [71]. The source, upon receiving the ROUTE ERROR packet, sends a JOIN QUERY for route recovery. In addition, it adjusts the next route refresh time to the current time plus the route refresh interval. Note that the ROUTE ERROR message does not exist in the ODMRP multicast operation since redundancy is created by multiple routes. In the unicast operation however, each <source, destination> pair maintains single path and no alternate route is available. Immediate route reconstruction is therefore necessary.

12.1.2 Adapting the Refresh Interval via Mobility Prediction

Since the mobility prediction mechanism and the route selection algorithm using it were introduced in Section 10.1 and Section 10.2, respectively, we omit the description in this chapter.

12.2 Simulation Model and Methodology

12.2.1 Simulation Environment

We implemented the simulator in PARSEC [10] within the GloMoSim library [160]. Our simulation modeled a network of 50 mobile hosts placed randomly within a 1000 meter $\times$ 1000 meter area. Radio propagation range for each node was 250 meters and channel capacity was 2 Mb/s. Each simulation executed for 600 seconds of simulation time. Multiple runs with different seed numbers were conducted for each scenario and collected data were averaged over those runs.

Our experiments used a free space propagation model [135] with a threshold cutoff. In the radio model, we assumed the ability of a radio to lock on to a sufficiently strong signal in the presence of interfering signals, i.e., radio capture. The IEEE 802.11 Distributed Coordination Function (DCF) [60] was used as the medium access control protocol. We developed a traffic generator to simulate constant bit rate sources. The sources and the destinations are randomly selected with uniform probabilities. Data payload size was 512 bytes. Each node moved continuously with the predefined speed between zero and 72 km/hr. Nodes randomly selected the moving direction, and when they reached the simulation terrain boundary, they bounced back and continued to move.

12.2.2 Methodology

To evaluate the unicast performance of ODMRP, we simulated and compared the following schemes:

- ODMRP (On-Demand Multicast Routing Protocol)
- ODMRP-MP (On-Demand Multicast Routing Protocol with Mobility Pre-

diction)

- LAR (Location Aided Routing) [78] : an on demand routing protocol that uses GPS location information
- WRP (Wireless Routing Protocol) [114] : a distance vector routing protocol for ad hoc networks

We evaluated all schemes as a function of speed and the number of unicast data sessions. In the experiments that we varied the mobility speed, the number of data sessions was set to five and speed was varied from zero to 72 km/hr. In the experiments that we varied the number of data sessions, mobility speed was set to 36 km/hr and the number of sessions was varied from 5 to 30. In another set of experiments, to assess the impact of the mobility prediction, we directly compare the performance of ODMRP-MP with ODMRP by varying the route refresh interval of ODMRP. Periodic ODMRP route refresh interval was varied from 0.5 second to 4.0 seconds. Remember that ODMRP-MP adapts the refresh interval based on mobility prediction. Mobility speed was set to 36 km/hr and the number of data sessions was set to five. In the first two set of experiments where ODMRP and ODMRP-MP performances were compared with LAR and WRP, the refresh interval of ODMRP was 1.5 seconds. In all the experiments, the total number of data packets were sent at the rate of 20 packets/sec.

The metrics of interest are packet delivery ratio, number of control bytes transmitted per data byte delivered, and number of total packets transmitted per data packet delivered

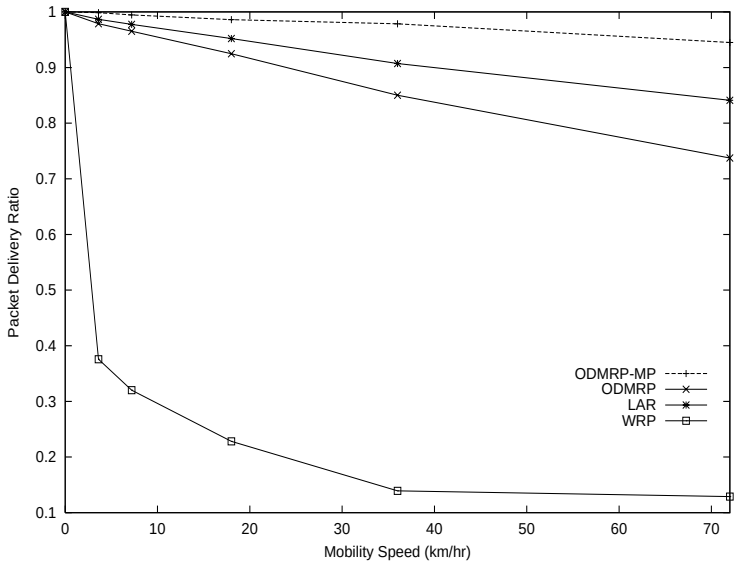


Figure 12.2: Packet delivery ratio as a function of speed.

12.3 Simulation Results

12.3.1 Packet Delivery Ratio

The packet delivery ratio as a function of mobility speed and the number of data sessions is shown in Figure 12.2 and Figure 12.3, respectively. We can observe from Figure 12.2 that as speed increases, the routing effectiveness of WRP degrades rapidly compared with other schemes. As nodes move faster, link connectivity changes more often and more update messages are triggered. For each triggered update, neighbor nodes are required to send back an acknowledgment. Moreover, temporary loops were being formed because the network view converged slowly, with many changes needing to be absorbed and propagated. Loops, triggered updates, and ACKs created an enormous amount of packets, contributing further to collisions, congestion, contention, and packet drops. ODMRP-MP is the least affected by the mobility speed. It is able to maintain delivery ratio

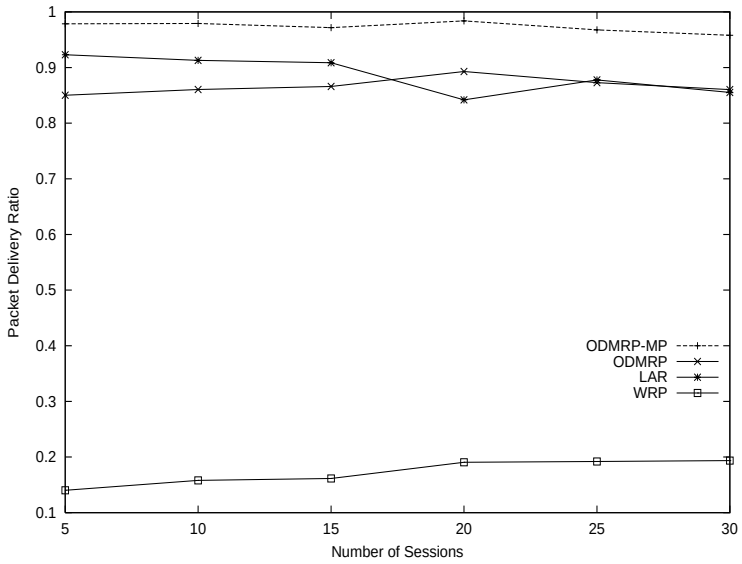


Figure 12.3: Packet delivery ratio as a function of number of sessions.

above 0.9 for all mobility speeds in our experiments. Performing rerouting prior to route disconnection minimized packet losses.

In Figure 12.3 ODMRP-MP outperforms the rest of the schemes again. ODMRP-MP shows no performance degradation when the number of sessions is increased. LAR also shows a high delivery ratio. The delivery ratio for WRP is significantly lower than other schemes because the mobility speed was relatively high (36 km/hr).

12.3.2 Number of Control Bytes Transmitted per Data Byte Delivered

Figure 12.4 shows the number of control bytes transmitted per data byte delivered as a function of mobility speed for each protocol. Remember that the control packet transmission in ODMRP is periodically triggered without adapting to mobility speed. The ratio for ODMRP hence does not increase as the mobility

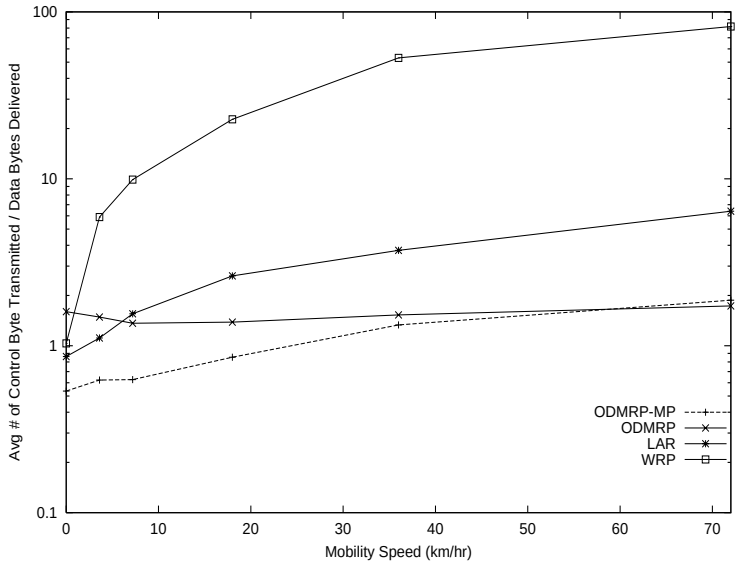


Figure 12.4: Number of control bytes transmitted per data byte delivered as a function of speed.

speed increases. On the other hand, the overhead of ODMRP-MP becomes larger as mobility speed increases. Since the scheme apples mobility prediction to adapt to mobility speed, more JOIN QUERY and JOIN REPLY packets are sent when mobility is high, thus resulting in more overhead. In WRP, route updates are produced more frequently in high mobility since there are more link changes. WRP has the highest ratio in mobile situations because of the small number of delivered data packets and the large number of triggered updates. LAR also shows more overhead as mobility speed increases because more route breaks occur and they invoke route recovery procedures.

In Figure 12.5, we can see that ODMRP and ODMRP-MP have better ratios than WRP and LAR. The ratios for ODMRP and ODMRP-MP grow slowly as the number of sessions is increased. ODMRP-MP's ratio is slightly lower than that of ODMRP because mobility prediction enables the efficient use of control packets.

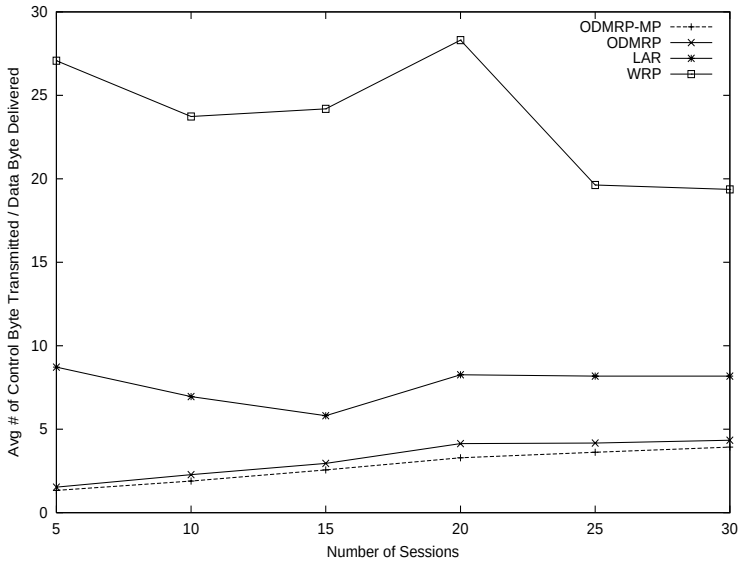


Figure 12.5: Number of control bytes transmitted per data byte delivered as a function of number of sessions.

12.3.3 Number of Total Packets Transmitted per Data Packet Delivered

The number of total packets (control packets and data packets) transmitted per data packet delivered is presented in Figure 12.6 and Figure 12.7. This measure shows the channel access efficiency and is very important in ad hoc networks since link layer protocols are typically contention-based. We can see that the numbers for ODMRP and ODMRP-MP remain relatively constant, with ODMRP-MP's ratio being lower than that of ODMRP. WRP has the highest ratio because of the same reasons described in Section 12.3.2. In Figure 12.7, ODMRP-MP again has the best performance compared to other schemes.

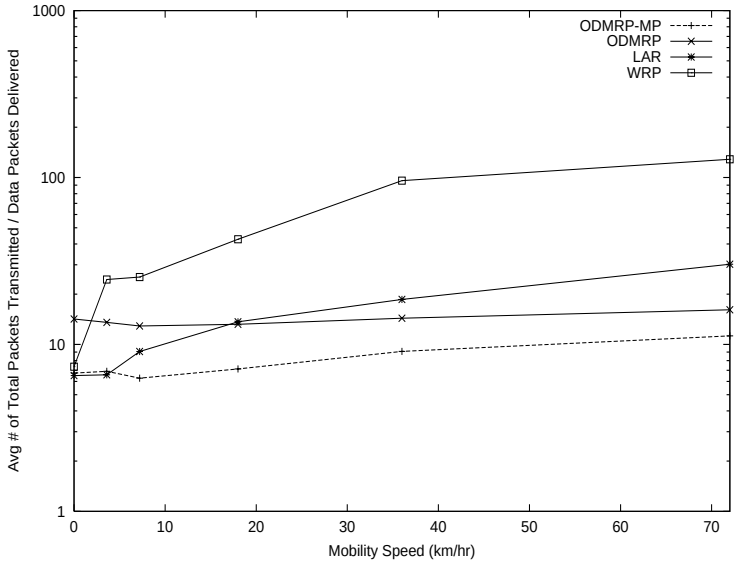


Figure 12.6: Number of total packets transmitted per data packet delivered as a function of speed.

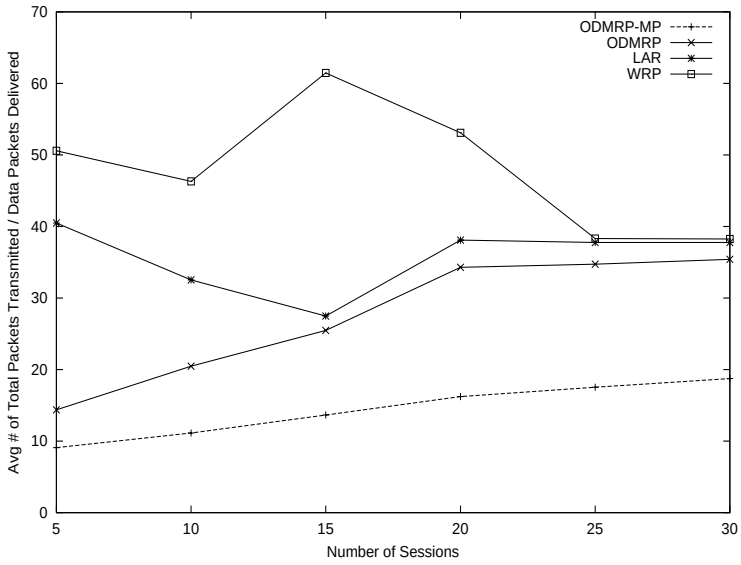


Figure 12.7: Number of total packets transmitted per data packet delivered as a function of number of sessions.

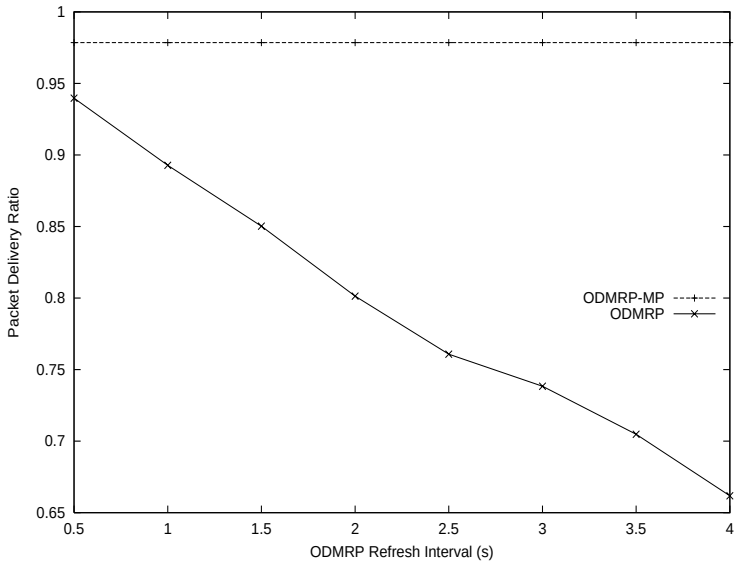


Figure 12.8: ODMRP packet delivery ratio with and without mobility prediction.

12.3.4 Mobility Prediction Effectiveness

Since the basic ODMRP scheme rediscovers routes periodically, the performance of the protocol is highly dependent on the route refresh interval. When we shorten the refresh interval, packet delivery ratio may improve. Nevertheless, since JOIN QUERY is flooded more often, routing message overhead increases. With mobility prediction, JOIN QUERY is flooded only when necessary. High packet delivery ratios can be maintained without yielding a large amount of overhead. To assess the improvement of mobility prediction, we vary the route refresh interval of ODMRP and compare the performance with ODMRP-MP. Figure 12.8 shows the packet delivery ratio as a function of route refresh interval. We can see that the ODMRP performance degrades rather rapidly when the refresh interval is increased. As we increase the route refresh interval, the routes are not updated quickly and more packets are dropped. ODMRP-MP performs better than ODMRP regardless of the ODMRP route refresh interval. Figure 12.9 shows the number of control bytes

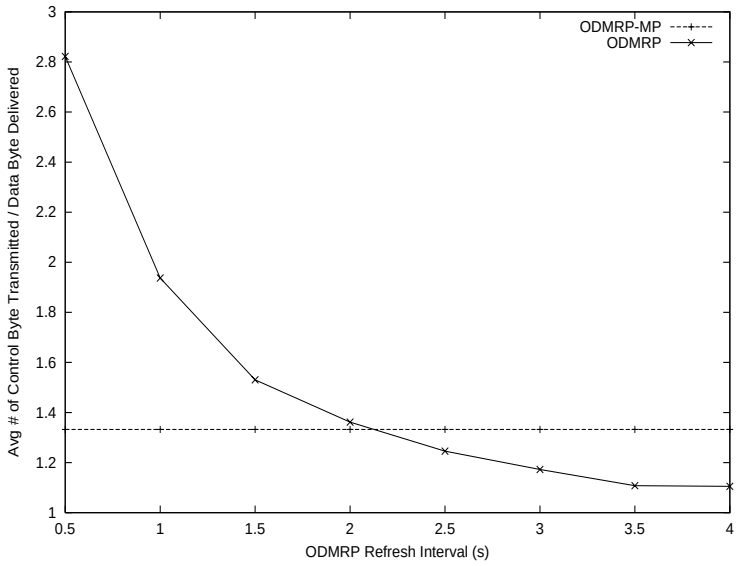


Figure 12.9: Number of control bytes transmitted per data byte delivered with and without mobility prediction.

transmitted per data byte delivered. We can see that at small refresh intervals, the overhead of ODMRP is significantly greater than that of ODMRP-MP, but it decreases as the refresh interval is increased. In fact, ODMRP generates less overhead than ODMRP-MP when refresh interval is greater than 2.1 seconds. As seen in Figure 12.8 however, packet delivery ratio of ODMRP drops as the interval is increased. We can analyze that the basic ODMRP does not efficiently utilize control packets when the route refresh interval is large.

12.4 Conclusion

ODMRP is an ad hoc routing protocol that is capable of routing both unicast and multicast data. We described ODMRP unicast operation in detail and evaluated its performance by comparing it with other ad hoc unicast routing protocols. We also examined the impact of the mobility prediction on ODMRP performance

to evaluate its effectiveness. Simulation results indicate that ODMRP is a competitive unicast protocol. The use of mobility prediction proved to be valuable and enhanced ODMRP performance. With mobility prediction, more data packets were delivered to destinations and the control packets were utilized more efficiently.

CHAPTER 13

ODMRP Implementation in Ad hoc Network Testbeds

The performance of ODMRP has been tested using a detailed simulator [95, 90]. The results obtained from simulation led us to take the next step in development: implement ODMRP in a real testbed to validate and fine tune the protocol. No prior work exists in building a *multicast* protocol in an ad hoc network testbed. Our implementation utilizes the multicast extension built into the Linux kernel. A wireless mobile testbed consisting of laptops with the Linux operating system has been organized to test the implementation. We also experiment the unicast capability of ODMRP in our testbed of seven laptop computers in an indoor environment. Both static and dynamic networks are deployed. We generate various topological scenarios in our wireless testbed by applying mobility to network hosts and study their impacts on our protocol performance. We believe that the performance study in a testbed network can help us analyze the protocol in a realistic way and point us to the future research direction.

13.1 Ad hoc Wireless Testbeds

There are several previous works that built ad hoc wireless testbeds. Monarch project team of CMU has developed a multihop wireless ad hoc network testbed

on existing BSD Unix network stack [106]. They implemented a unicast routing protocol DSR (Dynamic Source Routing) [69] and tested in an outdoor environment. UCSC has developed wireless IP routers, Wireless Internet Gateways (WINGS) [49] and Secure Protocols for Adaptive, Robust, Reliable, and Opportunistic WINGs (SPARROW) [48] for ad hoc networks. Using the C++ Protocol Toolkit (CPT), protocol softwares were transitioned from a simulation environment to an embedded system. University of Maryland has also developed an ad hoc network testbed on Linux 2.1 kernel [64]. Other works that built ad hoc network testbeds include SURAN project [17] and Task Force XXI [141].

13.2 Implementation

13.2.1 Implementation Platform

13.2.1.1 Operating System and Software

Our protocol is developed on Linux kernel version 2.2.12, the version provided by the Red Hat Linux version 6.1.[1] All tools and software packages used in our development originate from software bundle incorporated within the Red Hat Linux version 6.1 operating system package with the singular exception of Lucent WaveLan IEEE 802.11 device driver [161]. We chose the Linux operating system for its availability, familiarity, and most importantly, kernel level support for multicasting. The kernel support for multicast allows fast kernel level multicast packet switching and minimizes expensive delays caused by kernel-to-application and application-to-kernel level crossing.

[1]The implementation of ODMRP in a real ad hoc network testbed was mostly performed by my colleague Sang Ho Bae. Since the implementation of the protocol is relevant to this dissertation, it is included with the permission of Mr. Bae.

13.2.1.2 Hardware

Ad hoc network nodes consist of Intel Pentium II based Hewlett Packard Omnibook 7150 laptops and Texas Instruments Extensa 510 laptops equipped with Lucent IEEE 802.11 WaveLan radio devices. The WaveLan devices operate on 2.4 GHz bandwidth and communicate at the maximum capacity of 2 Mb/s with the semi-open space range of 150 meters. The WaveLan devices are operated in an ad hoc mode.

13.2.2 Software Architecture

ODMRP uses the kernel level multicast support option built into the Linux operating system. With the exception of a minor alteration made to allow single device forwarding, we did not make any changes at the kernel level. The Linux kernel supports multicast by performing the following procedures. The user enables the multicast option in the network interface driver. The multicast enabled interface accepts and sends all packets with multicast address to the kernel. The kernel accepts all multicast packets, stores them in the message cache, and starts the cache timer. The cached messages are discarded when the timer expires. The kernel then periodically looks for the cached group addresses in the kernel multicast routing table and decides whether to forward them or not. The messages are forwarded by altering their forwarding destination interfaces and buffering them to the corresponding interfaces. The messages are sent up to user level only if there exists a local multicast application that has joined the group. This process avoids the costly kernel-to-user crossing for store-and-forward packets and improves efficiency. The destination interface is changed in accordance with the listings in the kernel level multicast routing table. The kernel level routing table is updated and maintained by a user level routing daemon which keeps a

local image of the kernel level routing table. The user level table is copied to the kernel level table as soon as updates are made. We used this basic routing table interface to build and maintain "mirrored" ODMRP routing tables both at the kernel and user level.

13.2.2.1 Packet and Table Management

There are three types of control packets in ODMRP (JOIN QUERY, JOIN REPLY, and ROUTE ERROR). These packets are implemented as new types of Internet Group Management Protocol (IGMP) [45] packet with a data section. Existing IGMP packet structure and handler function are expanded to include functionalities for JOIN QUERY and JOIN REPLY. When a JOIN QUERY packet arrives at the router, the content of the packet is cached into temporary route table (`tr_table`) and the timer for the entry is started. If the router does not receive a corresponding JOIN REPLY in time, the timer expires and the cached entry is removed. If a JOIN REPLY which has a corresponding entry in the `tr_table` arrives before the timeout, the user level route table (`route_table`) is searched to find the <source, destination> pair that matches the `tr_table` entry. If such a pair is found, the soft state timer for the entry is reset and the router waits for the next event. If the pair can not be found in the `route_table`, a new entry is created and inserted into the table. The `route_table` is periodically checked for timer expiration and expired entries are removed. The trigger for the update of the kernel level route table (`kr_table`) is activated whenever an entry is inserted or deleted.

13.2.2.2 Forwarding on Virtual Interfaces

The DVMRP, PIM, and CBT based multicast routers are all built to be used over wired networks. Therefore, their frameworks are designed for routers with multiple network interfaces. The forwarding capability of these systems is limited strictly to passing the packets from one interface to another. In wired networks, having multiple interfaces does not cause any problems since the devices do not interfere with one another. This is not the case with our wireless network testbed because we use omni-directional antennas and a common broadcast channel. Having multiple wireless interfaces does not improve the performance. Unless specifically configured (for example, at different frequencies), the devices interfere with one another. The framework in Linux however, allows the forwarding between virtual interfaces (VIF) to support the tunneling among the multicast islands. A virtual interface can be created on a physical device in two ways. An IP (Internet Protocol) alias can be created on a physical device. In Linux version 2.0.36, we can create an IP alias by adding an interface entry with a new IP address and a network device alias onto the kernel interface configuration table. We can then use this interface in exactly the same manner as the original physical device with all its physical attributes. We can create a virtual interface by opening a tunnel between two multicast routers. Unlike VIF that is created with the aliasing method, only the multicast router can use a tunnel since the multicast routing daemon establishes the tunnel by opening a unicast socket to encapsulate the multicast streams. In our experiments, we make single device forwarding possible by aliasing the existing hardware interface to create VIFs and then enabling the forwarding of the multicast packets to VIFs corresponding to the `routing_table` entries.

13.2.3 ODMRP Agent for Nodes with Fixed Routes

Operating systems such as Microsoft Windows 95/98 and NT do not allow dynamic reconfiguration of the network routes. ODMRP Agent(ODA) was created to allow forwarding to the hosts with a static route. ODA operates on a Linux host serving as a gate way to ad hoc network for static-route node. Currently, ODA serves only the designated host and the host which employs ODA service must remain within its radio range. ODA performs routing tasks such as sending and receiving the JOIN QUERY and the JOIN REPLY, and updating the route table in behalf of the static-route node. Unicast traffic of static-route nodes can be forwarded through ODMRP ad hoc network with ODA. We experimented with existing Windows and Linux applications over the multihop testbed using ODA.

13.2.4 ODMRP Timers

For the ODMRP soft state timer values, we selected one second for route refresh interval and five seconds for forwarding group timeout interval.

13.3 Performance Evaluation

We present the radio channel performance of a basic wireless link with WaveLAN devices along with our channel experimental results. For multicast experiments, we created a testbed consisting of six nodes. We study the bandwidth utilizations of ODMRP and DVMRP. We intended to compare ODMRP with other ad hoc wireless multicast implementations, but no multicast testbed implementations are released to the public for testing. We were able to obtain the DVMRP implementation [113], and compare its performance with ODMRP in our study. For unicast experiments, we created a testbed consisting of seven nodes. The

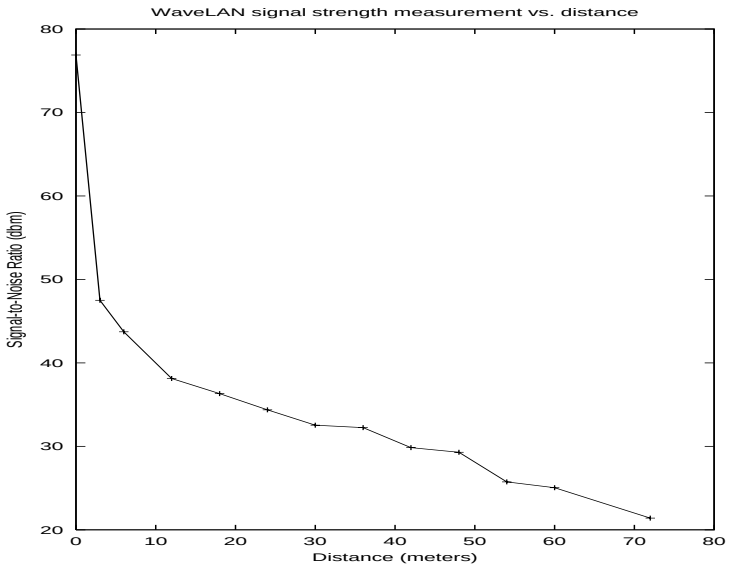


Figure 13.1: Signal-to-noise ratio vs. distance with WaveLAN radio device.

bandwidth utilizations of static-routing and ODMRP are studied

13.3.1 Radio Channel Evaluation

We collected the channel data by initiating a large scale UDP (User Datagram Protocol) packet transfer from one station to another. We chose the UDP as the transport layer protocol instead of TCP (Transmission Control Protocol) in order to study the behavior of packet loss without the intervention of TCP. We repeated the experiments for various distances at 6 meter increments until we reached the maximum line of sight distance within the building. At each trial, the source sends 13600 UDP packets of size 532 bytes. We programmed the receivers to collect the number of successfully received packets and record the signal strength for each packet. We report the signal strength as signal to noise ratio (in dbm). Measuring the pure signal strength alone would be meaningless unless the ambient noise level is known. We plot the average signal strength as

a function of distance in Figure 13.1. As expected, the signal strength degrades with increasing distance. Nevertheless, the packet loss rate remains nearly the same (below 0.5%) throughout the experiment. Packet loss depends more on the position of the node (next to a steal beam, near the elevator) than on signal strength as long as the signal to noise ratio remains above zero. This result indicates that in our environment, wireless communications suffer more losses from random noise than from signal degradation. We explain some of the effects of random channel loss in the next section.

13.3.2 Multicast Experiments

13.3.2.1 DVMRP Overview

DVMRP is based on the distance-vector routing algorithm. In this protocol, each router maintains a routing table with all reachable destinations. A typical routing table entry consists of destination address, metric to the destination (such as, distance, hop count), and the next hop to reach the destination. The router obtains the up-to-date routing information by periodically exchanging the route table with immediate neighbors. After each exchange, routers compute shortest paths and update new information to the route table. To accommodate the multicast, a route entry includes the multicast group address, children routers' membership information, and the local subnet membership information fields. The routing information distributed among the routers collectively creates a multicast tree for each multicast group. When a new router joins the network, the multicast streams and the route table of neighboring nodes are forwarded to the new router. To leave an unwanted multicast session, the router must send a prune message. A node is required to join the group if there exists a member among its descendent nodes. DVMRP relies on IGMP to request the routing

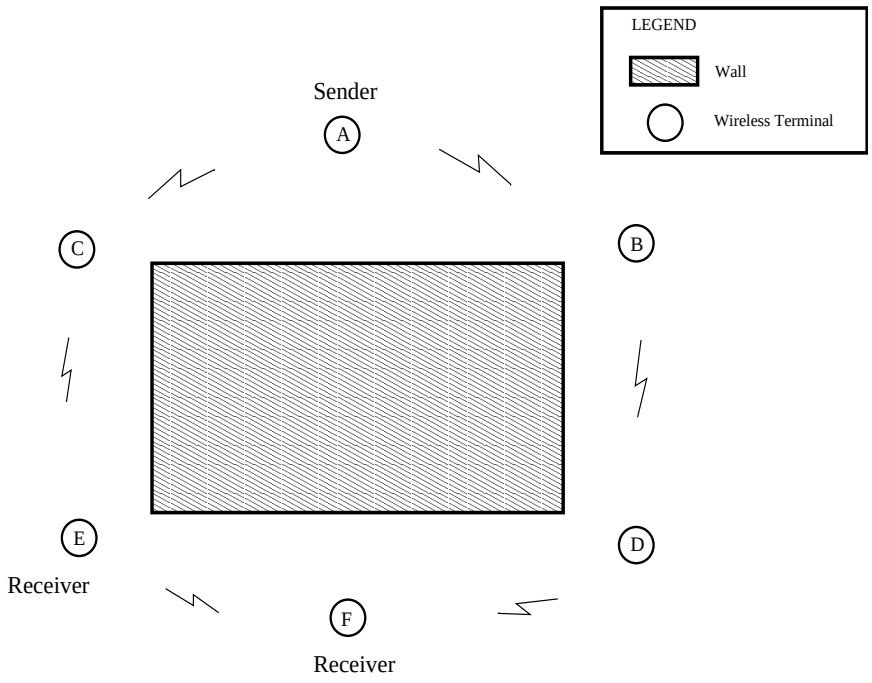


Figure 13.2: Our testbed topology.

information and to exchange the control messages and the route tables. IGMP control messages are also used for probing the neighboring routers for the active status of the next hop multicast daemon.

13.3.2.2 DVMRP Trace Analysis with MBone Sources

Figure 13.2 shows the basic testbed topology. The DVMRP based multicast routing daemon `mrouted` is installed in each node. The base station, marked as a sender, links wired and wireless networks. The Internet multicast backbone (MBone) is extended to the wireless network through the base station. Initially, the multicast daemons are activated one by one in each node. The time it takes for the routing daemon to stabilize as the routing updates from the parent node in the multicast tree are forwarded is variable. The duration of time depends

Table 13.1: DVMRP with MBone feed.

	Value	% of total
Avg. session length	178 sec	N/A
IGMP control packet O/H	15.58 kb/s	6 %
Avg. number of active multicast channels	2	N/A
O/H caused by multicast channels	26.41 kb/s	10.29 %
Effective data throughput	214.71 kb/s	83.71 %
Total throughput (data and control)	256.7 kb/s	100 %

on the number of (sender, destination) pairs in the network. We do not include the overhead required by the DVMRP initialization in the analysis because we cannot consider the initialization period as a part of the normal operation mode. The testbed is ready for measurement experiments only after the initial control packet rush subsides and a regular control packet traffic pattern emerges. We carry out the experiment in the following steps. Each router starts the traffic traces using `tcpdump`. The receivers at nodes E and F then join audio and video multicast sessions. We monitored the multicast for approximately three minutes. Table 13.1 reports the results.

DVMRP operates by first allowing all multicast streams to be forwarded downstream and then selectively "pruning" the unwanted streams bottom-up with IGMP messages. This practice works well in wired networks since very few control packets are lost. In wireless networks however, packet loss is frequent. When a prune message is lost at a router for instance, the corresponding multicast group is allowed to continue forwarding from the router down until the next prune cycle. This unnecessary forwarding causes the channel overhead listed in Table 13.1. If a more complex topology had been deployed, there would be even

more control packet losses, with higher bandwidth wastage.

13.3.2.3 DVMRP Trace Analysis with a Local Source

For this experiment, the base station that bridged the wired and wireless networks in the MBone experiment, is replaced with a wireless multicast source. We kept the rest of the topology the same. The multicast stream consists of a file multicast from sender node A to the receivers E and F. Table 13.2 shows the measurement results. The sessions, in this experiment, last only until the file is transfered, so the session length field in the table is left blank. Comparing with the results in Table 13.1, we note a much less control packet overhead. Recall that the DVMRP control overhead per node increases in proportion to the number of (source, multicast group) pairs in the network. Even though only two multicast streams were active in the MBone experiment, the multicast routing table for all active source group pairs was forwarded through control packets and caused high overhead. In fact, the multicast routing table refresh requires one IGMP poll per multicast group and per source in the group. In the local DVMRP experiment however, the control message overhead is much lower since there are no external multicast group pairs. There is also an increase in the total throughput, but this result is not an indication of the performance improvement. The two multicast channels (streams) that are propagated to the wireless network in the MBone experiment are source rate limited to conserve bandwidth. Our result reflects the source limiting.

13.3.2.4 ODMRP Trace Analysis

In the ODMRP experiments, we kept the topology the same as in the DVMRP experiments. The ODMRP multicast routing daemon does not need the sta-

Table 13.2: DVMRP with a local source.

	Value	% of total
Avg. session length	N/A	N/A
IGMP Control packet O/H	0.79 kb/s	0.13 %
Avg. number of active multicast channels	1	N/A
O/H caused by multicast channels	0 kb/s	0 %
Effective data throughput	608.5 kb/s	99.87%
Total throughput (data and control)	609.29 kb/s	100 %

bilizing period required by the DVMRP since no control packets are switched between routers to establish the initial state. Currently, the MBone traffic cannot be forwarded onto the ODMRP testbed, so we replicated only the single source multicasting of the Section 5.4 experiment. Table 13.3 shows the results.

By comparing Table 13.2 with Table 13.3, we note that neither DVMRP or ODMRP experiments achieve the full WaveLAN data rate of 2 Mb/s. This result is due to the multihop forwarding restrictions on a common channel along the path <*A*-*C*-*E*-*F*>. When the source node A sends the packet, node C receives the packet and forwards it to node E. This initial forwarding process reduces the throughput to half of the original. One half of the channel is used for receiving the packet at node C and the other half for sending the packet to node E. When node E forwards the data packet to node F, the available channel bandwidth is further reduced to a third of the original. Namely, in optimal conditions the channel operates as a TDM channel with 3 slots per frame. Only one slot is active in any frame (in correspondence with the active hop). This performance degradation was already observed in the early packet radios unicast experiments [71]. Note also that we are using UDP because of multicasting. If TCP were used then the

Table 13.3: ODMRP with a local source.

	Value	% of total
Avg. session length	N/A	N/A
Control packet O/H	1.12 kb/s	0.18 %
Avg. number of active multicast channels	1	N/A
O/H caused by multicast channels	0 kb/s	0 %
Effective data throughput	610.1 kb/s	99.82 %
Total throughput (data and control)	611.22 kb/s	100 %

multihop throughput would be further degraded by TCP acknowledgments [54].

Since the forwarding nodes relay the packets only if the `FG_FLAG` is set, no unnecessary forwarding exists. The dynamic adaptation scheme in ODMRP uses control packets to adapt to route changes caused by node movements or by changes in intermediate link quality. Recall that DVMRP, when there is a severe change in the link condition, simply discontinues the forwarding of the multicast streams. Typical DVMRP routing table update frequency is inadequate to track rapid topology changes of wireless networks. Because of this limitation, DVMRP can be used only in a relatively stable environment. Our experiments are based on a stationary network. Thus, as expected, ODMRP and DVMRP give comparable performances. If mobility were introduced into our testbed, link and topology changes would make ODMRP performance superior to DVMRP. Although we aimed the current stationary experiments at verifying the correctness of the ODMRP, future experiments will evaluate its efficiency in mobile scenarios.

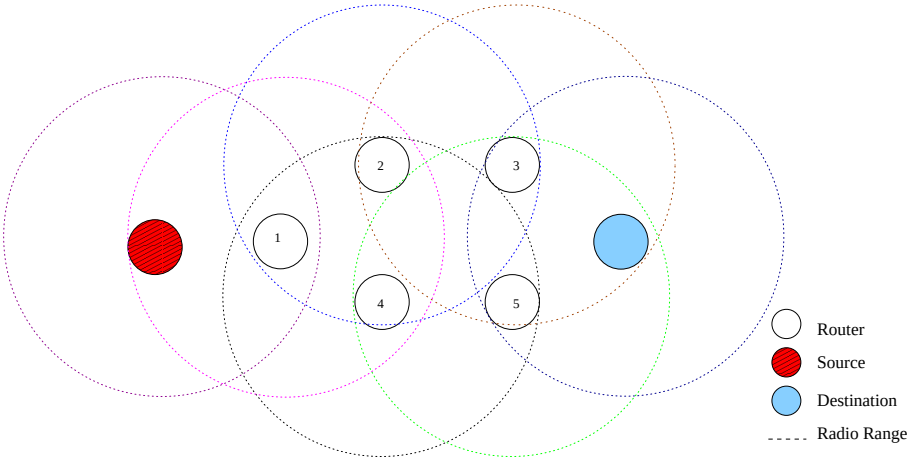


Figure 13.3: Multihop testbed topology in a static network.

13.3.3 Unicast Experiments

13.3.3.1 Efficiency Evaluation of Static Multihop Channel

A static multihop wireless network was constructed in a topology shown in Figure 13.3. The figure depicts the conceptual view of the building where the experiments were conducted. The router nodes were placed in each corner of the building and had line-of-sight accesses to two other routers. The walls of the building prevented the radio contact and the routers had accesses to one another only when the transceivers were in line-of-sight. A UDP packet transfer program described in the previous section was used to send 2307 packets of size 1556 bytes from the source node to the destination node. The results are presented in Table 13.4. The packet loss rate and ODMRP control overhead were measured to record the channel efficiency in static networks with no mobility. Even in a static network, the multihop channel suffers from large packet losses. Packets are lost because of the channel contention caused by the intermediate nodes competing to transmit, buffer overflow, channel noise, and packet collision.

Table 13.4: Unicast bandwidth distribution in a static wireless network.

	Value	% of bandwidth	% of packet loss
Avg. session length	65.64 sec	N/A	N/A
Control packet O/H	0 kb/s	0 %	0 %
Throughput	311.70 kb/s	100 %	28.75 %

13.3.3.2 ODMRP Performance in a Static Network

The initial experiment was conducted to investigate the performance of ODMRP in a non-mobile environment. We used the topology shown in Figure 13.3 to make performance comparison with the network running static routing. The same UDP packet transfer procedure was used. The efficiency of the channel was evaluated in Table 13.5. The result differs from that of the static network shown in Table 13.4 because of the control packet overhead and the packet loss caused by the route updates. The routes change frequently even when there is no mobility among the nodes. The JOIN QUERY packets arrive at the destination node through several alternate paths and the first arrived packet invokes a route update in the reverse path. The condition of the radio channel changes because of the ambient noise. Even when the network has no mobility, the optimal route may differ for each route update period due to the change in the channel condition. Once the packet transfer starts, UDP packets dominate the channel usage and often disrupts the route discovery sequence. The control packet has to contend for the channel with data packets and in the worst case, as little as one fourth of JOIN QUERY packets is forwarded all the way to the destination. This small JOIN QUERY delivery rate makes routes to change less frequently once the data transmission begins. Since there is no mobility in this experiment, low route change rate gives better performances as less packet losses are caused by route

Table 13.5: Unicast bandwidth distribution in a static ODMRP network.

	Value	% of bandwidth	% of packet loss
Avg. session length	66.76 sec	N/A	N/A
Control packet O/H	1.44 kb/s	0.47 %	28 %
Throughput	304.03 kb/s	99.5 %	29.32 %

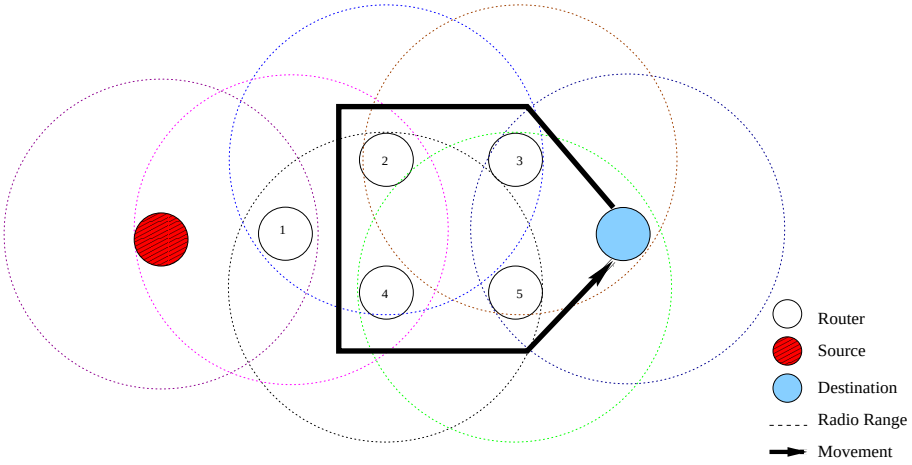


Figure 13.4: Multihop testbed topology with end node mobility.

updates.

13.3.3.3 ODMRP Performance with End-Node Mobility

In our second experiment, we measured the performance of the ODMRP ad hoc network when the end node was mobile. The basic topology remained the same as the previous sections, but mobility was introduced to the destination node. The destination node was transported following the path indicated in Figure 13.4 in an approximate speed of 1 meter/second. The UDP packet transfer was performed from the sender to the receiver in the same manner as in the previous experiments.

Table 13.6: Unicast bandwidth distribution in an ODMRP network with end-node mobility.

	Value	% of bandwidth	% of packet loss
Avg. session length	63.13 sec	N/A	N/A
Control packet O/H	1.53 kb/s	0.49 %	23 %
Throughput	307.72 kb/s	99.5 %	29.32 %

As the destination node moves, the routes are changed. Even though route update is disrupted by the normal flow of the data, it is not crucial in protocol performance. Data packets initially follow the route of <*source*-*router*1-*router*2-*router*3-*destination*>. As the destination moves closer to *router 3*, it enters into the radio range of *router 3*. The route change may take up to four seconds and by the time the actual route update occurs, the destination node may be within the transmission range of *router 2*. This delay in route update does not interfere with the flow of data as long as the destination is within the radio range of *router 3*. In the next successful route discovery sequence, the route is updated to <*source*-*router*1-*router*2-*destination*>. Because the route update has a minimal effect on data transmissions, the channel efficiency of ODMRP with the end node mobility is equivalent to the efficiency shown in no mobility case (shown in Tables 13.6 and 13.5, respectively).

13.3.3.4 ODMRP Performance with Intermediate Node Mobility

In this experiment, we measured the performance of the ODMRP ad hoc network when an intermediate node was mobile. The *router 4* was abruptly transported out of range of ad hoc network in path <*source*-*router*4-*router*5-*destination*> depicted in Figure 13.5. The UDP packet transfer was performed from the source

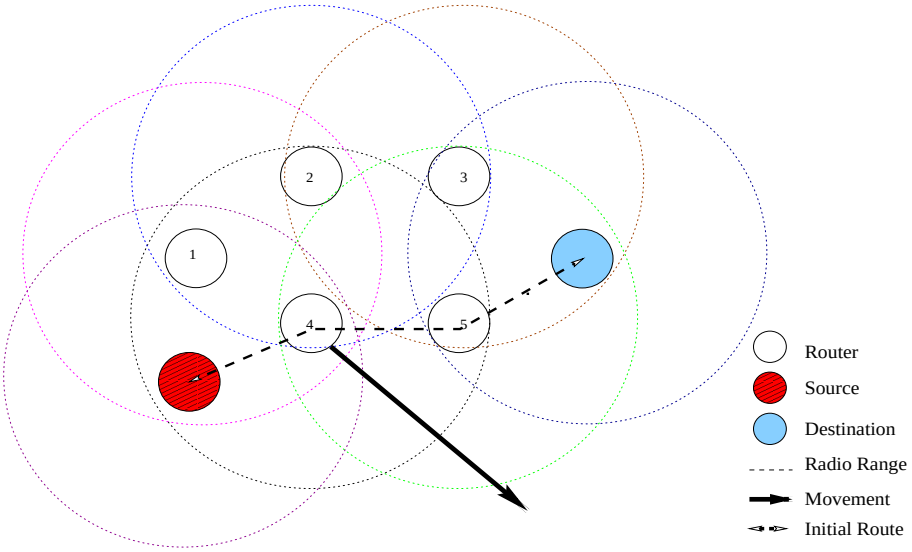


Figure 13.5: Multihop testbed topology with intermediate node mobility.

to the destination in the same manner as in the previous sections. As the router moves out of propagation range of its neighbors, one of two following scenarios occurs. If the mobile router was a part of the active route, the data transfer to the destination abruptly halts and packets are lost. Since the source is not immediately aware of the disruption in the data path, it continues to transmit data. Since *router 4* is now isolated from the network, there is less contention in the MAC layer. The source is able to send more packets quickly since no forwarding node exists to contend for the channel. There are much larger volume of packets flowing out of sender then there are from *router 1* and *router 1* cannot grab the channel. In this experiment, ODA is running on *router 1* with the sender as it is a client node. *Router 1* is the node which initiates the route refresh process so no new route can be discovered until it succeeds in transmission. The route update delay caused by the channel capture effect forces the low channel efficiency noted in Table 13.7. The re-established channel can slow down the transfer if the destination does not receive the JOIN QUERY packet from the

Table 13.7: Unicast bandwidth distribution in an ODMRP network with intermediate node mobility.

	Value	% of bandwidth	% of packet loss
Avg. session length	39.14 sec	N/A	N/A
Control packet O/H	.367 kb/s	0.12 %	77.06 %
Throughput	238.73 kb/s	99.85 %	67.46 %

path <*source-router*1-*router*2-*router*3-*destination*>, and receives the packet on its way back from *router 5* in the path <*source-router*1-*router*2-*router*3-*router*5-*destination*>, establishing a non-optimal route (in hop distance) as the new route. The second scenario is the case where the *router 4* that moved out of range was not part of the current forwarding path. In this scenario, the transmission continues without delay. The non-optimal route described above can also be built in this scenario. In Table 13.7, the result for the first scenario is collected and analyzed. The second scenario yields result almost identical to Table 13.5 and hence is not shown.

13.3.4 Experiences in Using Applications over Ad Hoc Networks

The testbed setup was operated with the existing applications to verify the reliability and robustness of ad hoc routing scheme in day to day operations. Virtual Network Computing (VNC) client-server [137] by AT&T was used to access and remotely control the end nodes. Telnet and FTP sessions were held to test the end-to-end TCP continuity. Live video streams were generated with Microsoft Netmeeting (Figures 13.6 and 13.7) to test the feasibility of multimedia application in multihop wireless networks. As expected, the performance of these applications were adequate, but less then spectacular. The applications often

Figure 13.6: Microsoft Netmeeting operating over the ad hoc testbed.

had the packet loss problem because of some random environmental interferences (e.g., pedestrians, elevator, cordless phone, etc.) even when there existed a strong radio channel. In a wired environment, packet loss indicates congestion along the route. The applications either compensate for the congestion by sending less packets or changing the data compression scheme, wait for a timeout and rerouting. When applied to the wireless environment, the heuristics built into the applications did not improve the performance. The concept of transparent layering dictates that the application layer should not be aware of layers underneath it. However, to optimize the performance, the application has to be keenly aware of its environment and take an active part in applying appropriate heuristics.

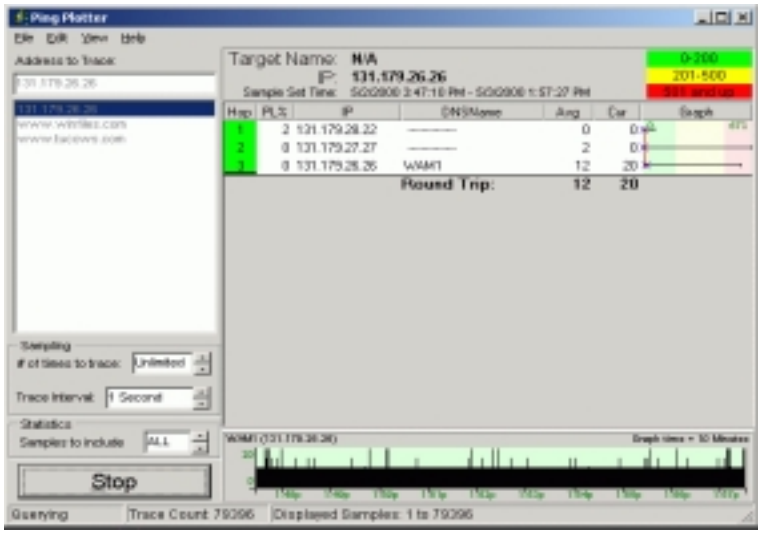


Figure 13.7: PingPlotter operating over the ad hoc testbed.

13.4 Conclusion

We presented our experimental analysis of ODMRP (On-Demand Multicast Routing Protocol) for an ad hoc wireless network. We implemented ODMRP and performed the multicast experiments on a six node wireless ad hoc testbed with Linux operating system. Our multicast experiments consisted of forwarding MBone traffic from wired to wireless network using DVMRP gateway and routers, and streaming multicast traffic in the wireless testbed using ODMRP and DVMRP routers. We analyzed and tabulated the traffic data for experiments. Our experiments confirmed the fact that DVMRP with MBone multicast feed introduces high channel overhead because of the forwarding of unnecessary data stream caused by the control message losses in wireless channels. When replacing the MBone feed with a local source, DVMRP multicast overhead disappeared since there is only one source. In the local source multicast scenario, ODMRP and DVMRP performance is almost identical since control packet overhead is very

low (and comparable). DVMRP however, is not expected to perform well in mobile networks due to lack of inherent fast adaptivity. Therefore, ODMRP is expected to outperform DVMRP in a mobile environment.

We also studied the unicast performance of ODMRP in a real ad hoc network testbed with seven networks hosts. We learned that protocols suffer from packet losses even in static networks because of channel contention, noise, and interference. We introduced various node mobility to the network and presented the throughput results. Our experiments demonstrated ODMRP's ability to dynamically adapt to a mobile routing environment. An end-to-end unicast connection was carried on with a minimal network overhead. We also discussed the need for application's awareness toward its environment to optimize network performances.

CHAPTER 14

Conclusion

Wireless mobile ad hoc networks present difficult challenges to routing and multicasting protocol designers. Mobility, constrained bandwidth, and limited power cause frequent topology changes. Routing protocols must construct and maintain multihop routes in dynamic ad hoc networks effectively and efficiently. We studied various routing and multicasting approaches in this dissertation. The main lesson learned from our studies is that on-demand protocols are well suited for mobile ad hoc networks, especially when the mobility rate is high. Efficient utilization of control packets is the primary reason of good performances. Providing alternate and multiple routes proved to be beneficial because they increase robustness to mobility and fading. Our simulation results showed that in both unicast and multicast situations, protocols that build multiple routes on demand perform well under mobility. We summarize our contribution in what follows:

- We conducted a performance evaluation of various routing and multicast protocols of different styles. Protocols were analyzed in diverse network scenarios to assess their relative strengths, weaknesses, and applications. Our results gave meaningful indications to protocol designers in this area.

- We performed a large scale simulation of up to 10,000 nodes and evaluated the routing protocol scalability. We also introduced schemes to enhance the routing performance in such large networks. Our study is the first to run

detailed ad hoc network simulations of that magnitude.

- We investigated the interaction between MAC and routing protocols in ad hoc network communications. It is determined that the choice of MAC layer protocol does, in fact, affect the relative performance of the routing protocols.

- We designed multipath routing schemes for ad hoc unicast communications. We studied two approaches; Ad hoc On-demand Distance Vector with Backup Routes (AODV-BR) uses alternate paths only when the primary route is not available, and Split Multipath Routing (SMR) builds maximally disjoint paths to distribute data traffic. Our study suggests that providing alternate and multiple paths is particularly useful in ad hoc networks.

- We introduced a novel route selection metric that considers the load of network hosts. We described three algorithms of Dynamic Load-Aware Routing (DLAR) protocol that utilize this metric. Our results indicate that the shortest delay route is not always optimal in ad hoc networks because it can easily cause congestion.

- We proposed the On-Demand Multicast Routing Protocol (ODMRP). The protocol reactively creates a mesh structure that provides multiple routes between multicast members. It also has the unicast capability. We evaluated the protocol by simulation and implemented in a real ad hoc network testbed. The protocol is currently one of the leading candidate for standardization of the IETF MANET working group.

Some of the future research directions in this ad hoc network area include:

- Reliable multicast
- Congestion and admission control
- Flow control
- Load balancing
- Quality of Service (QoS) provision
- Power efficient protocol design
- Security and privacy
- Interoperation with wired/cellular networks.

References

[1] A. Acharya and B.R. Badrinath, "A Framework for Delivering Multicast Messages in Networks with Mobile Hosts," *ACM/Baltzer Mobile Networks and Applications*, vol. 1, no. 2, October 1996, pp. 199-219.

[2] S. Agarwal, A. Ahuja, J.P. Singh, and R. Shorey, "Route-Lifetime Assessment Based Routing (RABR) Protocol for Mobile Ad-Hoc Networks," *Proceedings of the IEEE International Conference on Communications (ICC)*, New Orleans, LA, June 2000, pp. 1697-1701.

[3] G. Aggelou and R. Tafazolli, "RDMAR: A Bandwidth-Efficient Routing Protocol for Mobile Ad Hoc Networks," *Proceedings of the ACM International Workshop on Wireless Mobile Multimedia (WoWMoM)*, Seattle, WA, August 1999, pp. 26-33.

[4] P. Agrawal, D.K. Anvekar, and B. Narendran, "Optimal Prioritization of Handovers in Mobile Cellular Networks," *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communication (PIMRC)*, The Hague, Netherlands, Sepember 1994, pp. 1393-1398.

[5] I.F. Akyildiz, W. Yen, and B. Yener, "A New Hierarchical Routing Protocols for Dynamic Multihop Wireless Networks," *Proceeding of the IEEE Conference on Computer Communications (INFOCOM)*, Kobe, Japan, April 1997, pp. 1422-1429.

[6] V. Aravamudhan, K. Ratnam, and S. Rangarajan, "An Efficient Multicast Protocol for PCS Networks," *ACM/Baltzer Mobile Networks and Applications*, vol. 2, no. 4, January 1998, pp. 333-344.

[7] S. Bae, S.-J. Lee, and M. Gerla, "Unicast Performance Analysis of the ODMRP in a Mobile Ad hoc Network Testbed," *Proceedings of the IEEE International Conference on Communications and Networks (ICCCN)*, Las Vegas, October 2000, to appear.

[8] S. Bae, S.-J. Lee, W. Su, and M. Gerla, "The Design, Implementation, and Performance Evaluation of On-Demand Multicast Routing Protocol in Multihop Wireless Networks," *IEEE Network*, special issue on Multicasting Empowering the Next Generation Internet, vol. 14, no. 1, January/February 2000, pp. 70-77.

[9] S. Bae, S.-J. Lee, W. Su, and M. Gerla, "Implementation of a Multicast Routing Protocol in a Wireless Ad hoc Network Testbed," *Technical Re-*

port, Computer Science Department, University of California, Los Angeles, 990049, November 1999.

[10] 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.

[11] T. Ballardie, P. Francis, and J. Crowcroft, "Core Based Trees (CBT) - An Architecture for Scalable Inter-Domain Multicast Routing," *Proceedings of the ACM SIGCOMM Symposium on Communications Architectures, Protocols and Applications*, San Francisco, CA, October 1993, pp. 85-95.

[12] A. Bartoli, "Group-Based Multicast and Dynamic Membership in Wireless Networks with Incomplete Spatial Coverage," *ACM/Baltzer Mobile Networks and Applications*, vol. 3, no. 2, August 1998, pp. 175-188.

[13] S. Basagni, "Distributed Clustering for Ad Hoc Networks," *Proceedings of the IEEE International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN)*, Perth, Western Australia, June 1999, pp. 310-315.

[14] S. Basagni, I. Chlamtac, V.R. Syrotiuk, and B.A. Woodward, "A Distance Routing Effect Algorithm for Mobility (DREAM)," *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, Dallas, TX, October 1998, pp. 76-84.

[15] R.E. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, 1957.

[16] A. Bestavros and I. Matta, "Load Profiling for Efficient Route Selection in Multi-Class Networks," *Proceedings of the IEEE International Conference on Network Protocols (ICNP)*, Atlanta, GA, October 1997, pp. 183-190.

[17] D. A. Beyer, "Accomplishments of the DARPA Survivable Adaptive Networks SURAN Program," *Proceedings of the IEEE Military Communications Conference (MILCOM)*, Monterey, CA, October 1990, pp. 855-862.

[18] D. Beyer, M. Frankel, J. Hight, D. Lee, M. Lewis, R. McKenney, J. Naar, R. Ogier, N. Shacham, and W. Zaumen, "Packet Radio Network Research, Development and Application," *Proceedings of the SHAPE Packet Radio Symposium*, 1989.

[19] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LANs," *Proceedings of the ACM SIGCOMM Conference on Communications Architectures, Protocols and Applications*, London, UK, September 1994, pp. 212-225.

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

[21] L. Briesemeister and G. Hommel, "Role-Based Multicast in Highly Mobile but Sparsely Connected Ad Hoc Networks," *Proceedings of the ACM/IEEE Workshop on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Boston, MA, August 2000, pp. 45-50.

[22] J. Broch, D.B. Johnson, and D.A. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," *Internet Draft*, `draft-ietf-manet-dsr-00.txt`, March 1998. Work in progress.

[23] 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 the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, Dallas, TX, October 1998, pp. 85-97

[24] R. Castaneda and S.R. Das, "Query Localization Techniques for On-demand Routing Protocols in Ad Hoc Networks," *Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, Seattle, WA, August 1999, pp. 186-194.

[25] J.-H. Chang and L. Tassiulas, "Energy Conserving Routing in Wireless Ad-hoc Networks," *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, Tel Aviv, Israel, March 2000, pp. 22-31.

[26] J. Chen, P. Druschel, and D. Subramanian, "An Efficient Multipath Forwarding Method," *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, San Francisco, CA, March 1998, pp. 1418-1425.

[27] S. Chen and K. Nahrstedt, "Distributed Quality-of-Service Routing in Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, special issue on Wireless Ad Hoc Networks, vol. 17, no. 8, August 1999, pp. 1488-1505.

[28] T.-W. Chen and M. Gerla, "Global State Routing: A New Routing Scheme for Ad-hoc Wireless Networks," *Proceedings of the IEEE International Conference on Communications (ICC)*, Atlanta, GA, June 1998, pp. 171-175.

[29] C.-C. Chiang and M. Gerla, "On-Demand Multicast in Mobile Wireless Networks," *Proceedings of the IEEE International Conference on Network Protocols (ICNP)*, Austin, TX, October 1998, pp. 262-270.

[30] C.-C. Chiang, M. Gerla, and L. Zhang, "Adaptive Shared Tree Multicast in Mobile Wireless Networks," *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, Sydney, Australia, November 1998, pp. 1817-1822.

[31] C.-C. Chiang, M. Gerla, and L. Zhang, "Forwarding Group Multicast Protocol (FGMP) for Multihop, Mobile Wireless Networks," *Baltzer Cluster Computing*, special issue on Mobile Computing, vol. 1, no. 2, 1998, pp. 187-196.

[32] C.-C. Chiang, H.-K. Wu, W. Liu, and M. Gerla, "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel," *Proceedings of the IEEE Singapore International Conference on Networks (SICON)*, Singapore, April 1997, pp. 197-211.

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

[34] M. Correa, K. Tang, and M. Gerla, "Isolation of Wireless Ad hoc Medium Access Mechanisms Under UDP," *Technical Report*, Computer Science Department, University of California, Los Angeles, 990035, June 1999.

[35] M.S. Corson and S.G. Batsell, "A Reservation-Based Multicast (RBM) Routing Protocol for Mobile Networks: Initial Route Construction Phase," *ACM/Baltzer Wireless Networks*, vol. 1, no. 4, December 1995, pp. 427-450.

[36] M.S. Corson and A. Ephremides, "A Distributed Routing Algorithm for Mobile Wireless Networks," *ACM/Baltzer Wireless Networks*, vol. 1, no. 1, February 1995, pp.61-81.

[37] M.S. Corson and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," *Request For Comments 2501*, Internet Engineering Task Force, January 1999.

[38] S.R. Das, R. Castaneda, J. Yan, and R. Sengupta, "Comparative Performance Evaluation of Routing Protocols for Mobile, Ad hoc Networks," *Proceedings of the IEEE International Conference on Computer Communications and Networks (ICCCN)*, Lafayette, LA, October 1998, pp. 153-161.

[39] 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.

[40] S.E. Deering and D.R. Cheriton, "Multicast Routing in Datagram Internetworks and Extended LANs," *ACM Transactions on Computer Systems*, vol. 8, no. 2, May 1990, pp. 85-110.

[41] S. Deering, D.L. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu, and L. Wei, "The PIM Architecture for Wide-Area Multicast Routing," *IEEE/ACM Transactions on Networking*, vol. 4, no. 2, April 1996, pp. 153-162.

[42] C. Diot, W. Dabbous, and J. Crowcroft, "Multipoint Communication: A Survey of Protocols, Functions, and Mechanisms," *IEEE Journal on Selected Areas in Communications*, special issue on Multipoint Communications, vol. 15, no. 3, April 1997, pp. 277-290.

[43] R. Dube, C.D. Rais, K.-Y. Wang, and S.K. Tripathi, "Signal Stability-Based Adaptive Routing (SSA) for Ad Hoc Mobile Networks", *IEEE Personal Communications Magazine*, vol. 4, no. 1, February 1997, pp.36-45.

[44] H. Eriksson, "MBONE: The Multicast Backbone," *Communications of the ACM*, vol. 37, no. 8, August 1994, pp. 54-60.

[45] W. Fenner, "Internet Group Management Protocol, Version 2," *Request For Comments 2236*, Internet Engineering Task Force, November 1997.

[46] L.R. Ford and D.R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, NJ, 1962.

[47] C.L. Fullmer and J.J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access (FAMA) for Packet-Radio Networks," *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, Cambridge, MA, August 1995, pp. 262-273.

[48] J.J. Garcia-Luna-Aceves and D. Beyer, "SPARROW: Secure Protocols for Adaptive, Robust, Reliable, and Opportunistic WINGs," University of California at Santa Cruz and Rooftop Communications Corporation, http://www.cse.ucsc.edu/research/ccrg/projects/sparrow.html.

[49] J. J. Garcia-Luna-Aceves, C. L. Fullmer, E. Madruga, D. Beyer, and T. Frivold, "Wireless Internet Gateways (WINGS)," *Proceedings of the IEEE Military Communications Conference (MILCOM)*, Monterey, CA, November 1997, pp. 1271-1276.

[50] J.J. Garcia-Luna-Aceves and E.L. Madruga, "The Core-Assisted Mesh Protocol," *IEEE Journal on Selected Areas in Communications*, special issue on Wireless Ad Hoc Networks, vol. 17, no. 8, August 1999, pp. 1380-1394.

[51] J.J. Garcia-Luna-Aceves and E.L. Madruga, "A Multicast Routing Protocol for Ad-Hoc Networks," *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, New York, NY, March 1999, pp. 784-792.

[52] J.J. Garcia-Luna-Aceves and M. Spohn, "Source-Tree Routing in Wireless Networks," *Proceedings of the IEEE International Conference on Network Protocols (ICNP)*, Toronto, Canada, October, 1999, pp. 273-282.

[53] J.J. Garcia-Luna-Aceves and M. Spohn, "Scalable Link-State Internet Routing," *Proceedings of the IEEE International Conference on Network Protocols (ICNP)*, Austin, TX, October 1998, pp. 52-61.

[54] M. Gerla, R. Bagrodia, L. Zhang, K. Tang, and L. Wang, "TCP over Wireless Multihop Protocols: Simulation and Experiments," *Proceedings of the IEEE International Conference on Communications (ICC)*, Vancouver, Canada, June 1999, pp. 1089-1094.

[55] M. Gerla, C.-C. Chiang, and L. Zhang, "Tree Multicast Strategies in Mobile, Multihop Wireless Networks," *ACM/Baltzer Mobile Networks and Applications*, speical issue on Mobile Ad Hoc Networking, vol. 4, no. 3, October 1999, pp. 193-207.

[56] M. Gerla, G. Pei, and S.-J. Lee, "Wireless, Mobile Ad-Hoc Network Routing," *Proceedings of the ACM/IEEE WINLAB/Berkeley FOCUS*, New Brunswick, NJ, May 1999.

[57] M. Gerla and J.T.-C. Tsai, "Multicluster, Mobile, Multimedia Radio Network," *ACM/Baltzer Wireless Networks*, vol. 1, no. 3, March 1995, pp. 255-265.

[58] S.K.S. Gupta and P.K. Srimani, "An Adaptive Protocol for Reliable Multicast in Mobile Multi-hop Radio Networks," *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, New Orleans, LA, February 1998, pp. 111-122.

[59] Z.J. Haas and S. Tabrizi, "On Some Challenges and Design Choices in Ad-Hoc Communications," *Proceedings of the IEEE Military Communications Conference (MILCOM)*, Bedford, MA, October 1998, pp. 187-192.

[60] 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.

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

[62] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen, "Scalable Routing Strategies for Ad Hoc Wireless Networks," *IEEE Journal on Selected Areas in Communications*, special issue on Wireless Ad Hoc Networks, vol. 17, no. 8, August 1999, pp. 1369-1379.

[63] P. Jacquet, P. Muhlethaler, and A. Qayyum, "Optimized Link State Routing Protocol," *Internet-Draft*, `draft-ietf-manet-olsr-00.txt`, November 1998, Work in progress.

[64] L. Ji, M. Ishibashi, and M. S. Corson, "An Approach to Mobile Ad hoc Network Protocol Kernel Design," *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, New Orleans, LA, Sepember 1999, pp. 1303-1307.

[65] L. Ji and M.S. Corson, "A Lightweight Adaptive Multicast Algorithm," *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, Sydney, Australia, November 1998, pp. 1036-1042.

[66] M. Jiang, J. Li, and Y.C. Yay, "Cluster Based Routing Protocol (CBRP) Functional Specification," *Internet-Draft*, `draft-ietf-manet-cbrp-spec-01.txt`, August 1999, Work in progress.

[67] M. Joa-Ng and I.-T. Lu, "A Peer-to-Peer Zone-Based Two-Level Link State Routing for Mobile Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, special issue on Wireless Ad Hoc Networks, vol. 17, no. 8, August 1999, pp. 1415-1425.

[68] P. Johanson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, "Scenario-based Performance Analysis of Routing Protocols for Mobile Adhoc Networks," *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, Seattle, WA, August 1999, pp. 195-206.

[69] 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.

[70] J. Jubin, "Uphill Tier Routing, Less Frequent Tier Data Updating, and Larger Networks," *SURAN Program Technical Note (SRNTN) 2*, Richardson, TX, Rockwell Inc., 1983.

[71] 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.

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

[73] P. Karn. "MACA - A New Channel Access Protocol for Packet Radio," *Proceedings of the ARRL/CRRL Amateur Radio Ninth Computer Networking Conference*, September 1990, pp. 134-140.

[74] B. Karp and H.T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, Boston, MA. August 2000, pp. 243-254.

[75] L. Kleinrock and J. Silvester, "Optimum Transmission Radii for Packet Radio Networks or Why Six is a Magic Number," *Proceedings of the IEEE National Telecommunications Conference (NTC)*, Birmingham, AL, December 1978, pp. 4.3.2-4.3.5.

[76] L. Kleinrock and F.A. Tobagi, "Packet Switching in Radio Channels: Part I - Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics," *IEEE Transactions on Communications*, vol. COM-23, no. 12, December 1975, pp. 1400-1416.

[77] Y.-B. Ko and N.H. Vaidya, "Geocasting in Mobile Ad Hoc Networks: Location-Based Multicast Algorithms," *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, New Orleans, LA, February 1999, pp. 101-110.

[78] Y.-B. Ko and N.H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, Dallas, TX, October 1998, pp. 66-75.

[79] P. Krishna, N.H. Vaidya, M. Chatterjee, D.K. Pradhan, "A Cluster-based Approach for Routing in Dynamic Networks," *ACM SIGCOMM Computer Commnications Review*, vol. 27, no. 2, April 1997, pp. 49-64.

[80] R. Krishnan and J.A. Silvester, "Choice of Allocation Granularity in Multipath Source Routing Schemes," *Proceedings of the IEEE Conference on Computer Commuincations (INFOCOM)*, San Francisco, CA, March 1993, pp. 322-329.

[81] G. Lauer, "Hierarchical Routing Design for SURAN," *Proceedings of the IEEE International Conference on Communications (ICC)*, Toronto, Canada, June 1988, pp. 93-102.

[82] S. Lee and C. Kim, "Neighbor Supporting Ad hoc Multicast Routing Protocol," *Proceedings of the ACM/IEEE Workshop on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Boston, MA, August 2000, pp. 37-44.

[83] S.-J. Lee and M. Gerla, "SMR: Split Multipath Routing with Maximally Disjoint Paths in Ad hoc Networks," *Technical Report*, Computer Science Department, University of California, Los Angeles, August 2000.

[84] S.-J. Lee and M. Gerla, "Dynamic Load-Aware Routing in Ad hoc Networks," *Technical Report*, Computer Science Department, University of California, Los Angeles, August 2000.

[85] S.-J. Lee and M. Gerla, "AODV-BR: Backup Routing in Ad hoc Networks," *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Chicago, IL, Sepember 2000, to appear.

[86] S.-J. Lee, M. Gerla, and C.-C. Chiang, "On-Demand Multicast Routing Protocol," *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, New Orleans, LA, September 1999, pp. 1298-1302.

[87] S.-J. Lee, M. Gerla, and C.-K. Toh, "Simulation Study of Table-Driven and On-Demand Routing Protocols for Mobile Ad Hoc Networks," *IEEE Network*, vol. 13, no. 4, July/August 1999, pp. 48-54.

[88] S.-J. Lee, J. Hsu, R. Hayashida, M. Gerla, and R. Bagrodia, "Selecting Routing Strategies for Your Ad Hoc Networks," *Technical Report*, Computer Science Department, University of California, Los Angeles, 990045, October 1999.

[89] S.-J. Lee, E.M. Royer, and C.E. Perkins, "Ad hoc Routing Protocol Scalability," *Technical Report*, Computer Science Department, University of California, Los Angeles, July 2000.

[90] S.-J. Lee, W. Su, and M. Gerla, "Exploiting the Unicast Functionality of the On-Demand Multicast Routing Protocol," *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Chicago, IL, September 2000, to appear.

[91] S.-J. Lee, W. Su, and M. Gerla, "On-Demand Multicast Routing Protocol in Multihop Wireless Mobile Networks," *ACM/Baltzer Mobile Networks and Applications*, special issue on Multipoint Communication in Wireless Mobile Networks, 2000, to appear.

[92] S.-J. Lee, W. Su, and M. Gerla, "Wireless Ad hoc Multicast Routing with Mobility Prediction," *ACM/Baltzer Mobile Networks and Applications*, special issue on Routing and Multicasting in Wireless Networks, 2000, to appear.

[93] S.-J. Lee, W. Su, and M. Gerla, "Ad hoc Wireless Multicast with Mobility Prediction," *Proceedings of the IEEE International Conference on Computer Communications and Networks (ICCCN)*, Boston, MA, October 1999, pp. 4-9.

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

[95] S.-J. Lee, W. Su, J. Hsu, M. Gerla, R. Bagrodia, "A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols," *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, Tel Aviv, Israel, March 2000, pp. 565-574.

[96] S.-J. Lee, C.-K. Toh, and M. Gerla, "Performance Evaluation of Table-Driven and On-Demand Ad Hoc Routing Protocols," *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communication (PIMRC)*, Osaka, Japan, September 1999, pp. 297-301.

[97] J. Li, J. Jannotti, D.S.J. De Couto, D.R. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing," *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, Boston, MA. August 2000, pp. 120-130.

[98] C.R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 7, September 1997, pp. 1265-1275.

[99] C.R. Lin and S.-W. Chao "A Multicast Routing Protocol for Multihop Wireless Netorks," *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, Rio de Janeiro, Brazil, December 1999, pp. 235-239.

[100] C.R. Lin and J.-S. Liu, "QoS Routing in Ad Hoc Wireless Networks," *IEEE Journal on Selected Areas in Communications*, special issue on Wireless Ad Hoc Networks, vol. 17, no. 8, August 1999, pp. 1426-1438.

[101] E.L. Madruga and J.J. Garcia-Luna-Aceves, "Scalable Multicasting: The Core Assisted Mesh Protocol," *ACM/Baltzer Mobile Networks and Applications*, special issue on Management of Mobility in Distributed Systems, 2000, to appear.

[102] E.L. Madruga and J.J. Garcia-Luna-Aceves, "Multicasting Along Meshes in Ad-Hoc Networks," *Proceedings of the IEEE International Conference on Communications (ICC)*, Vancouver, Canada, June 1999, pp. 314-318.

[103] G. Malkin, "RIP Version 2 - Carrying Additional Information," *Internet Draft*, `draft-ietf-ripv2-protocol-v2-05.txt`, June 1998. Work in progress.

[104] G.S. Malkin and M.E. Steenstrup, "Distance-Vector Routing," In *Routing in Communications Networks*, edited by M.E. Steenstrup, Prentice Hall, 1995, pp. 83-98.

[105] D.A. Maltz, J. Broch, J. Jetcheva, and D.B. Johnson, "The Effects of On-Demand Behavior in Routing Protocols for Multihop Wireless Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, special issue on Wireless Ad Hoc Networks, vol. 17, no. 8, August 1999, pp. 1439-1453.

[106] D.A. Maltz, J. Broch, and D.B. Johnson, "Quantitative Lessons From a Full-Scale Multi-Hop Wireless Ad Hoc Network Testbed," *Proceedings of the IEEE Wireless Communications and Networking Conference*, Chicago, IL, September 2000, to appear.

[107] I. Matta and M. Krunz, "Packing and Least-Loaded Based Routing in Multi-Rate Loss Networks," *Proceedings of the IEEE International Conference on Communications (ICC)*, Montreal, Canada, June 1997, pp. 827-831.

[108] J.M. McQuillan, I. Richer, and E.C. Rosen, "The New Routing Algorithm for the ARPANET," *IEEE Transactions on Communications*, vol. COM-28, no. 5, May 1980, pp. 711-719.

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

[110] J. Moy, "OSPF Version 2," *Request For Comments 2328*, Internet Engineering Task Force, April 1998.

[111] J. Moy, "Link-State Routing," In *Routing in Communications Networks*, edited by M.E. Steenstrup, Prentice Hall, 1995, pp. 135-157.

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

[113] Multicast and MBONE on Linux - Application, http://www.teksouth.com/linux/multicast/applications.html.

[114] S. Murthy and J.J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks," *ACM/Baltzer Mobile Networks and Applications*, special issue on Routing in Mobile Communications Networks, vol. 1, no. 2, October 1996, pp. 183-197.

[115] S. Murthy and J.J. Garcia-Luna-Aceves, "Congestion-Oriented Shortest Multipath Routing," *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, San Francisco, CA, March 1996, pp. 1028-1036.

[116] B. Narendran, P. Agrawal, and D.K. Anvekar, "Minimizing Cellular Handover Failures Without Channel Utilization Loss," *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, San Francisco, CA, December 1994, pp. 1679-1685.

[117] A. Nasipuri, R. Castaneda, and S.R. Das, "Performance of Multipath Routing for On-Demand Protocols in Mobile Ad Hoc Networks," *ACM/Baltzer Mobile Networks and Applications*, 2000, to appear.

[118] A. Nasipuri and S.R. Das, "On-Demand Multipath Routing for Mobile Ad Hoc Networks," *Proceedings of the IEEE International Conference on Cmputer Communications and Networks (ICCCN)*, Boston, MA, October 1999, pp. 64-70.

[119] 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, March 1993, pp. 443-455.

[120] T. Ozaki, J.B. Kim, and T. Suda, "Bandwidth-Efficient Multicast Routing Protocol for Ad-Hoc Networks," *Proceedings of the IEEE International Conference on Computer Communications and Networks (ICCCN)*, Boston, MA, October 1999, pp. 10-17.

[121] V.D. Park and M.S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, Kobe, Japan, April 1997, pp. 1405-1413.

[122] M.R. Pearlman and Z.J. Haas, "Determining the Optimal Configuration for the Zone Routing Protocol," *IEEE Journal on Selected Areas in Communications*, special issue on Wireless Ad Hoc Networks, vol. 17, no. 8, August 1999, pp. 1395-1414.

[123] G. Pei, M. Gerla, T.-W. Chen, "Fisheye State Routing: A Routing Scheme for Ad Hoc Wireless Networks," *Proceedings of the IEEE International Conference on Communications (ICC)*, New Orleans, LA, June 2000, pp. 70-74.

[124] G. Pei, M. Gerla, and X. Hong, "LANMAR: Landmark Routing for Large Scale Wireless Ad Hoc Networks with Group Mobility," *Proceedings of the ACM/IEEE Workshop on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Boston, MA, August 2000, pp. 11-18.

[125] G. Pei, M. Gerla, X. Hong, and C.-C. Chiang, "A Wireless Hierarchical Routing Protocol with Group Mobility," *Proceedigns of the IEEE Wireless Communications and Networking Conference (WCNC)*, New Orleans, LA, September 1999, pp. 1538-1542.

[126] C. E. Perkins, "IP Mobility Support," *Request For Comments 2002*, Internet Engineering Task Force, October 1996.

[127] C.E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers," *Proceedings of the ACM SIGCOMM Conference on Communications Architectures, Protocols and Applications*, London, UK, September 1994, pp. 234-244.

[128] C.E. Perkins and E.M. Royer, "Ad-Hoc On Demand Distance Vector Routing," *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, New Orleans, LA, Febrary 1999, pp. 90-100.

[129] C.E. Perkins, E.M. Royer, and S.R. Das, "Ad Hoc On Demand Distance Vector (AODV) Routing," *Internet Draft*, draft-ietf-manet-aodv-05.txt, March 2000, Work in progress.

[130] R. Prakash, "Unidirectional Links Prove Costly in Wireless Ad-Hoc Networks," *Proceedings of the ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL M for Mobility)*, Seattle, WA, August 1999, pp. 15-22.

[131] M.B. Pursley and H.B. Russell, "Routing in Frequency-Hop Packet Radio Networks with Partial-Band Jamming," *IEEE Transactions on Communications*, vol. 41, no. 7, July 1993, pp. 1117-1124.

[132] J. Raju and J.J. Garcia-Luna-Aceves, "A New Approach to On-demand Loop-Free Multipath Routing," *Proceedings of the IEEE International Conference on Computer Communications and Networks (ICCCN)*, Boston, MA, October 1999, pp. 522-527.

[133] R. Ramanathan and M. Steenstrup, "Hierarchically-Organized, Multihop Mobile Wireless Networks for Quality-of-Service Support," *ACM/Baltzer Mobile Networks and Applications*, special issue on Mobile Multimedia Communications, vol. 3, no. 1, June 1998, pp. 101-119.

[134] S. Ramanathan and M. Streenstrup, "A Survey of Routing Techniques for Mobile Communication Networks," *ACM/Baltzer Mobile Networks and Applications*, special issue on Routing in Mobile Communications Networks, vol. 1, no. 2, October 1996, pp. 89-104.

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

[136] T.S. Rappaport, S.Y. Seidel, and K. Takamizawa, "Statistical Channel Impulse Response Models for Factory and Open Plan Building Radio Communication System Design," *IEEE Transactions on Communications*, vol. COM-39, no. 5, May 1991, pp. 794-807.

[137] T. Richardson, Q. Stafford-Fraser, K.R. Wood, and A. Hopper, "Virtual Network Computing," *IEEE Internet Computing*, Vol.2 No.1, January/February 1998, pp. 33-38.

[138] . E.M. Royer, S.-J. Lee, and C.E. Perkins, "The Effects of MAC Protocols on Ad hoc Communication Protocols," *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Chicago, IL, September 2000, to appear.

[139] E.M. Royer and C.E. Perkins, "Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Protocol," *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, Seattle, WA, August 1999, pp. 207-218.

[140] 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.

[141] P. Sass, "Communications Networks for the Force XXI Digitized Battlefield," *ACM/Baltzer Mobile Networks and Applications*, special issue on Mobile Ad Hoc Networking, vol. 4, no. 3, August 1999, pp. 139-155.

[142] N. Shacham, "Hierarchical Routing in Large, Dynamic Ground Radio Networks," *Proceedings of the Hawaii International Conference on System Sciences (HICSS)*, Honolulu, HI, January 1985, pp. 292-301.

[143] N. Shacham, E.J. Craighill, and A.A. Poggio, "Speech Transport in Packet-Radio Networks with Mobile Nodes," *IEEE Journal on Selected Areas in Communications*, vol. SAC-1, no. 6, December 1983, pp. 1084-1097.

[144] A. Shaikh, J. Rexford, and K.G. Shin, "Load-Sensitive Routing of Long-Lived IP Flows," *Proceedings of the ACM SIGCOMM Conference on Communications Architectures, Protocols and Applications*, Cambridge, MA, September 1999, pp. 215-226.

[145] D. Sidhu, R. Nair, and S. Abdallah, "Finding Disjoint Paths in Networks," *Proceedings of the ACM SIGCOMM Symposium on Communications Architectures, Protocols and Applications*, Zurich, Switzerland, September 1991, pp. 43-51.

[146] S. Singh, M. Woo, and C.S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks," *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, Dallas, TX, October 1998, pp. 181-190.

[147] P. Sinha, R. Sivakumar, and V. Bharghavan, "MCEDAR: Multicast Core-Extraction Distributed Ad hoc Routing," *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, New Orleans, LA, September 1999, pp. 1313-1317.

[148] R. Sivakumar, P. Sinha, and V. Bharghavan, "CEDAR: a Core-Extraction Distributed Ad hoc Routing Algorithm," *IEEE Journal on Selected Areas in Communications*, special issue on Wireless Ad Hoc Networks, vol. 17, no. 8, August 1999, pp. 1454-1465.

[149] A.J. Smith, "Cache Memories," *ACM Computing Surveys*, vol. 14, no. 3, September 1982, pp. 473-530.

[150] J. Stevens, "Spreading Connectivity Information out over Multiple PROP Periods, and Timeliness of Information," *SURAN Program Technical Note (SRNTN) 21*, Richardson, TX, Rockwell Inc., May 1985.

[151] W. Su and M. Gerla, "IPv6 Flow Handoff in Ad Hoc Wireless Networks Using Mobility Prediction," *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, Rio de Janeiro, Brazil, December 1999, pp. 271-275.

[152] W. Su, S.-J. Lee, and M. Gerla, "Mobility Prediction in Wireless Networks," *Proceedings of the IEEE Military Communications Conference (MILCOM)*, Los Angeles, CA, October 2000, to appear.

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

[154] N. Taft-Plotkin, B. Bellur, and R. Ogier, "Quality-of-Service Routing Using Maximally Disjoint Paths," *Proceedings of the IEEE International Workshop on Quality of Service (IWQoS)*, London, UK, June 1999, pp. 119-128.

[155] A.S. Tanenbaum, *Computer Networks*, 3rd Edition, Prentice Hall, Upper Saddle River, NJ, March 1996.

[156] F.A. Tobagi and L. Kleinrock, "Packet Switching in Radio Channels: Part-II - The Hidden Terminal Problem in Carrier Sense Multiple-Access Models and the Busy-Tone Solution," *IEEE Transactions on Communications*, vol. 23, no. 12, December 1975, pp. 1417-1433.

[157] H. Tode, Y. Sakai, M. Yamamoto, H. Okada, and Y. Tezuka, "Multicast Routing Algorithms for Nodal Load Balancing," *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, Florence, Italy, May 1992, pp. 2086-2095.

[158] C.-K. Toh, "A Novel Distributed Routing Protocol to Support Ad Hoc Mobile Computing," *Proceedings of the IEEE International Phoenix Conference on Computers and Communications (IPCCC)*, Scottsdale, AZ, March 1996, pp. 480-486.

[159] C.-K. Toh, "Associativity-Based Routing for Ad-Hoc Mobile Networks," *Wireless Personal Communications Journal*, special issue on Mobile Networking and Computing Systems, Kluwer Academic Publishers, vol. 4, no. 2, March 1997, pp. 103-139.

[160] 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.

[161] W. van der Moolen, "IEEE 802.11 WaveLAN PC Card User's Guide," Lucent Technologies, June 1998.

[162] S. Vutukury and J.J. Garcia-Luna-Aceves, "An Algorithm for Multipath Computation Using Distance-Vectors with Predecessor Information," *Proceedings of the IEEE International Conference on Computer Communications and Networks (ICCCN)*, Boston, MA, October 1999, pp. 534-539.

[163] Z. Wang and J. Crowcroft, "Shortest Path First with Emergency Exits," *Proceedings of the ACM SIGCOMM Symposium on Communications Architectures and Protocols*, Philadelphia, PA, September 1990, pp. 166-176.

[164] D.S.L. Wei and K. Naik, "An Efficient Multicast Protocol Using de Bruijn Structure for Mobile Computing," *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 3, July 1997, pp. 14-35.

[165] J.E. Wieselthier, G.D. Nguyen, and A. Ephremides, "Algorithms for Energy-Efficient Multicasting in Ad Hoc Wireless Networks," *Proceedings of the IEEE Military Communications Conference (MILCOM)*, Atlantic City, NJ, November 1999, pp. 1414-1418.

[166] C.L. Williamson, T.G. Harrison, W.L. Mackrell, and R.B. Bunt, "Performance Evaluation of the MoM Mobile Multicast Protocol," *ACM/Baltzer Mobile Networks and Applications*, special issue on Protocols and Software Paradigms of Mobile Networks, vol. 3, no. 2, August 1998, pp. 189-201.

[167] C.W. Wu and Y.C. Tay, "AMRIS: A Multicast Protocol for Ad hoc Wireless Networks," *Proceedings of the IEEE Military Communications Conference (MILCOM)*, Atlantic City, NJ, November 1999, pp. 25-29.

[168] W.T. Zaumen and J.J. Garcia-Luna-Aceves, "Loop-Free Multipath Routing Using Generalized Diffusing Computations," *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, San Francisco, CA, March 1998, pp. 1408-1417.

[169] H. Zhou and S. Singh, "Content Based Multicast (CBM) in Ad Hoc Networks," *Proceedings of the ACM/IEEE Workshop on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Boston, MA, August 2000, pp. 51-60.