ABSTRACT

JUN, JANGEUN. Networking in Wireless Ad Hoc Networks. (Under the direction of Assistant Professor Mihail L. Sichitiu.)

In modern communication systems, wireless ad hoc networking has become an irreplaceable technology where communication infrastructure is insufficient or unavailable. An ad hoc network is a collection of self-organizing nodes that are rapidly deployable and adaptable to frequent topology changes. In this dissertation, the key problems related to the network layer (i.e., forwarding, routing, and network-layer topology control) are addressed. The problem of unfair forwarding in ad hoc nodes is identified and cross-layer solutions are proposed. Because a typical ad hoc node functions both as a router and a host, severe unfairness occurs between originated and forwarded packets which eventually leads to a serious starvation problem. The results show that, to restore the fairness and enhance the capacity efficiency, non-traditional queueing schemes are required where both the network and the MAC layers should be considered together. Routing is a critical protocol, which directly affects the scalability and reliability of wireless ad hoc networks. A good routing protocol for wireless ad hoc networks should overcome the dynamic nature of the topology arising from unreliable wireless links and node mobility. In ad hoc networks, it is very important to balance the route accuracy and overhead efficiency. A number of routing protocols have been proposed for wireless ad hoc networks, but it is well known that current routing protocols scale poorly with the number of nodes, the number of traffic flows, the intensity of mobility. The main objective of this dissertation is to provide efficient routing protocols for different types of wireless ad hoc networks including wireless mesh networks (WMNs), mobile ad hoc networks (MANETs), and wireless sensor networks (WSNs). Since each category has different assumptions and constraints, different solutions should be considered. WMNs and WSNs have low mobility and centralized (one-to-any) traffic patterns while MANETs have relatively high mobility and uniform (any-to-any) traffic patterns. WSNs are highly resource-constrained while WMNs are not. A new routing protocol specially designed for WMNs is proposed. Simulation experiments show that the protocol outperforms existing generic ad hoc routing protocols. This improvement is enabled by the essential characteristics of WMNs, and as a result, the protocol does not rely on bandwidth-greedy flooding mechanism. For MANET routing, an existing de facto the standard Internet intra-AS (autonomous system)

routing protocol is extended to enhance the scalability in ad hoc environments. When extended for MANETs, Open Shortest Path First (OSPF) is expected to provide the benefits of maturity, interoperability, and scalability. The scalability extension is two-fold: the notions of distance effect and multiple areas are explored as extensions. Both approaches provide significant gain in scalability by efficiently reducing flooding overhead without compromising routing or forwarding performance. Finally, a new scalable and reliable sensor network routing is proposed. Since WSNs are the most resource-constrained type of ad hoc networks, the protocol should be very simple yet reliable. The proposed WSN routing protocol is designed to provide reliability (via multi-path redundancy), scalability (with efficiently contained flooding), and flexibility (source-tunable per-packet priority), which are achieved without adding protocol complexity or resource consumption. The protocol is implemented on real sensor motes and its performance is tested through outdoor sensor field deployments. The results show that the protocol outperforms even sophisticated link estimation based sensor network routing protocols.

Networking in Wireless Ad Hoc Networks

by

Jangeun Jun

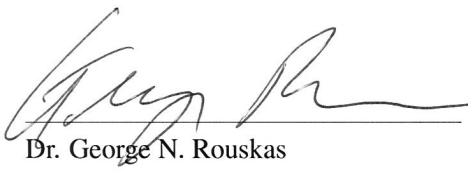
A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Computer Engineering

Raleigh

2006

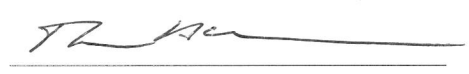
Approved By:



Dr. George N. Rouskas

Dr. Do Young Eun

Dr. Mihail L. Sichitiu
Chair of Advisory Committee



Dr. Arne A. Nilsson

To my parents, my parents-in-law, and my wife Jihae for their sacrifice and support.

To my two sons, Alex and Eric, who are the source of indescribable joy in my life.

Biography

Jangeun Jun was born in South Korea in 1971. He graduated from Pusan National University in February 1997 with the Bachelor of Science degree in Electronics Engineering. After spending a couple of years in a broadcasting company, he became a network planning and deployment engineer at the DACOM corporation, a leading ISP and telecommunications company in Seoul, South Korea. He participated in the Korean Information Superhighway project deploying national networking infrastructure using ATM and Frame Relay switches. He was engaged in a project to stabilize the backbone routers of the Internet. His experience includes engineering leased-line networks and dial-up access networks. He is also experienced in network security and traffic engineering in large-scale server farm networks for e-Commerce services. In 2000, he was selected as a recipient of the national scholarship, awarded by the Korean government which enabled him to pursue a graduate degree is the United States. He joined the Department of Electrical and Computer Engineering at North Carolina State University in Raleigh in Fall 2001. He earned the Master of Science degree in Computer Networking (CNE) in December 2002 with a thesis on the capacity of wireless mesh networks. After achieving his Master's degree, he continued his work as a doctoral student under the guidance of Dr. Sichitiu. His research was focused on discovering and solving problems in cutting-edge wireless communication networks including wireless mesh networks, mobile ad hoc networks, and wireless sensor networks. He was the first student member of WALAN (Wireless Ad hoc and Local Area Networks) research laboratory at NCSU. He also worked as a teaching assistant for the graduate and undergraduate courses of Wireless Communication Systems and Communication Engineering. Before finishing his doctoral degree, he spent his Summer at Los Alamos National Laboratory as a graduate research assistant, which enabled him to carry out sensor network deployment and ad hoc network simulation experiments using their advanced equipment. From Fall 2006, he is joining the faculty of West Virginia University - Institute of Technology as an assistant professor in the Department of Electrical and Computer Engineering.

Acknowledgements

Foremost, I would like to thank my advisor Dr. Mihail L. Sichitiu for his thoughtful directions and affectionate encouragements. This dissertation would not have been materialized without his careful guidance and continuous support. I owe Dr. Sichitiu every bit of my research accomplishments and exciting experiences I benefited from the doctoral work at NCSU. I would like to acknowledge my advisory committee members Drs. Arne A. Nilsson, George N. Rouskas and Do Young Eun for their enlightening comments and constructive suggestions on my work. I thank Dr. J. Keith Townsend for his kind guidance as the instructor of the courses for which I served as a teaching assistant. The IT national scholarship program of the Korean Government that partially supported my graduate study is gratefully acknowledged. In addition, most of my doctoral research projects were supported by the Center for Advanced Computing and Communication (CACC) and Cisco Systems. I sincerely appreciate their support that enabled the accomplishments presented in this dissertation. I would like to thank Drs. Nicolas W. Hengartner and Stephan J. Eidenbenz at Los Alamos National Laboratory (LANL) for having me in their research group. At LANL, I used their advanced equipment to carry out the important experiments required in my dissertation. I acknowledge all the staffs and students in LANL CCS-5 group for helping my research through constructive discussions and fruitful collaborations. Finally, I want to thank all the members of the Wireless Ad hoc and Local Area Network (WALAN) laboratory at NCSU for their heart-warming cooperation and family-like care. I miss those days when all the WALAN students got together every week to present the progress, discuss the problems, and share the ideas.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The origin of information network is inseparably related to that of humanity. Without exception, an efficient networking system was essential in the success of any colossal empires. Genghis Khan's rule of its vast territory was made possible by the network of fast post horses. The type of network system that relays messages over a "multihop" beacon poles even dates back to ancient Persia. Various network systems served as a nerve net of the society and quenched the thirst of plain people wanting to be "connected" with others. As well represented in the oriental word for "human" which literally means "relation among people", the innate sociality of mankind has led us to incessant invention of new networking technologies over the history, which in turn advanced the civilization and enriched its culture.

It is widely acknowledged that the two most successful modern networks are telephone network and the Internet. The impact of the information revolution brought about by the Internet is comparable to the invention of telephone network. The evolution of the Internet dramatically changed almost all the fundamental aspects of today's human life including economy, culture, society, education, and politics. Its application area is rapidly extending the boundary into various fields including conventional telephony and broadcasting. Clearly, the Internet is evolving towards convergence and ubiquity.

Although the original design of the Internet and telephone network are completely different (voice versus data, circuit-switched versus packet-switched, simple terminals with smart core versus the opposite, and etc.), they share an important property in common - both require establishment of fixed infrastructure with careful planning and centralized administration. Unfortunately, their dependence upon

infrastructure imposes limits on their usefulness in some situations. Such conventional infrastructure-oriented networks have improved their coverage and flexibility by advanced wireless technologies such as cellular networks, wireless local area networks (WLANs), wireless local loops (WLLs), and wireless Internet (e.g., Mobile IP). However, there still remains a significant gap between such infrastructure-oriented networks and networking demands in many scenarios.

In contrast to the type of networks that rely on pre-deployed communication infrastructure, ad hoc networks attempt to meet the networking demand where it is very difficult or inefficient, if not impossible, to establish a network infrastructure. Ad-hoc networks are formed by a group of mobile nodes with wireless transceivers that can dynamically organize themselves into a temporary network with an arbitrary topology. In such a rapidly-deployable and self-organizing network, the network topology is controlled in a distributed manner and user traffic is forwarded in multihop fashion by collaboration of participating nodes.

The earliest implementation of wireless ad hoc network, although it was stationary and single hop, dates back to the historic ALOHAnet in 1970s where several Hawaiian islands were connected by sending packets over a single-channel multi-access wireless link. The invention of ALOHAnet was followed by DARPA-sponsored PRNET project that proved the feasibility of ad hoc networking over unstable wireless links and dynamically changing topology. DARPA continued its support for multihop wireless networking with SURAN project for improved scalability and survivability. In 1997, IETF chartered the MANET working group to expedite the standardization efforts in ad hoc protocols and functionality. Recently, ad hoc networks are proving their great potential in commercial applications as well as the military arena. Some representative examples of ad hoc networks are wireless sensor networks, wireless mesh networks, networks for emergency rescue operations, and temporary networks for disaster-ridden cities as witnessed in the case of hurricane Katrina.

Ad hoc networks require solutions to the problems that are hardly found in conventional networks. There has been a lot of research effort in this direction. Despite the research activities, there are still a lot of open problems. The research challenges in ad hoc networking span almost all the layers of protocol stack. Some widely known issues are in the areas of reliable transport protocol, routing, medium access control (MAC), energy efficiency, security, time synchronization, cross-layer interaction, etc. Most of these issues arise from the mobility of nodes and high error rate of wireless links.

Among various research challenges in wireless ad hoc networking, we investigate multiple

key problems that primarily pertain to the network layer. We first investigate the problem of unfair forwarding in ad hoc networks. Since a typical ad hoc node functions as a router and a host, there can be a serious unfairness between originated and forwarded traffic. We propose several solutions to overcome the unfairness and analyze their performance. Then, we study routing protocols for ad hoc networks that scale well with the number of nodes. It is well known that the problem of accomplishing an efficient routing protocol is especially challenging for ad hoc networks, and it as the main issue of this dissertation.

In multihop wireless networks, fair allocation of bandwidth among different nodes is one of the critical problems that affects the serviceability of the entire system. Although there is significant research on the fairness issues in single-hop wireless networks, research on multihop forwarding fairness is rarely found in the literature. We study various queuing schemes for multihop wireless networks and examine the forwarding fairness and throughput performance of each scheme. Each scheme offers a different degree of fairness. While relatively simple queuing schemes require less hardware and processing budget, they inevitably lack good fairness and performance. In contrast, the scheme that provides fairness requires per-flow (i.e., network-layer flow) queuing. Furthermore, we show that in order to achieve the optimum bandwidth utilization, a cross-layer approach between the network and the medium access control (MAC) layer is critical where the MAC layer should be able to support different priorities based on the network-layer status. Without such a cross-layer scheme, in the worst case, the bandwidth utilization can be degraded by $O(N)$, where N is the number of end users. We theoretically investigate the pros and cons of different queuing schemes and validate the analytical results with detailed simulations.

Efficient routing in ad hoc networks is especially difficult to accomplish due to the dynamic and frequent change in its topology which makes traditional routing schemes inappropriate. The absence of centralized control (as found in cellular systems) makes it difficult to keep track of mobile nodes and ever-changing topology. Maintaining accurate routes in a distributed manner requires a significant control overhead. Unfortunately, wireless links in ad hoc networks are bandwidth-limited and error-prone, and mobile nodes have limited processing and energy resources. Therefore, it is very difficult to have an optimized ad hoc routing protocol with an ideal balance between the investment (i.e., control overhead) and the return (i.e., route optimality). The design of the protocol becomes even more difficult due to high error rates caused by fading, interference or collisions as well as hidden and exposed terminals.

A good ad hoc network routing protocol has to do more than just finding stable, loop-free,

low-cost paths. It is expected to be fully distributed, to rapidly adapt to topology changes, to allow small route setup delay, to be robust against fluctuating wireless links, to efficiently detect and remove stale routes, to suppress flooding (i.e., multihop broadcasting), and etc. Therefore, it has to be agile to topology changes and make fast decision while utilizing minimal bandwidth, computation, memory, and energy. Some further requirements are to be scalable to a large number of nodes or to provide QoS support to enable time-sensitive services.

A variety of routing protocols have been proposed for wireless ad hoc networks. The proposed ad hoc routing protocols can be classified into several categories. Table-driven (or proactive) protocols have the overhead of periodic and proactive routing updates while on-demand (or reactive) protocols have relatively larger route setup latency due to reactive approach. To achieve scalability, some protocols form and maintain hierarchical topologies while others operate on flat networks. There are protocols relying on additional information such as signal strength information passed from the physical layer, battery level of the node, or geographical information obtained from GPS. There are a class of protocols that make predictive decision. It is well known that current routing protocols scale poorly with the number of nodes, number of flows and increase in mobility.

The main objective of this dissertation is to provide efficient routing protocols for wireless ad hoc networks that scale well and outperform existing solutions. We propose new routing protocols or extensions to the existing ones to achieve the scalability and reliability in wireless ad hoc networks. First, we propose a new routing scheme specially designed for wireless mesh networks. Second, we propose an efficient routing overhead reduction scheme and apply to an existing Internet routing protocol as an extension for mobile ad hoc networks. Third, we explore the embedded network partitioning capability of the protocol to further enhance the scalability. And finally, we propose a scalable and reliable routing scheme for wireless sensor networks.

Wireless mesh networks are a special class of wireless ad hoc networks. Some characteristics distinguish wireless mesh networks from generic ad hoc networks. In mesh networks, one or more special nodes called gateways exist to provide user nodes with an exit or entry point to the outer world (e.g., to the Internet or to the backbone of corporate intranet). In such a network, the majority of traffic flows (i.e., source-destination pairs at the network layer) are assumed to be established between end users and the gateway (as opposed to the assumption of any-to-any traffic pattern in typical ad hoc networks). Wireless mesh networks are expected to have a significant number of fixed user nodes to

provide stable coverage and connectivity. The new protocol takes into account these particularities of wireless mesh networks and as a result, it forms and maintains a tree-structured route paths without relying on flooding. The protocol can be classified as distance vector and it operates in three different modes - proactive, reactive and hybrid (i.e., a combination of the two). In clear contrast to typical ad hoc routing protocols, none of the three modes relies on flooding, and thus, the protocol shows better overhead efficiency and scalability than using generic ad hoc routing protocols for wireless mesh networks.

Mobile ad hoc networks are another class of wireless ad hoc networks where all the participating nodes are uniformly assumed to be mobile and are expected to be equally likely to send data to any peer in the network. Therefore, the topology is more dynamic and the traffic flow pattern is any-to-any. These assumptions make the routing more difficult than wireless mesh networks or wireless sensor networks where mobility is relatively low and traffic pattern is one-to-any. In such a case, the inevitably large routing overhead is one of the most serious problems that impedes the scalability of the routing protocol. We focus our investigation on the so-called “distance effect” to achieve the scalability. Unless source routing is used, the typical store-and-forward operation on each user node relies on next-hop look-up in the forwarding table which is populated by the routing protocol. Therefore, there exists a high temporal and spacial locality in the requirement of route accuracy. Unlike existing schemes that typically turn to formation of a hierarchy that take complexity or external information such as location, we propose a novel way of suppressing the routing overhead at the expense of no additional control but simply the hop-count distance information which is readily available in most cases. The proposed scheme is applied to an existing Internet routing protocol as an extension for mobile ad hoc environments. The result from our study shows that the proposed scheme provides good scalability at small loss of route optimality.

We extend our exploration of adapting existing Internet routing protocol for mobile ad hoc environments. We focus on the embedded network partitioning capability of the Open Shortest Path First (OSPF) routing protocol as a scalability enhancement for mobile ad hoc networks. When adapted to an ad hoc environment, OSPF is expected to provide benefits of maturity, interoperability, and scalability. Existing scalability enhancements for OSPF are mostly based on single OSPF area. By introducing the notion of multiple areas, scalability can be dramatically increased. A novel analytical model is developed to capture the relationship between the number of areas and the flooding overhead. We perform

theoretical analysis to show that there exists an optimum area number that minimizes the overhead. When the optimum number of areas is used, the overhead is significantly reduced. The analytical results are validated with detailed simulation experiments. Several candidate dynamic area formation schemes are studied with different level of realism in the mobility model. The results show that the optimum area number has a greater impact on the scalability in more realistic scenarios.

Wireless sensor networks are a class of wireless ad hoc networks with a very constrained resource budget. Thus, a routing protocol's scalability and reliability is more important in sensor networks than in other classes of ad hoc networks. A number of approaches have been proposed for sensor network routing, but it is hard to meet the unique requirements of sensor networks. Furthermore, outdoor deployment studies with real implementation tends to be lacking in the literature. We address the problems of scalability and reliability in sensor network using a simple but powerful scheme implemented on real motes. We compare the performance of the proposed scheme with widely-used sensor network routing protocols. Experiments are performed in an outdoor sensor field, where detailed tests are carried out for the performance analysis. We present the implementation details and the experiment results obtained from head-to-head comparison of different routing protocols. The proposed protocol shows higher packet delivery ratio than existing protocols for the network of 10-hop diameter. The good performance of the proposed scheme can be explained by its key properties of high reliability (via multi-path redundancy), scalability (with efficiently contained flooding), and flexibility (source-tunable per-packet priority) which are achieved without adding protocol complexity or resource consumption. These strengths enable the protocol to outperform even more sophisticated protocols especially in adverse outdoor sensor field environments.

The remainder of this dissertation is organized as follows. Chapter 2 proposes various solutions for the forwarding unfairness problem in wireless ad hoc networks and presents the simulation results. Chapter 3 presents the design and performance analysis of a new wireless mesh networks routing protocol. In Chapter 4, a novel routing overhead reduction scheme using distance effect is discussed. Chapter 5 presents the study of multiple OSPF areas scheme as a scalability enhancement solution for ad hoc networks and discusses the impact of the optimum area number. Chapter 6 presents the implementation and performance analysis from the field deployment for a new sensor network routing protocol. Finally, Chapter 7 concludes this dissertation.

Chapter 2

Forwarding Fairness in Wireless Ad Hoc Networks

In this chapter, we propose various solutions for the forwarding unfairness problem in wireless ad hoc networks and present the analysis of performance results obtained from simulation experiments. In multihop wireless networks, fair allocation of bandwidth among different nodes is one of the critical problems that affects the serviceability of the entire system. Although there is significant research on the fairness issues in single-hop wireless networks, research on multihop forwarding fairness is rarely found in the literature. We study various queuing schemes for multihop wireless networks and examine the forwarding fairness and throughput performance of each scheme. Each scheme offers a different degree of fairness. While relatively simple queuing schemes require less hardware and processing budget, they inevitably lack good fairness and performance. In contrast, the scheme that provides fairness requires per-flow (i.e., network-layer flow) queuing. Furthermore, we show that in order to achieve the optimal bandwidth utilization, the medium access control (MAC) layer should be able to support different priorities. Without such a MAC-layer QoS scheme, in the worst case, the bandwidth utilization can be degraded by $O(N)$, where N is the number of end users. We theoretically investigate the pros and cons of different queuing schemes and validate the analytical results with detailed simulations.

2.1 Introduction to Multihop Fairness in Ad Hoc Networks

Fairness is one of the most important properties of a computer network: when network resources are unable to satisfy demand, they should be divided fairly between the clients of the network. Most often, *min-max fairness* [1] is the desired fairness scheme. Under this scheme, the clients are split into two groups: the first group consists of the clients that cannot be completely satisfied by network resources; they all receive the same share of bandwidth. The second group is made up of the clients that need less bandwidth than their fair share; they receive exactly the amount of bandwidth that they ask for. Either of the two groups can be empty. Considerable research has been done to ensure medium access control (MAC) layer fairness [2–4] and the result is that current wireless standards (e.g., IEEE 802.11 [2]) provide quite good MAC layer fairness at least in the long term. Unfortunately, as we will soon show, this does not ensure network layer fairness. An interesting practical mechanism providing proportional fairness is presented in [5]. With *proportional fairness*, clients that offer more load have a higher throughput, which may or may not be desirable depending on the particular applications of that network.

Another solution to the above problem can be provided by the transmission control protocol (TCP) [6] running over such an unfair network. TCP's behavior over ad-hoc networks was recently the subject of considerable research (see [7–14] and the references within). However not all traffic in multihop networks is TCP, and the fairness problem at the network layer can, and should be solved at the network layer. After all, it is the job of the network layer to provide fair resource allocations at the higher layers, and not vice versa. In this research, several schemes for providing fairness and differentiated services at the network layer are investigated.

2.2 Problem Formulation

A user node in a multihop network has to transmit both relayed and its own traffic. Therefore, besides the contention with other nodes for the same destination node, there is an inevitable contention between its own and relayed traffic. This contention does not occur in fixed wireless local loops or wireless LANs in infrastructure mode where user nodes are always at one-hop distance from the base station or the access point.

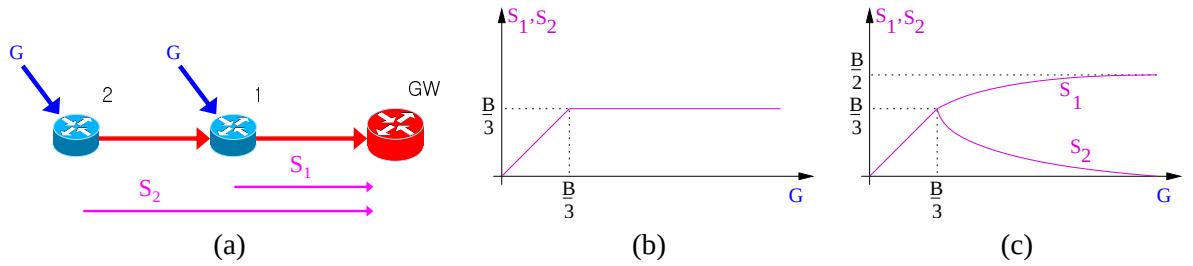


Figure 2.1: (a) Fairness study of a two-node network forwarding packets to a gateway GW. The ideal (b) and real (c) throughputs of nodes 1 and 2 as a function of the offered load G.

Consider the simple case depicted in Fig. 2.1-(a) where two nodes (1 and 2) are having the same offered load G to be sent to the gateway (GW). Ideally, as the offered load at each of the nodes (G) increases, both nodes will receive the same share of the MAC layer throughput, B (see Fig. 2.1-(b)). In practice, without a modified MAC or network layer, as the offered load increases, the node closest to the gateway (node 1 in Fig. 2.1-(a)) will gradually but completely starve the node further away from the gateway (as shown in Fig. 2.1-(c)). The results in Fig. 2.1-(c) are obtained under the assumptions that the MAC layer is fair and that the traffic to be forwarded by node 1 (from node 2 to the gateway) is queued together (either in the forwarding engine or at the MAC layer) with the traffic originating at node 1.

As can be seen in Fig. 2.1-(c), as the load at both nodes is increased, node 2 is gradually, but eventually completely starved by node 1. The overall system throughput (measured at the gateway) when the offered load is very high is $\frac{B}{2}$, where B is the throughput of the system when only node 1 forwards data to the gateway. So, the system operates at 75 % efficiency (because it gets $\frac{B}{2}$ while $\frac{2B}{3}$ is ideal) and is unfair.

The unfair behavior observed in Fig. 2.1-(c) is rooted in the fact that both the traffic originating at node 1, as well as the relayed traffic from node 2, are queued together at node 1. When the traffic load increases, the network cannot forward all data enqueued at node 1, and the queue starts to overflow. With a probability that increases with the offered load, the queue will be full when a new packet arrives from node 1, and it will be dropped immediately after it is received. The exact, expected throughput was determined theoretically, and it was verified using both OPNET and ns-2.

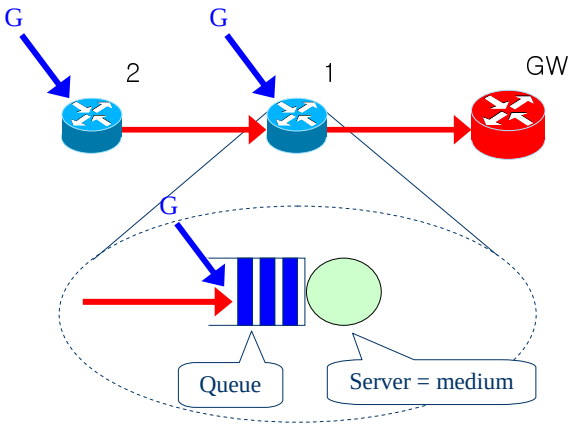
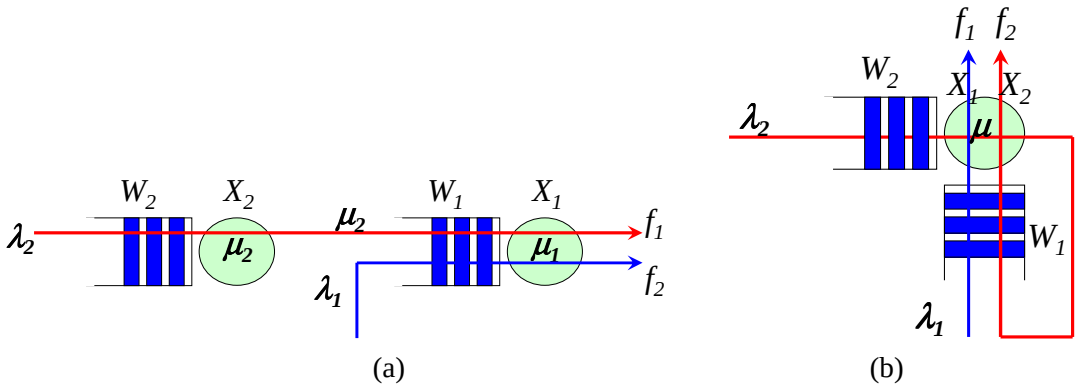


Figure 2.2: A node is modeled as a server with a queue.



(a) (b)

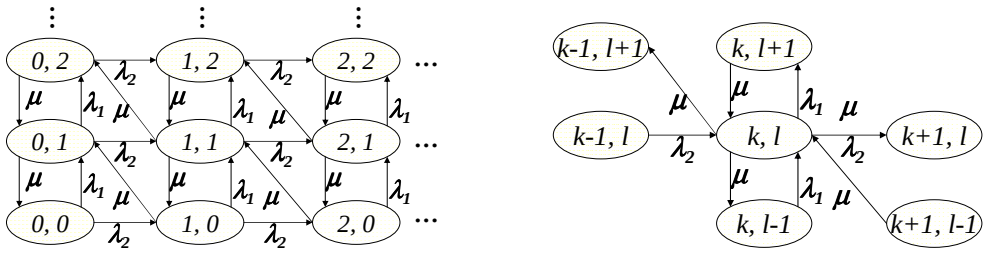
Figure 2.3: Modeling the two source chain network assuming (a) conventional network and (b) wireless network.

2.3 Analytical Modeling

In this section, an overview of the analytical model for the fairness study is presented. We consider only the MAC layer queue in the modeling. Figure 2.2 shows our model where a node is simplified as a server with a queue. The server represents the wireless medium.

If the network in Fig. 2.2 is assumed to be a conventional wired network, it can be modeled as Figure 2.3-(a) where each parameter is defined as follows:

- $\lambda_i \equiv$ arrival rate of i^{th} node (i.e., rate of the generated traffic)
- $\mu_i \equiv$ service rate of i^{th} node



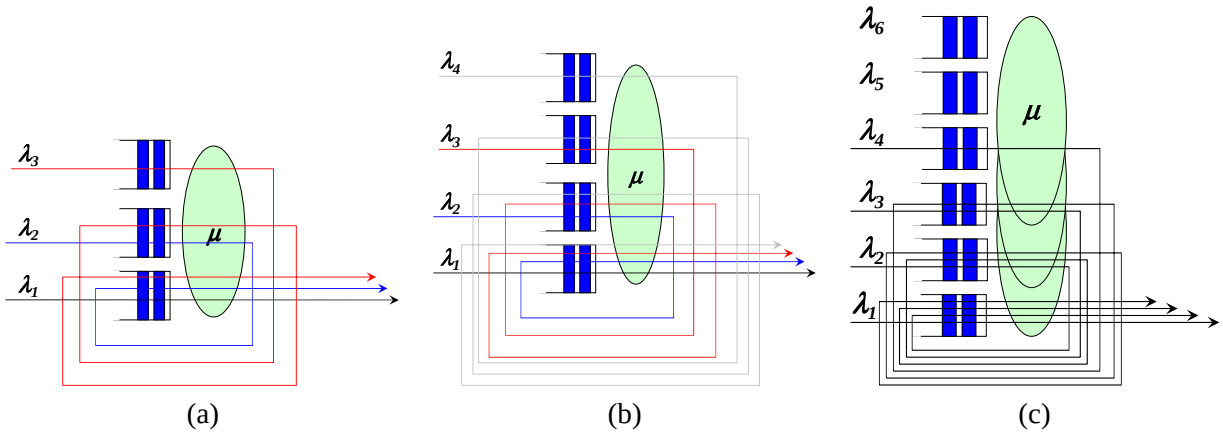
(a) (b)

Figure 2.4: (a) Two dimensional Markov chain and (b) its general state transition diagram for the two source chain network.

- $W_i \equiv$ queuing delay of i^{th} node
- $X_i \equiv$ service time of i^{th} node (= transmission delay + propagation delay + others (e.g., random back-offs))
- $f_i \equiv$ flow throughput of i^{th} node ($= \frac{1}{W_i + X_i}$)

However, the network in Fig. 2.2 is not correctly modeled as shown in Figure 2.3-(a) because it does not reflect the characteristics of real wireless environment. In a typical wireless ad hoc network, nodes are expected to have a single transceiver and share a common channel. Thus, the server in node 1 is not independent of the server in node 2. Assuming a con tension-based MAC protocol, the server in Fig. 2.3-(a) can be corrected as Fig. 2.3-(b) where two flows contend with each other to access the wireless medium. Figure. 2.3-(b) indicates that the traffic flow generated by node 2 has to go through two queues and get served twice by one server to reach the destination. In contrast, the traffic flow of node 1 goes through the queue and the server only once, which implies an unfair competition between the two flows unless otherwise controlled.

Two dimensional Markov chain can be drawn based on the model shown in Fig. 2.3-(b). Figure. 2.4-(a) and (b) show the Markov chain and its general state transition diagram, respectively. The solution to this model can be used to explain the unfair forwarding phenomenon observed in Fig. 2.1-(c). In the same manner, the analysis can be extended to the networks with longer chain. With more nodes, the complexity in traffic flow grows as shown in Fig. 2.5-(a) and (b). Spatial spectrum reuse makes the analysis even more difficult as shown in Fig. 2.5-(c) where multiple servers overlap with each other.



(a) (b) (c)

Figure 2.5: Generalization of modeling the chain network with an arbitrary length: (a) three sources, (b) four sources and (c) six sources.

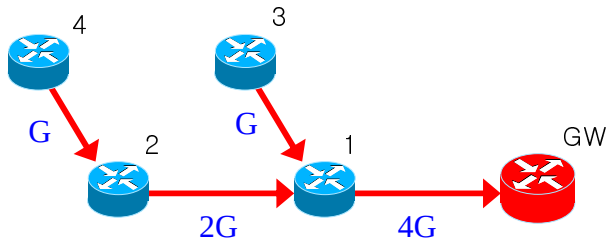


Figure 2.6: A simple multihop wireless network with four user nodes and a gateway.

2.4 Proposed Solutions

In this section, we propose various solutions to address the problem described in the previous section and explain the advantages and disadvantages of each scheme. For clarity, a simple multihop wireless network with four nodes as shown in Fig. 2.6 is used in the analysis. We assume that all the traffic flows are unidirectional toward the gateway. Each node in Fig. 2.6 can be modeled as a wireless router with network-layer queue(s) and MAC-layer queue(s) as shown in Fig. 2.7.

2.4.1 Isolate the Originating Traffic

Using the default queuing scheme in most routers, each node in Fig. 2.6 can be modeled as shown in Fig. 2.7-(a) (i.e., only one network-layer queue accommodates both the originating and relayed traffic flows). Assuming that Fig. 2.7-(a) shows the network and MAC queues of node 1 in Fig. 2.6, f_1 is the originating traffic flow and f_2, f_3 and f_4 are the relayed ones. The basic MAC layer

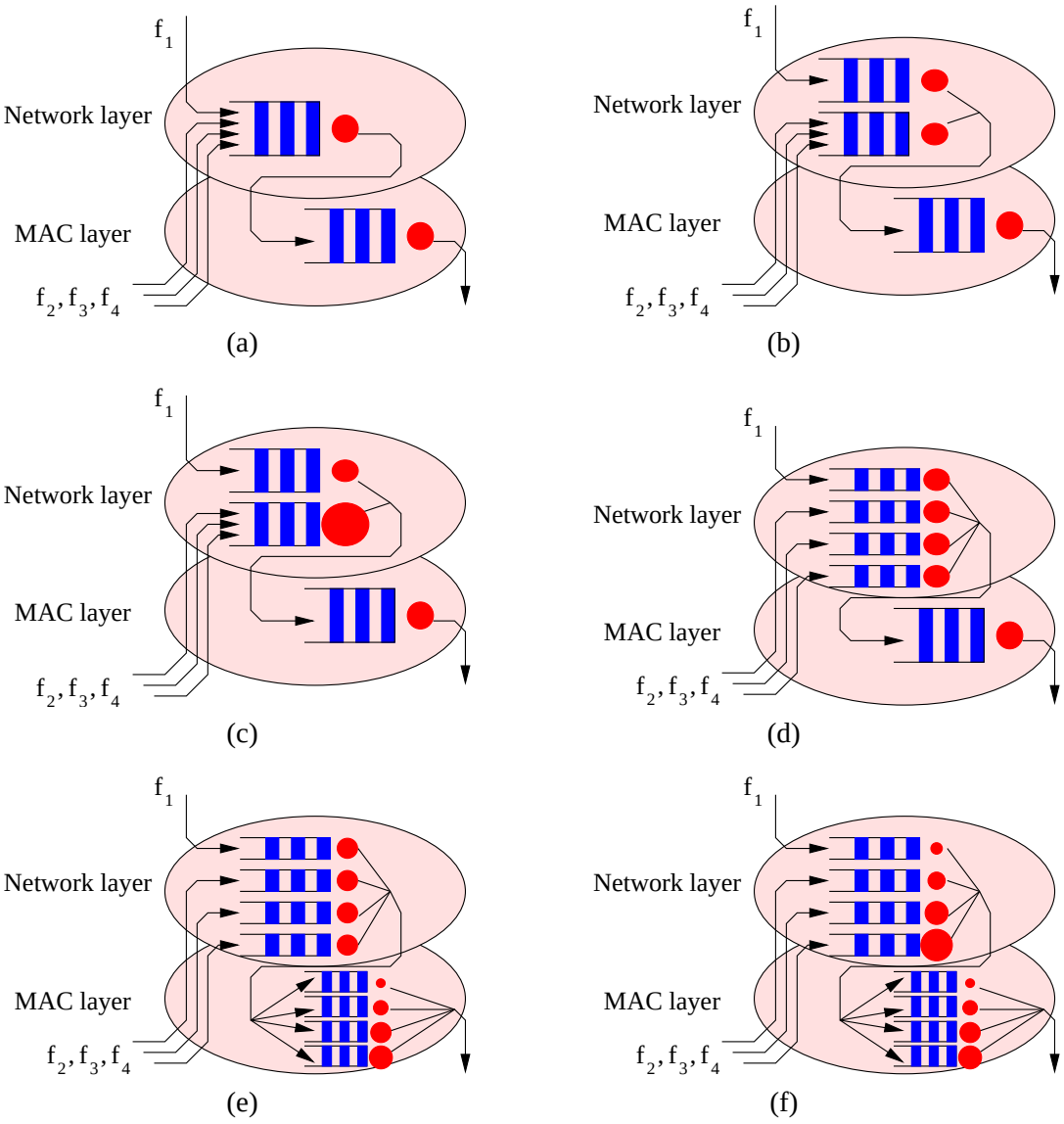


Figure 2.7: Candidate queuing schemes for multihop wireless networks. (a) Single network-layer queue. (b) Two fair queues at the network layer. (c) Two weighted queues at the network layer. (d) Per-flow (per-user) fair queues at the network layer. (e) MAC-layer QoS is added to the scheme (d). (f) Weighted per-flow queues at the network layer with MAC-layer QoS support.

commonly shown in Fig. 2.7-(a), (b), (c), and (d) models a common IEEE 802.11-like MAC layer. In such a model, no QoS is supported, but a good MAC-layer fairness is provided. With the basic queuing scheme in Fig. 2.7-(a), it is clear that the traffic flow f_1 will receive more bandwidth and eventually starve others due to the problem described in the previous section. Since this problem stems from the fact that both the relayed and the originating traffic share a common queue, the first solution that comes to ones mind is to use different queues for the relayed and for the originating traffic and to serve them in a round-robin fashion. This scheme will isolate the originating traffic which dominates the relayed traffic and protect the relayed traffic from being starved by the originating one.

In the network shown in Fig. 2.6, the maximum bandwidth share that each node can receive at the network layer (i.e., per-flow or per-user bandwidth) is G. Note that the term "flow" in this study represents the distinguishable traffic flow entity at the network layer between end users and the gateway and not the flow in the transport layer or above. Likewise, the term "per-flow" is used as opposed to the term "per-node" which indicates the traffic at the MAC layer.

Since some nodes have to carry relayed traffic as well as their own traffic, the bandwidth that nodes 1, 2, 3 and 4 need to use at the MAC layer are $4G$, $2G$, G and G, respectively. The network-layer throughput ratio of 1:1:1:1 and corresponding MAC-layer throughput ratio of 4:2:1:1 can be maintained when the offered load at each node is less than the capacity of the network; however, as the offered load at each node is increased to exceed the network capacity, and eventually saturates the network (i.e., all the nodes equally share the MAC-layer bandwidth due to the MAC layer fairness), the resulting per-flow throughputs for nodes 1-4 converge to $\frac{B}{2N}$, $\frac{B}{8N}$, $\frac{B}{4N}$ and $\frac{B}{8N}$ respectively, where B is the theoretical maximum throughput (TMT) of the network [15], and $N = 4$ is the number of nodes.

Therefore, when the network saturates, the per-flow throughput ratio of nodes 1, 2, 3, and 4 is 4:1:2:1 and the per-node MAC throughput ratio is 1:1:1:1. Isolating the originating traffic by putting two fair queues at the network layer still shows significant unfairness of the per-flow throughputs, although the scheme is simple to implement and prevents the severe starvation of relayed traffic. Fairness is guaranteed with this scheme only when the length of a chain does not exceed two hops.

2.4.2 Different Weight on the Relayed Traffic

To alleviate the unfairness shown in the previous section, one can assign different weights to each queue so that the queue for the relayed traffic will receive more bandwidth when needed. This

scheme is modeled as shown in Fig. 2.7-(c), where the larger disk indicates that a greater weight is given to the forwarding queue.

The weight of the forwarding queue can be fixed in all the nodes of the network, or different weights can be used, depending on the amount of relayed traffic at each node. The latter assumes that the amount of relayed traffic can somehow be determined in a distributed manner. For the network shown in Fig. 2.6, the desirable weight ratios of the originating versus relayed traffic are 1:3 and 1:1 for nodes 1 and 2, respectively. As in the case of the previous section, the network-layer throughput ratio of 1:1:1:1 and corresponding MAC-layer throughput ratio of 4:2:1:1 can be maintained when the offered load at each node is less than G. When the network saturates, the per-flow throughputs converge to $\frac{B}{4N}$, $\frac{3B}{16N}$, $\frac{3B}{8N}$ and $\frac{3B}{16N}$ respectively.

The resulting per-flow throughput ratio of nodes 1-4 is 4:3:6:3 and the MAC layer throughput ratio is 1:1:1:1, respectively. Giving more weights to the relayed traffic further alleviates the unfairness problem, and yet it does not achieve ideal fairness. The existing unfairness is the result of the asymmetric topology of the network, which is often the case in a real network. At node 1, incoming packets from nodes 2 and 3 are equally treated, although node 2 has to share its bandwidth with node 4.

2.4.3 Per-flow Queuing

A more general approach is to use per-flow queuing. As shown in Fig. 2.7-(d), packets of different flows are enqueued separately (based on their network-layer source address). We consider the unidirectional flows towards the gateway for now. When the network saturates, the throughput F_i of the i^{th} flow is:

$$F_i = \frac{B}{N} \times \frac{1}{4}. \tag{2.1}$$

There may exist networks lacking the resources to do per-flow queuing (e.g., in a large sensor network). Nevertheless, for many applications (especially in the access network where many multihop applications are encountered), per-flow queuing is a feasible strategy.

2.5 MAC Layer QoS and Bandwidth Efficiency

The per-flow fairness could be achieved by per-flow queuing at the network layer as shown in the previous section. But, when the offered load is high enough to saturate the network, a significant

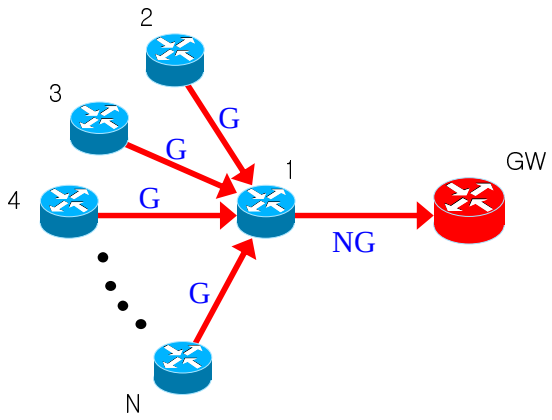


Figure 2.8: A worst-case scenario for the bandwidth waste due to MAC fairness (i.e., lack of MAC layer QoS).

amount of bandwidth is wasted due to over-injected packets. The over-injection problem cannot be solved with a pure IEEE 802.11-like MAC layer exactly due to its fairness.

If the bandwidth loss due to over-injection is considerable, the fairness guarantee at the network layer become irrelevant. Figure 2.8 shows a scenario where user throughput degrades as $O(N^2)$, where N is the number of user nodes. When the network saturates, all the nodes will have an equal share of bandwidth, $\frac{B}{N}$, at the MAC layer. At the network layer in node 1, this bandwidth is again fairly shared by N unique traffic flows. Thus, the worst-case per-flow throughput, F_{Worst} is

$$F_{Worst} = \frac{B}{N} \times \frac{1}{N}. \tag{2.2}$$

On the other hand, the best achievable per-flow throughput for the topology in Fig. 2.8, F_{Best} is

$$F_{Best} = \frac{B}{2N-1}. \tag{2.3}$$

In order to avoid such a drastic performance degradation, a MAC layer QoS scheme is required. A MAC priority scheme similar to the one proposed in IEEE 802.11e [16] can be used to allocate a different bandwidth to each node. The MAC layer model in Fig. 2.7-(e) shows the proposed scheme with MAC QoS enabled.

With QoS-enabled MAC layer, the throughput of i^{th} flow is calculated as

$$F_i = \frac{B}{2} \times \frac{1}{4}. \tag{2.4}$$

The factor N of the denominator in (2.1) is now replaced by a constant '2' because the MAC-layer bandwidth of $\frac{B}{2}$ is allocated to node 1 using the MAC-layer QoS. Compared to (2.1), the throughput is increased by 100

However, QoS schemes like IEEE 802.11e are designed for a single-hop network and may not work well in a multihop situation where bandwidth ratio between hidden nodes may not be maintained properly. A cross-layer scheme is also conceivable where the packet transmission is controlled at the MAC layer based on the information collected at the network layer. There are a number of different ways to do MAC QoS, but the details are beyond the scope of this research.

2.6 Differentiated Service and Asymmetric Traffic Flows

In this section, we examine how differentiated service can be provided by giving different weights to the network-layer queues. We also consider bidirectional traffic flows and explain how asymmetric bandwidth can be assigned.

2.6.1 Differentiated Service by Weighted Per-flow Queuing

Fairness is just a particular quality of service requirement. Fairness tries to ensure that all participants in the network receive the same share of the network bandwidth. Conceivably, for some networks some users should receive more bandwidth than others. For example, in a commercial wireless mesh network, some of the clients can have a "business class" subscription, and when the network is loaded, receive twice as much bandwidth as regular subscribers. Similarly, in a sensor network, some sensors can be more important than others (perhaps being closer to the observed phenomenon) and thus, should receive a larger share of the available bandwidth. The fairness schemes examined in the previous sections can be extended to provide differentiated service in multihop wireless networks.

The model shown in Fig. 2.7-(f) depicts how differentiated service can be provided. The larger disks indicate the greater amount of per-flow bandwidth allocation.

In the network shown in Fig. 2.6, when the per-flow bandwidth ratio of nodes 1, 2, 3 and 4 is desired to be 1:2:3:4, the per-flow throughputs at saturation are $\frac{1}{23}B$, $\frac{2}{23}B$, $\frac{3}{23}B$ and $\frac{4}{23}B$ respectively.

2.6.2 Bidirectional traffic flows and asymmetric bandwidth

Although we used unidirectional traffic flows for the clarity of our analysis, traffic flows are bidirectional in a typical network. Packets originated from and destined for the same user node should be placed in separate queues. Giving different weights will enable asymmetric bandwidth allocation. Typical Internet users will benefit from larger centrifugal bandwidth (i.e., traffic flow away from the gateway), and server sites or sensor nodes will prefer larger centripetal bandwidth.

2.7 Simulation

The analytical results presented above are verified through simulations. In this section, we describe the architecture of the simulator and present the simulation results for different queuing schemes.

2.7.1 Simulator Design

We designed our simulator architecture in a similar way to that which is presented in [17]. Their simulation design is simple but has almost all the important components required to perform the simulation experiments for fairness problem. Each node has a traffic generator, a sink, the forwarding module and multiple queues. The difference is that our model has queues at the MAC layer as well as the network layer while they have only network layer queue. The arrival rate at our MAC layer queue is associated with the contention among nodes while they assume con tension-free connection-oriented link model.

The queue size and the queuing scheme is configured to model the different schemes under study. We assume IEEE 802.11 standard as the basic MAC layer, and the TMT (i.e., B) is calculated based on the given MAC and physical (PHY) parameters. Specifically, we assume IEEE 802.11b, basic data rate of 11 Mbps and RTS/CTS scheme, and as a result, the TMT, B is 4.5 Mbps. To the best of our knowledge, there is no perfect MAC QoS scheme that works in a multihop wireless environment. Thus, we used a probabilistic approximation that models the ideal behavior of the MAC QoS. The network topology and the routing are manually configured to be the same as shown in Fig. 2.6. Exponential distribution is used for packet inter-arrival time, and the packet size is fixed to 1500 bytes.

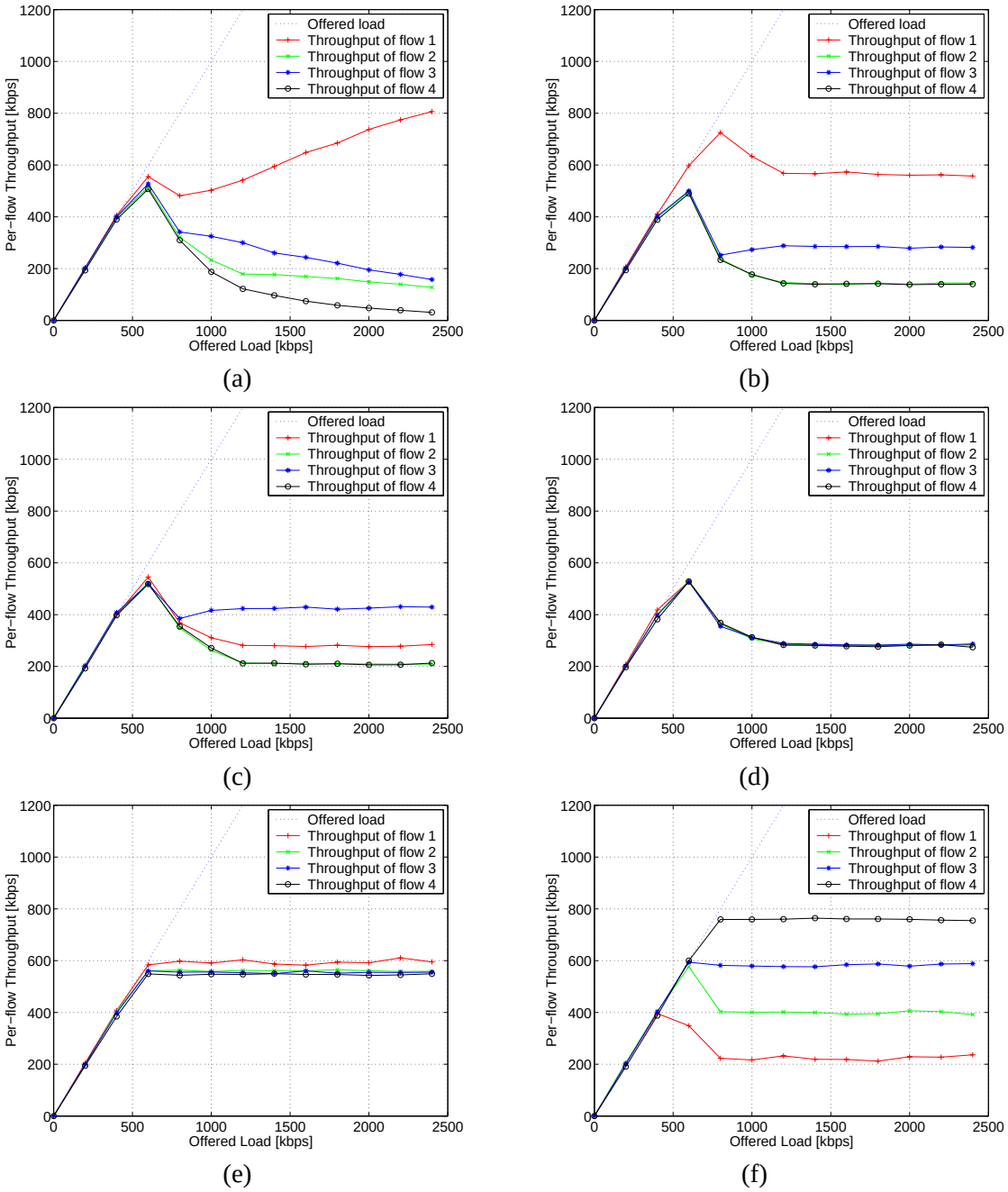


Figure 2.9: Simulation results for different queuing schemes. (a) Single network-layer queue. (b) Two fair queues at the network layer. (c) Two weighted queues at the network layer. (d) Per-flow (per-user) fair queues at the network layer. (e) MAC-layer QoS is added to the scheme (d). (f) Weighted per-flow queues at the network layer with MAC-layer QoS support.

2.7.2 Simulation Results

The simulation results for the queuing schemes in Fig. 2.7 are presented in Fig. 2.9. For all cases, the simulation result matches closely the analytically estimated values.

2.8 Summary

In this research we exposed a significant fairness problem existent practically in all wireless multihop networks. We proposed several network layer solutions to the fairness problem. The improvement in fairness is directly related to the resources investment. We showed that a network layer solution can restore fairness at the expense of bandwidth efficiency. This inefficiency can be dramatic in some topologies. It is shown that a MAC layer providing priorities is able to restore network efficiency while maintaining network layer fairness. Finally, we generalized the fairness concept to enable differentiated services in multihop networks. The results are validated using simulations.

2.9 Acknowledgement

The research presented in this chapter was supported by the Center for Advanced Computing and Communication.

Chapter 3

MRP: Wireless Mesh Networks Routing Protocol

In this chapter, we propose a new routing protocol for wireless mesh networks. Some characteristics of a wireless mesh network distinguish itself from generic ad hoc networks. In mesh networks, one or more special nodes called gateways exist to provide user nodes with connection to the infrastructure (e.g., the Internet). In such a network, the majority of traffic flows (i.e., source-destination pairs at the network layer) are assumed to be between end users and the gateway (as opposed to the assumption of any-to-any traffic pattern in typical ad hoc networks). Wireless mesh networks are expected to have a significant number of fixed user nodes to provide stable coverage and connectivity. The new protocol takes into account these particularities of wireless mesh networks and as a result, it successfully forms a tree-structured route paths without relying on flooding. The protocol operates in three different modes - proactive, reactive and hybrid (i.e., the combination of the two). In clear contrast to typical ad hoc routing protocols, none of three modes relies on flooding, and thus, the protocol shows better overhead efficiency and scalability than using generic ad hoc routing protocols for wireless mesh networks. The performance of the proposed routing protocol is analyzed through intensive simulation experiments.

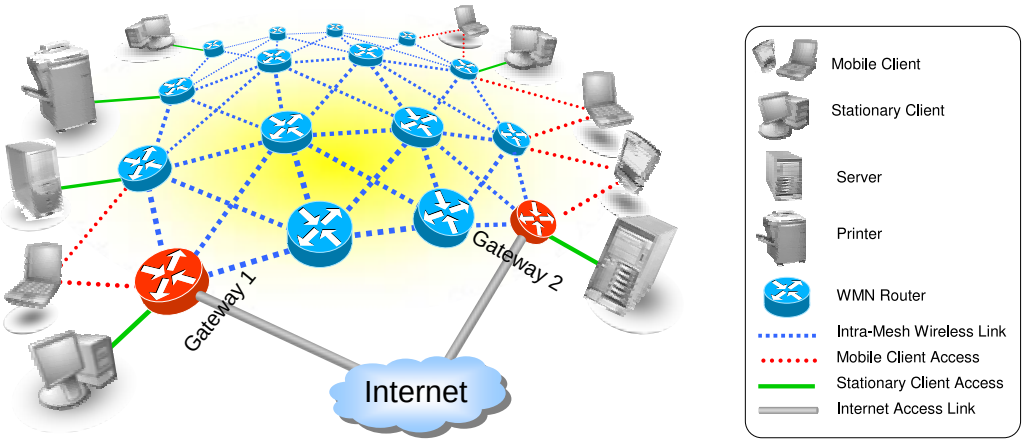


Figure 3.1: A wireless mesh network connecting several stationary and mobile clients to the Internet.

3.1 Introduction to Wireless Mesh Networks

Wireless Mesh Networks (WMNs) [18–20] are a relatively new wireless multihop technology that has much in common with the mobile ad hoc networks (MANETs). In its general form (Fig. 3.1), a WMN is a set of wireless nodes that can communicate with each other, forwarding each other's packets. Like in MANETs, each node is both a host and a wireless router. Clients can connect to the WMN routers using common networking interfaces (e.g., Ethernet, 802.11, Bluetooth) or, in some cases, a PCI or a PCMCIA bus (i.e., the WMN router is embedded in a network card). In most proposed applications, the WMN provides connectivity to an infrastructure network, typically connected to the Internet. We will call the nodes providing Internet connectivity *gateways*.

There are four types of links presented in Fig. 3.1. Except for the intra-mesh links (that have to be wireless), all other links can be either wireless or wired. The same or different technologies can be used for the four link types. The link choice usually represents a tradeoff between the cost/complexity and the performance of the WMN. Standardization efforts are initiated by IEEE (802.11s, 802.15.5, 802.16a).

Several companies [21–35] developed WMN products for different applications. One of the most popular WMN applications is providing broadband Internet access [26, 27, 29, 31, 32, 34, 35]. In this scenario, WMN routers are installed on the roofs of the clients and/or light poles in the coverage area of the WMN. Mobile clients may roam while being handed over from one wireless router to another

(some products may be able to handle highway speeds [27]). The main advantage of WMNs in comparison to traditional broadband Internet access technologies (cable-modem and xDSL) is the dramatically reduced initial investment and deployment time. The main advantage in comparison to fixed wireless metropolitan area networks (WMANs) (e.g., IEEE 802.16) is the market coverage [36] (especially in areas with significant obstructions - trees, high-rise buildings) and reliability (multiple available routes can avoid failed nodes and poor links). Furthermore, some implementations allow for mobile user access, while the current IEEE WMAN standard only allows stationary users (although work for mobility extensions is underway).

A second major class of WMN products targets the booming wireless local area network (WLAN) market (mainly IEEE 802.11) [21, 23–25, 29, 33]. An important drawback of WLAN technology in multi-access point (AP) deployments is the requirement to separately provide wired network connectivity to each AP, offsetting some of the advantages of a wireless network. WMNs in this category solve the problem by placing the APs in range of each other and allowing them to forward each other's packets to and from a common gateway (another AP connected to the distribution system). The main drawback of these deployments is the reduced bandwidth available to the users (this can become a major problem in scenarios with many active users [37]). Some companies are counteracting the reduction in bandwidth by using multiple radios on different channels and directional antennas [21, 25].

Other companies found niche applications for WMNs. Firetide [22] uses WMNs to provide network connectivity. Basically, the WMN cloud behaves like one big Ethernet switch that can be used to connect all devices plugged in any of the wireless routers (including, for example, 802.11 APs).

Routing is a fundamental characteristic of WMNs. The routing protocol's strengths and weaknesses are reflected directly in the WMN's characteristics. Several advantages of WMNs over competing technologies are directly enabled by the routing protocol:

- **Reliability:** The routing protocol should be able to reroute fast around failed nodes and broken links; upon the failure of a gateway, it should be able to redistribute the orphaned clients among neighboring gateways. For this property, fast reconfiguration and support of multiple gateways is essential.

- **Mobile user connectivity:** To ensure seamless mobile user connectivity, the routing protocol should enable fast hand-offs.

- **Scalability/Efficiency:** If the routing protocol has a high overhead, it will be impossible to scale the WMN to a large number of nodes.
- **QoS:** In addition to support from the medium access control (MAC) layer and the forwarding engine, selecting the "best" routes for different traffic classes is an essential ingredient for QoS support.

Taxonomically, WMNs are a particular type of mobile ad hoc network (MANET) [18, 38]. WMNs share the same multihop characteristics and mobility-related issues as MANETs. However, there are also significant differences between WMNs and general MANETs:

- **Gateways:** Most WMNs are designed to provide connectivity to a distribution system (usually connected to the Internet). Therefore, they have specialized nodes (the gateways) that provide connectivity to the distribution system.
- **Traffic pattern:** In WMNs, most of the traffic is expected to flow between the clients and the Internet (via the gateways). In general MANETs, the common assumption is that any node is equally likely to be the source or the destination of a traffic flow.
- **Mobility:** In most WMNs, nodes belong to two distinct categories: either stationary (e.g., on lamp poles, rooftops, etc.) or mobile, capable of roaming in the coverage area provided by the stationary nodes. In MANETs, it is often assumed that all nodes have homogeneous mobility characteristics.

In this research, we propose a new mesh routing protocol (MRP), specifically geared toward WMNs. There have been many routing algorithms proposed for MANETs [38–52]. However, the characteristics and requirements of WMNs are considerably different than those of general MANETs such that a new routing protocol can significantly outperform the general MANET routing protocols. The situation is similar to the case of wireless sensor networks, where differences from general MANETs prompted the development of specialized MAC and routing protocols [53].

MANET protocols rely on variations of a flooding mechanism for route discovery and recovery. Some of them attempt to reduce the flooding overhead by utilizing location information [49–52], limiting the flooding diameter [44, 54–56], or electing dedicated relay nodes [57]. However, none of them completely eliminates flooding. In contrast, our route discovery and recovery schemes do not

require flooding. The main difference between the proposed and existing routing protocols is the absence of flooding as well as a novel link failure detection scheme that takes into account the inherently unreliable nature of wireless links.

The rest of this chapter is organized as follows. Section 3.2 provides a brief overview of the related work. In Section 3.3, two versions of the proposed protocol are described, compared, contrasted, and merged into a third hybrid version. The performance of the three versions of the proposed protocol is evaluated via simulations in Section 3.4. Section 3.5 summarizes the results of this chapter.

3.2 Related Work

There are hundreds of proposed routing protocols. Many of them have been standardized by IETF and have been in use for many years. Some of those protocols have proven themselves in the Internet and are expected to continue to perform well for many years to come [58–62].

In the ad-hoc networking arena, several classes of routing protocols have been proposed and carefully analyzed [39, 40, 54]. The first class of proposed protocols was derived from existing "table-driven" protocols in the Internet (e.g., [42]) optimizing many of the aspects that reduced the efficiency of the existing Internet routing protocols in MANET environments.

In contrast to table-driven routing protocols, on-demand routing protocols (e.g., AODV [45, 63] and DSR [46]) were designed specifically for ad hoc networks with frequent disconnections (due to topology changes). They often outperform the table-driven routing protocols in scenarios with large networks with relatively few active connections.

In a class of their own, the geographical routing algorithms (e.g., LAR [49], DREAM [50] and ZRP [64]) take advantage of node location information (that can be relatively inexpensive to distribute [51]) to reduce routing overhead and improve the performance of the protocol. Geocasting [65, 66] takes a rather unique view to addressing, as it considers a valid destination any node within a given geographical region. Several specialized applications can greatly benefit from geocasting.

It is well known [54] that current routing protocols scale poorly with the number of nodes, number of flows and increase in mobility. Several protocols were designed for scalability [50, 55, 57, 67–69] and show significantly better performance in large ad hoc networks.

The WMN companies are using a variety of routing protocols to satisfy their needs. Some are

proprietary and held secret (e.g., [27]), while others use well-known ad hoc routing protocols (e.g., Firetide [22] uses Topology Broadcast based on Reverse-Path Forwarding (TBRPF) [67]). Other companies rely on the IEEE 802.1 spanning tree protocol for routing at layer 2 (e.g., MeshDynamics [25]).

It has long been recognized that many ad hoc networks would benefit from a connection to a fixed infrastructure, and several solutions have been developed to support this type of connectivity. In particular, the SURAN [70] and WINGS [71] projects solved many of the problems associated with these networks (MAC layer, channel access, integration with existing Internet protocols, mobility management, transport layer efficiency, etc.).

In [72], the authors propose extensions to existing routing protocols (AODV [45], DSR [46] and SOAR [73]) such that they optimize access to a set of nodes called netmarks (similar to what we call gateways). They evaluate the performance of the extended SOAR routing protocol (a link state routing protocol) and show via simulations that it outperforms both DSR and AODV. We do not propose to extend a link state protocol, but rather a new design specifically optimized for WMNs. Furthermore, our proposed protocol takes advantage of the fixed nodes in WMNs (by selecting stable routes).

In [74], the authors address routing issues in a hierarchical, two-level MANET. They propose a hybrid scheme using existing protocols in both hierarchies (DSDV [42] at the lower level, i.e., in the clusters connected to the gateways, and AODV in the backbone). They also propose an extension of AODV: Hybrid-AODV (H-AODV). Both approaches are shown to outperform classical AODV in terms of packet delivery ratio and delay for large networks. Our work is different in that it does not consider mobile backbones, and it does not use or extend existing protocols.

The LUNAR [75] system provides a simple routing protocol for small MANETs (limited to three hops). The protocol is predictable as well as easy to implement and analyze. It features autoconfiguration and gateway routing, thus, making it a complete set of protocols. Its goals and functionality are, however, very different from those of the protocol we propose in the following Section.

The K-hop routing protocol (KRP) [56,76] is designed to provide service in an ad hoc network with nodes at most K hops away from a gateway. An extension of AODV (the flooding mechanism in AODV is limited to K-hops) is used to discover routes to the gateways. The gateways themselves publish and update a bulletin board of reachable nodes ensuring network-wide dissemination of information. The approach allows for multiple routes to multiple gateways. Our approach is significantly different from KRP in that we use neither flooding not a bulletin for route discovery and maintenance.

Landmark Routing (LANMAR) [68,69], is an extension of the Fisheye State Routing (FSR) [55] that takes advantage of logical groups that tend to move together (e.g., soldiers in a platoon). Our approach is significantly different, as it is not a link state approach, and it does not take advantage of group mobility (unrealistic for general WMNs).

In [77], a gateway discovery scheme was proposed as a part of an architecture connecting generic MANETs and IPv6 networks. The scheme in [77] relies on flooding for route discovery, while the routing protocol proposed in this dissertation does not. Furthermore, the return paths (i.e., from the gateway to nodes in the WMN) are not established, a MANET routing protocol being assumed for this purpose.

Several articles considered routing-related issues for WMNs; in [78, 79], routing is considered in conjunction with channel assignment for multichannel WMNs based on commodity IEEE 802.11 compatible hardware. Several articles [80, 81] show that non-conventional metrics may result in an increased capacity of WMNs.

In MANETs, existing routing protocols employ flooding to discover and maintain routes between arbitrary pairs of nodes. In WMNs, most of the traffic is assumed to flow to and from the gateways connected to the Internet. With this assumption, we designed a protocol that does not use flooding either for route discovery or for recovery. Furthermore, we introduce a novel link failure detection procedure that is shown to effectively resolve the problem of misidentifying temporary link errors as permanent link breaks.

3.3 Protocol Description

In the design of the protocol, we decided to optimize the common case. In WMNs, the most common traffic flows are to and from the Internet (*downloads from* the Internet are by far the most common case, but TCP acknowledgments form streams in the opposite direction). Thus, we decided that any node in a WMN will only know how to reach one gateway and is, in general, reachable only from a gateway. Any small amount of client-to-client traffic can be routed through the common parent of the clients (potentially the gateway). In essence, the routes to and from the gateway form a tree routed at the gateway.

In this version of the protocol, every client chooses a single gateway to connect to the Internet.

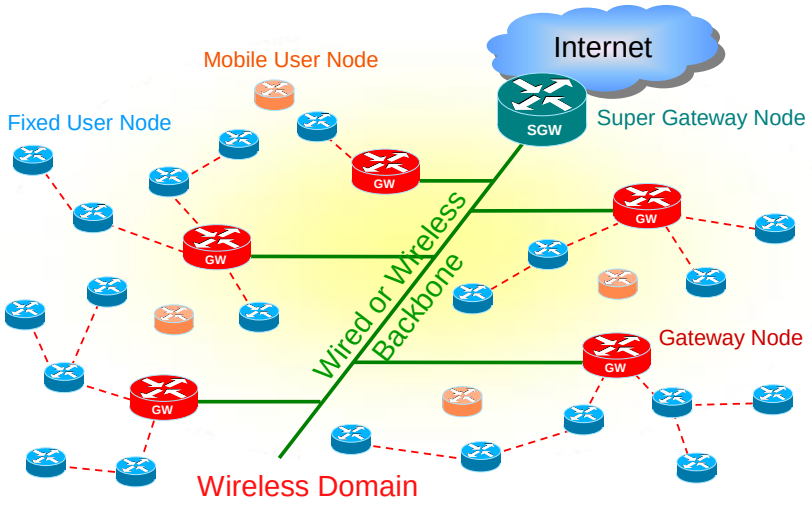


Figure 3.2: General WMN network topology.

If the node moves or if this gateway fails, the node may choose a different gateway. We, thus, assumed a network topology similar to the one shown in Fig. 3.2. The gateways are connected using wired or wireless links to a single super-gateway that is further connected to the Internet. All traffic is funneled through the gateways to the super-gateway and, eventually, to the Internet. Mobile users are free to roam as long as they are in the coverage area of the network. Small deployments may use a single gateway and forego the use of a super-gateway.

One of the design requirements we observed was the possibility to implement the protocol in the user space. In other words, the protocol does not have access to the forwarding engine other than through setting the routing table. We chose to transport the MRP messages using UDP packets for reasons of performance (although the choice also requires a loss-tolerant protocol design). All three versions of the protocol presented here interface with the kernel only through the routing table changing calls and by intercepting ICMP packet delivery failure messages.

3.3.1 MRP On-Demand

When designing the protocol, we faced the well-known proactive vs. on-demand question commonly encountered in MANETs [39]. The first version of the protocol is purely on-demand: when a node is joining the network, it will ask the closest gateway or neighboring user nodes for a route (we will shortly elaborate on the criteria used to choose a gateway). Thus, the first step for a node that

wishes to join the network is to broadcast *locally* a route discovery (RDIS) message. In contrast to existing MANET protocols (e.g., AODV and DSR), the RDIS packets in MRP are *not flooded* through the network, and are only received by the one-hop neighbors of the source. Initially, the joining node is in *disconnected* state.

We take advantage of the fact that all connected neighbors already know a route to the gateway and the significant routing metrics of those routes. Thus, all of the nodes receiving the RDIS message (one-hop neighbors of the joining node) will reply with a route advertisement (RADV) packet with the metrics of their current routes. The very first user node to join the network will receive this RADV packet only from the neighboring gateway(s).

The RADV packets are unicasted after a small random delay to avoid collisions. If there are no connected neighboring nodes, or all of the advertisements are lost, the joining node will periodically broadcast the RDIS message until it receives an advertisement. The new node, joining the network will store all RADV packets (wait for a time chosen slightly longer than the longest random delay used to generate RADVs). Once the joining node receives all the RADVs, it will select one or more upstream routes as a function of the node's requirements and the offered routes.

The current design of the protocol is open to a variety of metrics that can be used for differentiating routes:

- **Hop-count**: the basic and common metric in routing protocols.
- **Route stability:** this metric allows a joining node (given the choice) to select routes that include stationary nodes (those routes are presumably more stable than those including mobile nodes).
- **Minimum delay:** important for delay sensitive applications (e.g., VoIP, Telnet, etc).
- **Maximum bandwidth:** important for bandwidth intensive applications (e.g., ftp, peer-to-peer).
- **Minimum packet loss:** important for loss sensitive applications.

In contrast to the protocols recently suggested by others, we do not propose any new metrics or methods for computing the metrics above. Computing many of them is an area of active research [80, 81]; we simply include the possibility that they can be used in making routing decisions.

In the simulations, routes with high stability (i.e., employing stationary nodes) are preferred

to routes with minimum hop-counts. Preferring stability to hop-counts is expected to reduce routing overhead, increase the data throughput, and reduce the energy consumption of the mobile nodes.

Routing overhead and packet loss is reduced as the number of route breakage is minimized. In many applications of wireless mesh networks, the mobile nodes have limited energy resources (unless the nodes are mounted on vehicles); thus, power consumption becomes an important issue. If routes through stationary (and powered) nodes are preferred, packet forwarding of the mobile nodes is minimized. Other energy efficiency aspects of MRP are discussed in Section 3.3.3.

Once the joining node selects a route (based on the RADVs it received), the node practically has a route *toward* the Internet (as all of the neighbor nodes have routes toward the Internet). The node can start to send data immediately toward the Internet. We call this the *half-connected* state (as the node has a route to the Internet, but none of the nodes in the Internet can reach it). In the second phase, the joining node registers with the gateway. The registration has the main function of providing a reverse path from the Internet to the joining node. We could have simply used the reverse path of the packets going toward the Internet to provide the return path, but that would imply that the routing protocol can monitor the forwarded data packets (a requirement we eliminated for portability, as explained above).

The registration process proceeds in two steps. In the first step, a registration request (RREG) is unicasted to the gateway (it goes through the MRP layer at every intermediate hop). As the RREG travels toward the gateway, intermediate nodes set up an entry in their routing tables enabling the return path (*from* the Internet to the joining node).

The existence of the return path assumes bidirectional links, but as pointed out in the literature [82], unidirectional links automatically exclude many of the common MAC protocols (e.g., MACA, 802.11, 802.16, and certainly all of those that require acknowledgments). Upon receipt of the RREG packet, the gateway sends a registration acknowledgment (RACK) directly to the joining node, and it also forwards the RREG message to the super-gateway (such that the super-gateway will know to which gateway to forward the packets for the joining node). If any of the RREG or RACK messages are lost, the joining node reverts to disconnected state and re-initiates the route acquisition process (as very likely the route it initially chose just became disconnected).

Upon receipt of a registration acknowledgment, the joining node enters the *fully-connected* state in which it can send and receive data to and from the Internet.

Figure 3.3 depicts the sequence of messages discussed above, from route discovery to regis-

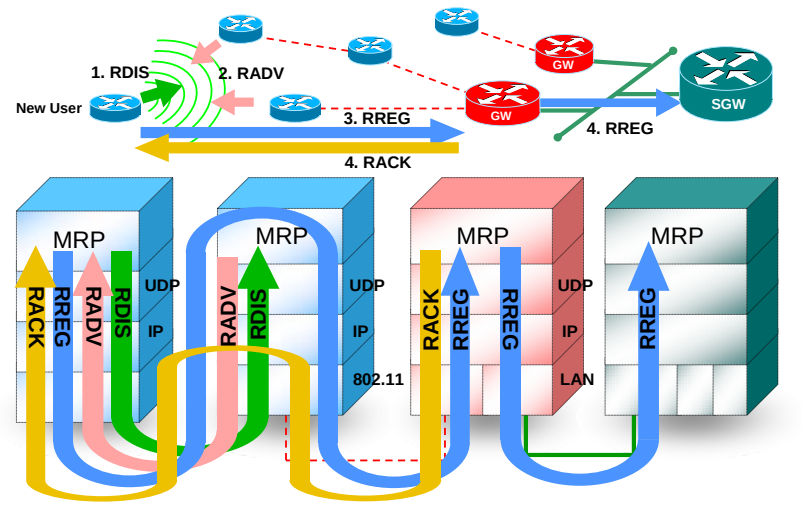


Figure 3.3: MRP route establishment message sequence.

tration acknowledgment. None of the four messages used to establish the routes (RDIS, RADV, RREG, and RACK) are *flooded* into the network. This is in clear contrast to other MANET routing protocols where the overhead/node ratio increases with the size of the network. Thus, we expect that the overhead of MRP will be significantly lower than that of existing MANET protocols, especially for large networks.

For wireless networks, the failure to forward a packet to the next hop may represent a route failure, but, more often than not, it might be simply the effect of interference or a temporary fading effect. Therefore, over-reacting to a packet loss (i.e., entering the disconnected state and re-initiating the route discovery process) may be detrimental to the protocol's performance. Hence, upon suspecting a link loss, instead of entering the disconnected state, the node enters a temporary state named *verify-link* state.

While in verify-link state, the node probes the availability of the suspected link by using special route-check packets (RCHKs) that are unicasted to the next hop. If a reply is received (we reused RADV for the reply), the node again becomes fully-connected; otherwise, it enters the disconnected state. If a node loses connection to the gateway, all of its children will lose connection to the gateway. Even if it discovers a new route to the gateway, the children will not be able to use this new route to *receive data from* the gateway (the reverse routes will be broken). Hence, when a node loses connection to the gateway, it will send a route error (RERR) message to all of its children. Each of those children

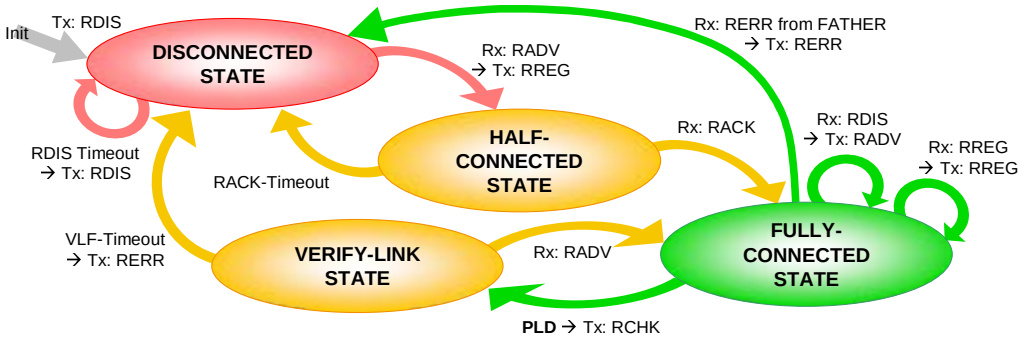


Figure 3.4: Finite state machine for MRP-OD.

will enter the disconnected state and re-initiate the route discovery process. A detailed analysis on the effect of the verify-link state on MRP's performance is presented later in this section.

Figure 3.4 depicts the finite state machine for the MRP protocol we just described. Since, in some ways, it resembles the on-demand protocols of MANET (route discovery initiated by the joining node and route errors initiated by packet loss), we will call this version *MRP on-demand* (MRP-OD).

Due to the tree structure of the routing graph (both to and from the gateways), and the use of hop-counts, the protocol is guaranteed to be loop free (it will be loop free as long as the routing metric is strictly positive).

3.3.2 MRP Beacon Mode

MRP-OD is fully functional; but, when a user node does not send any packets, it cannot detect that its route is no longer valid. Any packets from the gateway will be unable to reach the disconnected node.

A second version of MRP uses beacons to advertise routes: each node in fully-connected state (including the gateway) periodically sends beacons advertising the available routes. Each beacon acts as a gratuitous route advertisement (we used the same RADV packets as for MRP-OD). We will call this version *MRP beacon mode* (MRP-B).

In MRP-B, a node that wishes to join the network does not have to send an RDIS. Instead, it simply listens to the neighboring nodes slightly more than the beacon period and collects beacons (route advertisements). The route selection and registration proceeds identical to MRP-OD (the same states are used).

For detecting route disconnections, in addition to relying on packet forwarding failure, MRP-B utilizes the beacon messages originally intended for route setup. In MRP-B, each node monitors beacons sent from their parents. If a predefined number of beacons are missing, the node will enter the disconnected state and re-initiate the route discovery process. Also, as in the case of MRP-OD, if there is a packet forwarding failure, the node will enter the verify-link state and either become fully connected (if the RCHK is acknowledged) or will disconnect (if no reply is received). If link detection based only on beacons is employed, a real link breakage will be detected only after a considerable delay (several beacon periods).

The state diagram of MRP-B is practically identical to MRP-OD except for changing the triggers in some of the transitions, adding a periodic beaconing while in fully-connected state and a direct transition (triggered by beacon loss detection) from the fully-connected state to the disconnected state.

There is a trade-off between the beacon interval and the performance of the protocol: the more often a beacon is sent, the faster a node will be able to join the network and detect a route disconnection. However, it also increases the overall overhead of the routing protocol. Depending on the required performance, a reasonable range of beacon periods can be from tens of milliseconds to a few seconds.

We expect the MRP-B to exhibit better delay performance than MRP-OD because MRP-B is able to detect a route disconnection sooner than MRP-OD. This is true especially when traffic load is low and as a result, detecting forward failure does not promptly reflect a route change. On the other hand, MRP-B will have a higher overhead (due to the beacon messages).

3.3.3 Hybrid MRP

The two MRP versions we discussed are not mutually exclusive. They can be easily combined into a hybrid protocol (MRP-H). We simply use the same states as in MRP-OD and MRP-B and combine the state transitions from the two versions.

In MRP-H, the joining node broadcasts route discoveries (RDISs) and waits for route advertisements (RADVs) for a time equal to the minimum between the random delay of MRP-OD and the beacon period of MRP-B. The received RADVs include those generated in response to the RDISs, as well as the beacons. The joining node then selects the route and registers with the gateway. A route error can be discovered by either missing beacons or packet forwarding errors. The state diagram of MRP-H

is similar to the one corresponding to MRP-OD (Fig. 3.4), except that there are multiple triggers for the state transitions.

We expect MRP-H to perform better than either MRP-B or MRP-OD as MRP-H detects link failures faster than either of the two versions and is able to discover better routes faster: the pool of available routes at the moment of the route selection will be larger due to the redundancy offered by the beacons and triggered answers. (This feature only makes a difference if some of the RDIS or RADV packets are lost due to noise/interference.)

Since none of the three versions of the protocol uses flooding to establish or repair routes, all three should scale well to large networks.

With respect to energy consumption, a mobile user with limited energy resources can minimize the energy expended for routing by employing MRP-OD (where no periodic beacons are employed). The rest of the network may run the more energy-expensive MRP-H or MRP-B. Thus, for mobile nodes, MRP can minimize both the energy expended for *routing*, as well as for *forwarding* (by selecting routes that avoid the mobile nodes).

If the nodes in a large mesh network are powered up simultaneously (e.g., after a power outage), the first tier of nodes (with direct connectivity to the gateways) will discover routes first, then, the second tier (nodes two hops away from the gateway), and the process will continue from the center to the edges of the network.

The design of MRP is based on the assumption that most data flows are established between client nodes and the gateway. Routes between two arbitrary client nodes may be sub-optimal. In particular, two client nodes (unless they are parents of one another), will always communicate through their common parent (which can be several hops away from either of them, and possibly the gateway); however, client-to-client communication is assumed to be infrequent in WMNs (for most applications).

3.3.4 Verify-Link State

The verify-link state is designed to reduce the uncertainty introduced by wireless links and to avoid unnecessary disconnections and route re-establishments. We define a *false disconnection* as the case where an MRP node erroneously enters a disconnected state due to temporarily disabled links (e.g., caused by fading or interference). We define a *true disconnection* as the case where a node enters a disconnected state due to permanent link breakage (e.g., caused by mobility or node failure).

False disconnections and consequent route re-establishment may cause significant performance degradation. Frequent re-routing will also reduce the available bandwidth for end users due to increased routing overhead. Therefore, one of the design goals of MRP is to suppress false disconnections and to identify and promptly respond to true disconnections. To this end, we introduced the verify-link state whose state transition and packet exchange (i.e., RCHK and RADV) were detailed in Section 3.3.1.

One of the key parameters in the verify-link state mechanism is the verify-link-failure (VLF)-timeout value, which controls how long a node should stay in verify-link state before it transitions to disconnected state (assuming that no reply was received). The choice of a VLF-timeout value is a trade-off: a large VLF-timeout will reduce false disconnections but causes delay in true disconnection situations. On the other hand, a small VLF-timeout will ensure prompt handover in a true disconnection but increases the number of false disconnections. To study the efficiency of the verify-link state and to identify a working range of VLF-timeout values, we investigated the behavior of MRP through simulation experiments over a wide range of VLF-timeout values.

Since traffic load intensity and mobility intensity are expected to affect false disconnections and true disconnections, respectively, we tested VLF-timeout values for different traffic and mobility scenarios. Three performance metrics were used to measure how correctly and efficiently routes are maintained when different VLF-timeout values are used with the forward failure detection: routing overhead, packet delivery ratio (PDR), and end-to-end delay. The metrics are defined as follows:

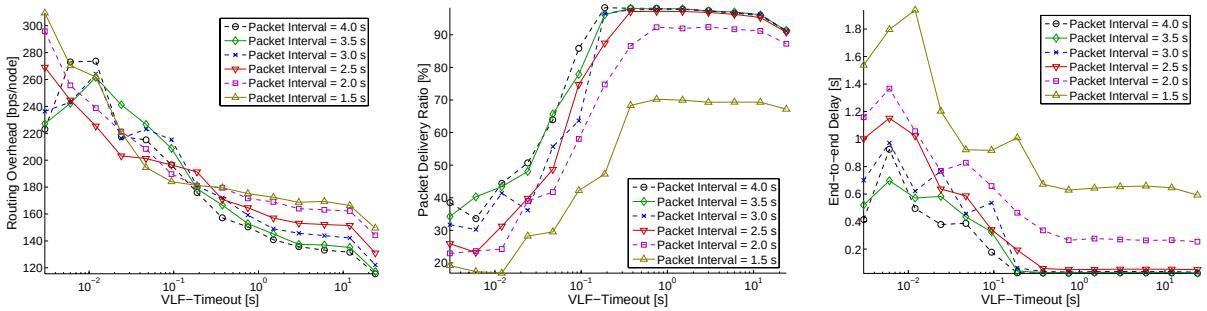
$$Routing\ Overhead \equiv \frac{\sum_{n=1}^{N}\sum_{p=1}^{P_n^{cs}} b_{n,p}}{T_{sim} \times N}\ \ (bps/node), \tag{3.1}$$

$$PDR \equiv \frac{\sum_{n=1}^{N} P_n^{dr}}{\sum_{n=1}^{N} P_n^{ds}}\ \ (\%), \tag{3.2}$$

$$Delay \equiv \frac{\sum_{n=1}^{N}\sum_{p=1}^{P_n^{dr}} D_{n,p}}{\sum_{n=1}^{N} P_n^{dr}}\ \ (ms/packet), \tag{3.3}$$

where:

N is the total number of nodes,



(a) (b) (c)

Figure 3.5: The effect of verify-link state as a function of VLF-timeout and different average packet intervals: (a) routing overhead, (b) packet delivery ratio and, (c) end-to-end delay.

P_n^{cs} is the total number of broadcast/unicast routing control packets sent (either generated or forwarded) by the n^{th} node,

$b_{n,p}$ is the number of bits in the p^{th} packet received by the n^{th} node,

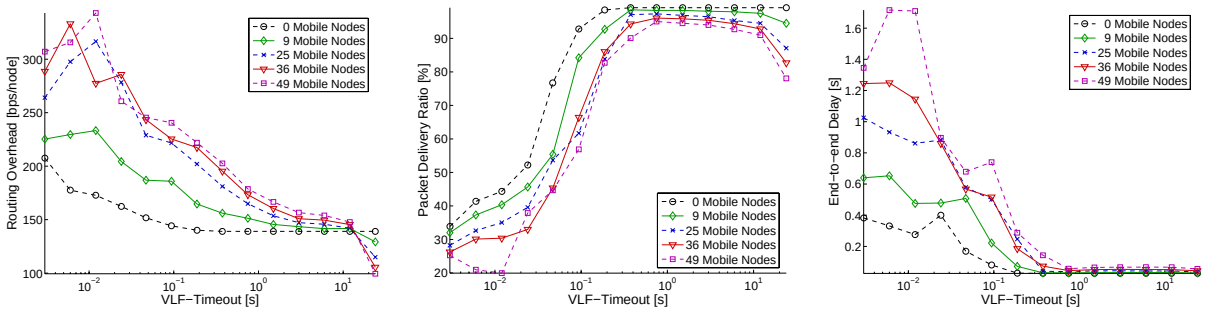
T_{sim} is the total simulation time, and

P_n^{dr} is the total number of unicast data packets received by the n^{th} node,

P_n^{ds} is the total number of unicast data packets sent by the n^{th} node, and

$D_{n,p}$ is the end-to-end delay experienced by the p^{th} packet received by the n^{th} node.

We use Poisson traffic with data packets serving as probes. Every user node has two flows: one toward the gateway and the other one from the gateway. Packet size is 1500 bytes. The traffic intensity is varied by changing the average packet interval ranging from 1.5 s to 4.0 s (in this case, the number of mobile nodes was fixed at 16). Mobility intensity is controlled by the number of mobile nodes from 0 to 49 (in this case, the packet interval was fixed to 3.0 s). Randomly-deployed mobile nodes move according to the random waypoint mobility model [83] with zero pause time, a maximum speed of 10 m/s, and a minimum speed of 1 m/s (to avoid the drawbacks highlighted in [84]). The mobile nodes move inside a rectangular area with a single gateway in the center and 144 uniformly deployed fixed nodes. Total simulation time is 400 s. The size of the rectangular area is determined by the number of fixed nodes to ensure constant node density (approximately 25 nodes per km^2).



(a) (b) (c)

Figure 3.6: The effect of verify-link state as a function of VLF-timeout and mobility intensity: (a) routing overhead, (b) packet delivery ratio and (c) end-to-end delay.

Figure 3.5 shows the performance of MRP as a function of VLF-timeout under different traffic intensity. Only the result for the MRP-H is presented to avoid cluttering. The VLF-timeout was increased from 6 ms to 12 s. The results for the first (0 s) and the last VLF-timeout value (400 s) are not shown to scale. They are included to provide insight on the MRP's behavior for extreme values. The VLF-timeout of 0 s represents the case where the verify-link state is completely disabled; in this case, one forwarding failure is interpreted as a route failure and, thus, it triggers a route rediscovery. The VLF-timeout of 400 s represents the case where packet forward failures are always ignored throughout the simulation time.

For small VLF-timeout values, the curves show low PDR and high end-to-end delay due to false disconnections. A false disconnection will trigger a route rediscovery that implies the exchange of several control packets (RDIS, RADV, RREG, and RACK). If the falsely disconnected node has children nodes, RERR packets are generated which leads to a cascade of false disconnections and route re-establishments.

As the VLF-timeout is increased, the overall performance is improved due to the reduced probability of false disconnections. However, if VLF-timeout is too large, true disconnections (from mobility) are temporarily ignored, leading to loss of packets for the mobile nodes.

Figure 3.6 shows the performance of MRP-H as a function of VLF-timeout and mobility intensity. When the number of mobile nodes is increased, the loss from ignored true disconnection increases.

Considering the results in Figures 3.5 and 3.6, the VLF-timeout should be at least 1 s (as for

lower values, the false disconnections considerably lower the PDR in Fig. 3.5-(b)). Figure 3.6-(b) shows that the PDR decreases steadily as the VLF increases over 1 s, as true disconnections are identified too late. In our simulations, we chose a VLF-timeout of 3 seconds. The performance could likely be improved if the value of the VLF-timeout is determined adaptively; but, we do not explore that option in our research.

3.4 Performance Evaluation

In this section, we present the results of the simulations which we performed using QualNet [85] to evaluate the performance of MRP. QualNet is the commercial version of GloMoSim [86] (which uses Parsec [87], a parallel discrete-event simulator). We chose QualNet because it provides accurate models (high level of detail) at each layer, including the physical and MAC layers. Results indicate that the accuracy of the models has a significant impact on the simulation results for wireless networks [88, 89].

3.4.1 Performance Metrics

We compared the performance of MRP with other routing protocols in terms of five metrics: routing overhead, packet delivery ratio (PDR), throughput, end-to-end delay, and average hop-count.

We employ Internet traffic as well as Poisson traffic in the performance evaluation; we use network throughput as a performance metric for Internet traffic and PDR as a performance metric for Poisson traffic. In the Internet traffic scenarios, the applications use mostly TCP as a transport layer, and the PDR does not capture the effect of packet loss on the TCP throughput.

The routing overhead, PDR, and the end-to-end delay are defined in Section 3.3.4. We define the network throughput and hop-count as:

$$Network\ Throughput \equiv \frac{\sum_{n=1}^{N} \sum_{p=1}^{P_n^{dr}} b_{n,p}}{T_{sim}}\ \ (bps), \tag{3.4}$$

$$Average\ Hop-Count \equiv \frac{\sum_{n=1}^{N} \sum_{p=1}^{P_n^{dr}} h_{n,p}}{\sum_{n=1}^{N} P_n^{dr}}\ \ (hops\ per\ packet), \tag{3.5}$$

where N, P_n^{dr}, $b_{n,p}$, T_{sim}, n and p are defined in Section 3.3.4, and $h_{n,p}$ is the number of hops traversed by the p^{th} packet of the n^{th} node.

We believe that the chosen metrics can be traded against each other (e.g., higher overhead for lower delays, lower throughput for lower delays, etc.), and a fair comparison has to consider all of the metrics.

3.4.2 Simulation Setup

For the physical and MAC layers, we used IEEE 802.11b in RTS/CTS mode. For the higher layers, we used the standard TCP/IP protocol stack, and we implemented MRP as described in Section 3.3. For MRP-B, we chose a beacon interval of 1 s. For MRP-OD and MRP-H, we chose the small random interval for sending route requests of 2 s. For MRP-H, we chose the beacon interval of 2 s (we shall see that the performance is comparable with MRP-B with 1 s beacon interval).

Simulations are run for two different traffic types: Poisson (Section 3.4.3) and Internet traffic (Section 3.4.4). For the Poisson traffic, user data packets serve as probes, and the performance metrics indicate how correctly and efficiently routes are maintained by different protocols. On the other hand, the Internet traffic model is used to evaluate the performance of the protocol in a more realistic environment. The Internet traffic is created by mixing several popular Internet applications based on real Internet traffic measurement. The traffic composition trend captured from the real Internet traffic [90] was accurately reproduced in our simulation using multiple application protocol models provided by QualNet [85]: HTTP, Telnet, FTP, DNS, and VBR models. Traffic composition ratio was reproduced at all detail levels: flow, packet, and byte.

Simulations were performed to evaluate the influence of the following parameters:

- traffic load,
- network size,
- degree of mobility,
- perturbation, and
- intra-mesh traffic ratio,

where the perturbation represents the degree of randomness in geographical node distribution, and the intra-mesh traffic ratio indicates the percentage of the user traffic flows whose both ends are user nodes (as opposed to the typical mesh network case where one end is a gateway.)

To avoid the large number of graphs that would result if we performed every experiment for every possible setting, we chose a *base case* and varied one parameter at a time. For each parameter variation, we present the four metrics defined in Section 3.4.1 for all of the protocols under consideration. For the base case, we used the following scenario:

- The Internet traffic has (on average) one flow for every user (in the proportion reported in [90], i.e., 75% HTTP, 5% FTP, etc.).
- The stationary network consists of 144 nodes uniformly distributed in a rectangular area of 2.4×2.4 km^2.
- There are 36 mobile nodes deployed in random locations. They move according to the random waypoint mobility model [83] with zero pause time, a maximum speed of 10 m/s, and a minimum speed of 1 m/s.
- There is a single gateway at the center of the network.
- The stationary nodes are deployed with zero perturbation (i.e., they are deployed in a grid of size 200 m).
- There is no intra-mesh traffic.
- Total simulation time is 400 s.

To evaluate the performance of MRP, we chose six well-known routing protocols representative of different classes of routing protocols: AODV [45,63], DSR [46], LANMAR [68,69], OLSR [57], RIP2 [62], and ZRP [64]. AODV and OLSR are standardized by the MANET workgroup of IETF. AODV and DSR are widely known *on-demand (reactive)* MANET routing protocols. LANMAR is designed for routing in large scale ad-hoc networks with group mobility. In QualNet implementation, LANMAR uses Fisheye State Routing (FSR) protocol for local scope routing. OLSR is based on the Inria implementation [91], and it is an optimized *link-state*, table-driven protocol designed for MANETs. RIP2 is a table-driven *distance-vector* protocol originally designed for wired Internet. The RIP2 model

in QualNet follows the Cisco implementation. ZRP is a hybrid ad-hoc routing protocol where proactive and reactive approaches are employed for intra- and inter-zone routing, respectively.

For each scenario, we simulated the network thirty times with different random seeds (resulting in different initial placements and mobility for the mobile nodes and in different inter-arrival times of the offered load). In every graph, we present the average of the thirty experiments. To avoid cluttering the figures, confidence intervals are not presented.

3.4.3 Poisson Traffic

In this section, simulation results for Poisson traffic are examined for five different scenarios where traffic load, network size, mobility, perturbation, and intra-mesh traffic ratio are varied from the base case. Except for the scenario of varied intra-mesh traffic, all nodes have packet streams flowing both to and from the gateway. The same traffic load is applied in both directions.

Traffic Load

We increase the load of the Poisson traffic by reducing the mean packet inter-arrival time from 5.0 s to 1.5 s (i.e., from 0.2 to 0.67 packets per second).

Figure 3.7 depicts the simulation results for the overhead, PDR, delay, and hop-count, respectively. In each graph, curves for nine routing protocols (including three versions of MRP) are plotted as the offered load increases.

Figure 3.7-(a) shows that the routing overhead of AODV rises as traffic load is increased. The main reason behind this increase is the corresponding increase in lost packets (that triggers the route discovery process). DSR shows lower overhead; however, the overhead of source route header is not included in the overhead calculation. In contrast, MRP and all the proactive protocols show overhead immunity to the traffic load. All three versions of MRP and LANMAR form the group of the lowest overhead of all considered protocols. MRP-H has a very low overhead, due to the larger beacon period (twice as large as MRP-B).

Figure 3.7-(b) shows that, as the offered traffic load intensifies, the PDR of AODV, DSR, OLSR, and ZRP drops faster than MRP. The drop in PDR is due to the packet loss in the queues, as well as lost routes due to the routing protocols' attempt to restore failed routes (or what are considered to be

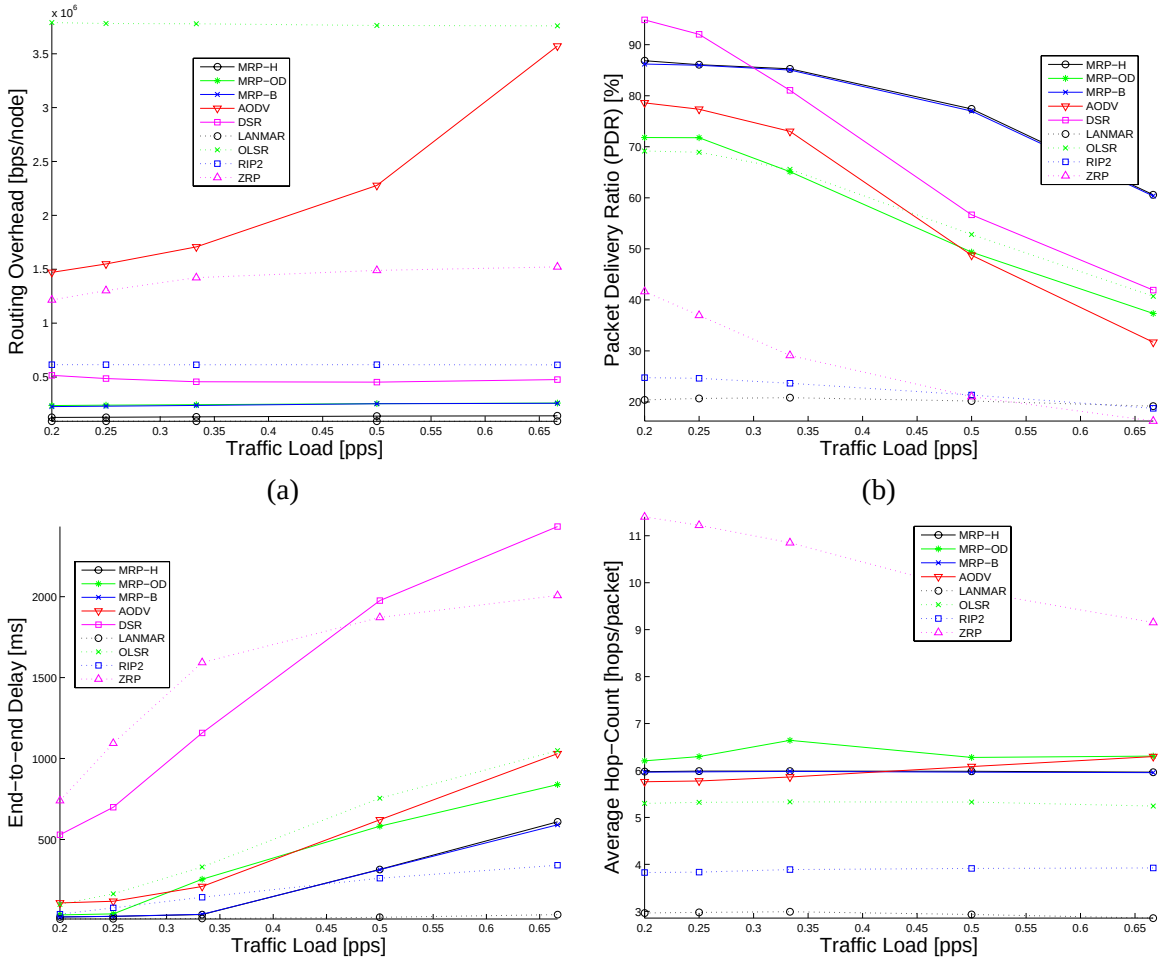


Figure 3.7: The performance of the routing protocols for Poisson traffic (a) routing overhead, (b) packet delivery ratio, (c) end-to-end delay, and (d) average hop-count as a function of traffic load.

failed routes due to packet drops). RIP2 shows the lowest PDR for the highest traffic load, indicating the unsuitability of wireline-oriented protocol for ad-hoc or mesh networks.

The delay increase in almost all of the protocols shown in Fig. 3.7-(c) is due to the larger queuing delays resulting from the increase in offered load. MRP-OD has a higher delay than MRP-B and MRP-H, as it discovers broken routes later than the other two. The low delays of RIP2 and LANMAR have to be considered with the very low PDR in Fig. 3.7-(b): the delay statistics consider only the few packets that reach their destinations.

Figure 3.7-(d) shows that the hop-count of each packet is almost insensitive to an increase in traffic load. Two on-demand protocols (AODV and MRP-OD) show a slight increase in the hop-count. The hop-count result of DSR is not included because measuring the TTL field of IP packets is not possible in QualNet due to its source routing implementation.

As expected, among three versions of MRP, MRP-H exhibits the best performance as it combines strengths from both MRP-B and MRP-OD.

Network Size

In this scenario, the number of fixed nodes is increased while keeping the network density constant (i.e., increasing the network diameter). The four graphs for overhead, PDR, delay, and hop-count are shown in Fig. 3.8, as a function of the network size.

In Fig. 3.8-(a), MRP shows almost constant overhead, while all of the other protocols show the overhead increasing with the network size. This is expected because for MRP, the overhead for each node does not increase with the total number of nodes (see Section 3.3). As a result, MRP also maintains the highest PDR and the lowest delay. This supports the claim that, for WMNs, MRP scales better than the existing protocols.

Fig. 3.8-(b) shows that the PDR of MRP-H and MRP-B slightly increases with the network size while all other protocols show decrease in PDR. Two main factors behind the increased performance of MRP are the absence of flooding and the route selection algorithm that prefers a more reliable path to shorter hop-count distance. These two factors enable more efficient mobility support in large networks than flooding-oriented protocols.

For large networks (i.e., over 100 fixed nodes), broken links due to mobility trigger route rediscoveries, which incur high overhead with flooding. The PDR of MRP-H and MRP-B increases because

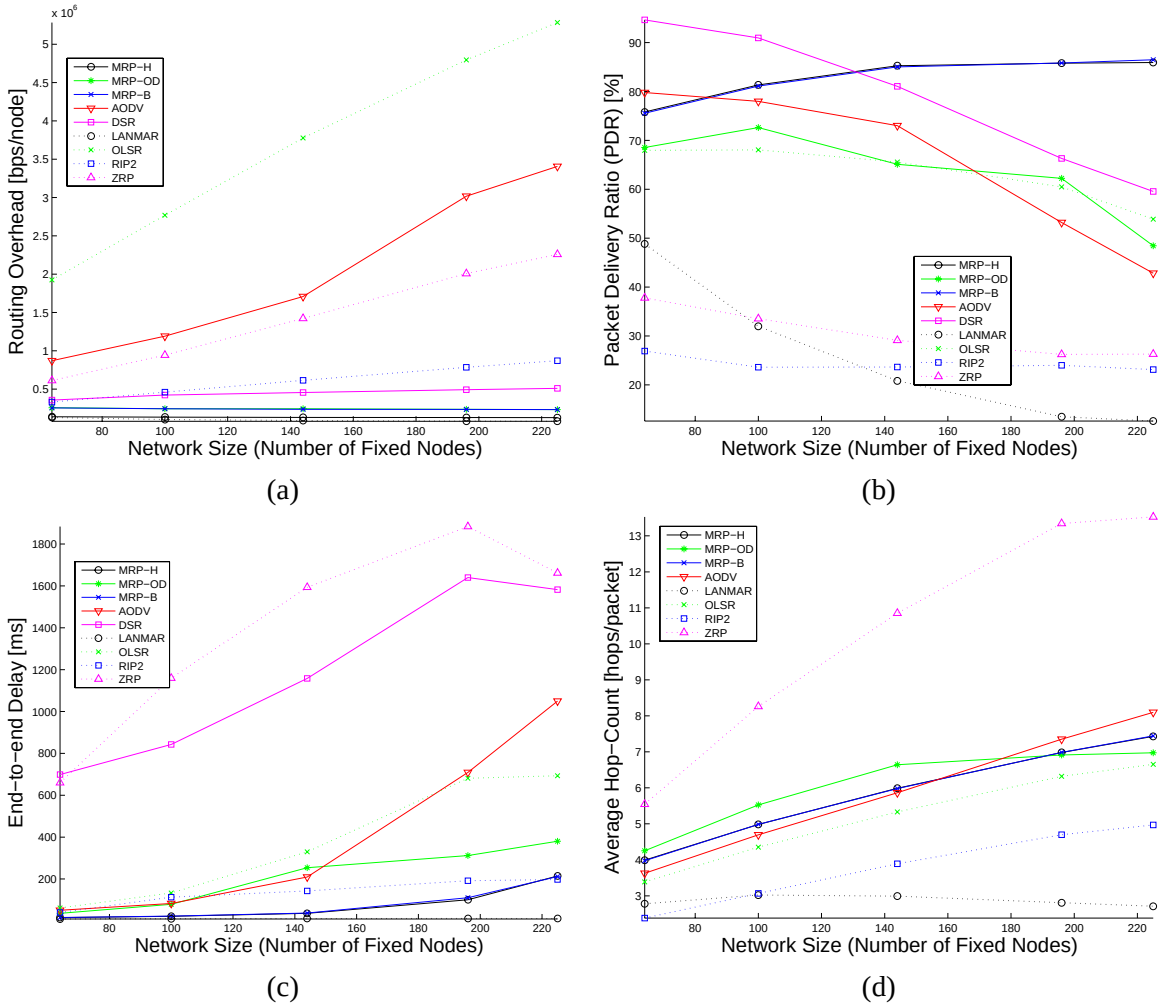


Figure 3.8: The performance of the routing protocols for Poisson traffic (a) routing overhead, (b) packet delivery ratio, (c) end-to-end delay, and (d) average hop-count as a function of network size.

the number of mobiles nodes are constant (i.e., 36 mobiles), while the number of fixed nodes increases, thus the ratio of mobile nodes decreases. Simulation results for varied mobility will be presented and discussed in detail in the following section.

Figure 3.8-(c) shows high delay for DSR and ZRP compared to other protocols. All three versions of MRP show good delay performance for scaled network size.

The hop-count shown in Fig. 3.8-(d) directly reflects the increase in network size. Three versions of MRP, AODV, and OLSR show hop-counts close to the ideal hop-count. ZRP shows longer than ideal routes indicating suboptimal routes, while LANMAR and RIP2 only route successfully to/from nodes closer to the gateway, and thus show shorter hop-counts.

Number of Mobile Nodes

In this scenario, we evaluate the performance of MRP and other protocols when the number of mobile nodes increases from 9 to 49. The results are shown in Fig. 3.9.

As in the cases of traffic load and network size, Fig. 3.9-(a) shows that MRP produces a low and steady overhead as the number of mobile nodes is increased. In Fig. 3.9-(b), MRP shows higher PDR than other protocols as the mobility intensifies. RIP2 shows a sharp drop in PDR as it was designed for stationary networks with sporadic topology changes. For a large number of mobile nodes, DSR shows rapidly increasing delay, as shown in Fig. 3.9-(c). The result supports the claim that, for WMNs, MRP supports mobility more efficiently than existing protocols.

Perturbation

The performance of MRP for increased randomness in the deployment of the stationary nodes is shown in Fig. 3.10. For this scenario, the stationary nodes are perturbed uniformly from their ideal grid positions. The degree of perturbation indicates by how much a node can deviate from its ideal position. A perturbation of 100% indicates that a node can be moved by up to one grid size in any direction.

As the perturbation increases, the network becomes less regular and hot spots are created. In Fig. 3.10-(a), DSR and AODV show increased overhead, while all three versions of MRP remain low and almost constant. OLSR and ZRP show a decrease in overhead; in OLSR, the multipoint relays (MPRs)

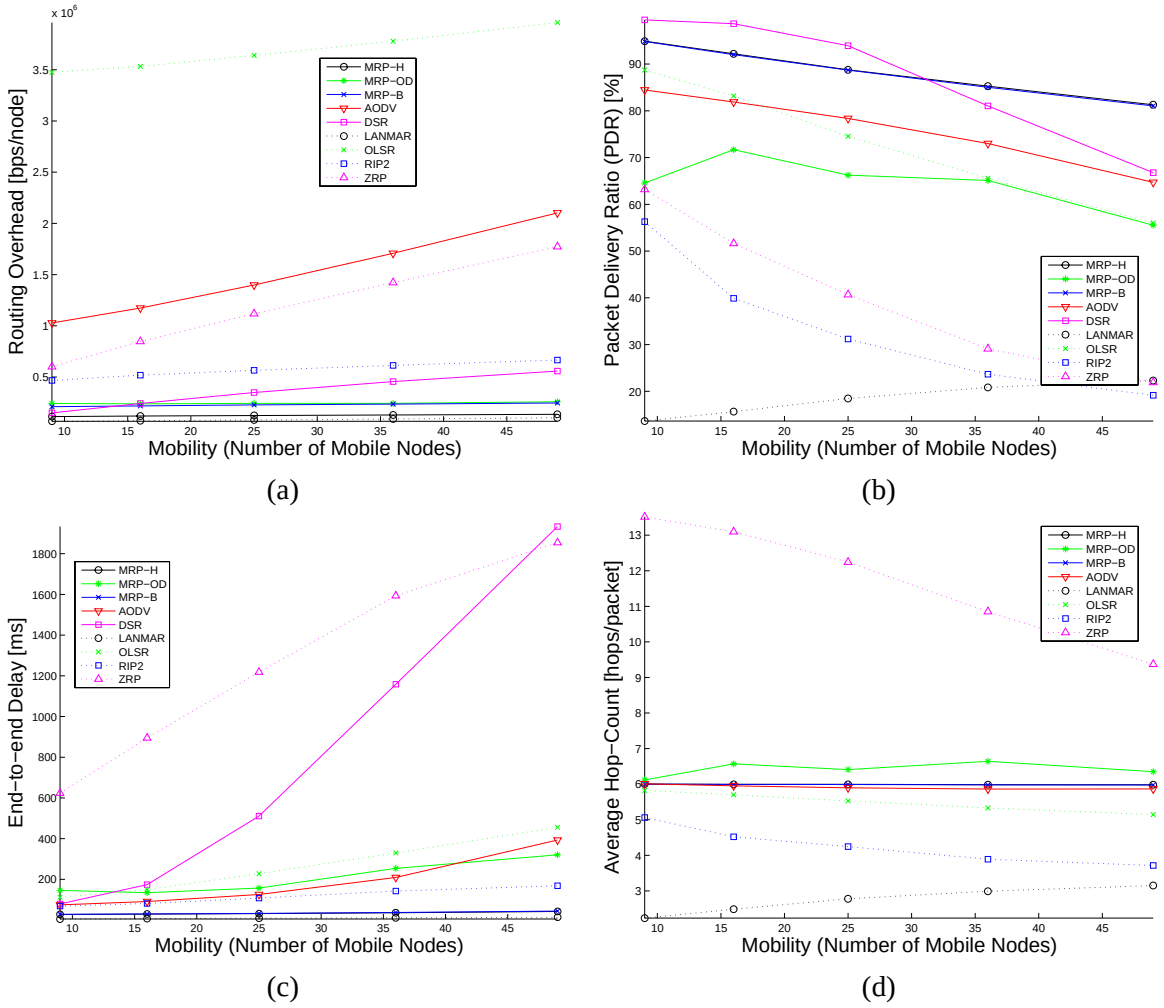


Figure 3.9: The performance of the routing protocols for Poisson traffic (a) routing overhead, (b) packet delivery ratio, (c) end-to-end delay, and (d) average hop-count as a function of mobility.

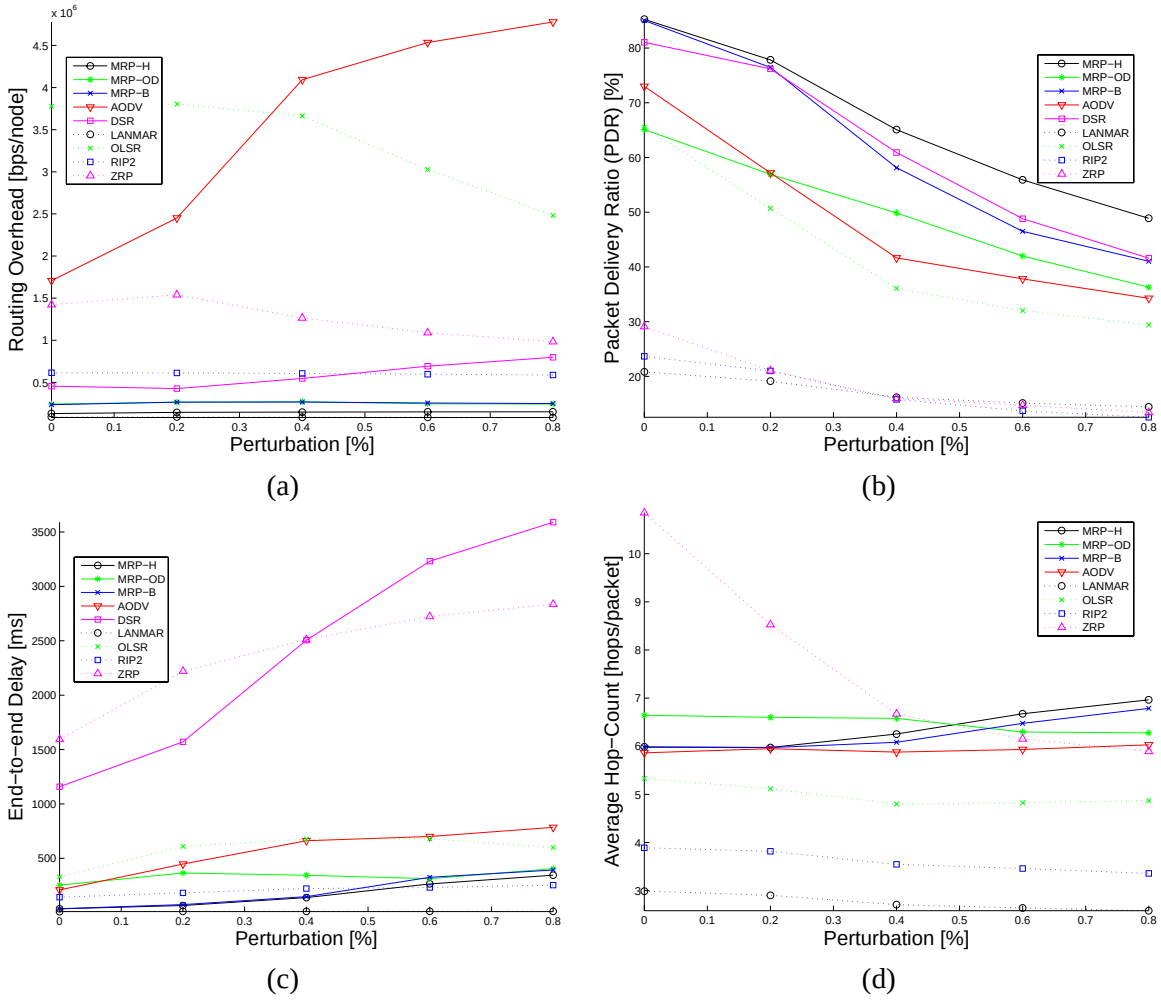


Figure 3.10: The performance of the routing protocols for Poisson traffic (a) routing overhead, (b) packet delivery ratio, (c) end-to-end delay, and (d) average hop-count as a function of the perturbation in stationary nodes grid deployment.

benefit from non-uniform node distribution. Likewise, in ZRP, as perturbation increases, fewer routing zones are required to cover the user nodes, and thus, less inter-zone, on-demand routing is needed.

Figures 3.10-(b) and (c) show decreased PDR and increased delays as the network becomes less regular. MRP shows better PDR and delay performance than other protocols. Among three versions of MRP, MRP-H shows the highest PDR and the lowest delay.

Figure 3.10-(d) shows that the hop-count for MRP increases slightly as the network becomes less regular; this is due to the route selection algorithm that favors stability to lower hop-counts.

Intra-mesh Traffic

In this scenario, the intra-mesh traffic ratio is increased from zero to 0.8. The ratio is calculated as the number of intra-mesh traffic flows over the total number of user traffic flows. For example, the ratio of 0.8 indicates that 80% of the user traffic flows are both originated and terminated within the mesh network (i.e., neither end of a traffic flow is a gateway.) The higher the intra-mesh traffic ratio, the stronger the characteristics of *ad-hoc* networks rather than those of *mesh* networks. The results are presented in Fig. 3.11.

In Fig. 3.11-(a), on-demand protocols such as AODV, DSR, and ZRP (ZRP has both on-demand and table-driven components) show a small increase in overhead as intra-mesh traffic ratio is increased. These protocols have to discover routes for destinations located farther away from the source due to increased intra-mesh traffic. Table-driven protocols show almost constant overhead.

Figure 3.11-(b) shows increases in PDR for AODV, DSR, and ZRP; while for MRP, the PDR decreases as the intra-mesh traffic ratio increases. This result is expected as AODV, DSR, and ZRP are designed for generic ad-hoc networks where all of the user nodes are assumed to be uniformly likely to send data to any destination in the network. In contrast, routes in MRP are always routed through the common parent of the source and the destination (often the gateway); thus, the routes are sub-optimal. The extent of this effect is shown in Fig. 3.11-(d).

3.4.4 Internet Traffic

In this section, simulation results for Internet traffic are examined for five different scenarios where traffic load, network size, mobility, perturbation, and intra-mesh traffic ratio are varied from

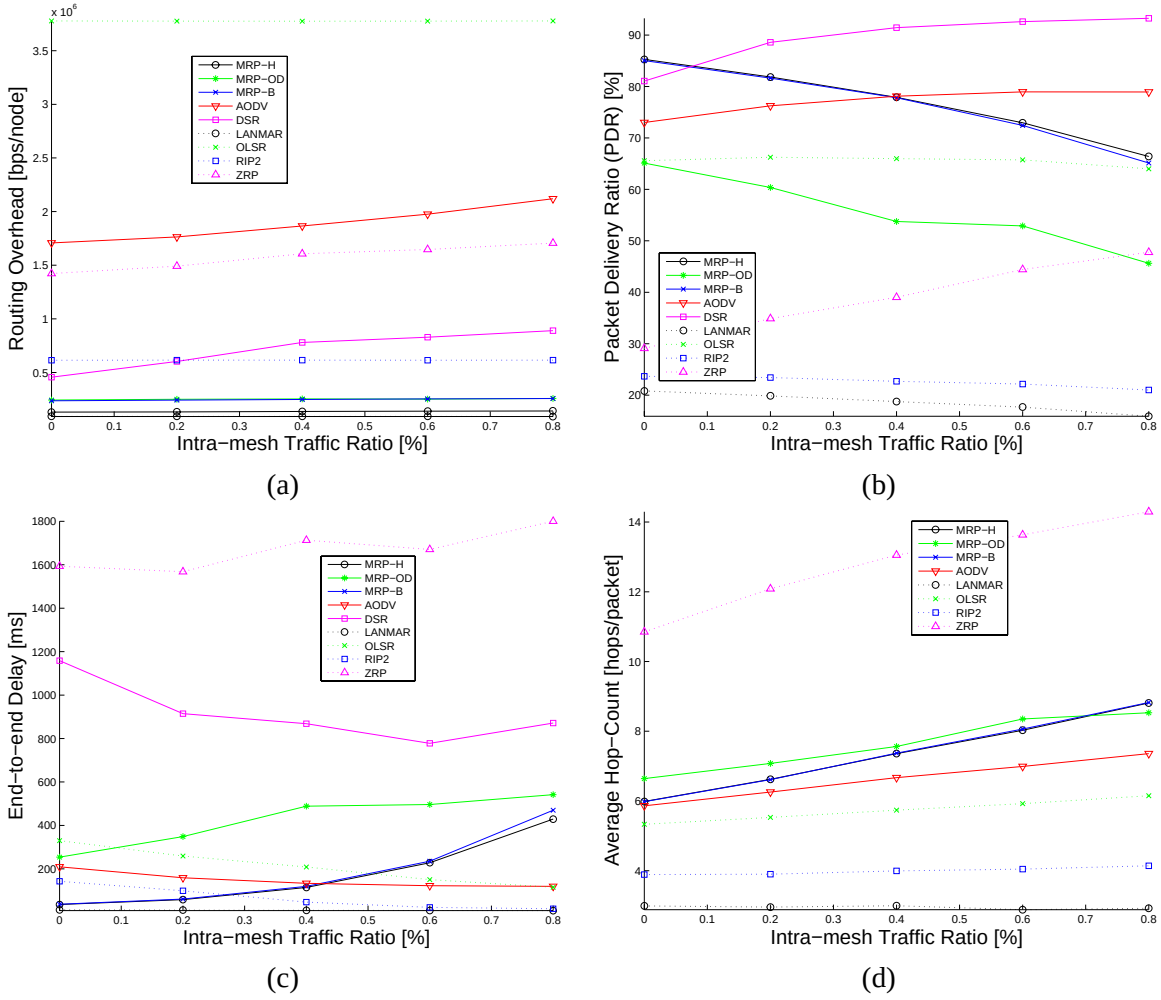


Figure 3.11: The performance of the routing protocols for Poisson traffic (a) routing overhead, (b) packet delivery ratio, (c) end-to-end delay, and (d) average hop-count as a function of the intra-mesh traffic ratio.

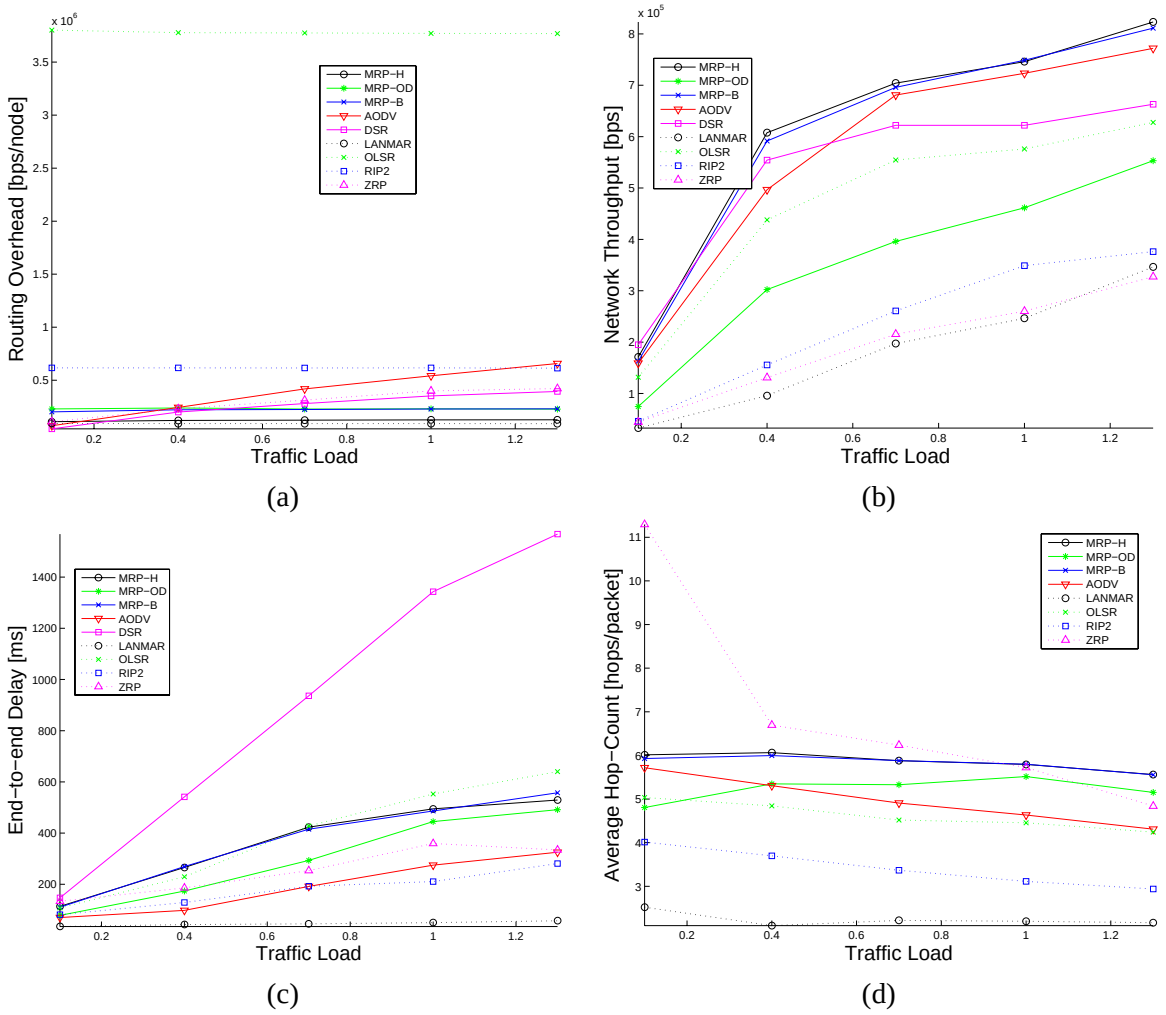


Figure 3.12: The performance of the routing protocols for Internet traffic (a) routing overhead, (b) network throughput, (c) end-to-end delay, and (d) average hop-count as a function of traffic load.

the base case. The purpose of using Internet traffic is to evaluate the performance of MRP and other protocols in a more realistic environment. The Internet traffic model is created by mixing several popular Internet applications based on real Internet traffic measurements as depicted in Section 3.4.2.

Traffic Load

In this scenario, we vary the relative traffic load of Internet traffic from 0.2 to 1.4. The relative traffic load is controlled by changing the number of active users while maintaining the ratio among different applications. For a traffic load of 1.0, every node is likely to have one active application. If the

load is greater than 1.0, some nodes are likely to have more than one active application.

Figure 3.12-(a) shows that the routing overhead of on-demand protocols (AODV, DSR, and ZRP) increases with the traffic load. This result and its cause is similar to the case of Poisson traffic: more packets tend to trigger more route discoveries for those protocols. In contrast, MRP and all of the table-driven protocols show overhead immunity to the traffic load.

As the offered traffic increases, the network throughput approaches network capacity due to the feedback loop of TCP that adjusts the transmission rate to the available bandwidth. Figure 3.12-(b) shows that AODV, DSR, and OLSR have lower saturation throughput than MRP. The lower throughput is due to the packet loss in the queues and failed routes due to packet drops. MRP-OD has lower throughput than MRP-H and MRP-B as it discovers broken routes later than the other two.

In Fig. 3.12-(c), the delay increase for all protocols is due to the larger queuing delays resulting from increased offered load. DSR shows higher delay than other protocols. The low delay of LANMAR has to be considered in conjunction with its very low throughput (the delay statistics only consider the few packets delivered to their destinations).

Network Size

We increase the network size in the same manner as in the Poisson traffic case, i.e., increase the number of fixed nodes while keeping the density constant. The results are shown in Fig. 3.13.

Figure 3.13-(a) shows that the overhead of MRP stays almost constant, while that of other protocols (especially OLSR) rises as the network size increases. As in the case of Poisson traffic, this result is expected in MRP as the overhead for each node does not increase with the total number of nodes.

Figure 3.13-(b) shows that MRP-H and MRP-B maintain the highest throughput. While packet loss simply reduces the PDR in Poisson traffic where UDP is used as its transport layer, the majority of the applications in Internet traffic rely on TCP. In response to packet loss and route rediscoveries, TCP congestion control reduces its transmission rate, thus decreasing the throughput.

In Figure 3.13-(c), it is shown that the delay increases for all of the routing protocols as the network diameter increases.

The increase in hop-count of MRP in Fig. 3.13-(d) is correlated to the increase in the network diameter. AODV has a lower hop-count than MRP, as AODV often chooses shorter (but possibly less

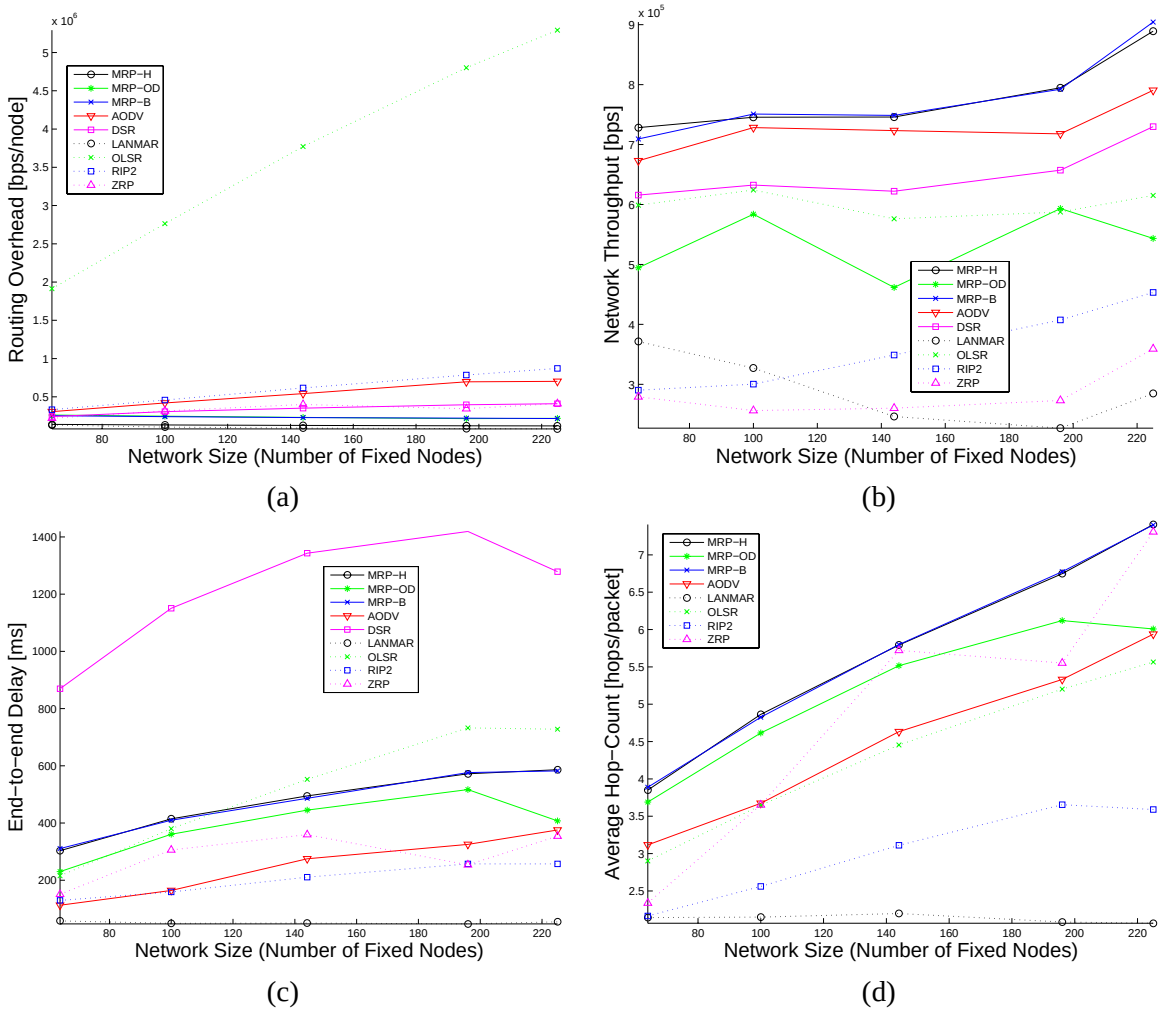


Figure 3.13: The performance of the routing protocols for Internet traffic (a) routing overhead, (b) network throughput, (c) end-to-end delay, and (d) average hop-count as a function of network size.

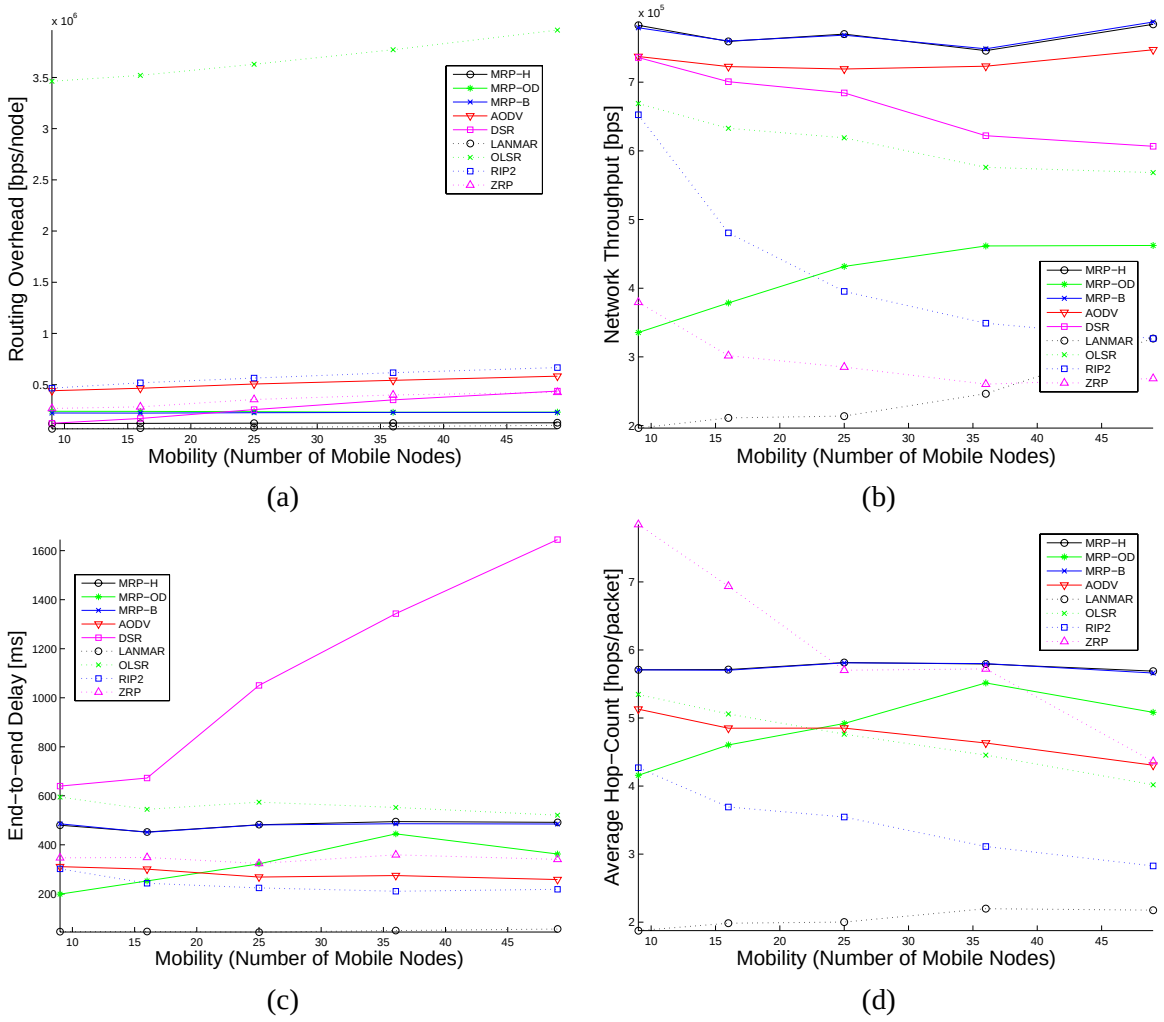


Figure 3.14: The performance of the routing protocols for Internet traffic (a) routing overhead, (b) network throughput, (c) end-to-end delay, and (d) average hop-count as a function of mobility.

reliable) routes, while MRP prefers longer routes through stationary nodes.

Number of Mobile Nodes

The number of mobile nodes is increased in the same manner as in the Poisson case. Figure 3.14-(a) shows that MRP produces a low and almost constant overhead as the number of mobile nodes is increased.

In Fig. 3.14-(b), MRP-H and MRP-B show the highest network throughput as the mobility intensifies. MRP-OD shows relatively low throughput because MRP-OD relies on packet loss to detect

a broken link; multiple lost packets drastically affect the TCP's congestion control mechanism (that times-out multiple times before a route is rediscovered). Table-driven protocols such as RIP2, ZRP (ZRP is table-driven for intra-zone routing), and OLSR show decrease in the network throughput.

In Fig. 3.14-(c), MRP shows higher delay than AODV. This result should be considered with AODV's lower throughput and shorter hop-count in Figures 3.14-(b) and (d), which indicates that AODV chooses shorter routes, but loses more packets than MRP due to less reliable routes. DSR shows higher delay than other protocols.

Perturbation

The deployment of the stationary nodes is perturbed in the same manner as in the Poisson case. The perturbation is increased from zero to 80%.

In Fig. 3.15-(a), MRP shows almost constant overhead. Interestingly, for Internet traffic, the control overhead of AODV and DSR does not increase as in the case of Poisson traffic. As perturbation is increased, the overhead of OLSR decreases for the same reason as for the Poisson case (Section 3.4.3).

Figures 3.15-(b) and (c) show decrease in the network throughput and increase in delay for all protocols as grid is perturbed. MRP shows better throughput and reasonable delay performance compared to other protocols. Among three versions of MRP, MRP-H performs best.

Figure 3.15-(d) shows that the hop-count of MRP slightly increases, while that of all others decreases. Again, this is the effect of the route selection procedure that prefers stable routes to lower hop-counts.

Intra-mesh Traffic Ratio

The intra-mesh traffic ratio is varied in the same manner as in the Poisson case. Intra-mesh traffic ratio is increased from zero to 0.8. The results are shown in Fig. 3.16.

In Fig. 3.16-(a), on-demand protocols such as AODV, DSR, and ZRP (ZRP uses on-demand approach for inter-zone routing) show an increased overhead because these protocols have to perform route discovery for destinations located farther away. As expected, MRP and table-driven protocols show almost constant overhead.

Figure 3.16-(b) shows increased network throughput for AODV, DSR, OLSR, and ZRP, while MRP shows a decrease. This result is expected because the former protocols are designed for generic ad-

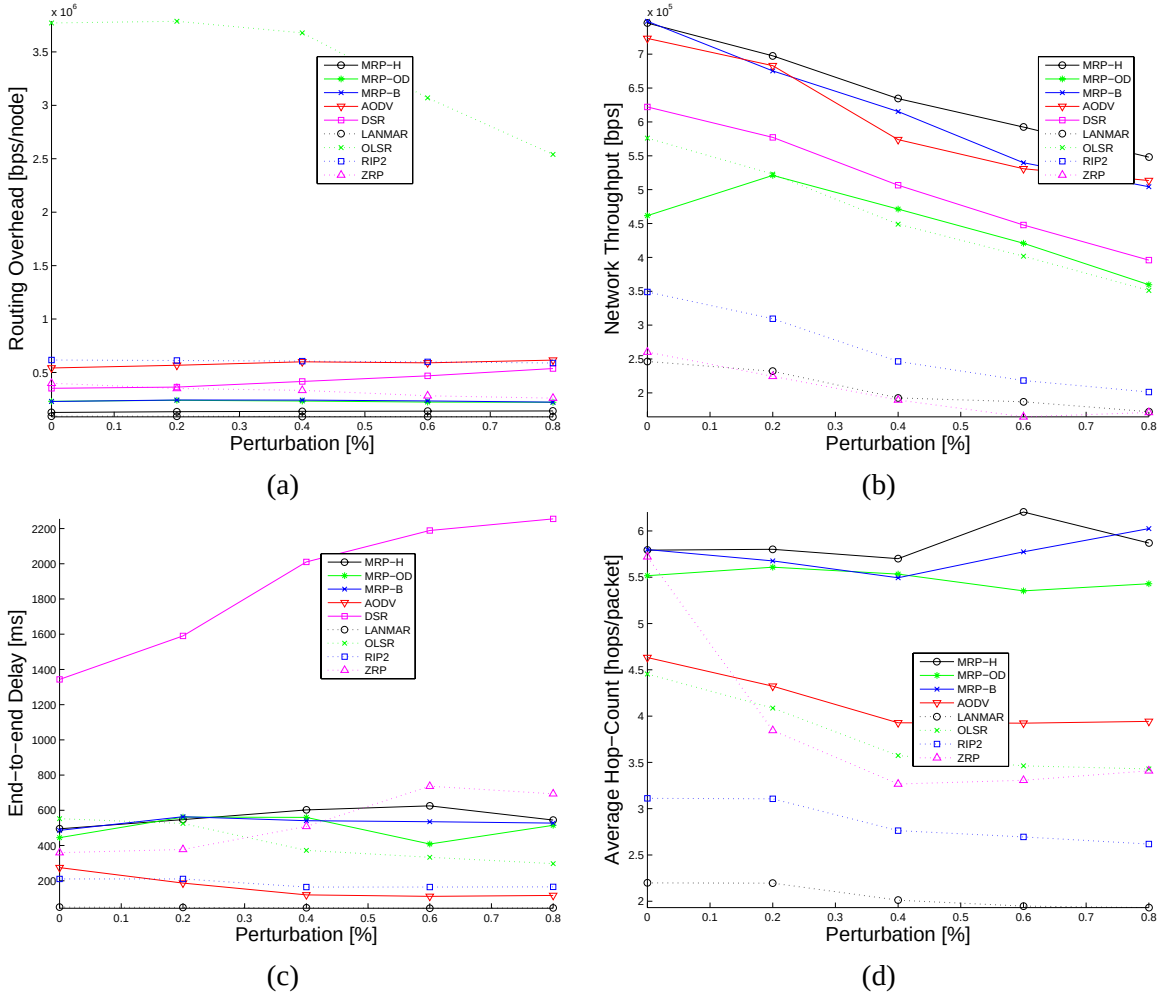


Figure 3.15: The performance of the routing protocols for Internet traffic (a) routing overhead, (b) network throughput, (c) end-to-end delay, and (d) average hop-count as a function of the perturbation in stationary nodes grid deployment.

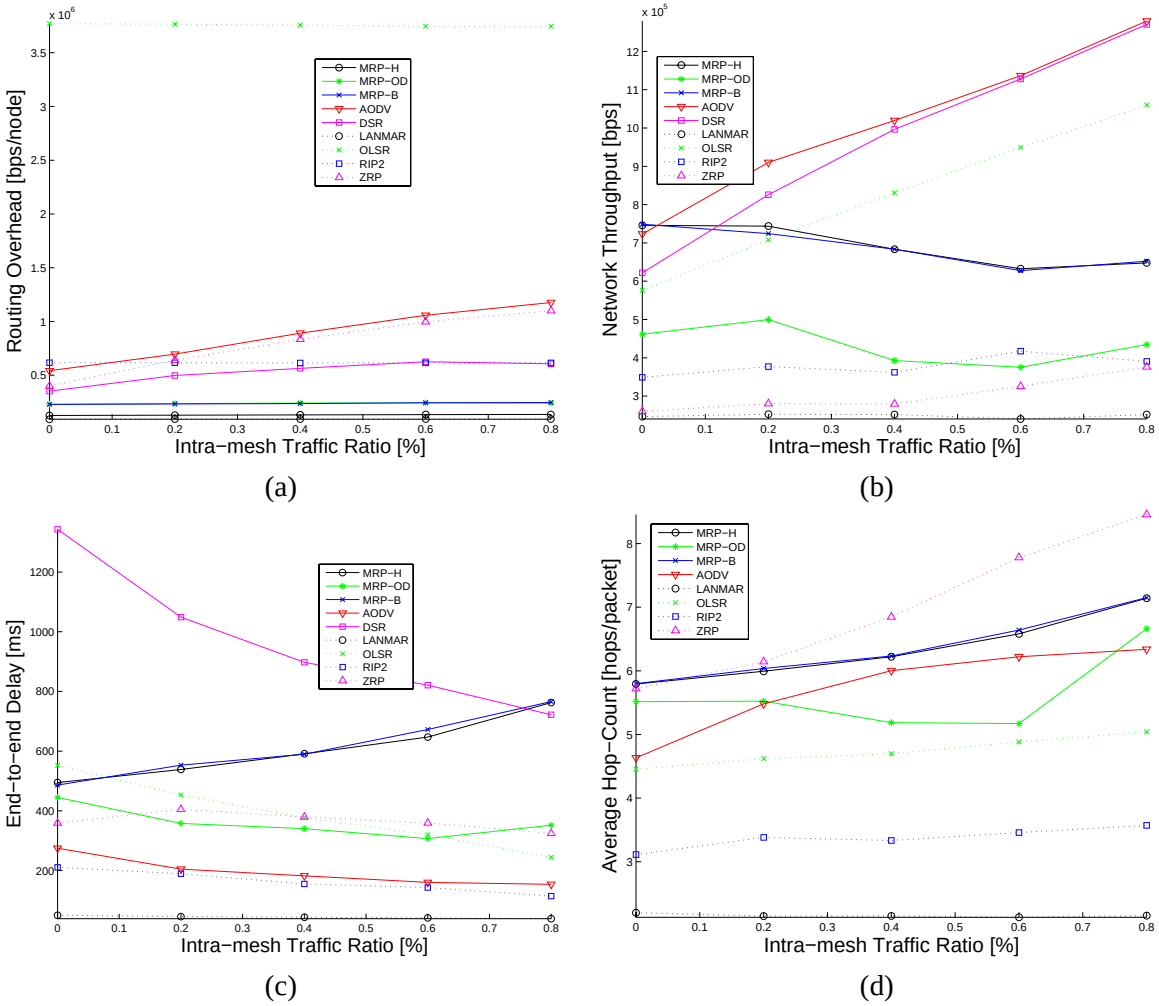


Figure 3.16: The performance of the routing protocols for Internet traffic (a) routing overhead, (b) network throughput, (c) end-to-end delay, and (d) average hop-count as a function of the intra-mesh traffic ratio.

hoc networks with destinations at arbitrary locations in the network. In contrast, routes in MRP always involve the common parent of the source and the destination. The resulting suboptimal routes are also reflected in the delay and hop-count performance shown in Figures 3.16-(c) and (d), respectively.

The simulation results show that MRP performs well in WMNs that have the most user traffic to and from the gateway (and intra-mesh traffic is a small percentage of the overall traffic).

In this section, we showed that *for WMNs*, MRP outperforms the other protocols for the considered metrics. MRP shows less overhead, higher PDR (for the Poisson traffic model), higher network throughput (for the Internet traffic model), and lower delay. MRP shows slightly higher hop-count than some of the other protocols as its route selection algorithm prefers stabler rather than smaller hop-count routes. MRP consistently performed better than others when traffic load, network size, mobility, and perturbation were increased. As MRP is optimized for WMNs, when intra-mesh traffic ratio is increased, MRP's advantage withers. Among the three MRP versions, MRP-H showed the best performance, followed closely by MRP-B (although with twice the overhead of MRP-H). MRP-OD showed relatively poor performance compared to the other two versions.

3.5 Summary of MRP

Wireless mesh networks are becoming increasingly popular as they have significant advantages over competing technologies. In this research, we introduce a new routing protocol specifically designed for those networks. The design of the proposed routing protocol takes advantage of the particularities of WMNs, only maintaining *routing trees* to and from the gateways. Simulation results show that, for WMNs, the proposed protocol outperforms existing Internet and MANET routing protocols.

3.6 Acknowledgement

The research presented in this chapter was supported by the Center for Advanced Computing and Communication.

Chapter 4

Scalable Updates in MANETs

In this chapter, an efficient routing overhead reduction scheme for mobile ad hoc networks (MANETs) is proposed. The proposed scheme is applied to OSPF, a widely-used Internet routing protocol as an extension for MANET environments. Recently, MANETs have evolved as an irreplaceable networking technology for situations with sparse or inexistent infrastructure. Adapting Open Shortest Path First (OSPF) to MANET environments has several advantages. The protocol has proven its maturity in the wired Internet and is now the de facto standard Intra-AS Internet routing protocol. When used in wireless ad hoc networks with proper extensions, OSPF can provide better interoperation between the ad hoc domain and the Internet. This is one of the essential requirements for wireless ad hoc networks that require connectivity to the Internet. The main challenge is that since OSPF is highly optimized for wired-oriented environment, it needs further enhancement to be fully functional in ad hoc networks. The most serious obstacle is the large routing overhead associated with the frequent topology changes due to volatile wireless links and node mobility. Various overhead reduction schemes have been proposed by focusing on how to reduce hello packets, flooding nodes, or the number of adjacencies. We propose a completely different approach by taking into account an important characteristics of wireless ad hoc networks - namely, the distance effect. We implemented the proposed scheme on a simulator that uses the real OSPF code in its routing module. The simulation experiments show that the proposed scheme results in a significant reduction in OSPF routing overhead at small loss of route optimality.

4.1 Introduction to Scalable Updates in MANETs

The main characteristic of ad-hoc networks is the absence of pre-planning. The topology of the network is discovered on the fly, after the network's deployment. Thus, such a network must exchange a large number of messages which are used to "set-up" various parameters in the network. Example of such parameters are the existence of other nodes in the network, their position, information about their neighbors, what services they offer (e.g., local maps, files, printing facilities). The location information and the neighbors will change with mobility, and thus will have to be updated. As the number of nodes and the mobility increases, the updates will start to be a significant percentage of the total traffic in the network, and at some point, the network will not be able to carry all the updates, let alone the "useful" user traffic.

Information about other nodes' neighbors and/or their location (as well as distance vector or link state information) is especially useful in routing. On one hand routing is essential for the functioning of the ad-hoc network, so one cannot do without it. On the other hand the updates consume available bandwidth in the ad-hoc network. The bandwidth consumed by the control overhead reduces the bandwidth for user applications. This leads to a significant problem in typical MANETs where bandwidth is a scarce resource.

4.1.1 Related Work

Various schemes have been proposed to alleviate the overhead problem in MANETs. Fisheye State Routing (FSR) [55] is a link state routing protocol for MANETs that attempts to address the scalability problem. The main idea of FSR is that the updates corresponding to closer nodes propagate more frequently. Since the topology map maintained at each node relies on periodic link state packets that are not flooded as in conventional protocols, it is expected to require less overhead than other link state protocols with full flooding. OLSR [57] is another link state routing protocol for MANETs that reduces the overhead by reducing the number of nodes participating in link state flooding. Some protocols (e.g., [44] [74]) rely on clustering scheme to contain the overhead within a certain boundary. DSR [46] uses aggressive caching to reduce the flooding. Location information is often incorporated into routing protocols to reduce the overhead and increase the scalability. Some examples are LAR [49], DREAM [50] and ZRP [64]. Existing solutions show improvements in flooding overhead using different

approaches. However, they require additional protocol complexity (e.g., formation and maintenance of routing hierarchy), storage (e.g., caching) or information (e.g., location).

4.1.2 Desired Properties

We aim at providing a distributed, efficient, and scalable scheme that can be either applied to existing protocols as an extension or implemented as a stand-alone protocol. The main characteristics of the proposed scheme can be described as:

- It does not rely on any external information other than simple network layer information readily available (e.g., hop count).
- It is fully distributed and simple enough to require no central control of the topology.
- It has the analytical property that the overhead asymptotically increases as $O(1)$ with the scaled network size.
- It is not protocol-specific and thus, it can be easily applied to almost all the classes of existing ad hoc routing protocols that employ flooding as its main or supplementary mechanism.

4.1.3 Extending OSPF for MANETs

OSPF [92] is by far the most popular intra autonomous system routing protocol in the Internet. In the latest version (OSPF for IPv6 [93]), the protocol was dissociated from the addressing scheme. This change allows the protocol to work not only with IPv6 addresses but with more general addressing schemes. Adapting OSPF to MANET environments has several advantages. The protocol is mature in the wired Internet and is now the de facto standard. When used in wireless ad hoc networks with proper extensions, OSPF can provide better interoperation between the ad hoc domain and the Internet. The most serious obstacle in using OSPF in MANETs is the large flooding overhead associated with the frequent topology changes.

4.1.4 organization

The presentation of new scalable update scheme for MANETs is organized as follows. Section 4.2 proposes a new scheme to solve the problem. Analytical results are provided in Section 4.2.1.

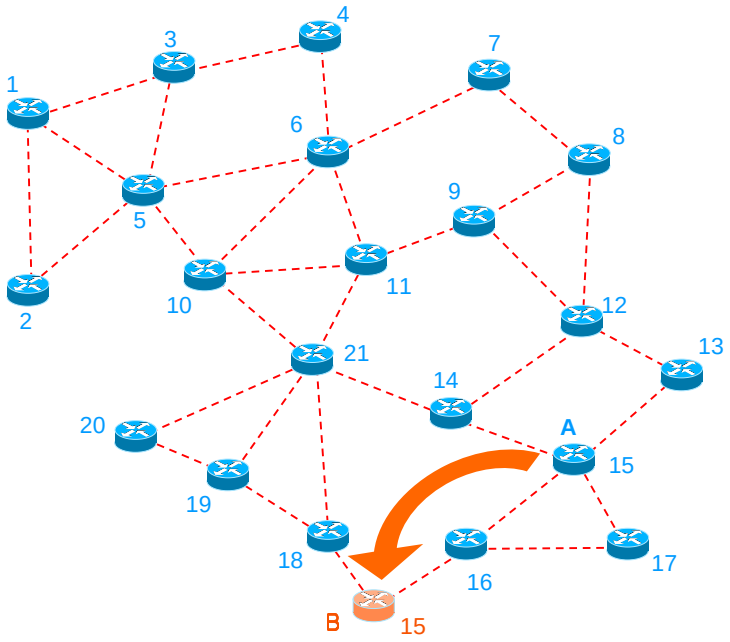


Figure 4.1: When node 15 moves from point A to point B, it must send updates to other nodes such that those nodes will be able to reach it.

Section 4.2.2 explains the flexibility of the scheme by presenting potential variations with different parameters. Section 4.2.3 discusses the possible trade-offs between the reduction in the overhead and the route optimality. Section 4.3 describes the implementation of the scheme in OSPF. Section 4.4 compares the efficiency of the scheme with pure OSPFv3 and its existing extensions through simulation experiments. Finally, Section 4.5 summarizes this chapter.

4.2 Proposed Solution

In many MANET protocols, the updates are done by flooding [42, 57]. This means, that when a node needs to update the others it will send (i.e., locally broadcast) the update to all its neighbors, and the neighbors to their neighbors, and so on. In this way, the update will eventually reach all nodes.

Assume that in Fig. 4.1, node 1 wants to send a message to node 15. When node 15 is in position A the messages from 1 to 15 will likely use the following route: 1-5-10-21-14-15. When node 15 moves to position B the route will likely be: 1-5-10-21-18-15. The first hop of node 1 is node 5 regardless of the position of node 15. Thus, we can say that 15's update going all the way to 1 is

```
char from_node[n] = {'y','y','y',....,'y'};

proces_update (received_update)
{
   if (from_node[received_update->source] == 'y')
   {
     from_node[received_update->source] == 'n';
     forward (received_update);
   }
   else
   {
     from_node[received_update->source] ← 'y';
   }
}
```

Figure 4.2: Pseudo-code for processing the received updates.

"useless". On the other hand, the route from node 12 to node 15 changes substantially when node 15 moves. The general rule is that the closer neighbors should be updated more often than the nodes which are far away. The idea is present in the literature [50, 55], but the implications were not fully explored.

A different way to update is not to forward all the updates, but only a few of them. For example, a node may forward only half of the updates it receives. Figure 4.2 depicts the pseudo-code of the routine processing the received updates. Under this policy, the nodes one hop away from the node sending the updates will get all updates, the nodes two hops away will get half of the updates, the nodes three hops away will get a quarter of the updates, etc. A node h hops away from the source of the updates will get $\frac{1}{2^{h-1}}$ of the updates.

4.2.1 Analysis

In this section, the asymptotic efficiency of the proposed scheme is presented. Two regular topologies are considered in this analysis. We first define some important variables used commonly in the analysis of the two cases as follows:

- $k \equiv$ the total number of updates generated at the center node
- $i \equiv$ the distance (i.e., number of hops) between the center node and another node in the network
- $N(i,k) \equiv$ the total number of updates (including both the generated updates at the center node and the relayed updates by other nodes) as a function of i and k

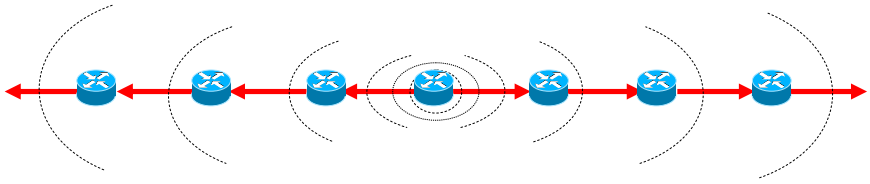


Figure 4.3: A regular chain topology chosen for the analysis of asymptotic efficiency.

- $R(i) \equiv$ the update generation rate induced by the center node in the network of diameter i, $R(i) = \frac{N(i,k)}{k}$
- $\beta \equiv$ the update relaying factor ($0 < \beta < 1$) at each node, e.g., one out of every $\frac{1}{\beta}$ arriving updates is relayed in exponential reduction

We investigate the overhead reduction efficiency for the chain topology and then extend the method to the grid topology. Figure 4.3 illustrates the chain topology under consideration. We focus on the effect of the updates generated at the node located in the center of the network, namely the center node. Assuming full flooding scheme is being used, we can express $N(i,k)$ and $R(i)$ as:

$$N_{flooding}(i,k) = 2k + ... + 2k = 2ik, \tag{4.1}$$

$$R_{flooding}(i) = \frac{2ik}{k} = 2i = \Theta(i). \tag{4.2}$$

If the proposed scheme is applied to the same scenario, the resulting $N(i,k)$ and $R(i)$ will become:

$$N_{new}(i,k) = 2k \sum_{l=1}^{i} \beta^l = 2k \left[\frac{\beta(1-\beta^i)}{1-\beta} \right] \tag{4.3}$$

$$R_{new}(i) = \frac{2\beta}{1-\beta} \left(1 - \beta^i\right) = \Theta(1). \tag{4.4}$$

The results from (4.2) and (4.4) indicate that the asymptotic update generation of the new scheme provides significant enhancement in scalability over the full flooding scheme.

Figure 4.4 illustrates the grid topology. We investigate the effect of the updates generated at the center node using the same variables used in the chain topology case. With full flooding scheme, $N(i,k)$ and $R(i)$ are:

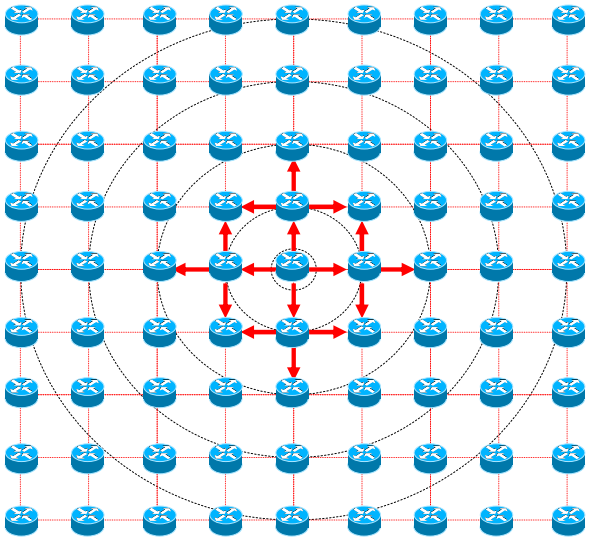


Figure 4.4: A regular grid topology chosen for the analysis of asymptotic efficiency.

$$N_{flooding}(i,k) = \left[\left(8\sum_{l=1}^{i} l\right) - 4i\right] k = 4ki^2, \tag{4.5}$$

$$R_{flooding}(i) = \frac{4ki^2}{k} = \Theta(i^2). \tag{4.6}$$

If the proposed scheme is used, the resulting $N(i,k)$ and $R(i)$ will become:

$$N_{new}(i,k) \;=\; 4k\sum_{l=1}^{i} l\beta^l$$

$$= \frac{4k\beta}{(\beta-1)^2}\left[(\beta i - i - 1)\beta^i + 1\right] \tag{4.7}$$

$$R_{new}(i) = \frac{N_{new}(i,k)}{k} = \Theta(1). \tag{4.8}$$

The results from the grid topology is consistent with the chain topology. Equations (4.6) and (4.8) show that the asymptotic update generation of the new scheme has significantly reduced overhead compared to the full flooding scheme. Note that the result holds for any β that satisfies $0 < \beta < 1$. Therefore, β can be varied depending on the performance requirement.

4.2.2 Parameters

The proposed scheme provides great flexibility because it turns into virtually infinite number of variations by simply changing a few parameters.

The scheme used for analytical result in Section 4.2.1 is a specific case where the number of relayed updates are exponentially reduced as they propagate from the center of the generating node. We define "prescaler" as the entity that performs the functionality of overhead reduction and β as the prime parameter that determines the intensity of reduction. Note that the prescaler need not always be exponential, rather, it can be hyperbolic or linear depending on the design of a certain routing protocol. Thus, the prescaler's "mode" is another parameter that can be used to customize the reduction behavior.

The two parameters of mode and β affect the functionality of the prescaler in a deterministic manner. That is, when a node receives an update, it determines whether or not to relay it based on a predefined schedule (e.g., relay two for every five received updates). On the other hand, the decision need not be hard fixed. Sometimes it is complicated to come up with a clear cut schedule for a given overhead reduction goal. In such a case, one can employ a scheme that makes the relaying decision based on a random function with parameter β. This provides another prescaler parameter that allows one to fine tune the reduction operation.

In many cases, the closest neighbors (e.g., within one or two hop distance) of a node require highly accurate update information. One example is that a node moves away from its neighbor and the neighbor does not realize the lost link until the update arrives detouring the previous link. This problem can be solved if the detouring update travels fast enough up to a certain distance. Therefore, it is useful to allow updates to be flooded to a predefined distance boundary. We define such a boundary as "inner flooding bound". The prescaler comes into play once the update travels beyond the inner flooding bound.

On the other hand, if a prescaler mode is used with a small β for a network with large diameter, the nodes far from the generating node will barely receive an update. Note that if the overhead is exponentially reduced with the distance, the time interval between updates is exponentially increased. Therefore, there has to be a boundary beyond which the prescaler decides to relay all the received updates. We define such a boundary as "outer flooding bound".

We have defined and discussed a few parameters that can be used for customizing the behavior

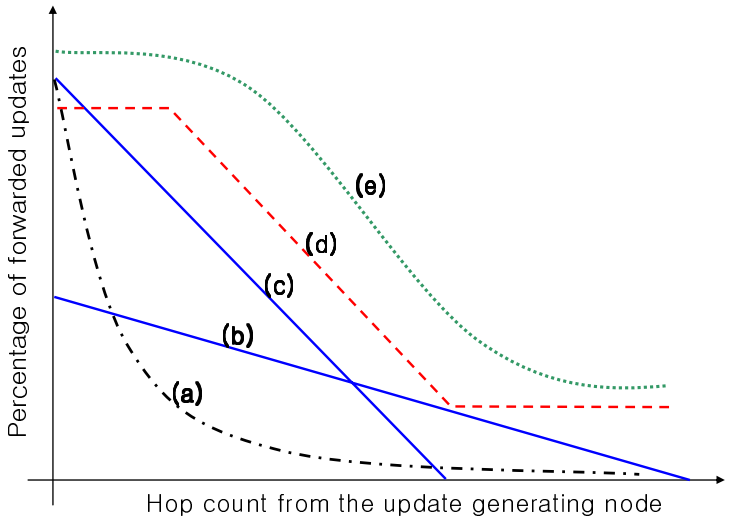


Figure 4.5: Overhead reduction behavior can be customized by configuring prescaler parameters.

of the prescaler. Following summarizes the parameters with brief descriptions:

- **Prescaler mode:** one of exponential, hyperbolic, discrete, linear, or any other overhead reduction scheme.
- **Prescaling intensity ($= \beta$):** defines the reduction intensity with which a prescaler operates.
- **Stateful relay decision:** deterministically relays update based on the state (i.e., reception counter) per destination.
- **Stateless relay decision:** relays update based on a random number generator with parameter β.
- **Inner flooding bound:** defines the distance range within which updates are unconditionally relayed.
- **Outer flooding bound:** defines the distance range beyond which updates are unconditionally relayed.

Figure 4.5 shows some examples of the configuration of these parameters. The x-axis of the graph represents the number of hops between the update generating (i.e., originating) node and other nodes in the network that receives the update. The y-axis represents the percentage of updates a node transmits because of the originating node. Curves (a) and (e) show the overhead reduction trend

obtained by setting the prescaler in exponential or hyperbolic mode with a certain β. In curve (a), the inner flooding bound is one hop and the outer flooding bound is infinity. Curves (b), (c), and (d) show the case of linear prescaler mode with different slopes (determined by β). Curves (d) and (e) show the effect of inner and outer flooding bounds. Smooth inflection in curve (e) indicates stateless (i.e., randomized) relay decision.

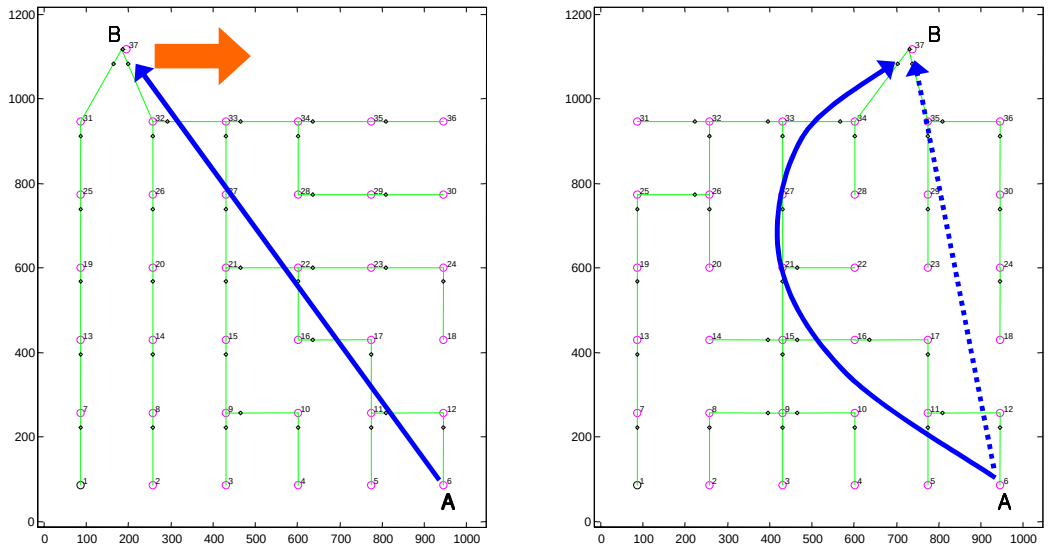
The proposed scheme assumes that all the nodes in the network are initially updated before the topology starts to change with mobility or node failure. In real situation, each node joins the network at different time which invalidates the assumption. This can be solved by setting the prescaler to always relay the initial updates it receives. Thus, the initial update of a node (no matter when it joins the network) will be flooded throughout the network and all the other nodes will receive the update.

4.2.3 Trade-offs

In this section, the trade-off between the overhead efficiency and route optimality is discussed. Because of the reduction in updates, it is possible that some routes become stale or suboptimal. If the update arrival interval at a receiving node is larger than the update change interval at the originating node, the information stored at the receiving node may become more and more incorrect until the correct update arrives. Although the node has incorrect information, the user traffic forwarded by the node may follow the correct path as it travels towards the final destination.

To visualize this phenomenon, a simple distance vector routing protocol with the proposed scheme is implemented in QualNet simulator [85]. We also implemented an animation tool in MATLAB that helps visualizing the dynamic change in topology and node movement based on the trace file generated by the simulator. The resulting route topology and the node position is shown in Fig. 4.6.

For clarity, all the nodes are assumed to be stationary except for node B that moves from left to right as indicated by the block arrow. All the fixed nodes are deployed in a grid to form a 4-regular graph. The route topology before and after the movement are shown in Fig. 4.6-(a) and (b), respectively. To avoid cluttering the figure, only the route vectors for the mobile node are presented. Thus, each solid line attached to every node in the figure is pointing at its next hop neighbor towards the destination node B. Figure 4.6-(a) shows that in the converged steady state, all the route vectors are optimal. The distance of multihop paths from every fixed node to node B is minimal. For example, node A reaches node B in ten hops. In contrast, Fig. 4.6-(b) shows transient suboptimal routes where node B reaches node A



(a) (b)

Figure 4.6: Two topologies showing (a) optimal route paths in steady state convergence, and (b) transient suboptimal route paths due to overhead reduction and mobility.

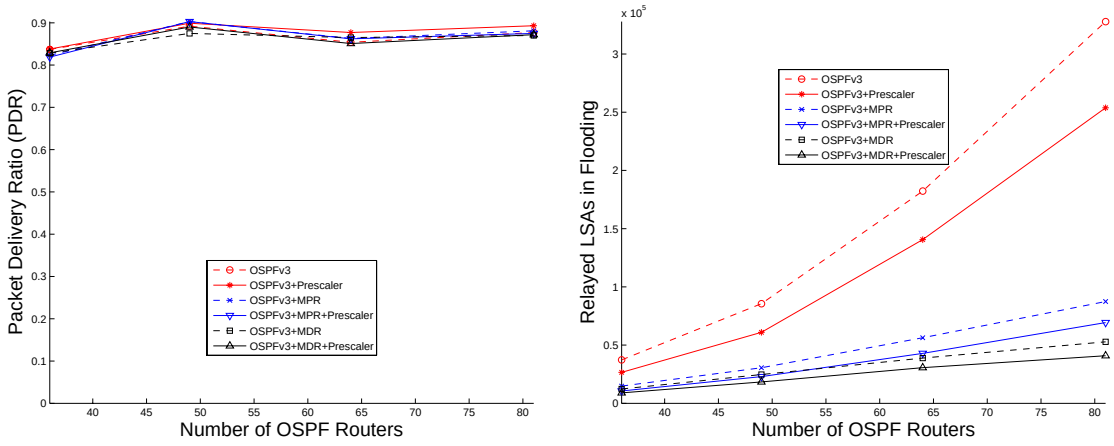
in ten hops (following the curved arrow) while the shortest path should be seven hops (following the dotted arrow).

4.3 Application to OSPF Protocol

This section describes how the proposed overhead reduction scheme can be applied to OSPF implementation.

The operation of OSPF protocol comprises three sub-protocols. First, hello protocol is used to establish or tear down adjacencies when an OSPF router arrives at or disappears from the network. Second, when building adjacencies, synchronization protocol makes sure that two new neighbors commonly share the most up-to-date link state database. Third, flooding protocol ensures that any change in the link state is immediately propagated throughout the network. Although OSPF supports hierarchical topologies with multiple areas, we assume single area scenario because the areas in legacy OSPF are manually configured and no dynamic OSPF area formation scheme is known to the best of authors' knowledge.

Among three sub-protocols, we focus on the link state flooding protocol that produces sig-



(a) (b)

Figure 4.7: The performance of different versions of OSPF under low mobility: (a) packet delivery ratio and (b) number of relayed LSAs for flooding.

nificant overhead in maintaining the topology under MANET environment. Link state update (LSU) packets are used in OSPF flooding, and one LSU packet may contain multiple link state advertisements (LSAs). When an OSPF router receives LSAs in a flooded LSU packet, it first updates its own link state database and then, determines whether or not to relay (i.e., reflood) each LSA to different neighbors. The updates are multicasted, but reflooding is made reliable by scheduling retransmission only for the intended neighbors. The overhead reduction scheme is inserted between these two steps. Although different types of LSAs exist, our scheme deals with only router-LSAs because all the LSAs in MANETs are router-LSAs.

Figure 4.8 shows the pseudo code with four prescaler modes that can be applied to the OSPF protocol. Unlike a distance vector protocol where each update element corresponds to a destination, a flooded update in OSPF is associated with the router that advertised (i.e., originated) the LSA. In addition to prescaler parameters, three pieces of information are required in making reflooding decision: advertising router (AdvRtr), hop distance (Dist) between the LSA receiver and AdvRtr, and state of the advertising router (AdvRtrState). The third variable is used only when the prescaler is in a stateful mode. Note that AdvRtr and Dist are readily available from the received LSA and SPF table, respectively. Thus, the scheme requires no extra control overhead.

When LSAs (in a LSU packet) arrive and if the prescaler is enabled, the procedure “Process-LSA” is called for each LSA. If the LSA is advertised by an unknown router indicating it is the initial

```
AdvRtrState[n] = {0,0,...,0}; // State per advertising router

Process-LSA (Received-LSA, SPF-Table) {
   AdvRtr ← Get-Advertising-Router (Received-LSA);
   If (AdvRtr is new) Then Reflood;
   Dist ← Get-Distance (AdvRtr, SPF-table);
   Else If (Dist ≤ FloodingInner) Then Reflood;
   Else If (Dist ≥ FloodingOuter) Then Reflood;
   Else If (FloodingMode == "Exponential & Stateful") {
      If (AdvRtrState[AdvRtr] == 0) Then Reflood;
      Else AdvRtrState[AdvRtr]←AdvRtrState[AdvRtr]+1)%(1/Beta);
   }
   Else If (FloodingMode == "Exponential & Stateless") {
      If (UniformRNG(0~1) ≤ Beta) Then Reflood;
   }
   Else If (FloodingMode == "Hyperbolic & Stateless") {
      Dist ← Max(0, Dist – FloodingInner);
      If (UniformRNG(0~1) ≤ (Dist/(Dist+1))^Beta) Then Reflood;
   }
   Else If (FloodingMode == "Linear & Stateless") {
      Dist ← Max(1, Dist – FloodingInner);
      If (UniformRNG(0~1) ≤ (Beta×10–Dist)/(Beta×10-Dist+1))
         Then Reflood;
   }
}
```

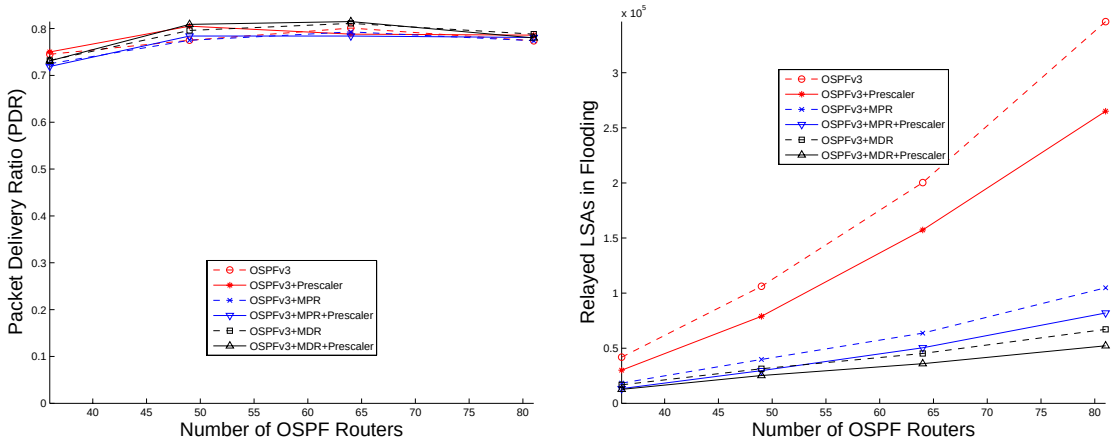
Figure 4.8: Pseudo-code for overhead reduction scheme applied to OSPF.

update, it gets reflooded unconditionally. Otherwise, if the advertiser is within the inner flooding bound (FloodingInner) or beyond the outer flooding bound (FloodingOuter), it gets reflooded. For other cases, a uniform random number generator (UniformRNG) is used for a stateless mode and state variable per advertising router (AdvRtrState) is used for a stateful mode. In hyperbolic and linear modes, generated random numbers are compared against the value calculated from the variables Dist and Beta to produce desired behavior.

4.4 Performance Evaluation

We implemented the prescaler described in Section 4.3 on a real OSPFv3 code ported from the Quagga routing software suite [94]. The experiments were performed using a unique simulator [95] that runs the Quagga OSPFv3 code on GTNetS simulator [96]. Implementing the prescaler on the Quagga software allows the proposed scheme to be quickly ported to and deployed on real OSPF routers. On the other hand, using GTNetS allows simulation experiments for large networks.

Six different versions of OSPFv3 protocol models are used for performance evaluation. Base-



(a) (b)

Figure 4.9: The performance of different versions of OSPF under high mobility: (a) packet delivery ratio and (b) number of relayed LSAs for flooding.

line OSPFv3, OSPFv3 with Multi-Point Relays (MPR) extension [97], and OSPFv3 with MANET Designated Routers (MDR) extension [98] are used with and without prescaler capability. Both MPR and MDR commonly attempt to solve the overhead problem [99] by reducing the number of the OSPF routers participating in flooding, but they differ in the algorithm that selects relays (i.e., flooding nodes). Their additional overhead reduction features such as differential hellos and smart peering are disabled for fair comparison of flooding suppression performance.

At each node, wireless signal range is set to 250 m and IEEE 802.11b is used for the physical and the MAC layers. Initially, nodes are randomly deployed in 1000×1000 m^2 area. Random Waypoint model is used with pause time 40 s. Simulations are run for 600 s for both low and high mobility scenarios. The maximum node velocity is 5 m/s for the low mobility and 10 m/s for the high mobility scenario, respectively. In each case, the number of nodes is incremented from 36 to 81. CBR traffic is offered to probe route correctness. Overall traffic load offered to the network is 5 p/s between randomly chosen pairs of sources and destinations, and packet size is 40 B. As performance metrics, we measure both packet delivery ratio (PDR) and the total number of relayed LSAs.

Figure 4.7-(a) and (b) show the PDR and flooding performance results of six protocols for low mobility and Fig. 4.9-(a) and (b) for high mobility. While maintaining almost the same PDR as shown in Fig. 4.7-(a) and Fig. 4.9-(a), the proposed scheme shows significant reduction in the number of flooded LSAs as shown in Fig. 4.7-(b) and Fig. 4.9-(b). The flooding reduction is around 30 % for the

network of 36 nodes. The amount of reduction grows as the number of nodes increases, but reduction ratio stays between 20 % and 25 % for the large network with 81 nodes. Note that our scheme almost equally affects all the three versions of OSPFv3. This is because MPR and MDR functionalities are independent of our scheme. While their schemes reduce flooding by selecting a subset of nodes that participate in flooding (where updates are still flooded throughout the network), ours reduce flooding by attenuation of flooded updates as they travel far from the source of change.

4.5 Summary

A distributed, efficient and scalable update scheme is proposed to alleviate the routing overhead problem in MANETs. The proposed scheme uses hop count distance to effectively reduce routing updates. The scheme does not require the formation of any hierarchy or externally obtained information. Asymptotically, it enables $O(1)$ overhead growth as the network is scaled. With various parameters that determine the behavior of overhead reduction, the scheme can be customized and applied to different classes of existing ad hoc routing protocols that rely on flooding. The overhead reduction is achieved at the cost of transient route suboptimality which should be considered when applying the scheme to a specific case.

4.6 Acknowledgement

The research presented in this chapter was supported by the Center for Advanced Computing and Communication.

Chapter 5

The Optimum OSPF Area Number for MANETs

This chapter presents the study of a partitioning scheme as a scalability enhancement solution for mobile ad hoc networks (MANETs). The embedded network partitioning capability of the existing Internet routing protocol is explored in theory and experiments. The dynamic topology of MANETs requires relatively large control overhead which impedes the scalability of the network. Recently, adapting a widely-used Internet protocol such as Open Shortest Path First (OSPF) for MANETs has been actively investigated to benefit from its maturity, interoperability, and scalability. To enhance the scalability of OSPF, several extensions have been proposed including flooding backbone, differential hellos, and smart peering. However, most of them assume single OSPF area. We present the surprisingly promising results from the groundbreaking effort to employ the notion of multiple OSPF areas as an efficient means to enhance the scalability of MANETs. A novel analytical model is developed to capture the relationship between the number of areas and the flooding overhead. We use theoretical analysis to show that there exists an optimum area number that maximizes the overhead reduction. The effect of the optimum area number is significant such that the overhead can be reduced to 5 % from the single-area case. The analytical results are verified with detailed simulation experiments. Several candidate dynamic area formation schemes are studies with different mobility models. The results show that the optimum area number has a greater impact on the scalability of the scenarios with higher realism.

5.1 Introduction to OSPF Areas for MANETs

Over the last couple of decades, mobile ad hoc networks (MANETs) have been an area of active research. With the unprecedented advancement in mobile devices and wireless communication technologies, MANETs are rapidly expanding applications in commercial and military environments. MANET is considered especially important for situations with sparse or inexistent infrastructure. Recently, wireless ad hoc networks witnessed the potential to fill the gap for the Internet, landline telephone, and cellular services in disaster-ridden areas [100, 101].

5.1.1 Scalability and Routing Overhead

In MANETs, due to node mobility, the topology of the network dynamically changes at a much higher rate than conventional wired/wireless networks. In addition to the topological instability, the traffic pattern of MANETs is considered to be any-to-any, i.e., any node in the network is uniformly likely to send packets to any other node. This is in clear contrast to wireless sensor or mesh networks where the traffic pattern is typically assumed to be one-to-any. The characteristics of dynamic topology change and any-to-any traffic pattern in MANETs raise the important issue of efficient unicast routing, i.e., how to keep up with rapidly changing topology in a distributed manner so that a correct forwarding decision can be made at each node. Of course, the topological instability can be dealt with by spending more control packets (e.g., more frequently flooded updates). However, considering the limited wireless bandwidth, a good MANET routing scheme should provide overhead efficiency so that it allows the network to scale with the number of nodes.

To enhance the scalability of MANETs by reducing the overhead without compromising the performance (i.e., route correctness), many schemes have been proposed. Fisheye State Routing (FSR) [55] is a link state routing protocol designed for MANETs. As opposed to using conventional full flooding, it attempts to achieve the scalability by allowing more frequent updates for the closer nodes. Optimized Link State Routing (OLSR) [57] is another link state routing scheme for MANETs that reduces the flooding overhead by allowing only a subset of nodes to participate in flooding. Containing the flooding overhead within a certain cluster or a hierarchical boundary is a popular scheme [44, 74]. In on-demand (or reactive) protocols such as AODV [63], route discovery packets are flooded only when necessary. DSR [46] is also an on-demand protocol that uses aggressive caching to reduce unnecessary

flooding. If available, location information can increase the scalability by suppressing the network-wide flooding [49, 50, 64].

5.1.2 Adapting OSPF for MANETs

Recently, adapting a widely-used Internet protocol such as Open Shortest Path First (OSPF) to MANET environment has been actively investigated as an efficient way to enhance the scalability [99]. The motivation for extending the standard OSPF protocol [92] for MANET routing is threefold. First, the OSPF protocol is the de facto standard and the most popular intra autonomous system routing protocol in the Internet. The protocol is mature and its performance is proven over decades of deployment. Futhermore, the latest version of the protocol (OSPFv3 [93]) allows the protocol to work with general addressing schemes including IPv6 addresses. Second, when used in MANETs with proper extensions, OSPF can provide seamless interoperability between the ad hoc and the infrastructured domain (e.g., the Internet). From the view point of users, nodes can easily switch from one domain to the other. Network-wise, the operation at the junction point (e.g., gateway in mesh networks) can be simplified as both domains use the same OSPF algorithm (with possibly different OSPF configuration parameters). Otherwise, redistributing routes between two routing protocols (one optimized for wired Internet and the other for MANETs) will introduce significant amount of complexity which is hard to solve if not impossible (e.g., interpreting and relaying route request packets of AODV or DSR to the Internet and vice versa). Third, legacy OSPF already supports the notion of multiple *areas* that partitions the network into pre-planned segments. This feature has a great potential to enhance the scalability of the protocol when applied to MANETs.

The main functionalities of OSPF protocol can be described in terms of three subprotocols, i.e., hello, synchronization, and flooding protocols as described as follows:

- **Hello protocol:** Establishes or tears down adjacencies with neighboring OSPF routers. This subprotocol relies on periodic OSPF hello packets to discover the new or lost (from hello timeout) neighbors.

- **Synchronization protocol:** When building adjacencies, this subprotocol ensures that two OSPF routers share the most up-to-date link state database with each other.

- **Flooding protocol:** This subprotocol makes sure that any change in the link state (detected by the hello protocol) is immediately propagated throughout the OSPF area.

Each of these subprotocols generates different types of overhead. The existing solutions attempting to reduce such overhead are presented in the following section.

5.1.3 Related Work

While the baseline OSPF provides sophisticated functionalities such that the protocol can be directly used for MANETs with minor extensions like defining new interface type [99], it requires relatively large amount of overhead compared to the protocols dedicated for MANETs. The large overhead is a serious problem because frequent changes of the topology will cause an implosion of flooded link state updates, which takes away user bandwidth and thus, impedes the scalability of the protocol.

The origin of OSPF overheads can be attributed to the three subprotocols and five different types of OSPF packets. Periodic hello packets are generated by the hello protocol. Database description (DBDesc), link state request (LSR), link state update (LSU), and link state acknowledgement (LSAck) packets are used by the synchronization protocol. The flooding protocols also uses the LSU packets to propagate link state advertisements (LSAs).

To achieve overhead efficiency of OSPF, various extensions have been proposed. One of the most commonly explored approaches is to reduce the number of the OSPF routers that participate in the flooding. Both the MANET Designated Routers (MDR) [98] and the Multi-Point Relays [97] extensions choose a connected dominating set (CDS) where only the nodes in the resulting CDS will perform flooding. They use different topology reduction algorithms (MDR is source-independent and MPR is souce-dependent) to form a flooding backbone. The simulation study on the two schemes reported in [95] shows that both schemes reduce a significant amount of overhead without compromising the routing/forwarding performance.

In addition to reducing the number of flooding nodes, reducing the size of each hello packet and formation of adjacencies are proposed. In differential hellos scheme, the size of the hello packet is minimized by including only new information (i.e., newly added or lost neighbors) in the periodic hello transmission. In Smart Peering algorithm [102], the reachability information available in the SPT is used to reduce the number of adjacencies without compromising reachability and routing paths redundancy.

In contrast to the schemes that reduce the overhead by employing flooding backbone or selective peering, the notion of *distance effect* is explored in the overhead reduction scheme proposed in Chapter 4 and [103]. In the distance effect scheme, whether to relay (or reflood) a link state update (LSU) at each node is determined by the hop-count distance the LSU traveled from the originating node (similar to what can be inferred from the TTL field in the IP packet header). As a result, the frequency of reflooding is attenuated as the LSU travels farther from the origination. The simulation results show that the scheme provides significant overhead reduction without compromising the packet delivery performance.

5.1.4 Gap

Although the proposed extensions provide efficient overhead reduction for MANETs, all of them are based on the assumption that there is only a single OSPF area in the network. We build on the fact that the baseline OSPF already supports hierarchical topologies with multiple OSPF areas. With proper extensions to the embedded multiple-area feature of OSPF, we can enhance the scalability of the protocol in a MANET environment. To the best of authors' knowledge, this capability has not yet been explored as a means to reduce the overhead and to enhance the scalability.

Obviously, multiple areas will dramatically reduce the flooding overhead. Flooding will be contained in each area as far as the change (mainly due to mobility) does not affect the route calculation at the nodes in other areas. However, a mobile node may cross the area border and as a result, the event has to be updated in all the areas including the two areas the node just left and entered. With increased number of areas for the given network, the range of intra-area flooding will be decreased. However, on the other hand, the frequency that a mobile node crosses the area border (which will trigger network-wide flooding) increases with the increased number of areas. This trade-off between the number of areas and the total flooding overhead raises a key question about the ideal number of areas for any given network. If the number of areas is resolved, the network can be partitioned into multiple areas to maximize the scalability.

5.1.5 Contribution

This study provides theoretical and experimental results obtained from applying the notion of multiple OSPF areas to MANETs to enhance the scalability. An analytical model is developed to investigate the scalability performance of multiple OSPF areas in MANETs. We perform theoretical analysis based on the model and show that there exists an optimum area number that maximizes the overhead reduction. The theoretical result is verified with detailed simulation experiments. The effect of the optimum area number is significant that for a medium (over 100 nodes) to large (over 1000 nodes) network, the overhead can be reduced to 5 % of the one for the single-area case. We extend our investigation for the area formation and maintenance schemes. Several candidate schemes are investigated with different mobility models. The results show that the optimum area number is again, a critical scalability factor for the scenarios with realistic mobility model and dynamic partitioning of areas.

5.1.6 Organization

The remainder of this chapter is organized as follows. In Section 5.2, the flooding mechanism of OSPF protocol is investigated to establish the analytical model that captures the impact of multiple areas on flooding overhead. Section 5.3 presents the analysis of the relationship between the flooding overhead and node mobility. Section 5.4 shows how the optimum area number of a given network is obtained by combining the results from Section 5.2 and 5.3. The theoretical optimum area number is verified with simulation experiments. Our study on multiple areas is extended in Section 5.5, where the potential area formation schemes are proposed and their performance is studied. Finally, Section 5.6 summarizes the study of the optimum OSPF area number for MANETs.

5.2 Model

In this section, we examine the flooding mechanism of OSPF protocol and establish an analytical model that will be used to evaluate the impact of multiple areas on the flooding overhead.

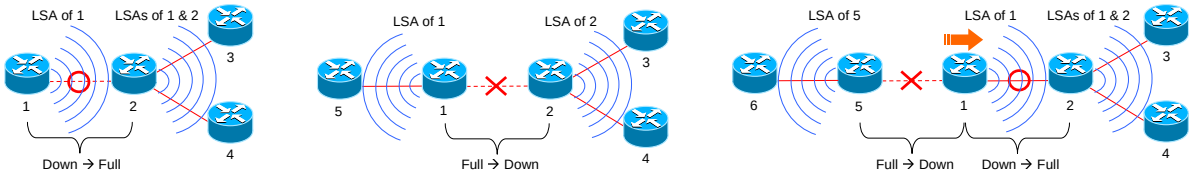
5.2.1 Origin of Flooding

We define some important elements in the modeling of OSPF flooding. First, we define the term “topology”. In general, topology is described as the configuration of a communication network or the method in which nodes of a network are connected by links. In terms of protocol layering stack, topology can be defined at different layers. For wireless networks, it is often defined as physical-layer connectivity among wireless nodes which is again determined by the signal reception range. Sometimes MAC-layer compatibility is considered together with physical-layer. In our study, toplogy is considered mainly at the network layer, and it is defined by OSPF routers and their relationship in terms of OSPF neighbor state. For example, two OSPF routers with different area IDs cannot become neighbors even if they are within the signal range. More precisely, the topology in our study is an undirected graph where vertices represent OSPF routers, and an edge represents fully adjacent neighbor relationship between two routers. All the terms used in graph theory may apply to the OSPF topology under consideration. For convenience, nodes and links will be interchangeably used for vertices and edges, respectively.

To estimate and measure the effect of topology changes on the OSPF flooding overhead, there should be an objective view on the origin of topology changes. Here, transient effects such as temporal fadings are not considered as the source of topology changes. The topology in our study is defined by OSPF routers and their neighbor states, and thus, such transient events at the physical layer will be automatically filtered out by periodically transmitted hello packets and associated time-outs. Three major events that contribute to topology changes can be summarized as:

- when a node comes up and starts transmitting hellos and establishes neighbor relationship through synchronization with adjacent nodes,
- a node goes down and stops transmitting hellos and as a result, existing full adjacency with each neighbor is torn down, and
- a node moves from one place to another and stops receiving hellos from the old neighbors and starts receiving hellos from new nodes, each causing break down and establishment of neighbor relationship with old and new neighbors, respectively.

Any of these events affects the topology by changing the OSPF neighbor state between the node that causes the event and its neighbor. In steady state analysis, where all the nodes converge and



(a) (b) (c)

Figure 5.1: Initiation of flooding due to different events: (a) a node comes up and establishes adjacency, (b) neighbor relationship breaks down between two nodes, and (c) neighbor relationship changes due to mobility.

share a common view on the network, the main contributor to the topology change will be mobility of the nodes. The atomic (i.e., the smallest scale) topology change involves two nodes and one link between them. Therefore, the origin of topology change should be viewed as creation or destruction of a link. However, a neighbor event (either entering or leaving full state) always involves two nodes and it is indeed these two nodes that generate the router-LSAs (link state advertisements) to be flooded.

In our study, mobility causes neighbor state changes, where each neighbor event triggers two nodes to initiate LSA flooding. We assume all the events occur sequentially, i.e., although a moving node leaves the reception range of its neighbor nodes almost at the same time, the loss of the node will be detected at different time due to jitter in hello transmission and associated router-dead time-out.

Figure 5.1 depicts flooding origination caused by different events. In Fig. 5.1-(a), node-1 comes up and establishes adjacency with node-2. Although both nodes-1 and 2 experience neighbor state change, since node-1 is a leaf node that just came up, the LSAs of both nodes are flooded towards node-3 and 4 (of course node-2's LSA reaches node-1 as well). In Fig. 5.1-(b), node-1 and 2 tear down the neighbor relationship after the link is disconnected. This neighbor state change triggers both node-1 and 2 to originate flooding. In Fig. 5.1-(c), node-1 moves from one position to another, tearing down the neighbor relationship with node-5 and establishing with node-2. In this case, two neighbor state changes occur and thus, four LSAs are originated. Each of node-5 and 2 generates one LSA, indicating the loss and discovery of node-1, respectively. Two LSAs are generated by node-1, one for the loss of node-5 and the other for the discovery of node-2. Note that in Fig. 5.1-(a) and (c), node-2's LSA will also reach node-1, but it is not shown to avoid cluttering the illustration.

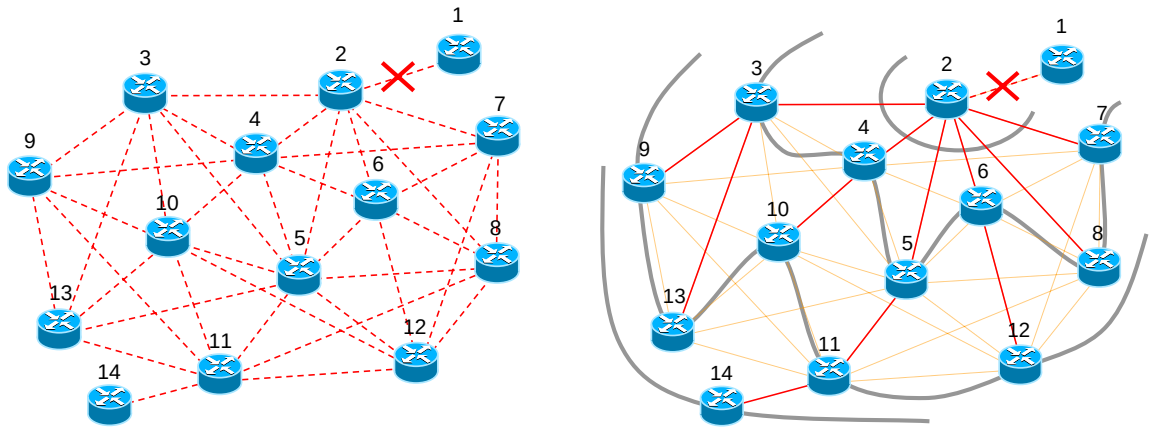


Figure 5.2: How LSAs are flooded in an area: (a) initial topology and (b) spanning-tree propagation of the first LSAs.

5.2.2 Estimation of Flooding per Neighbor State Change

To derive the formula that provides the amount of LSA flooding induced by a neighbor state change, we first define following variables:

- $F \equiv$ total number of flooded LSAs per neighbor state change in the area;
- $n \equiv$ total number of OSPF nodes in the area;
- $L \equiv$ total number of leaf nodes in the area;
- $I \equiv$ total number of nodes that are isolated from the origin of flooding.

Figure 5.2 depicts how LSAs are flooded in an area. The initial topology is shown in Fig. 5.2-(a). In Fig. 5.2-(b), the neighbor state change between node-1 and 2 triggers LSA flooding throughout the area. In Fig. 5.2-(b), red solid lines represent the first LSA to arrive at each node, forming a spanning tree rooted at node-2. The spanning tree can be different depending on the jitter at each node. The LSA originated by node-2 will first reach nodes-3, 4, 5, 6, 7 and 8. Grey curve represents a contour line where nodes have the same hop-count distance from the origin.

Although nodes-9, 10, 11, 12, and 13 appear as the leaves of the spanning tree in Fig. 5.2-(b), their degrees are greater than one as shown in the original topology in Fig. 5.2-(a). These nodes participate in flooding when receiving an LSA because they have neighbors other than the one that sent

the first LSA. In contrast, node-14 is a leaf node that has no other neighbors to relay the received LSA. Thus, it does not participate in flooding. Finally, node-1 is isolated from the origin of flooding and thus, it does not take part in the flooding. Therefore, the total number of LSA-floodings induced by single neighbor state change can be generalized as:

$$F = n - L - I \quad [LSAs]. \tag{5.1}$$

Applying (5.1) to Fig. 5.2 yields $F = 14 - 1 - 1 = 12$.

By choosing a reasonably high enough node density, we can assume that the network graph is connected within the area even after one edge is disconnected. This assumption will also eliminate the existence of a leaf node as shown in Fig. 5.2-(b). Note that if the network is connected after the neighbor state change, two different LSAs will be originated at both ends of the origin edge. Therefore, by assuming that the area is 2-edge-connected, we can rewrite (5.1) as:

$$F = 2 \times n \quad [LSAs]. \tag{5.2}$$

If nodes-1 and 7, and nodes-13 and 14 are connected in Fig. 5.2, the network satisfies the connectivity assumption, resulting in the total flooding as $F = 2 \times 14 = 28$.

Assuming C number of neighbor state changes (i.e., deletion of old neighbors and/or addition of new neighbors) experienced by one mobile node during the time duration of t, flooding rate of the network F_R can be derived from (5.2) as:

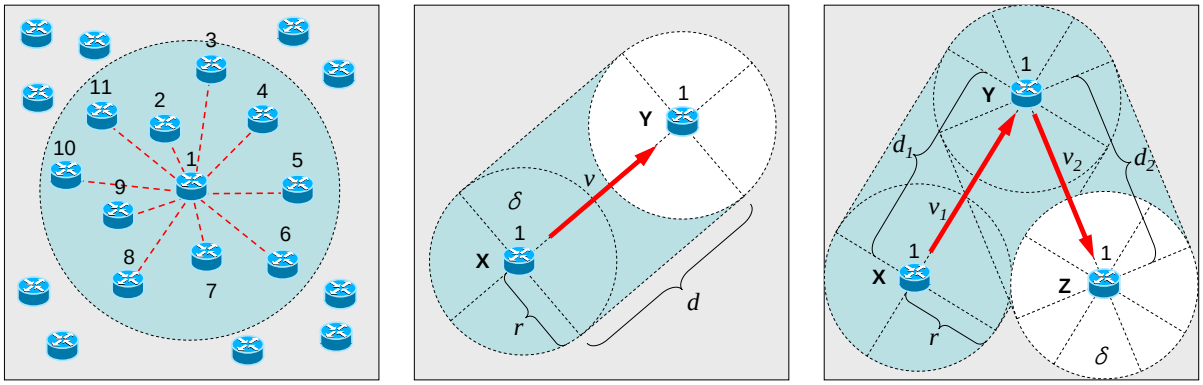
$$F_R = \frac{2nMC}{t} = 2nMR \quad [LSAs/s], \tag{5.3}$$

where M is the number of mobile nodes and R is the rate of neighbor state changes:

$$R = \frac{C}{t} \quad [changes/s]. \tag{5.4}$$

5.2.3 Estimation of Neighbor State Changes

Neighbor state changes are important because they are the origin of flooding as discussed in the previous section. Figure 5.3 illustrates the estimation of neighbor changes. In Fig. 5.3-(a), the relation between node degree (i.e., the number of full state neighbors) and the average node density is depicted. The shaded circle represents the signal reception range of node-1. All the nodes within the



(a) (b) (c)

Figure 5.3: Estimation of neighbor state changes: (a) average node degree and node density, (b) neighbor changes from node mobility, and (c) neighbor changes and change of movement direction.

circle are assumed to establish full adjacency with node-1, and thus, the number of node-1's neighbors is 10. In general, the degree (or neighbor density) of a node n_D is:

$$n_D = A_C\delta - 1 = \pi r^2\delta - 1 \quad [Neighbors/Node], \tag{5.5}$$

where A_C is the area of the signal range and r is the radius of the signal range. The average node density δ is defined as:

$$\delta = \frac{n}{S} = \frac{n}{l^2} \quad [Nodes/m^2], \tag{5.6}$$

where, n is the total number of nodes in the network, S is the total area (including all the OSPF areas if the network has multiple areas), and l is the length of a side of square-shaped network terrain (thus, $S = l^2$). We assume that the nodes are uniformly distributed in the network.

Figure 5.3-(b) shows the estimation of the amount of neighbor changes when node-1 moves from position X to Y. Node-1 moves d [m] at the velocity of v [m/s]. Along with node-1's movement, some new neighbors are added and some are deleted. The total number of newly discovered neighbors by the time node-1 reaches position Y pertains to the shaded area in Fig. 5.3-(b). Likewise, the total number of the deleted neighbors pertains to the same area because the area covered by the range of node-1 is symmetrical. Therefore, the sum of the added and deleted neighbors (i.e., the total number of neighbor state changes) C can be written as:

$$C = 2(A_S\delta - 1) = 4rd\delta - 2, \tag{5.7}$$

where A_S is the shaded area. From this result, (5.3) can be rewritten as:

$$F_R = \frac{2nMC}{t} = \frac{4nM(2rd\delta - 1)}{t} \quad [LSAs/s], \tag{5.8}$$

A leg is defined as a node movement in a direction between two consecutive destination selections in random waypoint mobility model. As we will focus on the flooding overhead per mobile node as a function of the expected distance a node travels at one leg, from (5.8), we define the flooding overhead per distance traveled at each leg $F(d)$ as follows:

$$F(d) = 4n(2rd\delta - 1) \quad [LSAs/leg]. \tag{5.9}$$

Figure 5.3-(c) considers the case where node-1 makes a turn. Node-1 moves from position X to Y, and then Y to Z. If the total distance in Fig. 5.3-(c) is the same as (b) (i.e., $d_1 + d_2 = d$), and the velocity is constant (i.e., $v_1 = v_2 = v$), then the resulting calculation of F_R for Fig. 5.3-(c) is the same as (b). To allow our model to be amenable to analysis, we do not consider the boundary condition where node-1 bounces on the border of the terrain.

5.3 Analysis

In this section, the analysis of the relationship between the flooding overhead and node mobility is presented.

5.3.1 Mobility Model and Expected Traveling Distance

The expected traveling distance of a mobile node is important because the induced flooding overhead will be proportional to the distance. For the analysis, the network is assumed to be partitioned into one or more OSPF areas based on geographical location of each node. Mobility of a node will contribute to flooding overhead in two ways:

- by causing neighbor state changes within the area it travels,
- and by leaving the old area and joining a new area (i.e., crossing the area border).

The former is handled by the intra-area routing and the latter by the inter-area routing of OSPF. We define the number of LSAs created by the former F and the latter F_A where the suffix A indicates the

flooding of LSAs caused by multiple "area" configuration. Note that F is a function of the traveling distance of the mobile node, and F_A is a function of the number of area borders crossed by the node. We define the former independent variable as distance d and and the latter as inter-area distance d_A, where suffix A indicates distance between two "areas".

$$F = f(d) \quad [LSAs], \tag{5.10}$$

$$F_A = g(d_A) \quad [LSAs]. \tag{5.11}$$

In legacy OSPF, summary-LSAs are generated by area border routers (ABRs) for the destinations outside the area. For MANETs, unless a proprietary address assignment scheme is employed, nodes are assumed to have unique addresses which cannot be aggregated at ABRs. Thus, two summary-LSAs are generated by each ABR and flooded by intra-area routers when a node crosses the area border: One LSA advertising the departure of the node from the old area and the other advertising the arrival at the new area. Here, since ABRs of the two areas (old and new) do not flood the summary-LSAs that they originated into their own areas, each summary-LSA gets flooded in $A - 1$ areas, where A is the number of areas in the network. Therefore, the total number of inter-area LSAs induced by a mobile node that crosses area borders d_A times can be written as:

$$F_A = 2d_A \left(\frac{A-1}{A}\right) n = 2d_A \left(1 - \frac{1}{a^2}\right) n \quad [LSAs]. \tag{5.12}$$

The mobility model assumed for the derivation is a special case of the random waypoint model and it can be decribed as follows:

- At each leg, from the position $X(x_1, y_1)$, it chooses a random destination $Y(x_2, y_2)$.
- The destination coordinate variables x_2 and y_2 are selected from a uniform distribution of $(0, l)$.
- The node moves from position X to Y at a constant velocity, and there is no pause time.

Based on the mobility model, we first derive the expected value of intra-area distance d. Figure 5.4-(a) shows the traveled distance by a node in the area of $S = l^2$.

$$E(d) = \int_l \int_l \int_l \int_l d(x_1, x_2, y_1, y_2) p_d dx_1 dy_1 dx_2 dy_2, \tag{5.13}$$

where $d(x_1, x_2, y_1, y_2)$ and p_d are:

$$d(x_1, x_2, y_1, y_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}, \tag{5.14}$$

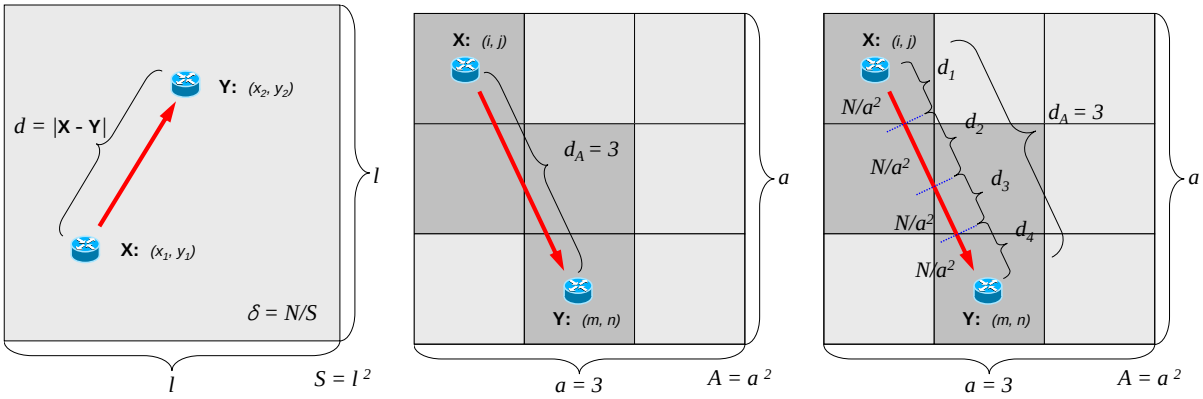


Figure 5.4: Analysis of distance traveled by a mobile node: (a) intra-area distance is measured in the Euclidean space R^2, (b) inter-area distance is measured as the area-count between two end areas, and (c) the effect of both intra- and inter-area distance.

$$p_d = \frac{1}{S^2} = \frac{1}{l^4}. \tag{5.15}$$

Equation (5.13) is a special case of the problem of evaluating the average distance in arbitrary bounded regions. The closed form solution for (5.13) is:

$$E(d) = \alpha l, \tag{5.16}$$

where $\alpha = \frac{2+\sqrt{2}+5\ln(1+\sqrt{2})}{15} \cong 0.521405433$ [104].

Now, we turn to inter-area distance d_A. Figure 5.4-(b) shows the traveled inter-area distance by a node in the network of nine areas (i.e., $A = a^2 = 9$). In this case, the node crosses area borders three times and thus, $d_A = 3$. Since our mobility model assumes uniform distribution in choosing the destination, the expected value of the inter-area distance d_A is:

$$E(d_A) = \sum_{i=1}^{a}\sum_{j=1}^{a}\sum_{m=1}^{a}\sum_{n=1}^{a}\frac{1}{A^2}D_A(X,Y), \tag{5.17}$$

where $X : (i,j)$ and $Y : (m,n)$ indicate the leaving and arriving areas in the form of a matrix. The area at the top left of the terrain is identified as area $(1,1)$ and the one at the bottom right as (a,a). $D_A(X,Y)$ represents the inter-area distance d_A between two specific areas $X : (i,j)$ and $Y : (m,n)$, and it is defined as follows:

$$D_A(X,Y) = |i-m| + |j-n|. \tag{5.18}$$

Table 5.1: Expected value of intra-area distance $E(d_A)$ as a function of the number of areas a.

a	1	2	3	4	5	6	7	8	9	10	11	12
$E(d_A)$	0	1	1.778	2.5	3.2	3.889	4.571	5.25	5.926	6.6	7.273	7.944

By applying (5.18) to (5.17), we can derive $E(d_A)$ as a function of the square root of the number of areas a as:

$$E(d_A) = \sum_{i=1}^{a}\sum_{j=1}^{a}\sum_{m=1}^{a}\sum_{n=1}^{a}\frac{1}{a^4}(\,|i-m|+|j-n|) = \frac{2}{a^2}\sum_{i=1}^{a}\sum_{m=1}^{a}|i-m| = \frac{2(a+1)(a-1)}{3a} \tag{5.19}$$

Table 5.1 shows the resulting $E(d_A)$ for increasing a. For single area (i.e. $a = 1$), mobile nodes will never cross the area border $E(d_A)|_{a=1} = 0$. For four-area configuration (i.e., $a = 2$), it is expected that a mobile node will cross one area border per leg on the average.

Note that for $a > 3$, there exist areas that do not share any area border with the backbone area. In legacy OSPF, all the areas are expected to connected to the backbone area, forming a spoke topology around the backbone. Otherwise, they should be connected to the backbone via virtual links. In our study, instead of proposing a specific scheme to connect remote areas to the backbone area, we assume that this requirement is eliminated. This assumption does not compromise the analysis of impact of multiple areas on flooding overhead.

5.4 The Optimum Area Number

In this section, the optimum number of areas for a given network is obtained by combining all the results from analysis and it is verified using both computation and simulation.

5.4.1 Analytical Results

The total number of flooded LSAs in the network Φ is the sum of both intra-area and inter-area LSAs:

$$\Phi = F + F_A \quad [LSAs]. \tag{5.20}$$

Since (5.9) assumes single area network, n should be replaced by n/a^2 for multiple areas, and d should be partitioned into multiple segments as d_1, ..., d_k, where $k = d_A + 1$. Figure 5.4-(c) presents the relationship among d, k, and d_A. By (5.6) and (5.16), F can be rewritten as follows:

$$\begin{aligned} F &= 4\frac{n}{a^2}(2rd_1\delta - 1) + ... + 4\frac{n}{a^2}(2rd_k\delta - 1) \\ &= 4\frac{n}{a^2}(2rd\delta - 1) = \frac{4n}{a^2}\left(\frac{2\alpha rn}{l} - 1\right) \quad [LSAs]. \end{aligned} \tag{5.21}$$

Turning to the total number of inter-area LSAs, (5.12) can be rewritten after applying the result from (5.19) as follows:

$$F_A = \frac{4(a+1)^2(a-1)^2}{3a^3}n \quad [LSAs]. \tag{5.22}$$

Finally, from (5.21) and (5.22), Φ in (5.20) can be expressed as a function of a and n:

$$\Phi(a,n) = \frac{8\alpha r}{a^2 l}n^2 + \frac{4(a^4 - 2a^2 - 3a + 1)}{3a^3}n \quad [LSAs], \tag{5.23}$$

where α is the constant parameter for the expected traveling distance $E(d)$, l is the length of one side of square terrain, r is the signal reception range, a^2 is the number of the OSPF areas, and n is the total number of nodes in the network.

The result of (5.23) shows that the number of LSAs grows as $O(n^2)$ where n is the total number of nodes. Note that this result represents the induced LSAs by one mobile node. Thus, it holds for the case where the number of mobile nodes is constant. If the number of mobile nodes is given as a certain fraction of the total number of nodes (e.g., 10 % of all the nodes are mobile), the number of induced LSAs will grow as $O(n^3)$.

Figure 5.5 shows the number of flooded LSAs as a function of the number of areas ($A = a^2$) and the number of nodes. In Fig. 5.5-(a), the area of the network is constant ($S = l^2 = 1\ km^2$) while the node degree grows with increasing number of nodes. In Fig. 5.5-(b), the node density is constant ($\delta = 20\ nodes/km^2$) while the network size grows with increasing number of nodes. Signal range r is $250\ m$. For fair comparison, both Fig. 5.5-(a) and (b) are given the same scale for y-axis.

Figure 5.5 shows that the maximum flooding reduction can be achieved by choosing appropriate number of areas for the given network. As the number of areas is increased in both Fig. 5.5-(a) and (b), the number of flooded LSAs decreases due to the “isolating” effect of areas where intra-area

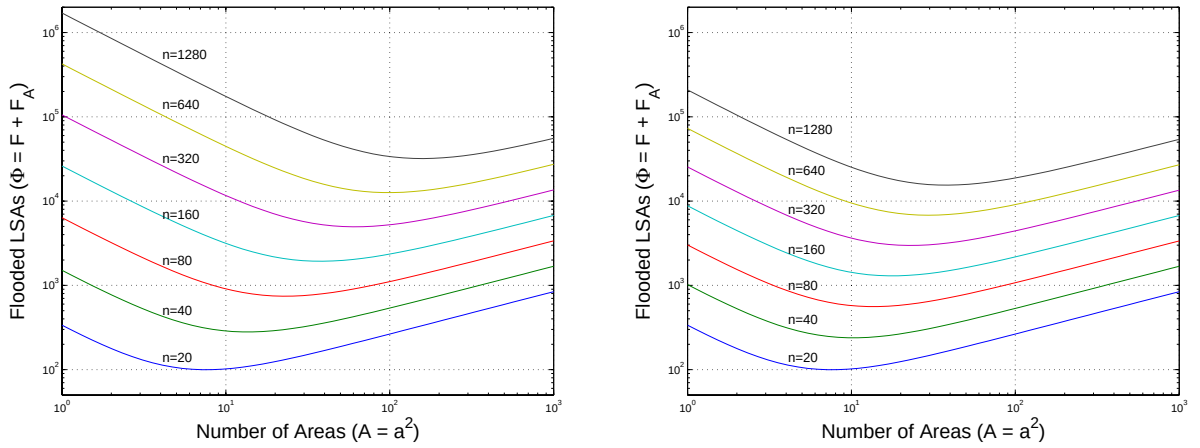


Figure 5.5: The number of flooded LSAs as a function of the number of areas ($A = a^2$) and number of nodes (n): (a) when the area of the network is constant ($S = l^2 = 1\ km^2$) and (b) when the node density is constant ($\delta = 20\ nodes/km^2$).

flooding involves $\frac{n}{A}$ nodes in each area. However, if A is increased beyond optimum point, the inter-area flooding will prevail, and eventually will increase the overall flooding overhead. We define the area number that provides the maximum flooding reduction as "optimum area number". It can be noted from the comparison of Fig. 5.5-(a) and (b) that for the same number of nodes, the optimum area number is larger for denser networks.

The significance of the optimum area number is in its overhead reduction performance. As shown in Fig 5.5-(b), for a network of 1280 nodes with the average node density of 20 $nodes/km^2$, the overhead can be reduced by 90 %. If the node density is scaled with the number of nodes, Fig 5.5-(a) shows even greater overhead reduction (by 97 %).

The optimum area number can be obtained by finding the natural number that is the closest to the solution of:

$$\frac{\partial \Phi(a,n)}{\partial a} = 0 \ , \tag{5.24}$$

which leads to the following equation:

$$\frac{(a^4 + 2a^2 + 6a - 3)}{a} = \beta \ , \tag{5.25}$$

where $\beta = \frac{12\alpha r n}{l}$. For example, if $n = 320$, $r = 250\ m$, and $l = 1000\ m$, the optimum area number can be obtained from (5.25) as $A = 64$ (or $a = 8$). Since the left-hand side of (5.25) monotonically

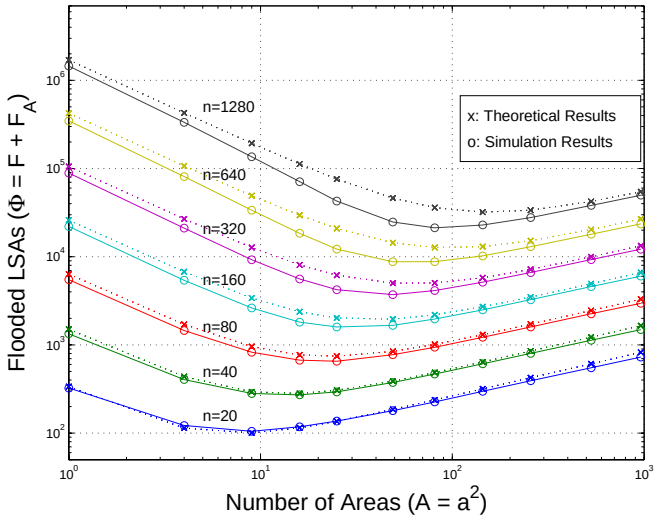


Figure 5.6: The comparison of the results from simulation and computation: the number of flooded LSAs with regard to the increased number of areas ($A = a^2$) and number of nodes (n) when the area of the network is constant ($S = l^2 = 1\ km^2$).

increases for $a > 0$, it can be inferred that the optimum area number grows with the node density and decreases with the dimensions of terrain.

5.4.2 Simulation Results

The theoretical results are verified with detailed simulation experiments. Since there is no simulator that supports multiple OSPF areas to the best of the authors' knowledge, a customized simulator is developed such that the overhead generation events of OSPF protocol is precisely captured in the scenarios with multiple areas and node mobility [105]. Simulation was performed with different number of nodes (from 20 to 1280) and areas. Simulation for each scenario is run for one hour and the result is averaged over 30 repetitions. The terrain is a square with area of $1km^2$ and the mobile node follows random waypoint model with constant velocity of $10m/s$ and zero pause time. The total number of LSAs are measured with regard to the increased number of areas. Figure 5.6 shows the comparison of the theoretical and simulation results. The overall simulation result closely matches the theorectically estimated values. The simulation results are sometimes lower than the theoretical estimations. This is because the model used in the theoretical analysis does not take into account the *boundary effect*. When the mobile node is close to a terrain border or an area border, the minimum number of neighbors will

be a half (when the mobile is close to the middle of the border) or a quarter (when the mobile is close to the corner) of normal situation. This effect becomes more significant for a large number of nodes and multiple areas. If the node density is constant (i.e., network terrain size is fixed) with increased number of nodes and areas, the size of each area will shrink and thus, the A_C value in (5.5) as illustrated in Fig. 5.3 will decrease (because the neighborhood indicated by A_C is not a disk any more). For this reason, the theoretical estimation tend to slightly over-estimate the overhead for a large number of nodes and areas; this can be verified from the curves shown in Fig. 5.6. For medium to small number of nodes, the boundary effect is not so significant.

5.5 Area Formation and Maintenance

In the previous sections, we showed the existence of the optimum area number with which the overhead can be dramatically suppressed. This promising result will naturally lead to the question of how to *automatically and dynamically* establish and maintain such multiple areas. To make the problem amenable to analysis, we assumed a simple mobility model and a regular grid partitioning of areas such that a network in a square terrain is partitioned into a^2 number of square areas. Obviously, such assumptions may not hold in real situations because the shape of the network and its areas will be irregular depending on the topology, and the mobility will be fairly different from the random waypoint model. Therefore, to shed light on the effect of dynamic area formation scheme and mobility model on the overhead reduction performance, several conceivable area formation schemes are investigated under different mobility models.

In real life scenarios, it can be efficient to map the area of a node to its physical location. For example, the whereabouts of a node will be highly correlated to a room, building, campus, district, or city. The topology change can be contained within each area if the area of a node is resolved by its location. One of the strengths of this approach is that it requires minimal computation and protocol complexity in the area formation and maintenance. Although it requires location information at each node, with the advancement in positioning solutions including both GPS and GPS-less technologies, relatively accurate location information will not be that expensive to afford in near future. We call this approach *Geographical Partitioning (GP)*. In many cases, the boundary between the geographically-derived areas can be unclear. The notion of activity points (e.g., center of a downtown square) can be

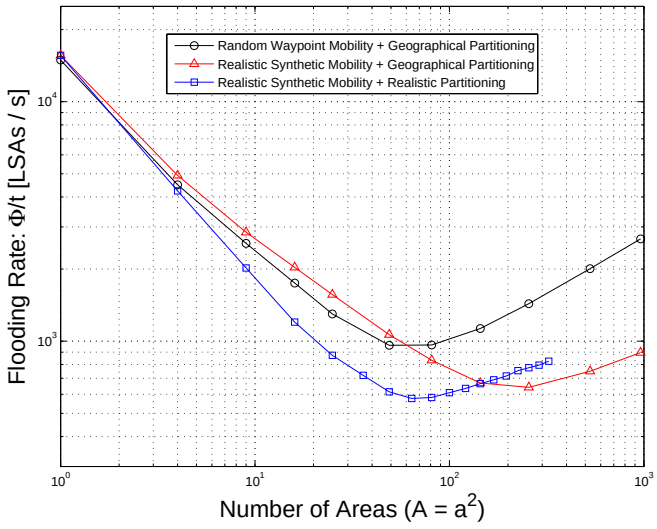


Figure 5.7: Comparison of flooding rates for different mobility models and area partitioning schemes.

more realistic in such a case. A node will resolve its area identity by selecting the closest activity point. By doing this, each node does not have to know the exact shape of the irregular areas. The resulting layout of area assignment will be a Voronoi diagram with the activity points around the center of each polygon cell. We call this scheme *Realistic Partitioning (RP)*. In contrast to the GP and RP schemes, a distributed partitioning algorithm that provides the minimum edge cut and balanced vertices can be realistic for the situation where location information is unavailable [106, 107]. We call this approach *Topological Partitioning (TP)* because the partitioning is done based on the topology of the network. An important drawback is that each node should know the global topology on which it will run the partitioning algorithm to resolve its area.

Figure 5.7 shows the scalability performance of the potential schemes compared through simulation experiments. Simulation parameters are similar to Section 5.4.2 except that we used Realistic Synthetic Mobility (RSM) model in addition to the Random Waypoint (RWP) model. RSM model is produced from the synthetic pedestrian movement data of the Rice University campus. One of the important properties of this model [108, 109] is the spatial-temporal correlation in node mobility. The model provides high level of realism which is lacking in the variations of random waypoint model. The measurement unit is changed from [LSAs/node/leg] to [LSAs/s] because the RSM does not have legs as RWP does. There are 320 mobile nodes in each scenario. For fair comparison, the velocity of RWP

is adjusted such that all the models generate the same overhead for the single-area case. In Fig. 5.7, the result from RWP and GP is similar to what is shown in Fig. 5.6. The combination of RSM and GP outperforms RWP and GP, but RSM requires a large number of areas. This is due to the irregular node distribution in RSM where many grid areas are wasted without containing any node. The most realistic scenario (i.e., RSM and RP) shows the maximum overhead reduction with the smallest number of areas because nodes tend to gather around each activity point that serves as a reference for the area identity. The maximum number of areas in RSM and RP is lower than others due to the limited activity points in the RSM model. Overall, the optimum area number shows greater impact on the scalability for more realistic scenarios.

5.6 Summary

This research shows promising results from the employment of multiple OSPF areas for MANETs. A novel analytical model is established to capture the effect of multiple OSPF areas. The theoretical and simulation results show that there exists an optimum number of areas and that it plays a critical role in the scalability. We also present several potential area formation and maintenance schemes. The findings from the study of the optimum area number can be applied to the existing table-driven protocols - especially to the link state routing protocols including OLSR and OSPF extensions (MPR, MDR, or distance effect) as the optimum area number is independent of such single-area approaches.

5.7 Acknowledgement

The research presented in this chapter was supported by the Center for Advanced Computing and Communication. The author also acknowledges Dr. Stephan J. Eidenbenz from Los Alamos National Laboratory (LANL) who guided the completion of this work. The author carried out the experimental part of this research at LANL as a Summer Intern Student. The author acknowledges Hector D. Flores from Rice University for providing the realistic mobility model used in this research.

Chapter 6

Performance Evaluation of a Scalable and Reliable Sensor Network Routing

Scalable and reliable routing is a critical issue in sensor network deployment. A number of approaches have been proposed for sensor network routing, but sensor field implementation tends to be lacking in the literature. In our study, the problems of scalability and reliability in sensor network routing are addressed through a simple, yet powerful scheme implemented on Mica2 motes running TinyOS along with other, more widely-used routing protocols. Motes are tested in an outdoor sensor field, and detailed experiments are carried out for performance analysis. This chapter presents the implementation details and the results obtained from head-to-head comparison of routing protocols. The proposed protocol delivers 93 % of packets injected at a rate of one packet per second in networks with end to end hop distances of over 10 hops – a result, which significantly improves upon results from the standard TinyOS routing implementation of MINTRoute. The promising results can be explained by the key protocol properties of reliability (via multi-path redundancy), scalability (with efficiently contained flooding), and flexibility (source-tunable per-packet priority) which are achieved without adding significant protocol complexity or resource consumption. These strengths enable the protocol to outperform even sophisticated link estimation based protocols especially in challenging outdoor sensor field environments.

6.1 Introduction to Scalable and Reliable Sensor Network Routing

Recognized as one of the emerging technologies for the 21st century, wireless sensor networks [110, 111] are undergoing rapid evolution with a tremendous amount of research effort in the networking community. Wireless sensor networks are considered to be a special type of wireless ad hoc network [38] with distributed, multi-hop, and self-organizing capabilities. Although sensor networks share many common aspects with generic ad hoc networks, several important constraints in sensor networks introduce a number of additional research challenges in providing viable deployments. In sensor networks, due to dense topologies and large numbers of nodes limited in energy budget, computational resources, storage capacities, and communication bandwidth, networking in sensor networks requires an approach that differs significantly from that of traditional wired/wireless networks.

6.1.1 Reliability and Scalability

This research project addresses one of the most important requirements that any large scale sensor network must meet, i.e. providing reliable and scalable data routing. The information collected at the sensor nodes close to the source of event should be *reliably* communicated to one or more centralized nodes (i.e., a base station or a sink) which may be preprocessed and relayed to the monitoring station over a backhaul. Here, we define reliability as resiliency against changes in network status due to various factors including but not limited to node failures (from battery outage or deadlocking), mobility, unreliable wireless links (due to transient interference or jamming), harsh environments, and malicious nodes. Obviously, the quickest solution to achieve high reliability is to use a sophisticated protocol like the ones proposed for generic ad hoc networks [39, 40, 54, 64]. Table-driven [41–44, 67, 112] and on-demand [46–48, 57, 63] schemes are two widely explored approaches. Some protocols rely on location information [49–52, 113], and some aim at providing QoS [114–117]. Unfortunately, most of these protocols may not be suitable for sensor networks due to the limited bandwidth, energy, and computational capacity of sensor nodes. For a sensor network to be *scalable*, the routing protocol has to be very simple and efficient and work well in most common sensor network scenarios. There is an important trade-off between reliability and scalability in sensor network routing and a good solution should not compromise one for the other.

6.1.2 Related Work

To achieve the two competing goals of scalability and reliability, various sensor network routing schemes have been proposed. Directed diffusion [118] is one of the early single-path sensor network routing scheme that explored the application-level data dissemination to improve the energy efficiency by employing data-centric dissemination, reinforcement-based adaptation of paths, duplicate suppression, and aggregation. In [119], node-disjoint and braided multi-path schemes are proposed to provide energy efficiency and resilience against node failures.

Gossiping has long been recognized as an efficient way of broadcasting or flooding of information in communication networks [120, 121]. The simplicity of its probabilistic transmission behavior inspired its application towards wireless ad hoc networks [122]. In [123], Gossiping is studied as a reliable multicasting scheme for ad hoc networks.

MINTRoute is the standard routing protocol software for TinyOS [124] and thus, it has been widely studied in the literature [125, 126]. In MINTRoute, route selection is mainly based on link-quality estimation, while its older version (i.e., Route software for TinyOS 1.1.0) is mainly based on the hop-count distance information. The performance of MINTRoute is known to depend on the snooping capability at the MAC layer [127].

The main routing protocol in this study is a real implementation of the parametric and probabilistic approaches in [128, 129]. The parametric probabilistic sensor network routing scheme proposed in [128] addresses the goal of reliability and scalability, where flooding is suppressed with a random probability similar to gossiping but the parameter for the non-constant relaying probability is determined per packet in a distributed manner using a few readily-available topological parameters. The scheme is very simple such that it only requires the distance between the source and the sink and the distance a packet traveled. Both of the parameters are available from the beacon originated by the base station and relayed by each sensor node. The simulation result shows that the parametric probabilistic scheme has higher immunity against misinformation under the presence of noisy data, and consumes less resource. Parametric probabilistic approach is further investigated in [129], where the probabilistic multi-path routing is compared with deterministic single-path routing with full-fledged simulation experiments. In a variety of realistic simulation scenarios, parametric probabilistic approach outperformed ad hoc routing protocols such as AODV and Gossiping.

6.1.3 Gap

Although there has been considerable work on reliable and scalable routing schemes for wireless sensor networks, it is hard to find routing solutions that provide simple and flexible parametric capabilities. Furthermore, most of the performance analyses of the proposed schemes are based on simulation or well-controlled indoor testbed experiments. In sensor networks, it is critical to investigate protocol performance in a real sensor field environment. This is because sensor nodes are prone to failures and various adverse factors including wireless communication phenomenon which are hard to predict and capture in simulation. A well-designed scheme tested in simulation may not scale with network size or may not live up to reliability expectations when applied to a deployed network. Following from this observation, the implementation and deployment experience along with the subsequent performance analysis of scenarios in a real sensor field is valuable for the advancement of sensor network technology.

6.1.4 Contribution

In our study, a simple and flexible parametric sensor routing protocols is implemented on real sensor motes along with existing representative protocols. A reasonably large number of motes with the implemented protocols are deployed in real sensor field for the performance analysis. The results from the implementation experience and experimental study of scalability and reliability are presented in detail. The experiments are designed such that the effect of various factors (e.g., traffic load, network size, perturbation in topology, and etc.) are observed. Route correctness (in terms of packet delivery ratio) is used to measure the reliability performance. Resource consumption (bandwidth, energy, and etc.) is measured by the MAC-layer load to the medium to check the scalability performance. The implementation design and parameter selection of each protocol are described in detail. Motes are tested in an outdoor sensor field, and detailed experiments are carried out for the performance analysis. The proposed protocol shows higher reliability and scalability against other widely-used sensor network routing protocols.

6.1.5 Organization

The remainder of this chaper is organized as follows. Section 6.2 describes the main functionality and the key properties of the protocol under study. Section 6.3 presents how we designed and implemented the protocol in TinyOS and on Mica2 motes [130]. In Section 6.4, we present the experiments carried out by deploying the motes on outdoor sensor field. The experimental setup and the analysis of collected results are provided in detail. Finally, Section 6.5 summarizes our study.

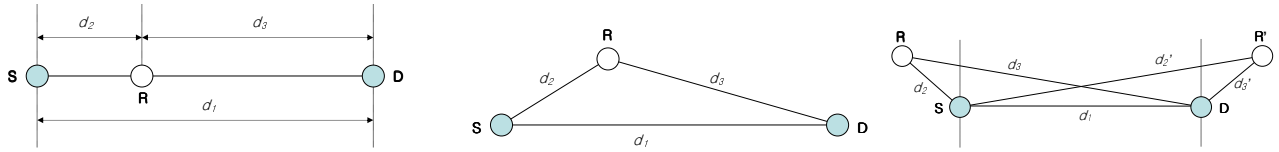
6.2 Protocol Description

In this section, the basic mechanism of the sensor network routing protocol under study is presented. Its expected properties of scalability and reliability are presented with theoretical and computational analysis.

6.2.1 Mechanism

Directed Transmission Routing Protocol (DTRP) is a multi-path proactive routing protocol specially designed for wireless sensor networks to provide improved scalability and reliability. The protocol can be classified as a parametric probabilistic routing protocol. It is parametric because it provides a tunable parameter with which packet-wise reliability (achieved by redundancy) can be flexibly varied in real time by the source node at no additional protocol complexity. It is probabilistic because the tunable parameter directly affects the probability on which each node bases its reflooding decision.

The protocol relies on beacon packets that are periodically originated at the sink node and flooded throughout the network by the sensor nodes. Note that unlike typical single-path distance vector routing protocols proposed for ad hoc wireless networks, DTRP does not use the beacon packet to select the next hop node towards the destination (i.e., the sink). Instead, the beacon is used to provide only the hop-count distance value between the sink and other sensor nodes. Thus, we assume that when a data packet arrives at a node, the packet contains the hop-count distance between the originating source and the destination sink. The node also knows the hop-count distance between itself and the sink which is obtained from a beacon as well. The protocol requires one more information: the hop-count distance a packet traveled from the originating source to reach the node, which is, in general, readily available in



(a) (b) (c)

Figure 6.1: Three cases for the location of the relay node R and the shortest path between the source node S and the destination sink D: (a) R is on the shortest path ($d_1 = d_2 + d_3$), (b) R is off the shortest path ($d_1 < d_2+d_3$), and (c) R is behind S ($d_1 < d_2 + d_3$) or R' is beyond D ($d_1 < d_2' + d_3'$).

the packet header field in the similar form as the TTL field in the IP packet header. The definition of these three values d_1, d_2, and d_3 for the protocol is summarized as follows:

- **d_1:** the shortest-path hop-count distance between the originating source and the destination sink (obtained from beacon packet).
- **d_2:** the hop-count distance a packet traveled before reaching a node (obtained from data packet header).
- **d_3:** the shortest-path hop-count distance between a relaying node the destination sink (obtained from beacon packet).

Using these three pieces of information, a node bases its decision whether or not to reflood (i.e. relay by rebroadcasting locally) the received packet on the transmission probability p^{tx} defined as follows:

$$p^{tx} = e^{k\alpha}, \tag{6.1}$$

where $\alpha = d_1-(d_2+d_3)$ and k is the tunable parameter that determines the reliability and the scalability of the packet transfer. Typically, it is assumed that $k \geq 0$. If $k = 0$, the protocol functions the same as the full flooding scheme. The behavior of the protocol is illustrated in Fig. 6.1, where S indicates the source node, D is the destination sink, and R is a relay node. Figure 6.1 includes three different cases with the relative location of R on the basis of the S-D segment. If R is on the S-D shortest path as shown in Fig. 6.1-(a), it causes d_1=d_2+d_3, $\alpha = 0$, and $p^{tx} = 1$. On the other hand, if R is off the S-D segment as shown in Fig. 6.1-(b), the farther R is from the shortest path, the lower the reflooding probability p^{tx}. The distance between R and S-D segment has the same effect when R is on the left-hand side of S or

on the right-hand side of D as shown in Fig. 6.1-(c). From Fig. 6.1-(b) and (c), the contour line for the equal reflooding probability p^{tx} is the collection of the nodes with the same $d_2 + d_3$ values.

6.2.2 Properties

To clarify the scalability and reliability characteristics of DTRP, the transmission and reception probabilities are investigated. The transmission (or reflooding) probability contour resulting from (6.1) is plotted in Fig. 6.2-(a). The source node S is placed at location (20,50) and the sink D at (80,50) in the network with a square grid topology of 100 by 100 nodes. Note that the probability values in Fig. 6.2-(a) have effect only when a node receives a packet. Therefore, Fig. 6.2-(a) does not represent how far a packet will propagate from the source.

This problem is addressed by introducing the new notion of *reception probability*. Figure 6.2-(d) describes the propagation model we use to estimate the reception probability. Two assumptions are made such that reflooding is done at each time unit and a small jitter is inserted before the transmission to minimize the collision. From these assumptions, the propagation can be modeled in diamond-shape tiers as shown in Fig. 6.2-(d). When the source S originates a packet, its four neighbors (first tier nodes marked 'a') will always receive the packet. Then, the first tier nodes will reflood the packet with the transmission probability shown in equation (6.1) and Fig. 6.2-(a). Among second tier nodes, the ones marked 'b' will receive from only one of their first tier neighbors, the ones marked 'c' will receive from two of them, and the one marked 'a' will always receive because it is on the S-D shortest path. Following this logic, the reception probability of the j^{th} node in $(i+1)^{th}$ tier $p^{rx}_{(i+1,j)}$ can be formulated as follows:

$$p^{rx}_{(i+1,j)} = \min(\sum_{\forall_j : j \leftrightarrow k} p^{rx}_{(i,k)} \cdot p^{tx}_{(i,k)},\ 1\ ), \tag{6.2}$$

where $p^{rx}_{(i,k)}$ and $p^{tx}_{(i,k)}$ are the reception and transmission probabilities of the k^{th} node in the i^{th} tier, respectively. The notation '$j \leftrightarrow k$' indicates adjacency. Now that we know the transmission (p^{tx}) and the reception (p^{rx}) probabilities, we can obtain the *real* transmission probability (q^{tx}) from $q^{tx} = p^{rx} \cdot p^{tx}$.

Figure 6.2-(b) shows the properties of reliability and scalability of DTRP. Although the transmission probability in Fig. 6.2-(a) tends to spread out, the actual packet propagation shown in Fig. 6.2-(b) is efficiently confined to a limited distance from the S-D path, which indicates DTRP has the de-

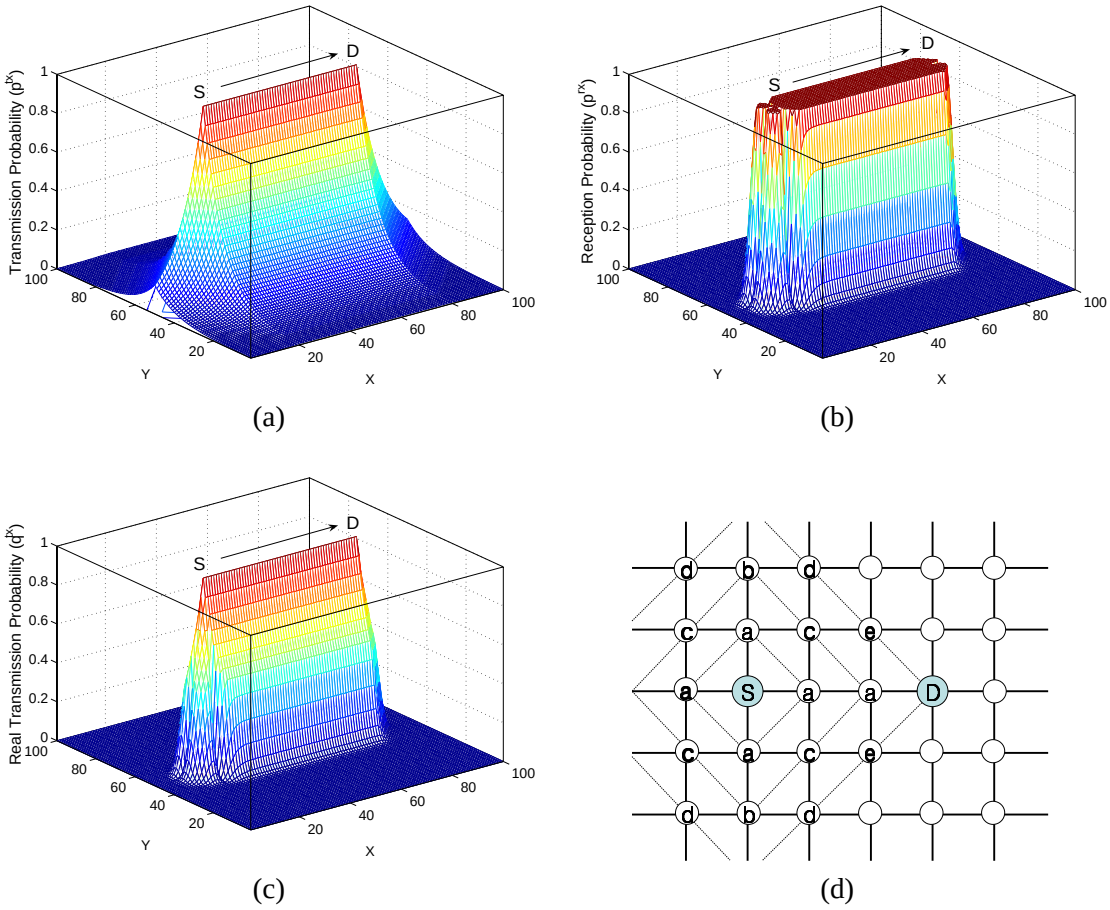


Figure 6.2: Scalability characteristics of Directed Transmission Routing Protocol (DTRP) for a source node S and the destination sink node D: (a) packet transmission probability, (b) packet reception probability, (c) the real packet transmission probability, and (d) the propagation model used for computing the reception probability of a packet that is originated by the source node S and destined for the sink node D.

Figure 6.3: Pictures from the field deployment: (a) overall network, (b) Mica2 motes, (c) Stargate monitor, and (d) tablet PC for data collection.

sirable scalability property. On the other hand, Figure 6.2-(b) shows multiple redundant paths running in parallel with the S-D shortest path, which provides high reliability against topological perturbation. It should be noted that DTRP provides very flexible *source-tunable* reliability and scalability. When a source node originates a packet, it can assign different k value (as shown in (6.1)) to the packet depending on the reliability and scalability requirements (e.g., sensed data of an explosion or an earthquake should be given higher priority than that of a by-passing vehicle).

To summarize, DTRP provides reliability (via multi-path redundancy), scalability (as the redundancy does not spread out to wide area), and per-packet source-tunable capability (which is hard to find in the existing protocols) at minimal protocol complexity and resource consumption (only simple beacon is sufficient).

6.3 Implementation on Sensor Motes

In this section, we describe the hardware, software, and protocol implementation. Pictures in Fig. 6.3 are taken from one of our field experiments.

6.3.1 Hardware: Mica2 Motes

Our mote testbed consists of 15 (small network scenario) to 30 (large network scenario) Crossbow MPR 400CB Mica2 motes covering about 6 and 11 network hops corner to corner, respectively. Each mote has an Atmel ATmega128L low-power microcontroller running TinyOS, with 128k of Flash, 4k of RAM, and a CC1000 Radio running at 38.4 Kbaud operating on channel 0 in the 916 MHz band (at 903 MHz).

The motes are enclosed in specially fabricated weather-resistant hard plastic boxes. The boxes are fitted with external quarter-wave rubber antennas equipped with conductive ground planes to lower the angle of antenna radiation for better performance. Mica2 motes are running TinyOS 1.1.15. They use the B-MAC CSMA based MAC layer, standard in TinyOS in releases of TinyOS 1.x 1.1.3 and later.

6.3.2 Software: TinyOS and Protocols

Directed Transmission

Development time for robust, scalable code in TinyOS must take its learning curve [131] into account. In-depth knowledge of all TinyOS components affected by the change is necessary to predict which small changes will require additional changes in other modules to ensure system reliability. To say the least, the development time for taking DTRP from working in the standard TinyOS Simulator (TOSSIM) to working in a large, heavily loaded network was much longer than expected. The completed application code is less than 500 lines long which, while not including changes made to the TinyOS kernel, illustrates the difficulty of having a working implementation of even a small application. The TinyOS SimpleCmd example code was modified to run with DTRP. Any mote generating a data packet must put its own distance (d_1) as a field in the packet. This value must stay with the packet for all transmissions. The hop count, d_2 and the k parameter (optionally) are stored in the packet. Also a packet sequence number is used to determine network statistics.

Packet variables have the advantage of being selected at the source and the disadvantage of decreasing the available payload size. If payload size is an issue, the k parameter can be a constant within the network. However, an application may wish to decrease this value for a critical packet to ensure its delivery, requiring it to be a packet variable. In the same vein, a source mote could artificially increase its distance from the destination, d_1, in a critical packet to effectively increase its packets' time-to-live and probability of delivery in addition to the total network strain for that packet. This flexibility makes DTRP adaptable to a variety of different applications and environmental conditions.

To make a relay decision, a mote must only know its current distance from the destination, d_3. With knowledge of d_3 and packet variables d_1 and d_2, nodes have all the information required to decide whether to relay the packet, broadcasting it to all nodes in range. Also, all nodes maintain a table of the 4 previously heard source, sequence number pairs. If a received packet matches a packet that has

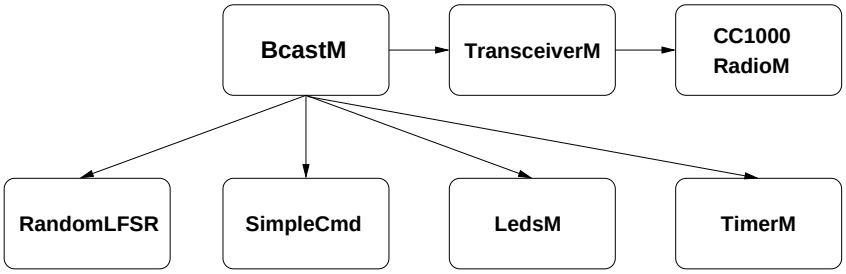


Figure 6.4: The hierarchy of key TinyOS modules for DTRP and Gossiping. Commands travel in the direction of the arrows down the hierarchy, and events travel in the opposite direction.

previously been heard, it it not relayed to avoid network overload. We choose a k parameter of one for all packets generated by all nodes to lower the relaying probabilities to e^{-1} if a packet is one hop off of the shortest path and to e^{-2} if a packet is two hops off of the shortest path. This value is a baseline value.

Disseminating the hop distance d_2 is the responsibility of the destination. In our experimental implementation, we broadcast a d_2-beacon packet from the destination (i.e., the sink) every 10 seconds. Every node that hears a beacon packet checks to see if it has heard a beacon with the same sequence number before. If it has not, it updates its distance from the destination, d_2, and re-broadcasts to all nodes in reception range. An averaging window of previous beacons could be kept in stable network scenarios, however we chose to leave this implementation for future work. In the case it has heard this beacon previously, it disregards it with an exception if the beacon has a shorter hop distance than all previously received copies. In this case, it updates its d_2 (lowering the value) and re-broadcasts the packet. This same method can easily be expanded to multiple destinations, where all motes in the network must keep track of their distance to each destination which has sent out a beacon.

As shown in Fig. 6.4, a message is generated when the Timer module sends an event to the Bcast module. The Bcast Timer.fired() event handler generates a packet and fills the packet's fields before sending a command along with the packet to the Transceiver module. For debugging purposes, the Bcast module then toggles a LED by sending a command to the LED module. After the Transceiver module has the packet, it places it in a queue and sends a command to the CC1000 radio module to send a packet. When the CC1000 radio module determines that it is ready to send the packet, it sends an event back to the Transceiver module requesting the packet, and the packet is returned to the CC1000 radio module and sent over the wireless medium.

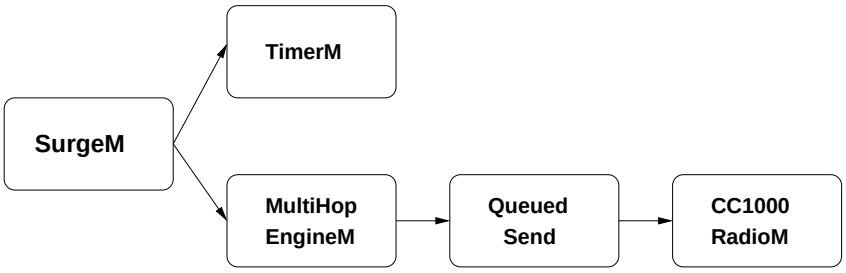


Figure 6.5: The hierarchy of key TinyOS modules for MINTRoute.

When a message is received, it is passed from the CC1000 radio module to a received message buffer in the Transceiver module. From there, it is passed to the Bcast module. In DTRP, a forwarding probability value is calculated based on mote-local information and/or packet variables. This value is checked against a random number (0,1) generated by the RandomLFSR module to determine whether or not the packet will be forwarded. If it is determined that the packet will be forwarded, the packet is sent over the ProcessCmd interface to the SimpleCmd module, which decodes the command (`led_on`, `led_off`) and toggles the LED. Finally, the packet is returned to the Bcast module and sent to the CC1000 radio module via the Transceiver module for wireless transmission. The full code is available from the authors upon request.

MINTRoute

MINTRoute, the TinyOS standard multi-hop routing protocol, is a single-path protocol, which uses point to point transmissions to move packets through the network, relying heavily on link estimates and parent selection. MINTRoute was tested using a modified TinyOS example Surge program. The Surge program (Fig. 6.5) is designed to collect sensor readings at any of the network nodes and relay those readings to a single base station. We modified Surge to generate dummy data packets at selected source nodes. MINTRoute has several parameters that can be set based on the expected traffic in the network. To put MINTRoute on a level playing field with DTRP, the data to route ratio was set to 10. This effectively has each node perform route discovery once for every 10 data packets generated by the node. With each generating source node inserting packets into the network at one packet per second, this roughly equates to the DTRP implementation of a beacon signal being sent through the network every 10 seconds. In addition, in order to track packets, the ID of the source node is inserted into the

data field of each packet.

A message is generated when the Timer sends a fired event to the Timer.fired() event handler in SurgeM. From here, the packet travels to MultiHopEngineM, which abstracts all multi-hop functions away from the application. MultiHopEngineM handles link quality estimation and parent selection by sending control packets through the QueuedSend module to the CC1000 radio module. Control packets are received through this same chain as well, giving the MultiHopEngineM all necessary information to relay packets from the application and from the network towards the destination at the base node.

Gossiping

Gossip was tested using a slightly modified version of DTRP. Beaconing from the destination and beacon-dependent forwarding decision calculations at each routing node are removed, and a decision to forward a received packet and broadcast it to all nodes in transmission range is based on a single forwarding probability value. This value is a packet variable, and can be adjusted by the packet source node depending on the importance of the packet data.

In our experiments, the Gossip forwarding probability is set to 0.7 to reflect the value found to work best in terms of maximizing PDR and minimizing load in simulation work [128, 129]. As with DTRP, a table filled with pairs of source IDs and packet sequence numbers of the four most recently received packets are kept at each node. If the incoming ID matches a packet in the table, the packet is not forwarded.

Gossip executes nearly identically to DTRP (see Fig. 6.4). The only major difference lies in determining whether or not to forward a packet. Here, a forwarding probability value is not calculated, but is instead in the packet already or is available as a local mote variable (0.7 in our experiments). This number is checked against the random number (0,1) generated by RandomLFSR to determine if the packet will be forwarded.

6.3.3 Monitoring System: Stargates

In order to extract simulation-comparable Rx/Tx detail from our testbed without burdening our network nodes with additional processing and storage requirements, we choose to passively monitor experiments with a separate network of higher order wireless nodes. This network is comprised of

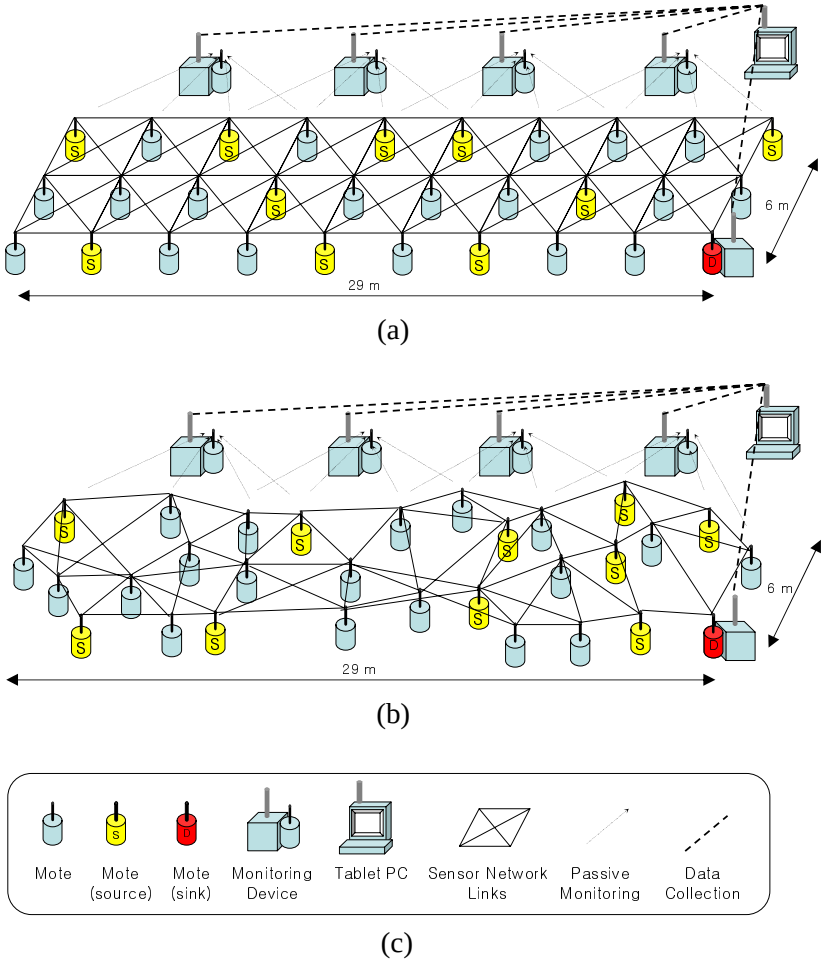


Figure 6.6: The environment setup for the outdoor sensor field experiments: (a) grid topology, (b) random topology, and (c) devices and links.

Crossbow Stargates with an Intel 400MHz RISC processor running Embedded Linux. Each Stargate node is powered by a 4200 mAh Ni-MH rechargeable battery pack. The Stargates have access to a 802.11 wireless card through a Compact Flash port in addition to a Mica2 mote connector on the main SPB400CB Processor Board. A mote running the TOSBase standard base station code is connected to this mote connector, and forwards all packets decoded by the radio to the Stargate. From there, the packets are forwarded over 802.11 to a Tablet PC, which compiles often redundant data from multiple Stargates into a file of unique and ordered packet transmissions. See Fig. 6.6 for an illustration.

Coverage is not perfect, but fell between 94 % and 98 % in tests with each Stargate assigned to monitor a unique set of about 4 neighboring nodes. Since we are only concerned with relative (to other

protocols and scenarios) load and not with measuring exact network load, this coverage is sufficient for comparison. This density of coverage equates to 4 Stargates for a 15 node network and 7-8 Stargates for a 30 node network. The Stargates are placed in the same locations for all experiments. The locations are chosen such that they are evenly spread throughout the network and provide near full coverage for grid and random topologies.

6.4 Performance Evaluation

In this section, we describe the environment for performance evaluation and provide the results and analyses for all experimental scenarios.

6.4.1 Setup

All experiments are run outdoors at the bottom of Los Alamos Canyon in a relatively quiet RF environment. Experiments are run for 10 minutes. We ran all experiments early in the morning to avoid the heat related node failures we encountered in preliminary tests. In each experiment, motes are tuned to 10 μW transmit power, giving each node a tested communication radius of 11-13 feet in their enclosures. We equipped all motes with fresh batteries before the first experiment, and voltage tested each at the conclusion of each set of experiments to ensure batteries are nearly at full charge. We discarded and replaced batteries reading less than 1.45V.

We chose two topologies, identical for each routing protocol, for the small and large network scenarios. A grid topology, either 3x5 or 3x10 for small and large scenarios respectively, and a random topology. The grid topology is 3 nodes wide due to flat-ground width constraints at our test site. However, this creates a network with 9-11 hops end-to-end which allows us to test the reliability of each routing protocol over long hop distances. It has been suggested that network reliability drops significantly as network size increases beyond 5-6 hops [132]. For a random topology, we split the grid of 15 or 30 nodes into 4 or 8 cells, respectively, and randomly place 4 nodes in each cell. We randomly disregard 1 or 2 of those random placements to come to a final count of 15 or 30 nodes depending on the desired network size. This cell-based placement is an effort to keep network connectivity, while introducing bottlenecks and thinly covered areas at random. For fairness, we use the same random topology for all tests.

Source nodes are chosen such that sources are the in the same locations for all protocols in a given topology setup of a given size. Sources are chosen randomly in each of these scenarios from the total set of nodes available with the following exception: Node with ID 1 (the node furthest from the destination) is chosen as the first source in all topology setups of all sizes. After Node 1 is run as a source, 2 nodes are converted from routing to source nodes in each iteration until our experimental limit of 9 sources are being used concurrently, generating 9 pps (packets per second) in the network. This was chosen as an experimental limit following experimental evidence suggesting 16-18 pps is the maximum limit for a 20 node Mica2 network running B-Mac. With our topology, a relatively unconnected destination, and a larger network, 9 pps is enough to induce the significant congestion necessary for high traffic routing conclusions. For example, in the small network scenario Node 1 is chosen as the first node and nodes with ID 5 and 13 are randomly selected as the next two nodes. The 3 source scenario will be run with sources as Nodes 1, 5, and 13 for all routing protocols (Gossiping, DTRP, and MINTRoute) and both topologies (grid and random) to be sure to run all protocols in as identical conditions as possible.

All sources inject packets into the network at the constant rate of one pps. Load on the network is increased by adding more sources. The decision was made to run, for example, three unique sources at one pps instead of one source at three pps in an attempt to look at the performance of the network as a whole rather than the performance of a subset of total network nodes connecting a single source to the destination. In this way, the PDR numbers gathered represent a delivery percentage more representative of the network as a whole.

6.4.2 Results and Analysis

Overview

We choose to focus on two specific network measures: PDR and total network load. PDR, in this case, is the percentage of the summation of all unique injected packets from all destinations which are received by the destination. For the PDR graphs, the X-axis is the total number of sources, which is directly proportional to the total injection load per second, and the Y-axis is the PDR. In the total network load graphs, the X-axis is also total number of sources, and the Y-axis is the total network load (representing all control and data packets sent in the network) in pps.

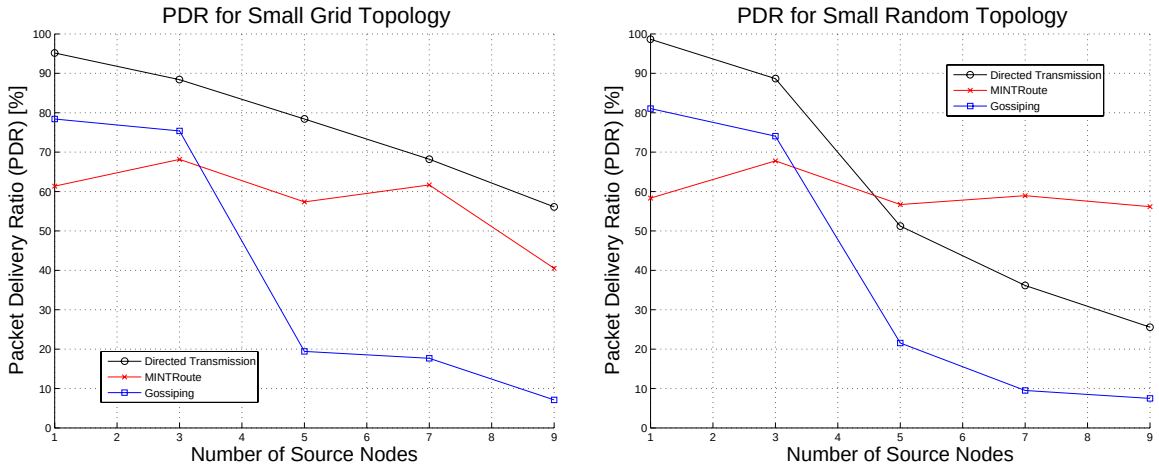


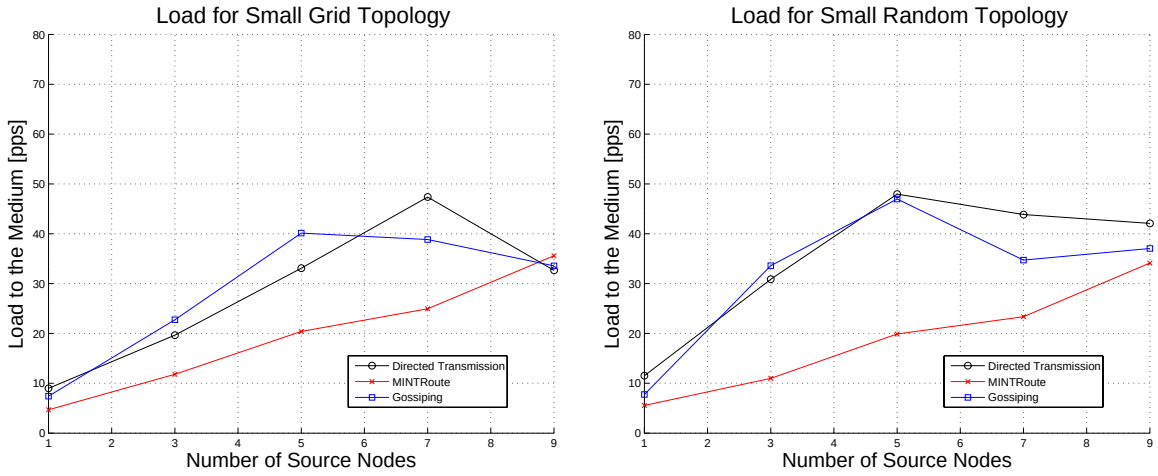
Figure 6.7: Packet delivery ratio (PDR) for the small networks with (a) grid and (b) random topologies.

Small Network Scenario

Figure 6.7 shows the PDR for small grid and small random topologies. DTRP decidedly outperforms MINTRoute and Gossiping in all low load (1-3 sources) scenarios. It is also interesting to note that Gossiping, which requires no control packets, outperforms MINTRoute for these low load scenarios as well. Since MINTRoute uses passive link estimation for parent selection and routing paths, this is somewhat expected. With a lower amount of network traffic, there are not as many packets being sent, and link quality estimates will be less accurate and more dated than in a high traffic scenario.

DTRP delivers the highest percentage of packets in all grid scenarios, delivering 96 % of packets with one source (1 pps) and 56 % of packets with nine sources (9 pps). Gossiping performs fairly well with low load, but effectively breaks down with loads of greater than 3 pps. MINTRoute performs almost as well in high load scenarios as it does in low load scenarios due to the tradeoff between good link quality estimation (good parent selection) and contention for the wireless medium.

In the random topology, DTRP and Gossiping still outperform MINTRoute in low load scenarios, with DTRP delivering 98 % of packets in the single source scenario. In high load scenarios (7-9 sources), MINTRoute delivers more packets than DTRP and Gossiping. This is due to an induced bottleneck in the random topology between some of the more distant source nodes and the destination. MINTRoute appears to overcome this obstacle. It is interesting to see that Gossiping appears to follow the same PDR trends, most notably in the random topology, as DTRP. This follows directly from the



(a) (b)

Figure 6.8: Aggregate load to the medium for the small networks with (a) grid and (b) random topologies.

fact that DTRP is effectively Gossiping, only DTRP will not diffuse packets away from the source. Gossiping, therefore, will have nearly identical behavior to DTRP with a single source at the maximum distance from the destination. Both protocols will effectively diffuse the packet towards the source. However, a breakdown occurs with higher injection load due to unnecessary load stemming from packet propagation away from the destination.

Figure 6.8 shows the total network load for the small grid and random topologies. DTRP and Gossiping have notably higher load numbers than MINTRoute in all but the highest load grid scenario. This is not surprising, as our multi-path protocols depend on the limited branching factor in packet broadcast diffusion to achieve higher PDR. MINTRoute has a total load which increases nearly linearly with offered load. DTRP and Gossiping also increase linearly before reaching a maximum load point and decreasing thereafter.

Interestingly, load appears to meet a network area (since both random and grid topologies cover roughly the same area) limit of 48 pps at an offered load of 5 or 7 pps, with the total load decreasing at higher offered load. This load decrease corresponds to a PDR fall-off at the same breaking point in all Gossiping scenarios and in the DTRP scenarios with a random topology. The DTRP scenario with a grid topology has a total load which peaks at a higher offered load and does not show the characteristic breakdown at the maximum total load point, suggesting the grid topology is used more efficiently and

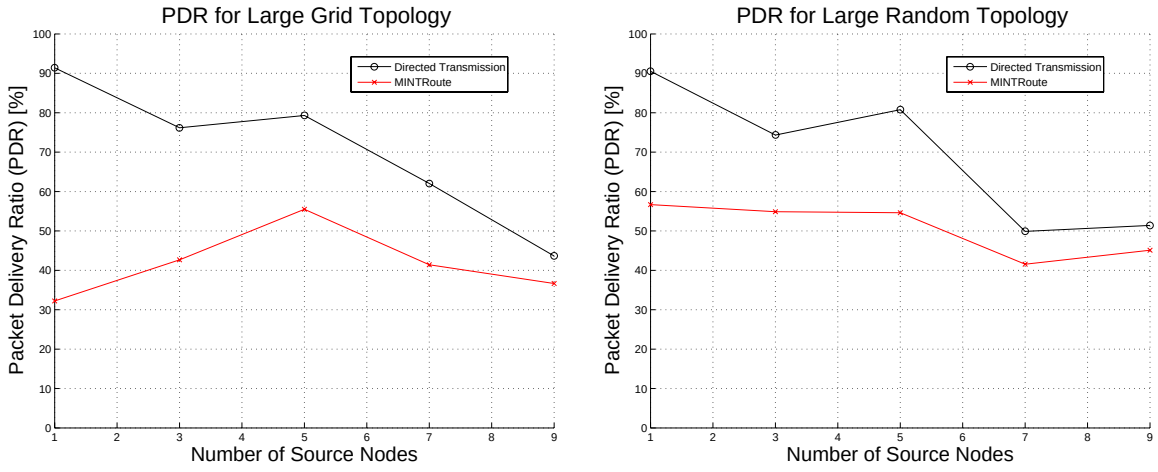


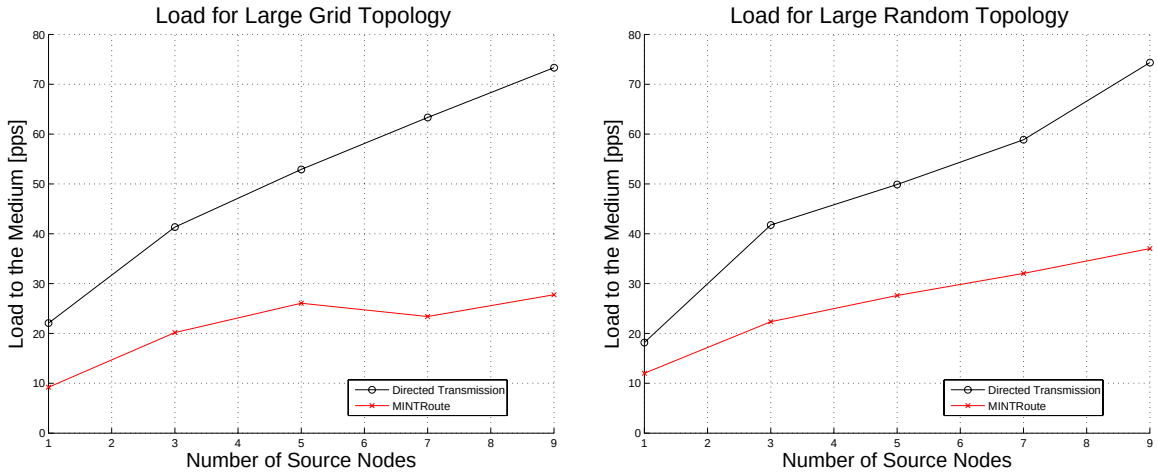
Figure 6.9: Packet delivery ratio (PDR) for the large networks with (a) grid and (b) random topologies.

effectively by DTRP in high load scenarios.

Large Network Scenario

After analyzing results in the small network scenario, we narrowed our experimental focus to two protocols: DTRP and MINTRoute. While Gossiping has the advantage of being a protocol void of control packets, it does not offer a significantly lower load profile, a better high traffic handling capability, or a better level of low traffic efficiency than DTRP. Simply, it is an inferior derivative of DTRP which, considering the experimental overhead involved, can be removed from contention and omitted from larger scale tests. We will consider only DTRP and MINTRoute in the large network scenario.

In the large network (Fig. 6.9) , much of the same results carry over from the small network scenario. DTRP has a higher PDR than MINTRoute in all load and topology scenarios. This is due to the selection of a fresh set of randomly chosen source points. The set selected for the small network, random topology scenario is simply a bad set for DTRP in high load conditions. DTRP delivers over 91 % of packets in both single source scenarios, with packets arriving in 9-11 hops. The network still manages to deliver 80 % of all source generated packets in both topologies at 5 pps. MINTRoute again maintains a relatively steady PDR for all offered load scenarios, with PDR values ranging from 31 % to 56 %.

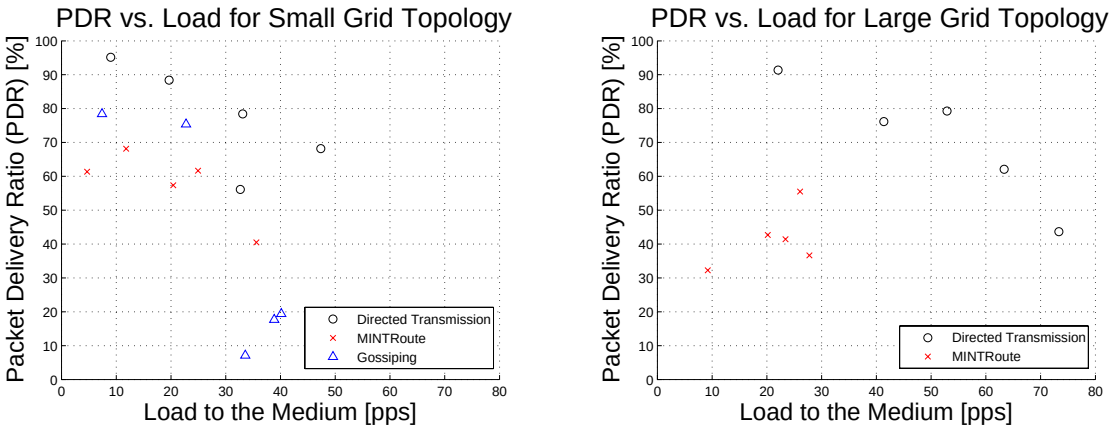


(a) (b)

Figure 6.10: Aggregate load to the medium for the large networks with (a) grid and (b) random topologies.

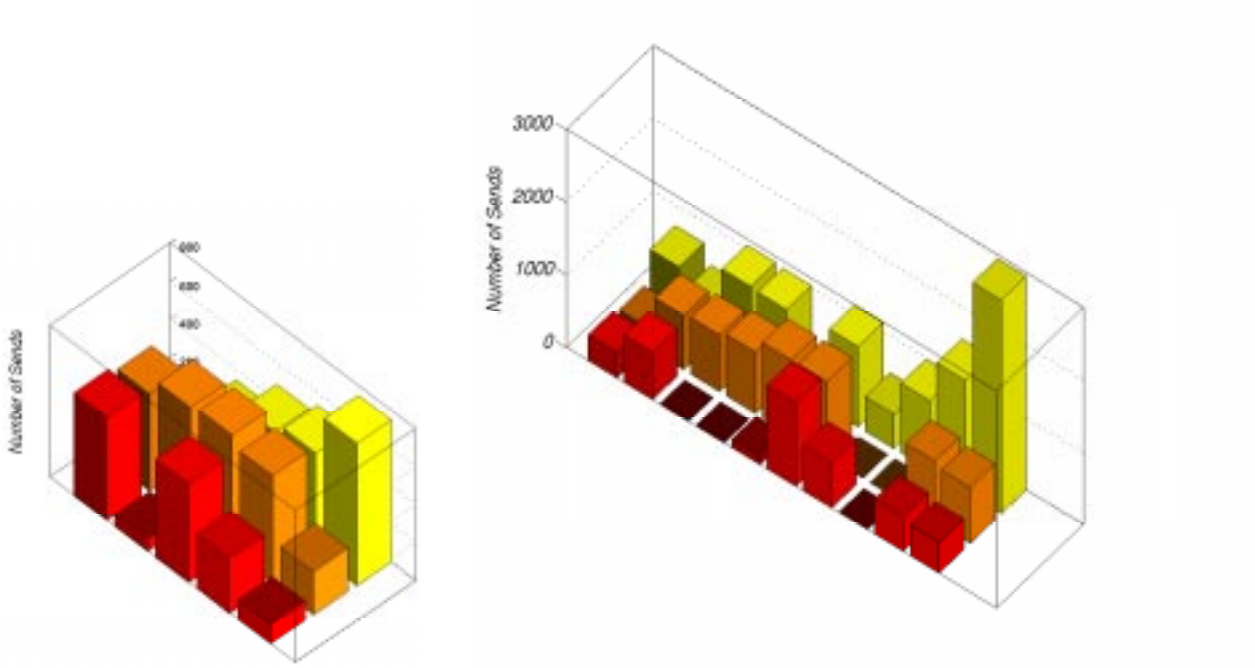
The consistently good PDR numbers for DTRP in the small and large network scenarios show its scalability and reliability. Running DTRP, the average PDR over all topologies and all levels of offered load is 69 % in the small network and 71 % in the large network. MINTRoute doesn't perform quite as well, with the same aggregate PDR numbers being 59 % in the small network and 46 % in the large network.

Total network load (Fig. 6.10) is lower using MINTRoute in all large network scenarios. Again, DTRP is a multi-path protocol, so this is to be expected. It appears that the maximum number of sources tested (9 pps) does not yet reach the network area traffic ceiling observed in the small network scenario. This makes sense, since we would expect the large network, with double the area of the smaller network, to have a ceiling of about double that of the smaller network, or about 100 pps. Our max offered load only produces about 75 pps in the network. However, all load lines are still linear, with DTRP still growing at a faster rate than MINTRoute. Figure 6.11-(a) and (b) show DTRP to be in a better operating zone than MINTRoute in both the small and large networks, delivering a high percentage of packets with relatively low load. Figure 6.12-(a) and (b) show the per-node cumulative load over the course of experiment with one source and five sources respectively in grid topologies. The graph coordinates represent physical placements and the destination sink node is located on the right hand side of the figure.



(a) (b)

Figure 6.11: PDR versus laod for the (a) small and (b) large grid topologies.



(a) (b)

Figure 6.12: Per-node cumulative load for (a) small and (b) large grid topologies.

6.5 Summary

It is very clear that DTRP (and Gossiping) deliver a markedly greater percentage of packets than MINTRoute in low load scenarios in large and small networks . Though the network load is greater with DTRP, the primary concern of most applications is the highest PDR with a minimal amount of overhead. MINTRoute cleverly uses the transmission of data packets as control packets for nodes not involved in the route path, implicitly requiring there to be ambient traffic in the network. In many applications with low data rates, this is not the case. However, this is not to say DTRP performs badly in high traffic scenarios. We show DTRP to deliver 10 % to 20 % more packets than MINTRoute in nearly every scenario tested, outside of one bottlenecked random topology with a small set of 15 nodes. We conclude that DTRP delivers more packets and shows to be more scalable than MINTRoute in low and high load scenarios running in large and small networks.

6.6 Acknowledgement

The research presented in this chapter was performed at Los Alamos National Laboratory (LANL). The author gratefully acknowledges Dr. Stephan J. Eidenbenz from LANL who advised this research. The author also acknowledges Matthew S. Nassr for his efforts in outdoor sensor deployment and programming routing protocols on sensor motes.

Chapter 7

Conclusion

Recently, wireless ad hoc networking has become an irreplaceable technology to fill the gap where communication infrastructure is insufficient or inexistent. Ad hoc networking protocols should be self-organizing, rapidly-deployable, and adaptable. These goals are hard to achieve in ad hoc environments due to volatile wireless links and node mobility. A number of research problems should be solved to turn the potential of wireless ad hoc networks into reality. Among various research challenges in wireless ad hoc networking, we investigate the key problems pertaining to the network layer. We focus on forwarding and routing issues that are critical to the performance of such networks. A serious forwarding unfairness problem is identified and various solutions are proposed with theoretical and experimental analyses. For efficient routing, we propose new routing schemes for different types of wireless ad hoc networks. A novel routing protocol for wireless mesh networks is proposed. Unlike typical ad hoc routing protocols, the proposed protocol does not rely on flooding by taking into account the unique properties of wireless mesh networks. An efficient update reduction scheme is proposed for mobile ad hoc networks where the notion of distance effect is introduced to suppress the flooding overhead. The proposed scheme is applied to OSPF protocol, the de facto standard Internet protocol, as an extension for mobile ad hoc environments. To further enhance the scalability of mobile ad hoc networks, the embedded notion of multiple OSPF areas is explored. We prove that there exists an optimum area number that minimizes the overall flooding overhead and maximizes the scalability. The results from the simulation experiments show that partitioning the network into multiple areas provides more significant impact for realistic mobility scenarios. Finally, we address the routing problem in wireless

sensor networks. Sensor networks are highly resource-constraint type of ad hoc networks and thus, the protocol should be simple yet reasonably reliable. This requirement is met by a multi-path and parametric probabilistic routing protocol. The proposed protocol outperformed widely-used sensor network routing protocols in outdoor field deployment experiments.

Bibliography

[1] L. Kalampoukas, A. Varma, and K. K. Ramakrishnan, “An efficient rate allocation algorithm for ATM networks providing min-max fairness,” tech. rep., UCSC-CRL-95-29 Computer Engineering Dept. University of California, Santa Cruz, June 1995.

[2] IEEE, “Wireless LAN medium access control (MAC) and physical layer (PHY) specification.” IEEE Std. 802.11, June 1999.

[3] V. Bharghavan, A. J. Demers, S. Shenker, and L. Zhang, “MACAW: A media access protocol for wireless LAN’s,” in *SIGCOMM*, pp. 212–225, 1994.

[4] C. L. Fullmer and J. J. Garcia-Luna-Aceves, “Floor acquisition multiple access (FAMA) for packet-radio networks,” in *SIGCOMM*, pp. 262–273, 1995.

[5] A. Woo and D. E. Culler, “A transmission control scheme for media access in sensor networks,” in *Mobile Computing and Networking*, pp. 221–235, 2001.

[6] J. Postel, “Transmission control protocol.” RFC 793, Sept. 1981.

[7] M. Gerla, K. Tang, and R. Bagrodia, “TCP performance in wireless multihop networks,” in *Proc. of IEEE WMCSA 99*, (New Orleans, LA), Feb. 1999.

[8] S. Xu and T. Saadawi, “Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks,” *IEEE Communications Magazine*, June 2001.

[9] S. Bae, K. Xu, S. Lee, and M. Gerla, “Measured analysis of TCP bahavior across multihop wireless and wired networks,” in *In Proc. IEEE Globecom 2002*, (Taipei, Taiwan, R.O.C.), Nov. 2002.

[10] K. Xu, S. Bae, M. Gerla, and S. Lee, “TCP behavior across multihop wireless and wired networks,” in *Proc. of WoWMoM 2002*, (Atlanta, GA, USA), Sept. 2002.

[11] K. Xu and M. Gerla, “TCP over an IEEE 802.11 ad hoc network: Unfairness problems and solutions,” Tech. Rep. 020019, UCLA Computer Science Department, May 2002.

[12] K. Tang and M. Gerla, “Fair sharing of MAC under TCP in wireless ad hoc networks,” in *Proc. of IEEE MMT 99*, (Venice, Italy), Oct. 1999.

[13] G. Holland and N. H. Vaidya, “Analysis of TCP performance over mobile ad hoc networks,” in *Proc. the ACM Intern. Conf. on Mobile Computing and Networking, (MobiCom)*, (Seattle, WA, USA), pp. 207–218, Aug. 1999.

[14] P. Sinha, T. Nandagopal, N. Venkitaraman, R. Sivakumar, and V. Bharghavan, “WTCP: A reliable transport protocol for wireless wide-area networks,” *Mobile Computing and Networking*, pp. 231–241, 1999.

[15] J. Jun, P. Peddabachagari, and M. Sichitiu, “Theoretical maximum throughput of ieee 802.11 and its applications,” in *The 2nd IEEE International Symposium on Network Computing and Applications*, Apr. 2003.

[16] S. M. Sunghyun, “Ieee 802.11e wireless lan for quality of service.”

[17] R. Luo, D. Belis, R. M. Edwards, and G. A. Manson, “A simulation design for link connection-oriented wireless mesh networks,” in *4th International Workshop on Mobile and Wireless Communications Network*, pp. 665–669, Sept. 2002.

[18] M. L. Sichitiu, “Wireless mesh networks: Opportunities and challenge,” in *Proc. of the Wireless World Congress*, (Palo Alto, Ca), May 2005.

[19] I. F. Akyildiz, X. Wang, and W. Wang, “Wireless mesh networks: a survey,” *Computer Networks*, 2005.

[20] R. Bruno, M. Conti, and E. Gregori, “Mesh networks: Commodity multihop ad hoc networks,” *IEEE Communications Magazine*, vol. 43, pp. 123–131, Mar. 2005.

[21] “Belair Networks website.” http://www.belairnetworks.com.

[22] “Firetide website.” http://www.firetide.com.

[23] “Intel Wireless Mesh Networks website.” http://www.intel.com/update/contents/nc11032.htm.

[24] “Kiyon website.” http://www.kiyon.com.

[25] “Mesh Dynamics website.” http://www.meshdynamics.com.

[26] “Microsoft’s Wireless Mesh Networks website.” http://research.microsoft.com/mesh/.

[27] “MeshNetworks website.” http://www.meshnetworks.com.

[28] “Nokia RoofTop website.” http://www.nwr.nokia.com.

[29] “Nortel Networks website.” http://www.nortelnetworks.com/solutions/wrlsmesh/.

[30] “PacketHop website.” http://www.packethop.com.

[31] “Radiant Networks website.” http://www.radiantnetworks.com.

[32] “SkyPilot network website.” http://www.skypilot.com.

[33] “Strix systems website.” http://www.strixsystems.com/.

[34] “Telabria website.” http://www.telabria.com/.

[35] “Tropos networks website.” http://www.troposnetworks.com/.

[36] D. Beyer, “Fundamental characteristics and benefits of wireless routing (“mesh”) networks,” in *Proc. of the International Technical Symposium of the Wireless Communications Association*, (San Jose, CA), Jan. 2002.

[37] J. Jun and M. L. Sichitiu, “The nominal capacity of wireless mesh networks,” *IEEE Wireless Communications Magazine, Special Issue on: Merging IP and Wireless Networks (to appear)*, Oct 2003.

[38] I. Chlamtac, M. Conti, and J. J.-N. Liu, “Mobile ad hoc networking: imperatives and challenges,” *Ad hoc Networks*, vol. 1, no. 1, pp. 13–64, 2003.

[39] E. Royer and C. Toh, “A review of current routing protocols for ad-hoc mobile wireless networks,” *IEEE Personal Communications*, Apr. 1999.

[40] J. Raju and J. Garcia-Luna-Aceves, “A comparison of on-demand and table-driven routing for ad-hoc wireless networks,” in *Proc. of IEEE ICC*, June 2000.

[41] S. R. Das, C. E. Perkins, and E. E. Royer, “Performance comparison of two on-demand routing protocols for ad hoc networks,” in *Proc. of INFOCOM*, pp. 3–12, 2000.

[42] C. Bhagwat, “Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers,”

[43] S. Murthy and J. Garcia-Luna-Aceves, “A routing protocol for packet radio networks,” in *Proc. of Mobicom*, Nov. 1995.

[44] C. Chiang, H. K. Wu, W. Liu, and M. Gerla, “Routing in clustered multihop mobile wireless networks with fading channel,” in *Proc. of IEEE Singapore International Conference on Networks*, 1997.

[45] C. E. Perkins and E. M. Royer, “Ad hoc on-demand distance vector routing,” in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, (New Orleans), 1999.

[46] D. B. Johnson and D. A. Maltz, “Dynamic source routing in ad hoc wireless networks,” in *Mobile Computing* (Imielinski and Korth, eds.), vol. 353, Kluwer Academic Publishers, 1996.

[47] V. Park, M. Scott, and Corson, “A highly adaptive distributed routing algorithm for mobile wireless networks,” in *Proc. of IEEE INFOCOM*, 1997.

[48] R. Dube, C. Rais, K. Wang, and S. Tripathi, “Signal stability based adaptive routing (SSA) for ad hoc mobile networks,” *IEEE Personal Communication*, Feb. 1997.

[49] Y. B. Ko and N. H. Vaidya, “Location aided routing (LAR) in mobile ad hoc networks,” *Wireless Networks*, vol. 6, pp. 307–321, Sept. 2000.

[50] S. Basagni, I. Chlamtac, V. Syrotiuk, and B. Woodward, “A distance routing effect algorithm for mobility (DREAM),” in *Proc. of ACM Mobicom’98*, (Dallas, TX), pp. 76–84, Oct. 1998.

[51] J. Li, J. Jannotti, D. S. J. DeCouto, D. R. Karger, and R. Morris, "A scalable location service for geographic ad-hoc routing," in *Proc. of ACM Mobile Communications Conference*, (Boston, MA), Aug. 2000.

[52] K. Amouris, S. Papavassiliou, and M. Li, "A position-based multi-zone routing protocol for wide area mobile ad-hoc networks," in *Proc. of IEEE Vehicular Technology Conference*, (Houston, TX), pp. 1365–1369, 1999.

[53] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Mobile Computing and Networking*, pp. 56–67, 2000.

[54] H. Xiaoyan, X. Kaixin, and M. Gerla, "Scalable routing protocols for mobile ad hoc networks," *IEEE Network*, vol. 16, July-Aug 2002.

[55] M. G. G. Pei and T. Chen, "Fisheye state routing: A routing scheme for ad hoc wireless networks," in *Proc. of ICC 2000*, (New Orleans, LA), June 2000.

[56] W. List and N. Vaidya, "A routing protocol for K-hop networks," in *Proc. of WCNC 2004*, Mar. 2004.

[57] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)." RFC 3626, Oct. 2003.

[58] J. Moy, "OSPF version 2." RFC 2328, Apr. 1998.

[59] D. Oran, "OSI IS-IS intra-domain routing protocol." RFC 1142, Feb. 1990.

[60] Y. Rekhter and T. Li, "A border gateway protocol." RFC 1771 (BGP version 4), Mar. 1995.

[61] C. Hedrick, "Routing information protocol." RFC 1058, June 1988.

[62] G. Malkin, "RIP version 2 - carrying additional information." RFC 1388, Jan. 1993.

[63] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing." RFC 3561, July 2003.

[64] Z. J. Haas and M. R. Pearlman, "The zone routing protocol (zrp) for ad hoc networks," tech. rep., IETF MANET Working Group.

[65] Y. Ko and N. H. Vaidya, "GeoTORA: A protocol for geocasting in mobile ad hoc networks," in *Proc. of the 8th International Conference on Network Protocols (ICNP)*, (Osaka), Nov. 2000.

[66] Y. Ko and N. Vaidya, "Geocasting in mobile ad hoc networks: Location-based multicast algorithms," in *Proc. of WMCSA*, (New Orleans), 1999.

[67] R. Ogier, F. Templin, and M. Lewis, "Topology dissemination based on reverse-path forwarding (TBRPF)." RFC 3484, Feb. 2004.

[68] G. Pei, M. Gerla, and X. Hong, "LANMAR: landmark routing for large scale wireless ad hoc networks with group mobility," in *Proc. of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, (Boston, MA), Aug. 2000.

[69] M. Gerla, X. Hong, and G. Pei, "Landmark routing for large ad hoc wireless networks," in *Proc. of IEEE GLOBECOM 2000*, (San Francisco, CA), 2000.

[70] D. Beyer, "Accomplishments of the DARPA SURAN program," in *Proc. IEEE MILCOM 90 Conference*, (Monterey, California), Oct. 1990.

[71] J. Garcia-Luna-Aceves, C. Fullmer, E. Madruga, D. Beyer, and T. Frivold, "Wireless internet gateways (WINGs)," in *Proc. IEEE MILCOM '97*, (Monterey, California), Nov 1997.

[72] S. Roy and J.J.Garcia-Luna-Aceves, "Node-centric hybrid routing for ad-hoc wireless extensions of the internet," in *Proc. of Globecom*, 2002.

[73] S. Roy and J.J.Garcia-Luna-Aceves, "Using minimal source trees for on-demand routing in ad hoc networks," in *Proc. IEEE INFOCOM*, (Anchorage, Alaska), Apr. 2001.

[74] K. Xu and M. Gerla, "A heterogeneous routing protocol based on a new stable clustering scheme," in *Proc. of the Military Communications Conference (MILCOM)*, (Anaheim, CA), Oct. 2002.

[75] C. Tschudinand and R. Gold, "LUNAR: Lightweight underlay network ad-hoc routing," tech. rep., University of Basel, Switzerland, Jan. 2002.

[76] M. J. Miller, W. D. List, and N. H. Vaidya, "A hybrid network implementation to extend infrastructure reach," tech. rep., Coordinated Science Laboratory, University of Illinois at Urbana-Champaign., Jan. 2003.

[77] J. Xi and C. Bettstetter, “Wireless multi-hop internet access: Gateway discovery, routing, and addressing,” in *Proc. Intern. Conf. on Third Generation Wireless and Beyond (3Gwireless)*, (San Francisco, CA), May, 28–31 2002.

[78] A. Raniwala and T. Chiueh, “Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network,” in *Proc. of IEEE Infocom ’05*, 2005.

[79] J. Tang, G. Xue, and W. Zhang, “Interference-aware topology control and QoS routing in multi-channel wireless mesh networks,” in *Proc. of Mobihoc’05*, May 2005.

[80] R. Draves, J. Padhye, and B. Zill, “Routing in multi-radio, multi-hop wireless mesh networks,” in *Proc. of Mobicom’04*, (Philadelphia, PA), sep 2004.

[81] L. Iannone, R. Khalili, K. Salamatian, and S. Fdida, “Cross-layer routing in wireless mesh networks,” in *Proc. of the 1st International Symposium in Wireless Communication Systems*, (Mauritius), Sept. 2004.

[82] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar, “Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW protocol,” in *Proc. of European Wireless 2002. Next Generation Wireless Networks: Technologies, Protocols, Services and Applications*, (Florence, Italy), pp. 156–162, Feb. 25-28 2002.

[83] T. Camp, J. Boleng, and V. Davies, “A survey of mobility models for ad hoc network research,” *Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, no. 5, pp. 483–502, 2002.

[84] J. Yoon, M. Liu, and B. Noble, “Random waypoint considered harmful,” in *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies – INFOCOM 2003*, pp. 1312–1321, Mar. 2003.

[85] Scalable Network Technologies, Inc., “QualNet User’s Manual, version 3.8.” http://www.scalable-networks.com/, 2003.

[86] “GloMoSim.” http://pcl.cs.ucla.edu/projects/glomosim/.

[87] R. Bagrodia, R. Meyer, M. Takai, C. Yu-An, Z. Xiang, J. Martin, and H. Y. Song, "Parsec: a parallel simulation environment for complex systems," *IEEE Computer*, vol. 31, Oct. 1998.

[88] M. Takai, J. Martin, and R. Bagrodia, "Effects of wireless physical layer modeling in mobile ad hoc networks," in *Proceedings of MobiHoc 2001*, Oct. 2001.

[89] J. Hsu, S. Bhatia, M. Takai, R. Bagrodia, and M. J. Acriche, "Performance of mobile ad hoc networking routing protocols in realistic scenarios," in *Proceedings of MILCOM 2003*, Oct. 2003.

[90] K. Thompson, G. J. Miller, and R. Wilder, "Wide-area internet traffic patterns and characteristics," *IEEE Network*, vol. 11, Nov-Dec 1997.

[91] "Inria OLSR." http://menetou.inria.fr/olsr/.

[92] J. Moy, "OSPF version 2." RFC 2328, Apr. 1998.

[93] R. Coltun, D. Ferguson, and J. Moy, "OSPF for IPv6." RFC 2740.

[94] "GNU quagga routing software." http://www.quagga.net/.

[95] T. R. Henderson, P. A. Spagnolo, and G. Pei, "Evaluation of OSPF MANET extensions." Boeing Technical Report, Jul. 21 2005.

[96] G. F. Riley, "Large-scale network simulations with GTNetS," in *Proceedings of the 2003 Winter Simulation Conference*.

[97] M. Chandra, "Extensions to OSPF to support mobile ad hoc networking." Internet Draft draft-chandra-ospf-manet-ext-03, Apr 2005.

[98] R. Ogier and P. Spagnolo, "MANET extensions of OSPF using CDS flooding." Internet Draft draft-ogier-manet-ospf-extension-04, Jul 2005.

[99] F. Baker, M. Chandra, R. White, J. Macker, T. Henderson, and E. Baccelli, "Problem statement for OSPF extensions for mobile ad hoc routing." draft-baker-manet-ospf-problem-statement-00, Sep. 23 2003.

[100] "Hurricane Katrina reveals strengths of emerging technologies," *IEEE IT Professional*, vol. 7, Sep 2005.

[101] J. S. Granelli, “Wireless broadband rises to challenge landlines,” *Los Angeles Times*, Sep 2005.

[102] A. Roy, “Adjacency reduction in OSPF using SPT reachability.” Internet Draft draft-roy-ospf-smart-peering-00, Jan 2006.

[103] Jangeun Jun and Mihail L. Sichitiu, “Scalable OSPF updates for MANETs,” in *Proc. of the IEEE Globecom 06 - WASNet (to appear)*, Nov 2006.

[104] “Distances in bounded regions.” http://www.mathpages.com/home/kmath324/kmath324.htm.

[105] “OASISv1.o: OSPF Areas Simulation System, Version 1.0.” http://www4.ncsu.edu/~jjun/oasis1.0.zip, Jul 2006.

[106] M. E. J. Newman, “Detecting community structure in networks,” *Eur. Phys. J. B*, no. 38, pp. 321–330, 2004.

[107] L. Danon, J. Duch, A. Diaz-Guilera, and A. Arenas, “Comparing community structure identification,” *J. Stat. Mech.*, p. P09008, 2005.

[108] S. Eidenbenz, H. Flores, N. Hengartner, and R. Riedi, “Describing MANETS: Principal component analysis of sparse mobility traces,” in *Proc. of PE-WASUN’06 (to appear)*, Oct. 2006.

[109] H. Flores, S. Eidenbenz, N. Hengartner, and R. Riedi, “PedSims: Building towards realism of mobility models for wireless networks,” in *(submitted)*, 2006.

[110] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Communications Magazine*, August 2002.

[111] C.-Y. Chong and S. P. Kumar, “Sensor networks: Evolution, opportunities, and challenges,” *Proceedings of the IEEE*, vol. 91, August 2003.

[112] M. Gerla, “Fisheye state routing protocol (FSR) for ad hoc networks,” June 2001.

[113] M. Mauve, J. Widmer, and H. Hartenstein, “A survey on position based routing in mobile ad-hoc networks,” *IEEE Network*, Nov 2001.

[114] S. Chen and K. Nahrstedt, “Distributed quality of service routing in ad hoc networks,” *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 1488–1505, Aug. 1999.

[115] P. Sinha, R. Sivakumar, and V. Bharghavan, "CEDAR: a core-extraction distributed ad hoc routing algorithm," in *Proc. of INFOCOM*, pp. 202–209, 1999.

[116] S. Chen, K. Nahrstedt, and Y. Shavitt, "A QoS-aware multicast routing protocol," in *Proc. of INFOCOM*, pp. 1594–1603, 2000.

[117] C. Lin and J.-S. Liu, "QoS routing in ad-hoc wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, Aug. 1999.

[118] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Mobile Computing and Networking*, pp. 56–67, 2000.

[119] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly resilient, energy efficient multipath routing in wireless sensor networks," *Mobile Computing and Communications Review*, vol. 1, no. 2.

[120] S. M. Hedetniemi, S. T. Hedetniemi, and A. Liestman, "A survey of gossiping and broadcasting in communication networks," *Networks 18*, 1988.

[121] D. Kempe, J. Kleinberg, and A. Demers, "Spatial gossip and resource location protocols," in *Proceedings of 33rd ACM Symposium on the Theory of Computing (STOC)*, 2001.

[122] Z. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," in *Proceedings of INFOCOM 2002*, 2002.

[123] J. Luo, P. T. Eugster, and J.-P. Hubaux, "Route driven gossip: Probabilistic reliable multicast in ad hoc networks," in *Proceedings of INFOCOM 2003*, 2003.

[124] "Tiny Operating System. UC Berkeley." http://tinyos.net/.

[125] A. Woo, T. Tong, , and D. Culler, "Taming the underlying challenges of reliable multhop routing in sensor networks," in *SenSys 2003*, (Los Angeles, CA), 2003.

[126] R. Jurdak, P. Baldi, and C. V. Lopes, "Energy-aware adaptive low power listening for sensor networks," in *the Second International Workshop on Networked Sensing Systems*, (San Diego, CA), June 2005.

[127] U. Malesci and S. Madden, “A measurement-based analysis of the interaction between network layers in TinyOS,” in *EWSN*, 2006.

[128] C. Barrett, S. Eidenbenz, L. Kroc, M. Marathe, and J. Smith, “Parametric probabilistic sensor network routing,” in *Proceedings of The Second ACM International Workshop on Wireless Sensor Networks and Applications (WSNA’03)*, (San Diego, CA), Sept. 2003.

[129] C. Barrett, S. Eidenbenz, L. Kroc, M. Marathe, and J. Smith, “Probabilistic multi-path vs. deterministic single-path protocols for dynamic ad-hoc network scenarios,” in *Proceedings of The 2005 ACM Symposium on Applied Computing (SAC’05)*, (Santa Fe, NM), Mar. 2005.

[130] “Mica 2 Motes. Crossbow Technology Inc..” http://www.xbow.com/.

[131] K. Langendoen, A. Baggio, and O. Visser, “Murphy loves potatoes: Experiences from a pilot sensor network deployment in precision agriculture,” in *WPDRTS*, 2006.

[132] S. Bapat, V. Kulathumani, and A. Arora, “Analyzing the yield of exscal, a large-scale wireless sensor network experiment,” in *ICNP 2005*, 2005.