CERIAS Tech Report 2004-51

ADAPTIVE AND HETEROGENEOUS MOBILE WIRELESS NETWORKS

by Yi Lu

Center for Education and Research in
Information Assurance and Security,
Purdue University, West Lafayette, IN 47907-2086

ADAPTIVE AND HETEROGENEOUS MOBILE WIRELESS NETWORKS

A Thesis

Submitted to the Faculty

of

Purdue University

by

Yi Lu

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2004

To my parents Zongfu Lu and Chunfang Cui and my wife Yuhui Zhong.

献给我的父母卢宗福和崔春芳以及我的妻子钟昱晖。

ACKNOWLEDGMENTS

I would like to acknowledge the effort of my major advisor, Professor Bharat Bhargava, in motivating and guiding me during the years that I spent in his research lab at Purdue. Professor Bhargava's expertise and insight are the keys for the success of his students. I would like to thank Professor Michael Zoltowski for giving me ideas for wireless networking research. I thank Professor Sonia Fahmy and Professor Aditya Mathur for answering my questions and recommending research papers. I enjoyed the discussion of research problems with Professor Dongyan Xu. Without the support and encouragement of my advising committee, this thesis would not have been complete.

I thank my colleagues in the RAID lab for their help and support. Dr. Leszek Lilien, Dr. Xiaoxin Wu, and Dr. Ahsan Habib gave me valuable comments on research ideas and papers. The discussions with Weichao Wang and Mohamed Hefeeda have inspired me a lot.

Finally, I greatly appreciate the help, support, and encouragement of my wife Yuhui Zhong.

TABLE OF CONTENTS

Page

Page

Page

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABSTRACT

Lu, Yi. Ph.D., Purdue University, August, 2004. Adaptive and Heterogeneous Mobile Wireless Networks. Major Professor: Bharat Bhargava.

This dissertation investigates two research problems: (a) designing ad hoc routing protocols that monitor network conditions, select routes to satisfy routing requirements, and adapt to network topology, traffic load, and congestion; (b) building an integrated infrastructure for heterogeneous wireless networks with movable base stations and developing techniques for network management, routing, and security.

The experimental study of ad hoc routing protocols shows that the on-demand approach outperforms the proactive approach in less stressful situations, while the later one is more scalable with respect to the network size. Mobility and congestion are the primary reasons for the packet loss for the on-demand and proactive approaches respectively. Self-adjusting congestion avoidance (SAGA) routing protocol integrates the channel spatial reuse with the multi-hop routing to reduce congestion. Using the intermediate delay as the routing metric enables SAGA to bypass hot spots where contention is intense. An estimate of the transmission delay is derived based on local information available at a host. Comparison of SAGA with AODV, DSR, and DSDV shows that SAGA introduces the lowest end-to-end delay. It outperforms DSDV in the measured metrics. SAGA can sustain heavier traffic load and offers higher peak throughput than AODV and DSR. It is shown that considerations of congestion and the intermediate delay can enhance the routing performance significantly.

Hierarchical mobile wireless network is proposed to support wireless networks with movable base stations. Mobile hosts are organized into hierarchical groups. An efficient group membership management protocol is designed to support mobile hosts roaming among different groups. Segmented membership-based group routing protocol takes ad-

vantage of the hierarchical structure and membership information to reduce overhead. A secure packet forwarding algorithm is designed to protect the network infrastructure. The roaming support algorithm cooperates with the proposed mutual authentication protocol to secure both the foreign group and the mobile host. The evaluation shows that the computation overhead of the secure packet forwarding is less than 2% of the CPU time, and that of the secure roaming support ranges from 0.2% to 5% of the CPU time depending on the number of hosts and their motion. This justifies the feasibility of the security mechanisms.

1 INTRODUCTION

1.1 Problem statement

The research problem is *how to provide continuous connectivity for a mobile unit to a network in which every node is moving*. We investigate this problem in two network environments shown in Figure 1.1: (a) mobile ad hoc networks that have no centralized control; (b) large scale heterogeneous wireless networks with control in movable base stations. The major challenges are *dynamic topology*, *decentralized control*, and *limited bandwidth*. We concentrate on research problems at the network layer.

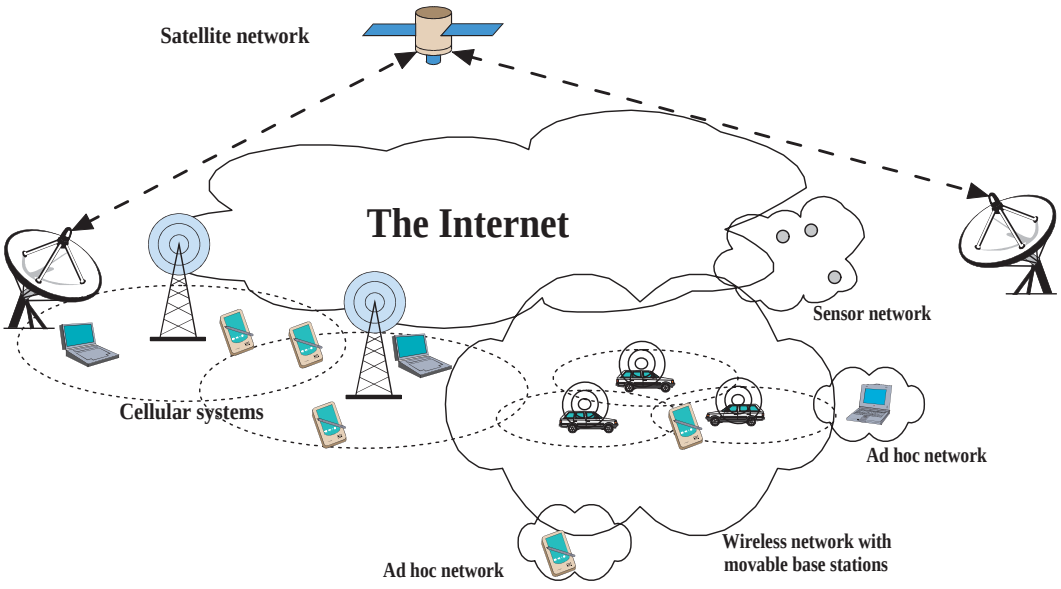


Figure 1.1. Network environments

1.1.1 Adaptive routing in mobile ad hoc networks

A mobile ad hoc network (MONET) is a collection of mobile hosts that are deployed as a multi-hop wireless network without the aid of any preexisting infrastructure or centralized administration. It relies on hosts cooperation to maintain network connectivity and functionality. The salient characteristics of ad hoc networks, including highly dynamic topologies, low bandwidth, energy-constrained operations, and limited computation capability, make the design of routing protocols a challenging problem. The protocols must be capable of keeping up with the drastically and unpredictably changing network topology, with minimized message exchanges, in a fully distributed way. Most proposed ad hoc routing protocols, such as destination-sequenced distance vector (DSDV) [1], ad-hoc on-demand distance vector (AODV) [2], and dynamic source routing (DSR) [3], adopt the content of routing information from the Internet protocols and react to topology changes. Research is needed to develop a protocol that is able to adapt to various network conditions such as traffic load and congestion. This requires the following:

- Identifying the network parameters that impact the performance of routing protocols and evaluating their effects through experiments.

- Determining the appropriateness of on-demand and proactive approaches to maintain network connectivity, given specific routing requirements and network parameters.

This research provides a better understanding of the advantages and disadvantages of different routing approaches in various network contexts that will lead to the development of new adaptive routing protocols. It offers guidelines on identifying and avoiding the performance bottleneck of routing protocols, and choosing proper parameters in future simulation and analytic studies. Based on this research, a congestion avoidance routing protocol is developed, which is capable of adapting to congestion, traffic load, and network topology.

1.1.2 Large scale heterogeneous wireless networks

The mushrooming of heterogeneous wireless technologies and the need of robust and efficient communication systems call for the ubiquitous and integrated wireless infrastructure. While the existing wireless networks have been individually studied, the integrated wireless system brings new challenges in network management, protocol design, and performance evaluation.

In a heterogeneous wireless network, there are base stations that have more resources than mobile hosts in terms of energy supply, processing power, etc. These base stations that have multiple wireless interfaces (each interface may use different wireless technology) connect different wireless networks. Most existing heterogeneous wireless network models assume base stations are stationary [4–6]. They are not able to adapt to dynamic movement. We study the case where even base stations are moving. We refer this kind of network as *wireless network with movable base stations* (WNMBS). The following research problems have been investigated.

- How to organize the network in an efficient way so that the effect of motion on topology is minimized without loss of network connectivity? How to minimize the involvement of resource-poor mobile hosts?

- How to build efficient routing protocols for WNMBS? The protocols should be IP-compliant and transparent to upper layer services such as TCP and UDP. It should be capable of cooperating with various routing protocols used by different sub-nets.

- What cryptography mechanism should be used to secure communications? How to authenticate a mobile host? How to maintain authentication when a host is roaming among the system? How to handle agent failures when authentication is required?

In addition to the commercial 3G wireless system that provides different mobile services, many existing and future military networks that consist of highly dynamic autonomous topology segments require integration of heterogeneous wireless technologies. This research have impacts on the development of a framework to seamlessly support

IP-compliant data services over heterogeneous wireless networks and new security mechanisms that fit into the mobile wireless world.

1.2 Thesis contributions

1.2.1 Mobile ad hoc networks

We investigate the performance issues of DSDV and AODV routing protocols for mobile ad hoc networks. Four performance metrics are measured by varying the maximum speed of mobile hosts, the number of connections, and the network size. The correlation between the network topology change and the host mobility is investigated by using linear regression analysis. The simulation results indicate that AODV outperforms DSDV in less stressful situations, while DSDV is more scalable with respect to the network size. It is observed that network congestion is the dominant reason for packet drop for both protocols. We propose a new routing protocol, congestion-aware distance vector (CADV), to address the congestion issues. CADV outperforms AODV in delivery ratio by about 5%, while introduces less protocol load. The result demonstrates that integrating congestion avoidance mechanisms with proactive routing protocols is a promising way to improve performance.

We study the impact of congestion and mobility on the packet loss in various network contexts. The results indicate that DSDV loses 10% to 20% more packets than AODV for UDP traffic. For TCP traffic, the packet loss for DSDV is a half of that for AODV. Mobility is the dominant cause for AODV, which is responsible for more than 60% of total packet loss. For DSDV, more than 50% of total packet loss is congestion-related. This work provides guidelines for the design of routing and flow control algorithms and insights in choosing proper parameters in future simulation and analytic studies.

Congestion in ad hoc networks is a serious problem. Contention among neighbors for the access to the shared media is the primary cause for the network congestion. We consider congestion in the design of the routing protocols. The main thrust is to avoid congestion by minimizing contentions for channel access. The intermediate delay (IMD)

is proposed as a routing metric. It enables routing protocols to select routes that bypass mobile nodes in contention. IMD characterizes the impacts of channel contention, traffic load, and the length of a route. The packet transmission procedure of the distributed coordination function (DCF) in the IEEE 802.11 standard is analyzed and used as a study case for evaluation and experimentation. An estimate of the transmission delay is derived based on local information available at a node. The estimation takes the impact of active traffic in the neighborhood into account without exchanging messages with neighbors.

The self-adjusting congestion avoidance (SAGA) routing protocol is designed with IMD as the routing metric. The performance of SAGA is evaluated and compared with that of AODV, DSDV, and dynamic source routing (DSR) protocols using simulation. Two types of UDP traffic, constant bit rate traffic and traffic exhibiting long range dependency, as well as the TCP traffic are considered. SAGA can sustain heavier traffic load and offers higher peak throughput than AODV and DSR. The overhead of SAGA can be as low as 10% as that of AODV and 12% as that of DSR. The average end-to-end delay of SAGA is the lowest among the protocols. It is shown that considerations of congestion and intermediate delay instead of hop count can enhance routing performance significantly.

1.2.2 Wireless networks with movable base stations

Wireless networks with movable base stations combine the advantages of mobile ad hoc networks and wireless LAN to achieve both flexibility and scalability. We present the hierarchical mobile wireless network (HMWN) to support movable base stations. HMWN may be applied to ad hoc networks as well to build a virtual hierarchy. In such a system, mobile hosts are organized into hierarchical groups. Four basic operations for setting up and maintaining the network structure are grouping, registration, leaving, and migration. An efficient group membership management protocol is developed to support mobile hosts roaming among different groups. The segmented membership-based group routing (SMGR) protocol is proposed to take advantage of the hierarchical structure and membership information. In this protocol, only local message exchanging is required

for maintaining network topology and routing information. Simulation-based experiment demonstrates the scalability of the design in terms of protocol overheads.

Security, flexibility, and scalability are critical to the success of wireless communications. In a HMWN system, the group agents serve as a distributed trust entity. A secure packet forwarding algorithm and an authentication and key exchange protocol are developed to protect the network infrastructure. A roaming support mechanism and the associated mutual authentication protocol are proposed to secure the foreign group and the mobile host when it roams within the network. The computation overhead of secure packet forwarding and roaming support algorithms is studied via experiments. The results demonstrate that the computation overhead of secure packet forwarding is less than 2% of the CPU time, and that of secure roaming support ranges from 0.2% to 5% of the CPU time depending on the number of hosts and their motion.

1.3 Thesis organization

Chapter 2 briefly introduces the two ad hoc routing protocols, DSDV and AODV, that are used in this research. The network simulator ns2 and the specifications of the physical and MAC layers of the wireless networks are described. The random waypoint mobility model is used to generate movements for mobile hosts. Its parameters and characteristics are outlined.

Chapter 3 presents the study of ad hoc routing protocols. The correlation between the network topology change and the host mobility is investigated. The results indicate that the topology change may be a linear function of the maximum speed and the pause time, respectively. DSDV and AODV are taken as the representatives of the proactive and on-demand routing approaches in this study. The performance of these protocols are evaluated by varying the maximum speed, the number of connections, and the number of mobile hosts. The measurements include delivery ratio, average end-to-end delay, protocol overhead, and power consumption. Further investigation on DSDV is conducted by

reducing the broadcast interval and increasing the queue length. Based on the analysis of the experimental study, the congestion-aware routing protocol CADV is proposed.

Chapter 4 studies the causes for packet loss in ad hoc networks. The causes are categorized as congestion-related and mobility-related. The effects of congestion and mobility on DSDV and AODV in various network contexts are explored. The results indicate that mobility is the dominant cause for AODV, which is responsible for more than 60% of total packet loss. For DSDV, more than 50% of total packet loss is congestion-related.

Based on the guidelines obtained from the experimental studies. The self-adjust congestion avoidance (SAGA) routing protocol is presented in Chapter 5. The characteristics of contention-based access to wireless channels and their impact on ad hoc routing are discussed. The intermediate delay (IMD) is proposed as a routing metric and the ideas of ad hoc routing based on IMD are illustrated. The delay estimation is critical in SAGA. When a node has active traffic, statistical methods are used to evaluate the mean of the delay. Otherwise, the underlying MAC protocol is analyzed and probability methods are applied to compute the expectation of the delay. The packet transmission procedure of the distributed coordination function (DCF) in the IEEE 802.11 standard is analyzed and used as a study case for evaluation and experimentation. The performance of SAGA is evaluated and compared with that of AODV, DSR, and DSDV protocols.

In Chapter 6, a hierarchical network architecture is proposed to support movable base stations in heterogeneous wireless networks. The design considerations include the asymmetric capacity and responsibility between base stations and mobile hosts, the coordinated movement of hosts, and the localized traffic. Four basic operations, grouping, registration, leaving, and migration, are defined for setting up and maintaining the network structure. The details of the membership management scheme and the segmented membership-based group routing protocol are presented. Experiments are conducted to study the protocol overhead.

Chapter 7 presents mechanisms for securing wireless networks with movable base stations. The base stations serve as a distributed trust entity for key management and authentication. A secure packet forwarding algorithm and an authentication and key exchange

protocol are developed to protect the network infrastructure. A roaming support mechanism and the associated mutual authentication protocol are proposed to secure the foreign group and the mobile host when it roams within the network. The computation overhead of secure packet forwarding and roaming support algorithms is studied via experiments.

Chapter 8 concludes this dissertation and outlines directions for extending the research.

2 BACKGROUND

2.1 Destination-sequenced distance vector routing protocol (DSDV)

DSDV routing protocol is one of the first routing protocols designed specially for ad hoc networks. It extends the basic Bellman-Ford mechanism by attaching a sequence number, which is originated by the destination, to each distance. This destination sequence number is used to determine the "freshness" of a route. Routes with more recent sequence numbers are preferred for making packet forwarding decisions by a host, but not necessarily advertised to other hosts. For routes with the equal sequence number, the one with the smallest distance metric is chosen. Each time a host sends an update to its neighbors, its current sequence number is incremented and included in the update. The sequence number is disseminated throughout a network via update messages. The DSDV protocol requires each host to periodically advertise its own routing table to its neighbors. Updates are transmitted immediately when significant new routing information is available. Routes received in broadcasts are used to update the routing table. The receiver adds an increment to the metric of each received route before updating.

In DSDV, the broken link may be detected by the layer-2 protocol, or may be inferred if no broadcast has been received from a former neighbor for a while (e.g., three periodic update periods). A broken link is assigned a metric of ∞ (i.e., a value greater than the maximum allowed metric). When a broken link to a next hop is detected, the metric of any route through that next hop is immediately assigned ∞, and the sequence number associated with it is incremented. Such modified routes are immediately broadcast in a routing update packet. Handling broken links is the only situation when a sequence number is generated by a host other than the destination. To distinguish this situation, sequence numbers generated by the originating hosts are even numbers, while sequence

numbers generated to indicate the ∞ metric are odd numbers. Any *real* sequence number will supersede an ∞ metric.

Two types of updates are defined in DSDV protocol. One, called “full dump”, carries all the available routing information. The other, called “incremental”, carries only information changed since the last full dump. Full dumps are generated relatively infrequently. If the size of an incremental approaches the size of a packet, a full dump can be scheduled so that the next incremental will be smaller.

Since all mobile hosts periodically advertise their routing information, a host can almost always locate every other host when it needs to send out a packet. Otherwise, the packet is queued until the routing information is available. DSDV guarantees loop-free paths to each destination [1].

2.2 Ad hoc on-demand distance vector routing protocol (AODV)

AODV routing protocol is also based upon distance vector, and uses destination sequence numbers to determine the freshness of routes. It operates in the on-demand fashion, as opposed to the proactive way of the DSDV protocol. AODV requires hosts to maintain only active routes. An *active route* is a route used to forward at least one packet within the past *active timeout* period. When a host needs to reach a destination and does not have an active route, it broadcasts a Route Request (RREQ), which is flooded in the network. A route can be determined when RREQ is received either by the destination itself or by an intermediate host with an active route to that destination. A Route Replay (RREP) is unicast back to the originator of RREQ to establish the route. Each host that receives RREQ caches a route back to the originator of the request, so that RREP can be sent back. Every route expires after a predetermined period of time. Sending a packet via a route will reset the associated expiry time.

Every host monitors the link status of next hops in active routes by listening for “Hello” messages from its neighbors or for any suitable link layer notification (such as those provided by IEEE 802.11). When a link break in an active route is detected, a Route Error

(RERR) is sent back along the path to the source. All hosts on that path notice the loss of the link. In order to report errors, every host maintains a *precursor list* for each route, containing the neighbors that are likely to forward packets on this route.

To prevent unnecessary network-wide dissemination of route request messages, the source may use an *expanding ring search* technique as an optimization. The search range is controlled by the time-to-live (TTL) field in the IP header of the RREQ packet. The search process is repeated with an incremented TTL (thus expanding the ring) until a route is discovered.

Another optimization is *local repair*. When a broken link in an active route is detected, instead of sending back RERR, the host first tries to locally repair the link by broadcasting RREQ for the destination. Although local repair is likely to increase the number of deliverable data packets, it may result in increased delay as well.

2.3 Simulation environment

This research involves extensive experimental studies using ns2. ns2 is an event-driven network simulator targeted at networking research. It is a widely used tool for simulating inter-network topologies to test and evaluate various networking protocols. It supports simulations of wireless networks and interconnecting wired and wireless networks.

In simulation, each mobile host uses an omni-directional antenna having unity gain. The wireless interface works like the 914 MHz Lucent WaveLAN direct-sequence spread-spectrum (DSSS) radio interface [7]. WaveLAN is modeled as a shared-media radio with a nominal bit rate of 2 Mb/s, and a nominal radio range of 250m [8]. The IEEE 802.11 distributed coordination function (DCF) is used as the MAC layer protocol. A unicast data packet destined to a neighbor is sent out after handshaking with request-to-send/clear-to-send (RTS/CTS) exchanges and followed by an acknowledgement (ACK) frame. The broadcast packets are simply sent out without handshake and acknowledgement. The implementation uses carrier sense multiple access with collision avoidance (CSMA/CA).

The implementations of DSDV and AODV provided by ns2 are used in the studies. They closely match the specifications [1] and [2]. The implementation of AODV enables expanding ring search and local repair.

2.4 Mobility model

The *random waypoint* model [9] is used to generate movements for mobile hosts. At the beginning of a simulation, mobile hosts are randomly placed on a 1000m x 1000m square field. Each host randomly chooses its destination in the field, and a moving speed that ranges from 0 to the given maximum speed. All destinations and speeds are independent and identically distributed. After a host reaches the destination, it waits for a specified time (i.e., pause time), and then repeats the above steps. According to this model, the speed and direction of the next movement have no relation to those of the previous movement. As indicated in [10], the pause time and the maximum speed have similar impacts on the mobility with respect to link change or route change. Thus the mobility is varied by changing the pause time or the maximum speed in the simulation.

3 STUDY OF AD HOC ROUTING PROTOCOLS

3.1 Introduction

3.1.1 Problem statement

The high mobility, low bandwidth, and limited computing capability characteristics of mobile hosts make the design of ad hoc routing protocols challenging. The protocols must be able to keep up with the drastically and unpredictably changing network topology, with minimized message exchanges, in a computation efficient way.

The routing protocols may be categorized as *proactive*, *on-demand*, and *hybrid*, according to the way in which the mobile hosts exchange routing information. The proactive protocols, such as DSDV [1] and source tree adaptive routing (STAR) [11,12], periodically disseminate routing information among all the hosts in the network, so that every host has the up-to-date information for all possible routes. On-demand routing protocols, such as AODV [2] and dynamic source routing (DSR) [3], operate on a need basis, discover and maintain only active routes that are currently used for delivering data packets. Hybrid routing protocols, such as zone routing protocol (ZRP) [13, 14] and Core Extraction Distributed Ad Hoc Routing (CEDAR) [15], maintain a virtual routing infrastructure, apply proactive routing mechanisms in certain regions of a network and on-demand routing in the rest of the network.

An ad hoc routing protocol tends to be well-suited for some network contexts, yet less suited for the others [16]. A better understanding of the advantages and disadvantages of different routing approaches in various network contexts will serve as a cornerstone for the development of new adaptive routing protocols. However, ad hoc networks are too complex to allow analytical study for explicit performance expressions. We use the means

of simulation to evaluate the routing approaches numerically and gather data to estimate their characteristics.

We study the performance of DSDV and AODV in a wide range of network contexts with varied network size, mobility, and traffic load. Both protocols utilize distance vector coupled with destination sequence number, and choose routes in the same manner. They are differentiated by the way in which they operate (i.e., proactive versus on-demand). Studying these two protocols gives insights into the differences between proactive and on-demand approaches. This analysis provides guidelines to improve these two specific protocols as well.

3.1.2 Our contributions

The linear dependence between network topology change and host mobility is investigated by using statistical analysis. The suitable network contexts for DSDV and AODV are identified. We discover that AODV introduces 1.5 to 5 times protocol load as DSDV does, which contradicts the motivation for the on-demand approach. The major causes for packet drop are investigated by exploring packet traces. We argue that DSDV is plagued by network congestion. Based upon the idea of integrating congestion avoidance mechanisms with proactive routing protocols to improve routing performance, we propose congestion-aware distance vector (CADV) routing protocol. The preliminary study of CADV shows positive results. To our knowledge, it is the first research effort to take the power consumption as a routing performance metric.

3.2 Related work

Several simulation-based performance comparisons have been done for ad hoc routing protocols in the recent years. Das et al. evaluate performance of ad hoc routing protocols based on the number of conversations per mobile node using Maryland Routing Simulator (MaRS) [17]. The performance comparison of two on-demand routing protocols DSR and AODV is presented in [8], using ns2 (network simulator) [18] for the simulation. The

pause time and the offered traffic load are taken as parameters. In [19], GloMoSim [20] is used for the performance study of the STAR, AODV, and DSR routing protocols, taking the pause time as the parameter. The authors point out that simulating the same protocol in different simulators may produce differences in the results. The performance of two location-based routing protocols for ad hoc networks is investigated by using ns2 and the effect of average moving speed in different scenarios is presented in [21]. An adaptive distance vector routing algorithm is proposed in [22], and its performance, compared with AODV and DSR, is studied. The offered traffic load and the simulation time are the input parameters.

Our work is to comprehensively investigate the characteristics of proactive and on-demand approaches by studying DSDV and AODV. In addition to identifying the suitable network contexts for each approach, we explore the causes for performance degradation. Based on the investigation, a new distance vector based routing protocol is proposed.

The rest of the chapter is organized as follows. In Section 3.3, the correlation between topology change and mobility is investigated. Section 3.4 describes the simulation environment, including the mobility, traffic, energy models, and performance metrics. Section 3.5 presents the experiment results and analysis. Improvements of DSDV are discussed in Section 3.6. Section 3.7 introduces the proposed CADV routing protocol and presents preliminary results of performance comparison of CADV, DSDV, and AODV. Section 3.8 concludes this chapter.

3.3 Correlation between topology change and mobility

The performance of a routing protocol is effected by the rate of topology change (i.e., the speed at which a network's topology is changing). The topology change can be represented as link change or route change. It is difficult to control the either of them directly in simulations. Our study demonstrates that:

- The link change and route change can be perfectly fitted into linear functions of the maximum speed when the pause time is 10 seconds.

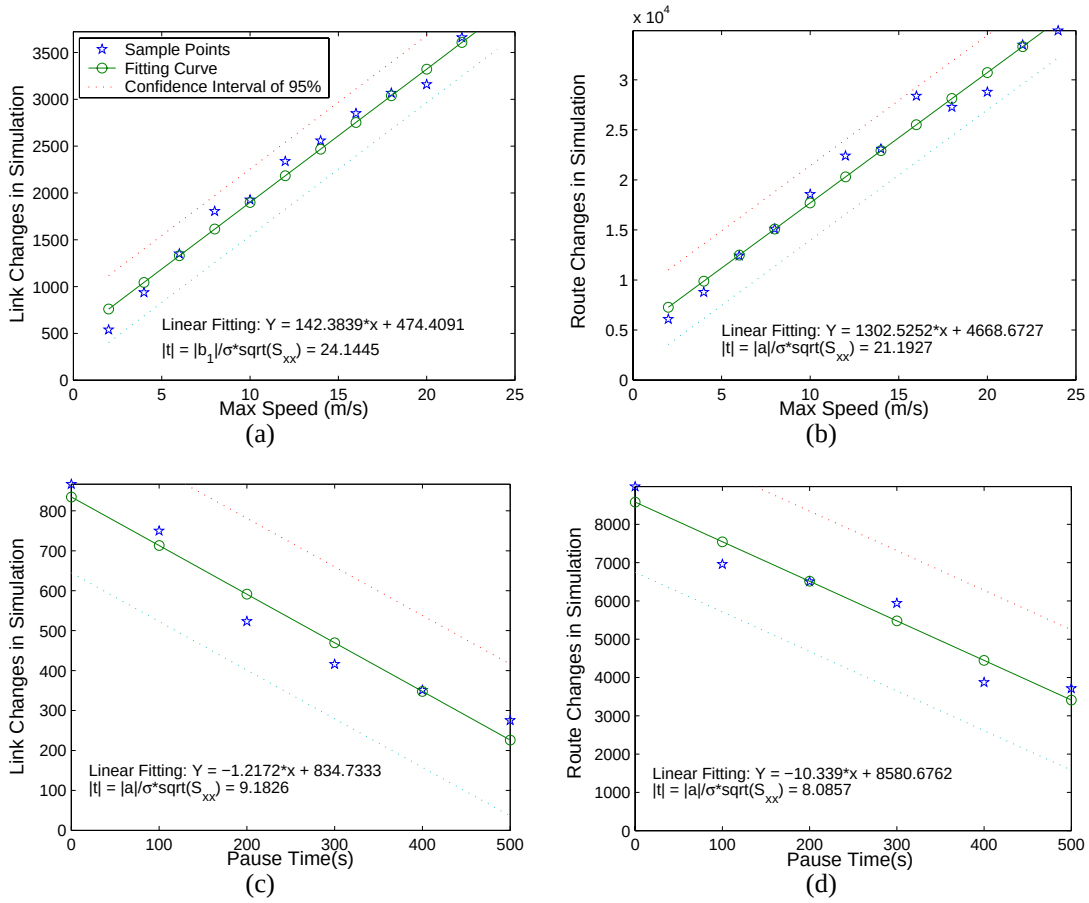


Figure 3.1. Topology change vs. mobility

- The link change and route change can be perfectly fitted into linear functions of the pause time when the maximum speed is 4 m/s.

Thus, the topology change can be indirectly controlled by varying mobility.

As shown in Figure 3.1a and 3.1b, the maximum speed is treated as the predictor variable, and link changes and route changes as the response variables (with the pause time to be 10 seconds). The fitting curve is obtained by using linear regression with least squares [23].

$$\begin{aligned} \hat{Y} &= b_0 + b_1 X \\ b_1 &= \frac{\sum_{i=1}^{n} X_i Y_i - n\overline{XY}}{\sum_{i=1}^{n} {X_i}^2 - n\overline{X}^2} \end{aligned}$$

$$b_0 = \overline{Y} - b_1\overline{X}$$

If we assume that the variations of the sample points about the line are normal, we can test the null hypothesis:

$$H_0 : b_1 = 0$$

using the *t-test* [23].

$$t = \frac{b_1\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2}}{\hat{\sigma}}$$
$$\hat{\sigma^2} = \frac{\sum_{i=1}^{n}(Y_i - \hat{Y_i})^2}{n-2}$$

For the link changes versus the maximum speed, $|t|$ = 24.1445. For the route changes versus the maximum speed, $|t|$ = 21.1927. Both of them exceed the appropriate critical value of $t_{0.995}(10)$ = 3.169 [1] (because 12 sample points are used for the linear regression, the degree of freedom is 10 = 12 - 2). Thus the hypothesis H_0 that the linear relationships between the link changes and the maximum speed, the route changes and the maximum speed do not exist is rejected with 99% confidence. The dotted lines in Figure 3.1 indicate the confidence interval of 95%. In plain words, the values of the link changes and the route changes lie within the specified intervals, respectively, and the statement is made with 95% confidence.

Figure 3.1c and 3.1d show the linear regressions of the link change versus the pause time and the route change versus the pause time. H_0 hypothesis is also verified with t-test. Because only 6 sample points are used, the degree of freedom is 4. $t_{0.995}(4)$ = 4.604, while the observed $|t|$ is 9.1826 and 8.0857 respectively. Thus H_0 is rejected with 99% confidence as well. The dotted lines in figure 3.1c and 3.1d show the confidence intervals of 95%.

3.4 Simulation settings and performance metrics

The constant bit rate (CBR) traffic is used in the simulation. Each connection is specified as a randomly chosen source-destination (S-D) pair. The packet sizes are fixed as 512

[1] The percentage points for the t-distribution are obtained from [23], using the two-tailed table.

Table 3.1
Power requirements

State	Documented Requirements	Measured
suspended	0.00 W	0.00 W
receiving	1.48 W	1.52 W
transmitting	3.00 W	3.10 W

bytes. The packet sending rate is 4 packets per second. Each connection starts at a time randomly chosen from 0 to 100 seconds.

Every host has an initial energy level at the beginning of a simulation. For every transmission and reception of packets, the energy level is decremented by a specified value, which represents the energy usage for transmitting and receiving. When the energy level goes down to zero, no more packets can be received or transmitted by the host. According to the manufacturer specifications [7], the power requirements of the WaveLAN card are shown in Table 3.1, column 2. Column 3 shows the actual power requirements measured in [24], without any power management mechanism. In the simulations, we use the values in column 3. We let the initial energy of each host to be 4000 joules so that the energy level does not reach zero in the simulation period.

The following four quantitative metrics are used to assess the performance:

- *Delivery Ratio*: The ratio of the data delivered to the destinations (i.e., throughput) to the data sent out by the sources.

- *Average End-to-end Delay*: The average time it takes for a packet to reach the destination. It includes all possible delays in the source and each intermediate host, caused by routing discovery, queueing at the interface queue, transmission at the MAC layer, etc. Only successfully delivered packets are counted.

- *Protocol Overhead*: The routing load per unit data successfully delivered to the destination. The routing load is measured as the number of protocol messages trans-

mitted hop-wise (i.e., the transmission on each hop is counted once). A unit data can be a byte or a packet.

- *Power Consumption*: The total consumed energy divided by the number of delivered packets. We measure the power consumption because it is one of the precious commodities in mobile communications. Wireless devices may consume over 50% of total system power for current handhold computers, and up to 10% for high-end laptops [24]. This poses challenging demands on the design of power-efficient routing protocols.

In the simulation, five scenarios are generated using the random waypoint model for each experiment, and the average values are used for analysis.

3.5 Results and analysis

To comprehensively measure the performance of a protocol, various network contexts are considered. The following parameters are varied in the simulation.

- *Host mobility* is determined by the maximum speed (with 10 seconds pause time).
- *Traffic load* is the number of the CBR connections.
- *Network size* is measured as the number of mobile hosts. Since the simulation field is fixed, the network size also measures the density of mobile hosts.

3.5.1 Varying maximum speed

This set of experiments study the impact of mobility on the performance metrics. The number of mobile hosts and the number of connections are both 30. The maximum speed ranges over $\{4, 8, 12, 16, 20, 24\}$ m/s.

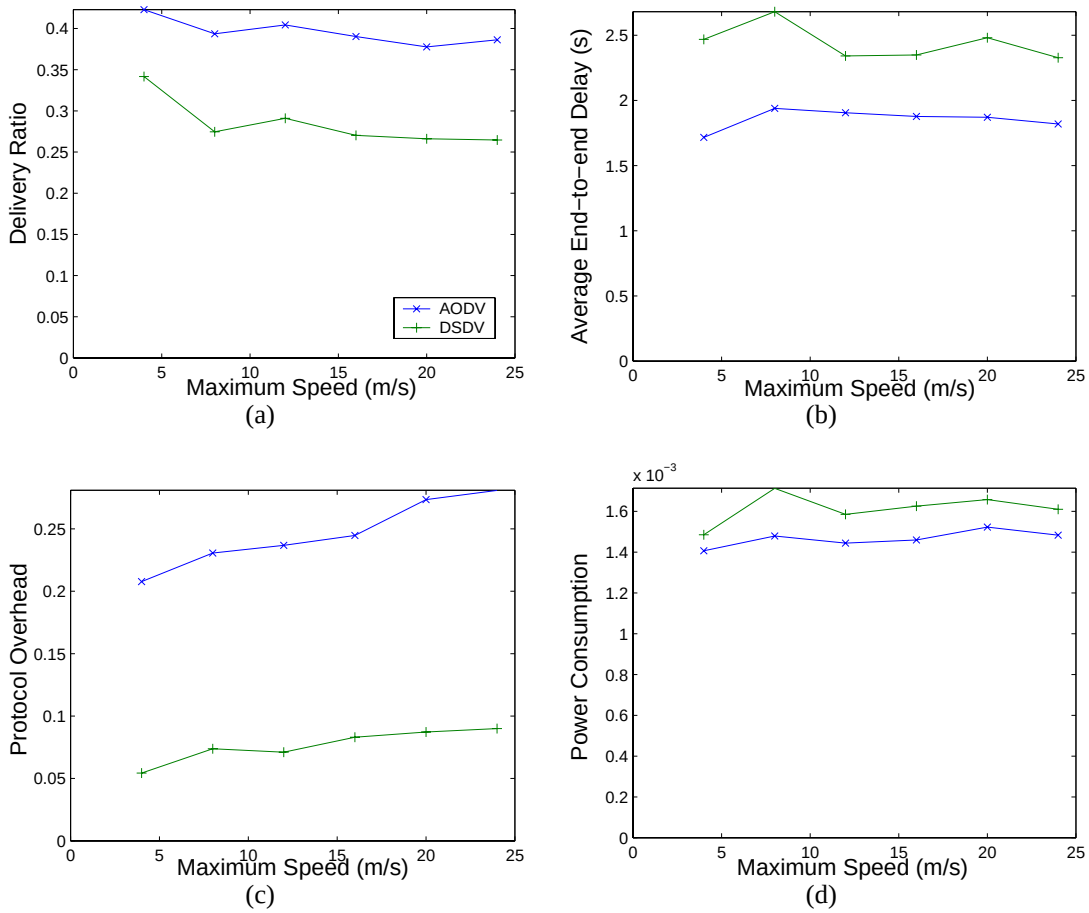


Figure 3.2. Varying maximum speed

As Figure 3.2a shows, the packet delivery ratios for both protocols are less than 50% [2]. When mobility is low (i.e., the maximum speed is 4 m/s), AODV delivers about 43% of total packets, while DSDV delivers about 34%. As the mobility increases, the delivery ratios of both protocols drop gradually, but DSDV has a little bigger drop.

It is interesting that DSDV has a higher delay than AODV does in all cases, which seems to contradict to the advantage of the proactive approach. It results from the implementations of the protocols. Although both implementations apply the drop-tail approach

[2]The implementation of IEEE 802.11 has been revised in ns2 since version 2.1b9. The default wireless bandwidth is set to 1 Mb/s. It, however, does not affect the performance comparison in this chapter, because it has same impact on different routing protocols.

for packet queues, AODV poses a limit on the time a packet can be queued, which currently is 30 seconds. Thus the delay of any received packet is bounded. DSDV keeps packets in queues no matter how long they have stayed. It delivers the older packets rather than the younger ones, and therefore increases the average delay.

Because a DSDV protocol packet contains many routes, while an AODV protocol packet contains at most one route (e.g., RREQ), we compare the byte-wise protocol overhead. DSDV introduces a significantly (3-4 times) lower protocol overhead than AODV (Figure 3.2c). The bad performance of AODV results from the following factors:

- Each host discovers routes individually.

- Unicasting RREP to the originator of the RREQ prevents valuable routing information from being propagated to other hosts.

- AODV treats network topology as a directed graph. It might need to discover two different directions for the same path twice due to a short reverse route lifetime.

As illustrated in Figure 3.2d, the power consumptions for both protocols are rather stable. Although DSDV introduces a much lower protocol overhead, it consumes more power. AODV “wins” in the way it handles link breaks. When a broken link of a route is detected, a route error (RERR) packet is sent to the source. Every host along the path notices the broken link immediately, and drops or queues packets locally. DSDV treats a broken link as a significant routing information and triggers a routing update. There is a minimum time interval between two triggered updates. The information about a broken link is delayed at each host. In the meantime, those hosts that have not received this information keep sending packets that will be dropped eventually to their next hops. A remarkable amount of power is consumed unnecessarily.

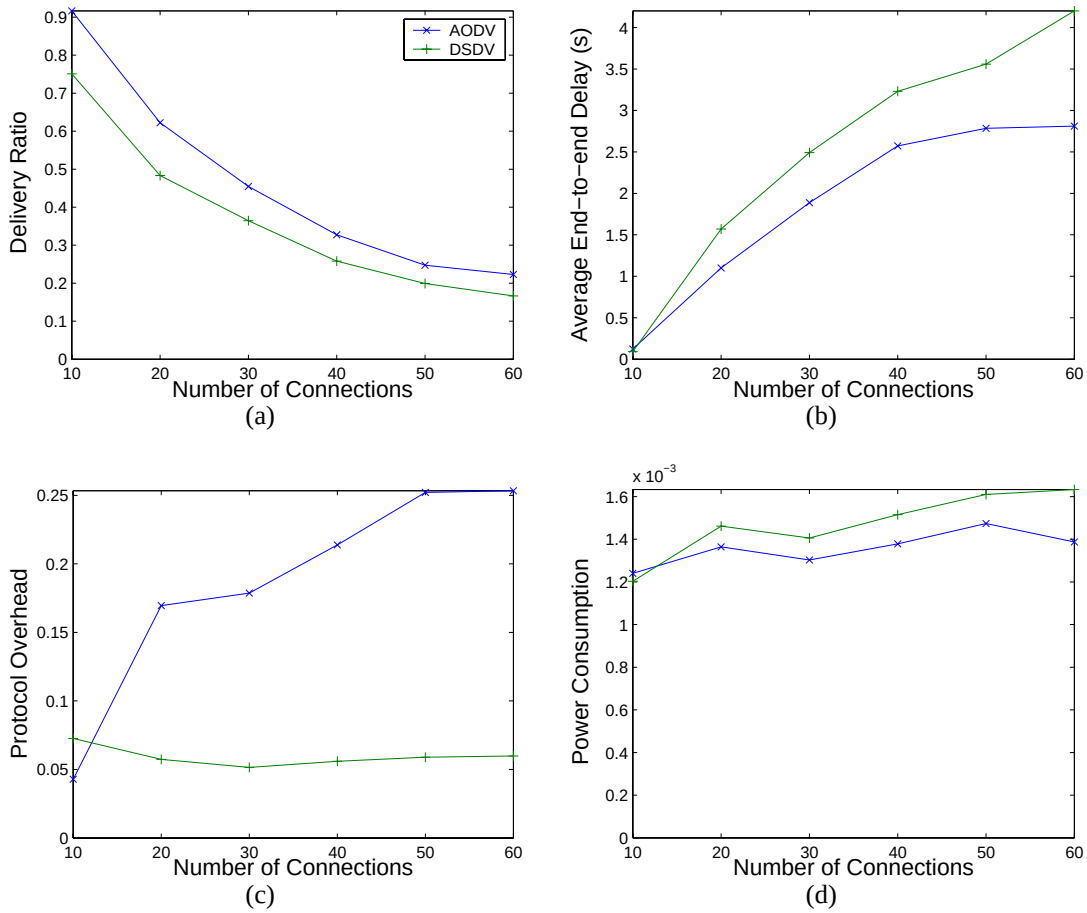


Figure 3.3. Varying number of connections

3.5.2 Varying number of connections

The next set of experiments demonstrate the effect of the traffic load. The number of mobile hosts is 30, the maximum speed is 4 m/s, and the pause time is 10 seconds. The number of connections varies from 10 to 80, increasing by 10 each time.

The delivery ratio of AODV (Figure 3.3a) drops dramatically from more than 90% to about 28% when the number of connections increases from 10 to 50, while that of DSDV drops from about 80% to about 20%. For more than 50 connections, the ratios of both DSDV and AODV drop more gradually because the network has already been fully loaded.

As Figure 3.3b shows, for 10 connections, DSDV and AODV have similar delay. The delays for both protocols increase rapidly with the number of connections (from about 0.1 second to 3 and 2.5 seconds for 40 connections, respectively). After the number of connections reaches 40, the delay of AODV grows gradually, while that of DSDV increases almost as fast as before.

For DSDV, the number of protocol packets is determined mostly by the network size and mobility. The protocol overhead stays fairly stable at 0.06 with an increasing number of connections (figure 3.3c). The protocol overhead of AODV increases sharply as the number of connections increases. AODV performs better than DSDV at 10 connections. At 80 connections, the protocol overhead for AODV is about 4 times higher than for DSDV.

As shown in Figure 3.3d, DSDV consumes more power than AODV does except for 10 connections. The power consumptions for both protocols increases gradually from 10 connections to 80 connections (the increase is about 50% for DSDV, and about 25% for AODV).

3.5.3 Dropped packets

Since the delivery ratio drops dramatically with an increase in traffic load, we are interested in investigating the reasons for packet drop. We check this by studying the ns2 trace files.

Figure 3.4 shows the number of packets dropped for four reasons. A packet is dropped due to congestion if the packet buffer at MAC layer is full when it arrives. When a collision is detected, CSMA/CA does a exponential backoff, which increases the delay for sending the packet. It makes the packet buffer to be full quickly.

For DSDV, no packet is dropped due to “no route” to the destination. It is guaranteed by the design the protocol. For AODV, the number of packets dropped due to “no route” increases from 2000 to 10000, as shown in Figure 3.4a.

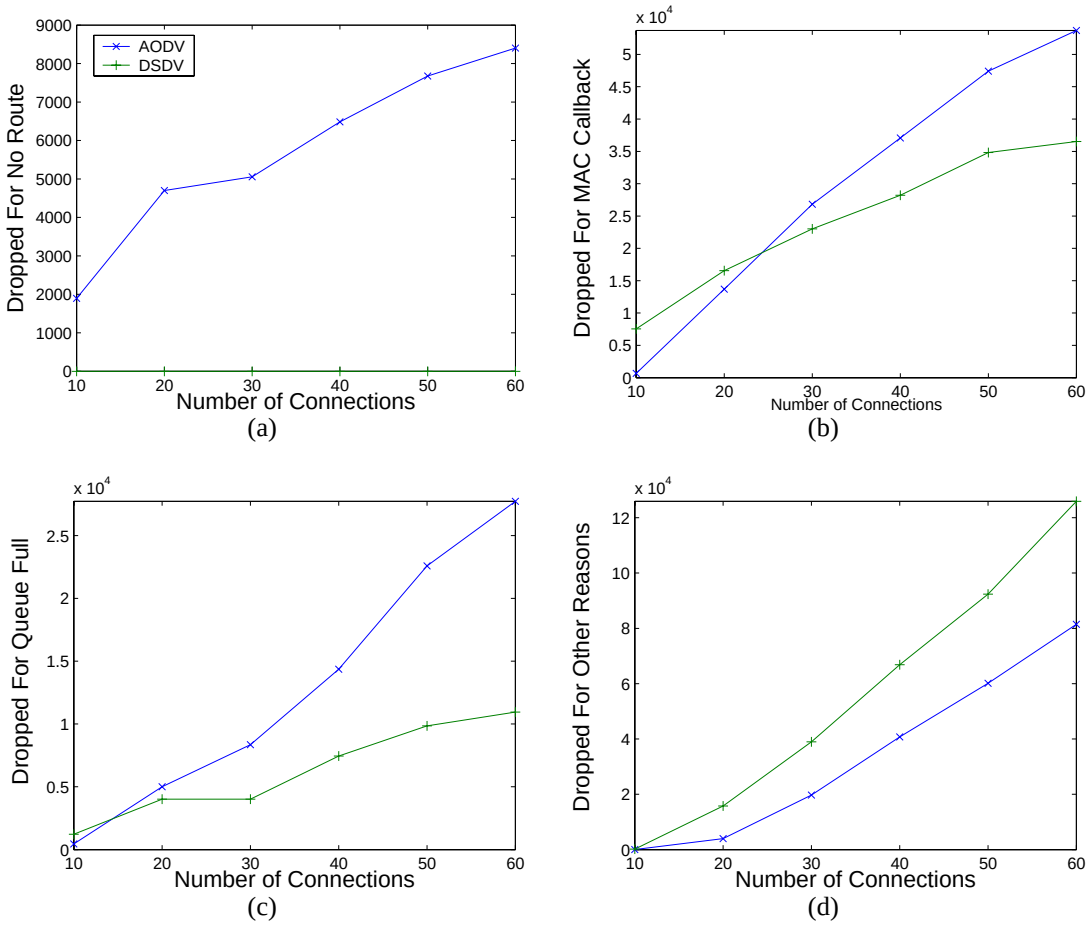


Figure 3.4. Dropped packets

As Figure 3.4b and 3.4c show, for 10 connections, AODV almost does not drop packets due to a MAC callback (i.e., the next hop is not a neighbor now), or queue being full. However, the number of packets dropped for AODV increases with the number of connections at a rate higher than DSDV. DSDV drops fewer packets than AODV for the above two reasons in most cases.

From Figure 3.4, we can calculate that more than half of the dropped packets result from congestion. DSDV performs better for the first three reasons, but worse than AODV for avoiding congestion. Although both DSDV and AODV do not utilize any congestion control or avoidance mechanism to balance traffic load, AODV in fact distributes the data

traffic more evenly in the network. AODV tries to build the shortest route when it originates a request, but it keeps the route as long as it does not break, even if a shorter route is available at a later time. In contrast, DSDV tends to always send packets via the shortest routes. Forwarding packets through the shortest routes will likely push traffic to several heavily burdened hosts and congest the network.

3.5.4 Varying number of mobile hosts

The last set of experiments investigate the effect of the network size. All hosts move randomly at the maximum speed of 4 m/s. The pause time between two movements is 10 seconds. The number of mobile hosts increases from 20 to 70 by 10s. The number of connections is equal to the number of hosts.

The delivery ratio of AODV decreases faster than that of DSDV with the number of mobile hosts (Figure 3.5a). AODV has a better performance in a sparser network (fewer than 40 hosts), and worse performance in a denser one. Figure 3.5b indicates that AODV outperforms DSDV in terms of end-to-end delay.

DSDV and AODV have similar protocol overhead for 20 mobile hosts. Both of them introduce more overhead as the number of hosts increases, with the overhead for AODV growing faster than for DSDV (Figure 3.5c).

Both DSDV and AODV have similar power consumption in a sparse network (Figure 3.5d). For DSDV, the increase of power consumption is nearly linear with the host number. The power consumption for AODV increases faster than for DSDV. For 70 hosts, AODV consumes 33% more energy than DSDV does per 1k-byte delivered data.

From the results provided in Figure 3.5, we can tell that DSDV is more scalable with respect to the number of hosts. It seems that 40 hosts per square kilometer is the turning point. For more than 40 hosts, DSDV equals or outperforms AODV for all metrics (the average delay is an exception that should not be considered).

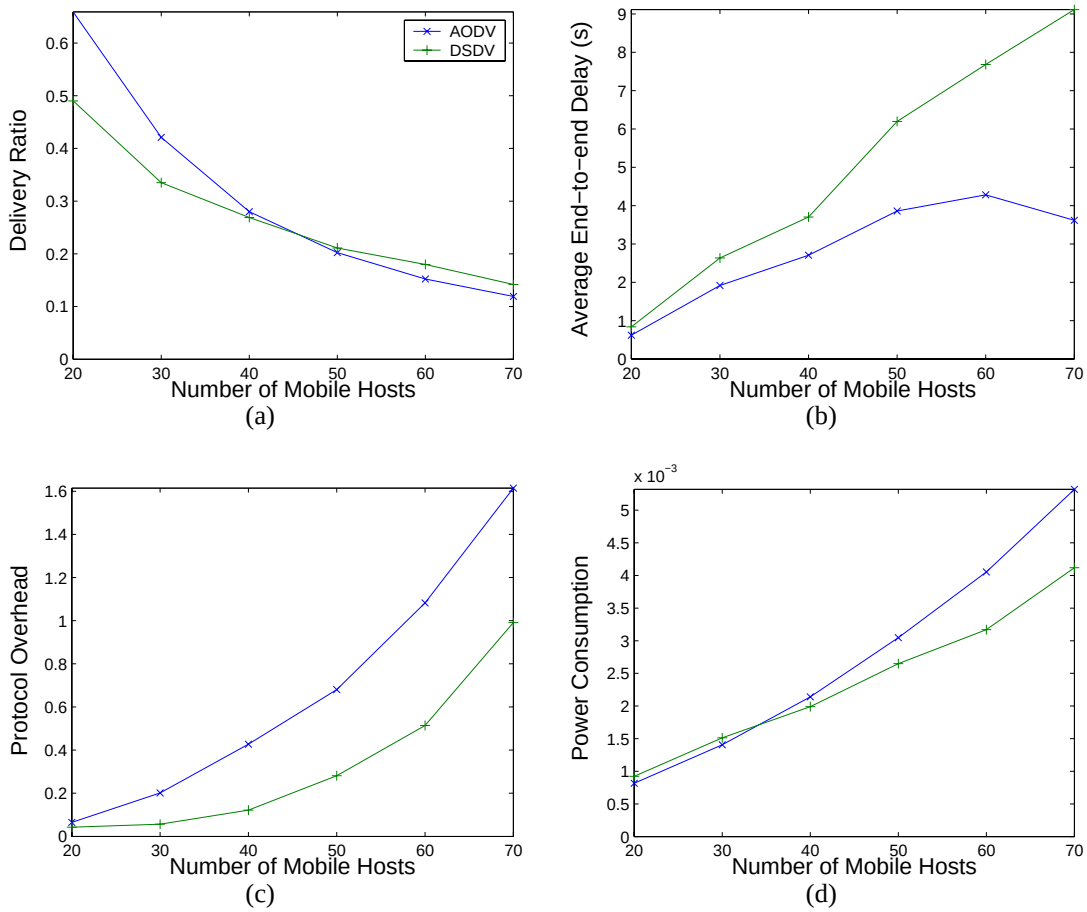


Figure 3.5. Varying number of mobile hosts

3.6 Further discussion about DSDV

3.6.1 Reduce broadcast interval of DSDV

The time interval between broadcasting routing information is one of the most important parameters of DSDV [1]. As shown in figure 3.4, in total about $5.5 * 10^4$ packets are dropped for 80 connections due to either a MAC callback or a full queue, which means that the outgoing links are broken or the routes are not established timely. Some of these situations could be avoided by broadcasting routing information more frequently, at the cost of

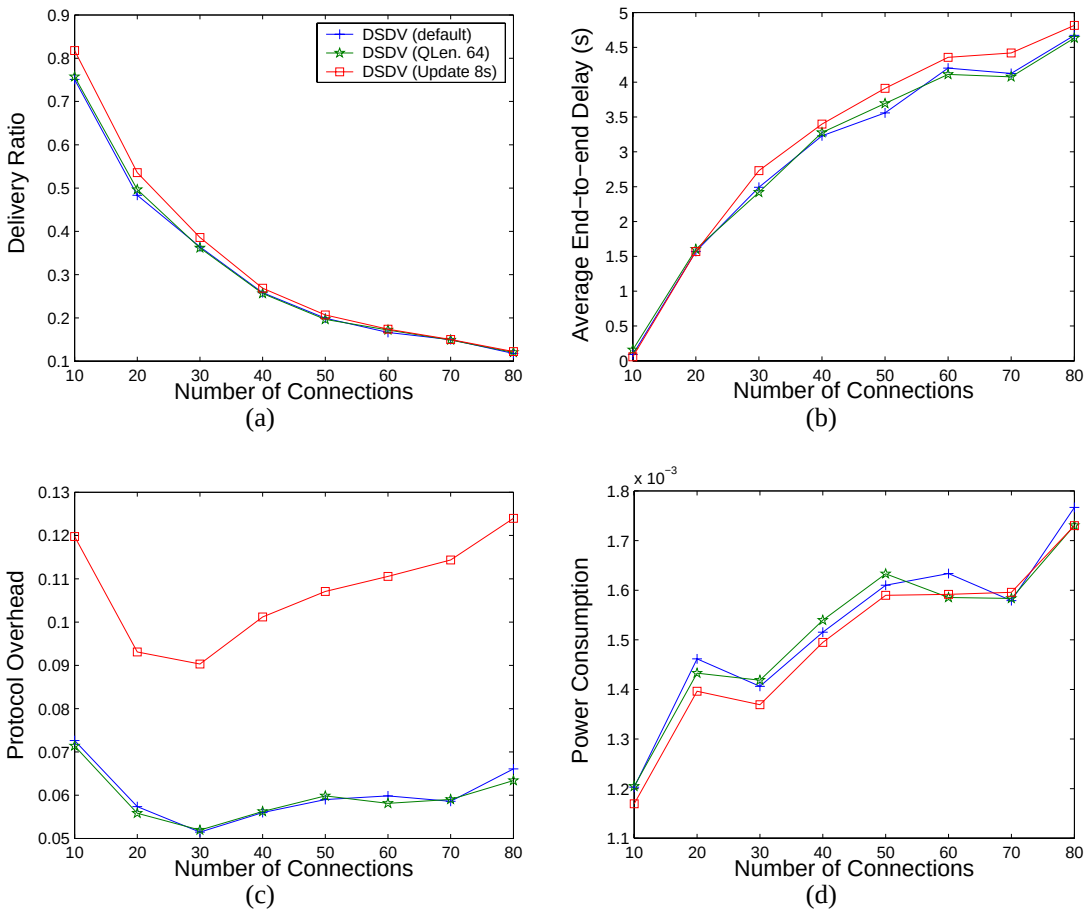


Figure 3.6. Performance comparison of different DSDV implementations

a higher protocol overhead. The questions are: How much improvement of performance can be obtained? How much will it cost?

We reduced the broadcast time interval from 15 seconds to 8 seconds, and rerun the set of experiments described in Section 3.5.2, using the same settings, parameters, scenarios, and connections.

Figure 3.6a (the "Update 8s" curve) shows that the throughput increases about 10% for less stressful cases (i.e., for fewer than 50 connection). The average delay is almost the same (Figure 3.6b). The protocol overhead doubles as we expect (Figure 3.6c). The

power consumption slightly decreases, because packets are dropped earlier as we explain in Section 3.5.1.

3.6.2 Increase the queue length of DSDV

Figure 3.4c shows that about $1.5 * 10^4$ packets are dropped due to a full queue. Since the queue length for DSDV is only 5, much smaller than that for AODV, it is natural to ask this question: Will a longer queue increase the throughput of DSDV?

We set the queue length to 64 and rerun the set of experiments again. The results are shown in Figure 3.6 (the "QLen. 64" curve). The performance metrics are almost the same as those measured for the original DSDV implementation. Thus, the longer queue does not help in improving performance of DSDV.

3.7 Congestion-aware routing protocol – CADV

Although the published result [25] showed that on-demand protocols outperform proactive protocols and are better suited for mobile ad hoc networks, the proactive protocols have the following advantages.

- *Better support for Quality of Service (QoS):* Proactive protocols timely propagate network conditions (available bandwidth, delay, etc.) throughout the system, so that appropriate QoS decisions, including admission control, traffic shaping, and route choosing, can be made.

- *Better support for anomaly detection:* Proactive protocols constantly exchange the network topology information. It enables real-time detection and reaction to malicious behaviors and attacks such as the false distance vector attack and the false destination sequence attack [26, 27].

As shown in Section 3.5.4, DSDV performs better than AODV in denser networks, which demonstrates potential scalability of the proactive approach with respect to the number of mobile hosts. Figure 3.4 reveals that this approach is plagued by congestion, the dominant

reason of performance decrease. To address the congestion issues, we propose a new proactive distance vector based ad hoc routing protocol called congestion-aware distance vector (CADV).

3.7.1 Overview

A mobile host in an ad hoc network can be viewed as a single server queueing system. The delay of sending a packet is positively correlated with congestion. In CADV, each routing entry is associated with an *expected delay*, which measures congestion at the next hop. Every host estimates the expected delay based on the mean of delay for all data packets sent in a past short period of time. Currently, the length of the period is equal to the interval between two periodical updates. The expected delay is computed as $E[D] = \frac{\sum D_i}{n} L$, where n is the number of sent packets and L is the length of MAC layer packet queue. $E[D]$ estimates the time a newly arrived packet has to wait before it is sent out. When a host broadcasts an update to neighbors, it specifies the delay it may introduce. A routing decision is made based on the distance to the destination as well as the expected delay at the next hop. CADV tries to balance traffic and avoid congestion by giving priority to a route having low expected delay. For example, hosts A and B both advertise a route to the destination. If the expected delay at host A is significantly less than that at host B, A will be chosen as the next hop (given B is not A's next hop), even if the route via A is one hop longer than the one via B. When making routing decisions, a function $f(E[D], distance)$ is used to evaluate the value of a route. Various routing policies can be implemented by replacing this function.

A CADV routing module consists of three components: (a) *Traffic Monitor* monitors traffic going out through the link layer. It keeps track of the average delay for sending one data packet in recent period of time. The time period is specified by the route maintenance component. (b) *Traffic Control* determines which packet is the next to send or drop, and reschedules packets if needed. It supports a drop tail FIFO queue and provides functionality to re-queue packets. (c) *Route Maintenance* is the core component. Its functionalities

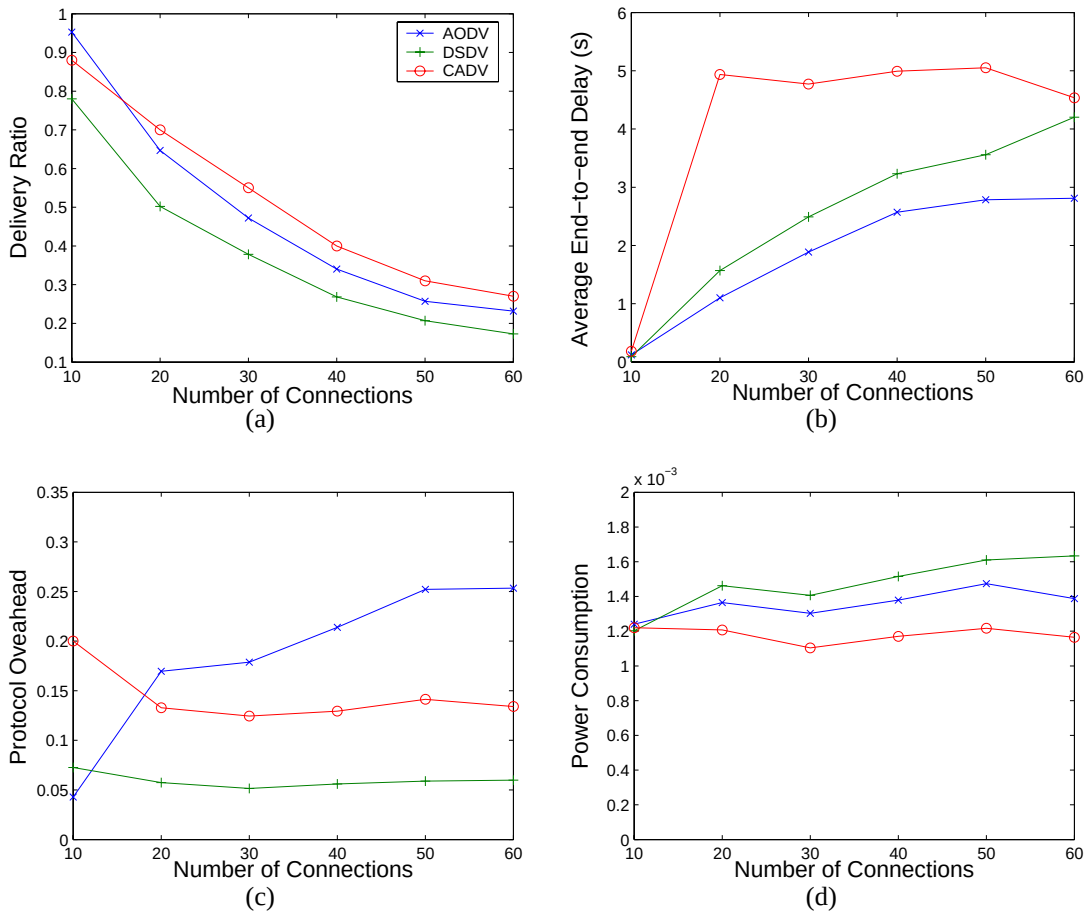


Figure 3.7. Comparison of three protocols

include exchanging information with neighbors, evaluating and maintaining routes, managing the traffic monitor and traffic control components.

3.7.2 Preliminary results

A preliminary study is conducted to investigate the performance of CADV with the number of connections. The maximum speed is 4 m/s and the number of mobile hosts is 30. Figure 3.7 illustrates the performance comparison of CADV, DSDV, and AODV. AODV performs better than CADV only for 10 connections, where congestion is not likely to occur. For other cases, as shown in Figure 3.7a, CADV outperforms AODV by about

5% in terms of packet delivery ratio. The tradeoff for the improvement is shown in Figure 3.7c. CADV introduces about 2.5 times protocol overhead as DSDV does. However, the protocol overhead is still lower than that introduced by AODV when the number of connections is greater than 10. CADV introduces higher end-to-end delay than AODV and DSDV when the number of connections is greater than 10 (figure 3.7b), because it may choose longer route to forward packets. The delay is rather stable with the increase of the number of connections. Figure 3.7d shows that CADV consumes less power. It results from packet rescheduling done by the traffic control component. When a neighbor becomes unreachable, all packets in the MAC layer packet buffer whose next hop is that neighbor will be rescheduled. This mechanism saves power by preventing a host from sending unnecessary Request-To-Send (RTS) messages.

3.8 Conclusion

Conclusion 1: For the movements of mobile hosts generated by the random waypoint model, with a very high probability, the link change and route change are, linear functions of the maximum speed, and linear functions of the pause time, respectively. The maximum speed does not affect much the performance of DSDV and AODV at the range from 4 m/s to 24 m/s.

Conclusion 2: In less stressful situations, AODV outperforms DSDV for all metrics except for normalized protocol load. DSDV performs better than AODV does in denser networks with a higher traffic load. In general, we can state: (1) The protocol load for the proactive routing protocols (such as DSDV) grows as the number of hosts increases, while that of the on-demand routing protocols (such as AODV) increases with the number of source-destination (S-D) pairs. The proactive approach performs better when the number of S-D pairs is close to the number of hosts. (2) The on-demand approach consumes less power, because it propagates the link break information faster, thus it avoids sending packets that are dropped eventually. (3) Network congestion is the dominant reason for packet drop for both proactive and on-demand approaches.

Conclusion 3: The preliminary study of CADV routing protocol demonstrates that the performance of proactive routing protocols can be improved by integrating with congestion avoidance mechanisms. Currently, only delay at the next hop and distance to the destination are considered when making routing decisions. We are working towards a complete version of CADV that takes advantage of other information such as available queue length, delay on a path, etc. A comprehensive study will be conducted to investigate how different congestion predication and load balancing mechanisms can cooperate with CADV to reduce congestion in ad hoc networks.

4 PACKET LOSS IN AD HOC NETWORKS

4.1 Introduction

Throughput is generally accepted as one of the most important metrics to evaluate the performance of a routing protocol. Several simulation-based performance comparisons have been done for ad hoc routing protocols in the recent years. S.R. Das et al. evaluate performance of ad hoc routing protocols based on the number of conversations per mobile node [17]. The performance comparison of two on-demand routing protocols: dynamic source routing (DSR) [3] and AODV [2] is presented in [8]. The performance of two location-based routing protocols for ad hoc networks is investigated in [21]. An adaptive distance vector routing algorithm is proposed in [22], and its performance, compared with AODV and DSR, is studied. Although various throughput results in different network contexts have been obtained, the causes for throughput variation in ad hoc networks have not been deeply understood. Packet loss is one thrust to study throughput, since throughput is determined by how many packets have been sent and how many packets have lost.

Packet loss in wired network has been investigated. For example, a single server queueing system with a finite buffer capacity is used to analyze packet loss processes in high-speed networks in [28]. The end-to-end packet delay and loss behaviors in the Internet are studied using the UDP echo tool in [29]. These work target at the packet loss due to buffer overflow (congestion), which is the major loss in wired networks.

Packet loss problem is much more complicated in mobile ad hoc networks, because wireless links are subject to transmission errors and the network topology changes dynamically. A packet may lose due to transmission errors, no route to the destination, broken links, congestions, etc. The effects of these causes are tightly associated with the network context (e.g., host mobility, number of connections, traffic load, etc.). Even building an approximate model to analytically evaluate packet loss is difficult. We investigate the

problem via simulations. Data is gathered from more than 1000 individual experiments to estimate the desired true characteristics of packet loss in ad hoc networks.

In mobile ad hoc networks, wireless link transmission errors, mobility, and congestion are major causes for packet loss. Packet loss due to transmission errors is affected by the physical condition of the channel, the terrain where networks are deployed, etc. They can not be eliminated or reduced by improving the routing protocols. This chapter only addresses congestion-related and mobility-related packet loss. Congestion in a network occurs whenever the demands exceed the maximum capacity of a communication link, especially when multiple hosts try to access a shared media simultaneously. Mobility may cause packet loss in different ways. A packet may be dropped at the source if a route to the destination is not available, or the buffer that stores pending packets is full. It may also be dropped at an intermediate host if the link to the next hop has broken. We study the effect of congestion and mobility on packet loss in various network contexts. AODV and DSDV are chosen as representatives of on-demand and proactive routing protocols respectively.

This work can benefit the design of routing and flow control algorithms, the dimensioning of buffers, identifying and avoiding the performance bottleneck of current routing protocols, and choosing proper parameters in future simulation and analytic studies.

The rest of the chapter is organized as follows. Section 4.2 introduces the related work. The simulation settings, including traffic, routing protocols, congestion-related and mobility-related packet loss, are discussed in Section 4.3. Section 4.4 presents two sets of experiments and the results. The relations between the shortest path and congestion is discussed in section 4.5. Section 4.6 concludes the chapter.

4.2 Related work

There has been some recent work on addressing packet loss issues in wireless networks. S. Biaz and N.H. Vaidya investigate the ability of three loss predictors to distinguish congestion losses from wireless transmission losses [30]. They use a wireless link with transmission loss rate r_w in the simulations. F. Anjum and L. Tassiulas analytically

study the performance of different TCP algorithms over a wireless channel with correlated packet losses [31]. A simple two-state Markov chain is used to model the correlated fading channel. T.V. Lakshman et al. also analyze the impact of random packet loss at a wireless link on the performance of TCP/IP in [32]. They indicate that bidirectional congestion increases TCP's sensitivity to loss. These efforts assume transmission losses on a single wireless link follow a simple model and focus on how losses effect the performance of TCP.

Even if wireless transmission is loss-free, packet loss still exists in ad hoc networks. Our work is to understand the major causes for packet loss and to capture its characteristics.

4.3 Simulation settings

4.3.1 Traffic

To investigate the impact of traffic load and congestion control mechanisms on packet losses, both unresponsive traffic and responsive traffic are studied.

- *Unresponsive traffic* only consists of UDP connections, each of which is specified as a source-destination (S-D) pair. Every source is associated with a constant bit rate (CBR) traffic generator, which sends out packets at the given rate. The source of each S-D pair is randomly chosen from all hosts, and the destination is randomly chosen from all hosts other than the source. All S-D pairs are mutually independent. The packet size is fixed at 512 bytes. The start time of each connection is uniformly distributed between 0 to 100 seconds.

- *Responsive traffic* is comprised of TCP connections. Each connection has a Tahoe TCP [1] sender and a TCPSink receiver. The sender window size is decreased by half when packet losses are detected. The retransmission starts from the first lost packet. Tahoe TCP enters the slow start when an ACK for a new packet is received.

[1]The TCP performs congestion control and round-trip-time estimation in a way similar to the version of TCP released with the 4.3BSD Tahoe UNIX system from UC Berkeley, so it is called Tahoe TCP.

Table 4.1
Packet loss at MAC and network layers

	Mobility-related	Congestion-related
MAC Layer	√	√
Network Layer	√	

All TCP packets have the same size of 512 bytes. The initial sender window size is 1 and the maximum bound on the window size is 32. TCPSink is responsible for returning ACKs to the sender. It generates one ACK per packet received. The ACK packet size is 40. The data of each connection is generated by an attached FTP application, which simulates a bulk data transfer. Every FTP application starts at a time randomly chosen from 0 to 100 seconds.

4.3.2 Differentiated packet losses

Packet loss is measured at all mobile hosts. Every host monitors the networking layer and the MAC layer for all kinds of packet losses. The layers of the protocol stack and the modules that are responsible for mobility-related and congestion-related packet loss are shown in Table 4.1.

Mobility-related packet loss may occur at both the network layer and the MAC layer. When a packet arrives at the network layer, the routing protocol forwards the packet if a valid route to the destination is known. Otherwise, the packet is buffered until a route is available. A packet is dropped in two cases:

- The buffer is full when the packet needs to be buffered.
- The time that the packet has been buffered exceeds the limit. (The AODV implementation in ns2 poses a 30-second limit on the time a packet can be buffered. The DSDV implementation does not have a limit.)

The MAC layer mobility-related packet loss occurs when the next hop of a packet is out of range at the moment the packet is transmitted. The reason is that the routing information is obsoleted. It occurs frequently in a high mobility network than in a low mobility network.

Congestion-related packet loss only occurs at the MAC layer. Because CSMA/CA is used in the simulation, a packet may be dropped due to congestion for two reasons:

- The wireless channel is so busy that the number of *backoff* procedures exceeds the limit.
- The channel is associated with a queue that buffers all the packets waiting to be transmitted. A packet is dropped if the queue is full when it arrives.

4.4 Experiments

A series of experiments have been conducted to investigate the mobility-related and congestion-related packet losses in different network contexts. The network configuration for the experiments is a 1000m x 1000m square field with 30 hosts. The buffer size is 64-packet for each route and the MAC layer. Each data point in the result figures represents an average of 5 runs with identical traffic but different mobility scenarios, which are randomly generated with the same parameters (i.e., same maximum speed and pause time). Every experiment runs for at least 1000 seconds.

4.4.1 Varying mobility and the number of connections

The purpose of the first set of experiments is to study the impact of host mobility. The pause time is varied over the range of {0, 50, 100, 200, 300, 500} seconds. Zero pause time results in the highest mobility since hosts keep moving without a pause. For these experiments, 10, 20, and 30 connections, which represent light, moderate, and heavy communication requests respectively[2], are used. The packet sending rate for each connection is 4 packets/s. The results are shown in Figure 4.1.

[2]Traffic load is represented by the sending rate in this chapter. It has different effect on packet loss compared with communication request.

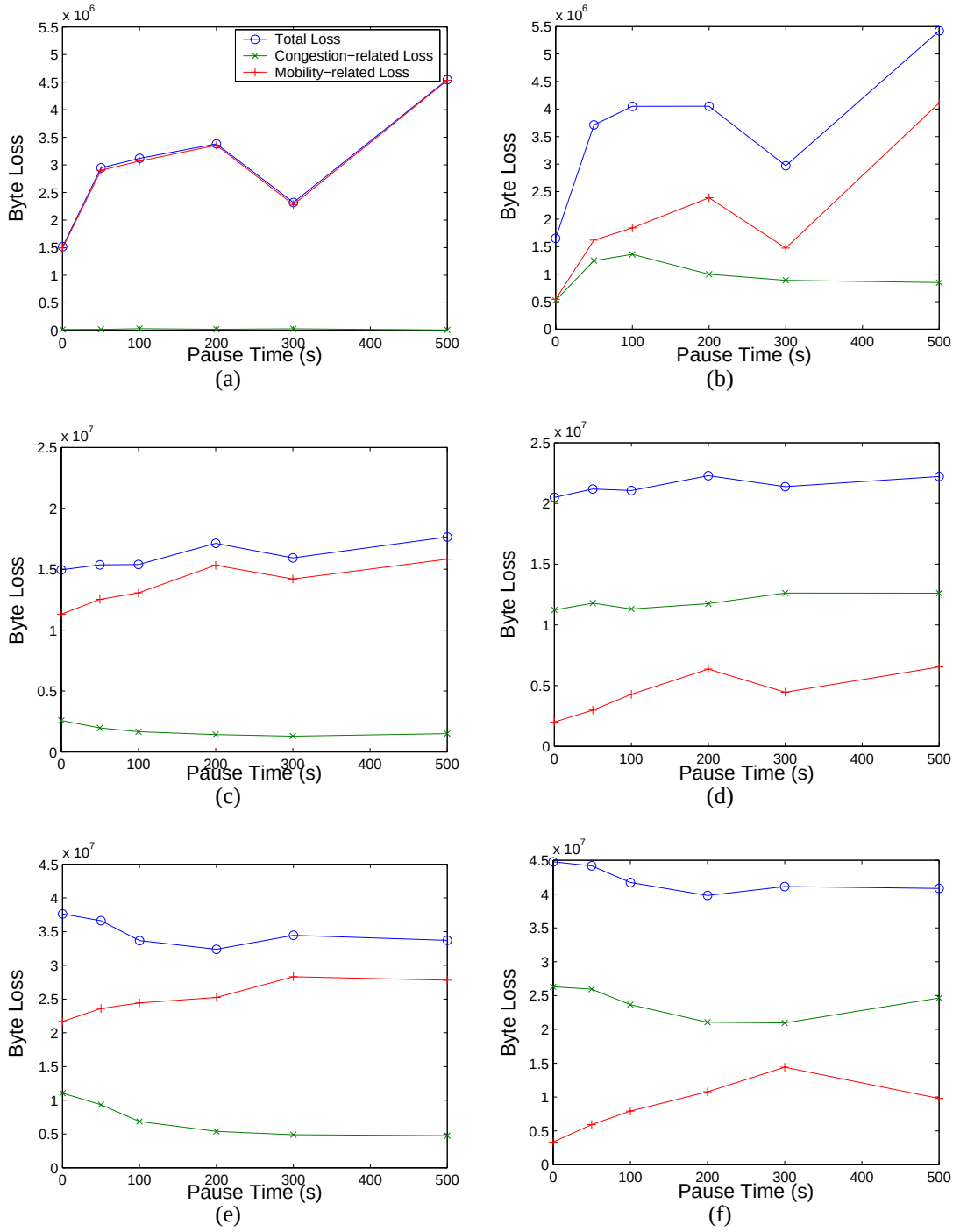


Figure 4.1. Packet loss for 4 packets/s CBR connections

Packet Loss for AODV

The total packet loss grows from about 3000 to 8000 with the increase of the pause time from 0 to 500 seconds for 10 connections, as shown in Figure 4.1a. In the case there are 20 connections, the total packet loss gradually increases by 10% (Figure 4.1c). For 30 connections, it gradually decreases by 10% (Figure 4.1e). As the communication request grows from 10 to 20, the total packet loss increases by 9 times when the pause time is 0 seconds, and by 3 times when the pause time is 500 seconds. The increase of the communication request from 20 to 30 results in doubled total packet loss.

There is almost no congestion-related packet loss when the communication request is 10. In the other two cases, the packet loss gradually decreases by about a half as the pause time increases from 0 to 500 seconds. From 10 to 20 and 30 connections, with no pause time, the packet loss increases to 5000 and 20000 respectively. The percentage with respect to the total loss increases as well, to 20% and 30% respectively.

Mobility is always the dominant cause for the packet loss. However, the majority decreases as the communication request increases. When the pause time is 0 seconds, the percentage of mobility-related loss decreases from about 100% to 70% and 60%, for 10, 20, and 30 connections. The absolute value and the percentage of the mobility-related packet loss increase with the pause time.

Packet Loss for DSDV

The growth of the total packet loss with the pause time for DSDV follows a similar pattern as that for AODV. For 10 connections, the total packet loss increases from about 3000 to 10000 as the pause time increases from 0 to 500 seconds (Figure 4.1b). It is nearly unchanged with the pause time for 20 and 30 connections as shown in Figure 4.1d and 4.1f (gradually increases by 5% for 20 connections and decreases by 5% for 30 connections). Increasing the communication request from 10 to 20 makes the total packet loss grow 10 times and 4 times for 0 and 500 seconds pause time, respectively. The increase of the communication request from 20 to 30, however, only doubles the total packet loss.

The percentage of the congestion-related packet loss increases with the communication request. Congestion begins to be the dominant cause for the packet loss after the communication request reaches 20 (it results in approximate 50% and 60% of the total packet loss with 20 and 30 connections, respectively). The loss is fairly stable with the pause time, but jitters exist.

The mobility-related packet loss increases with the communication request, but slower than the congestion-related packet loss.

Comparison between AODV and DSDV

The comparison of different packet losses for AODV and DSDV is as follows.

- *Total packet loss:* The total packet loss for DSDV is always 10% to 20% higher than that of AODV, regardless the pause time or the number of connections. For the moderate and heavy communication requests, the total packet loss for DSDV is more stable than that of AODV with the increase of the pause time.
- *Congestion-related packet loss:* DSDV loses more packets due to congestion than AODV. The gap of the congestion-related packet loss between DSDV and AODV decreases with the growth of the communication request.
- *Mobility-related packet loss:* AODV has more mobility-related packet loss than DSDV.

4.4.2 Varying traffic load and traffic type

The second set of experiments illustrate the effect of traffic load and traffic type. The pause time ranges over $\{0, 50, 100, 200, 300, 500\}$ seconds. 10, 20 and 30 connections are used. Both unresponsive traffic and responsive traffic are studied. The packet rate for CBR connections is 8 packets/s, which injects a reasonable heavy load to the network. We use the same mobility scenarios and connection configurations for this set of experiments as for the previous set of experiments to compare the results with the previous ones.

CBR connections with 8 packets/s

As shown in Figure 4.2, each curve that represents a different type of packet loss has similar shape compared with the corresponding one in the previous experiments, but flatter (i.e., increase and decrease are more gradual).

For AODV, mobility is still the major cause for the packet loss. Congestion plays a more important role compared with CBR connection with 4 packets/s rate. Increasing the number of connections has less effect in this set of experiments than in the previous one. From 10 connections to 20 connections, the total packet loss increases by only about 3 times. From 20 to 30 connections, the increase is less than 2 times. Comparing Figure 4.2a with Figure 4.1a, the total loss increases by 660% for 0 second pause time, and by 200% for 500 seconds pause time. For the moderate and heavy communication requests, the total packet loss is only tripled or doubled as the packet rate increases from 4 to 8 packets/s.

For DSDV, congestion dominates the packet loss even when there are only 10 connections. The total packet loss increases the same amount as that for AODV when the communication request increases, with respect to percentage. The total packet loss with traffic load is almost the same for DSDV and AODV.

For both AODV and DSDV, increasing the communication request has similar impact on the total packet loss (i.e., more losses) as increasing the traffic load. The increase of either parameters will result in decreasing the impact of the other parameter. Heavier communication request or traffic load introduces more congestion-related packet loss.

TCP connections

The number of bytes (the total size of all lost packets), instead of the number of packets, is used in the experiments with TCP connections. Because both the application data and the ACK packets, which have different sizes, are treated as data packets by the routing protocol, the number of bytes is more comprehensive than the number of packets.

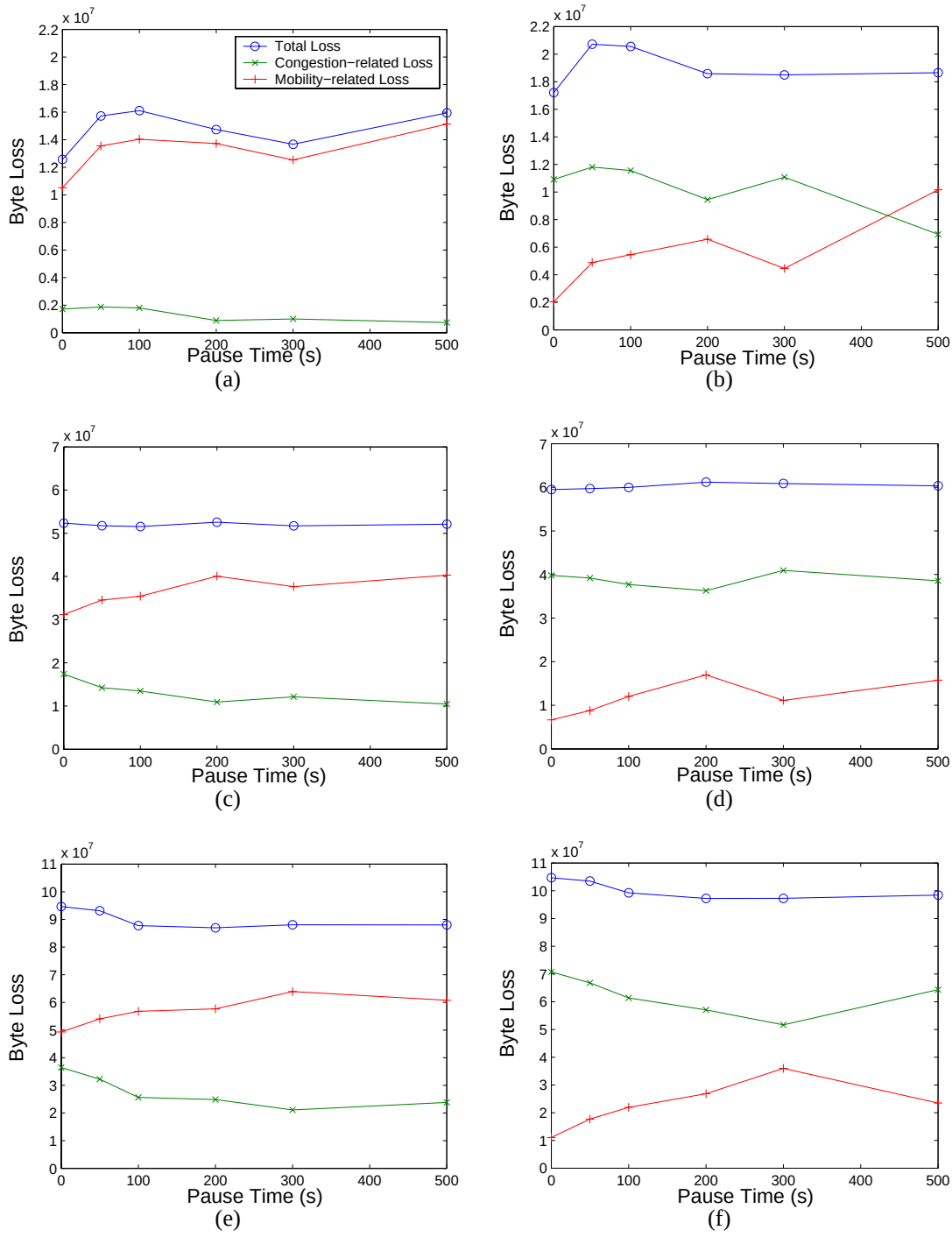


Figure 4.2. Packet loss for 8 packets/s CBR connections

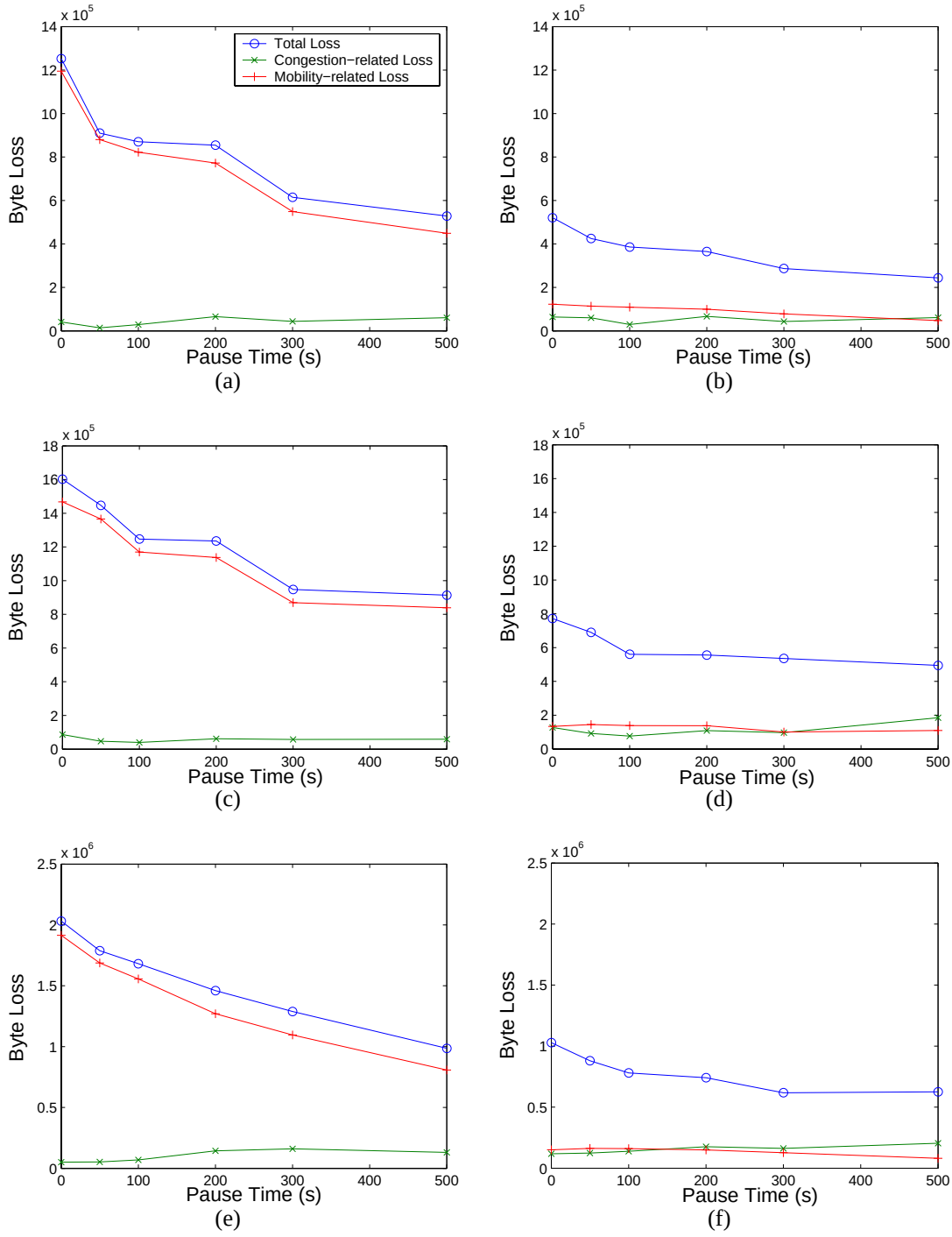


Figure 4.3. Packet loss for TCP connections

Figure 4.3 demonstrates byte loss in TCP connections[3]. It shows that the congestion-related loss for both protocols is greatly reduced by the congestion control mechanism. The total loss decreases with the decrease of mobility. DSDV outperforms AODV in terms of the total loss. The total loss of DSDV is only half of that of AODV in all test cases, because the effect of the major cause for DSDV to lose packets (i.e., congestion) is diminished by the congestion control mechanism.

For AODV, with the decrease of the congestion-related loss, more than 90% of the total loss is mobility-related. The total effect of mobility and congestion is less than 20% for DSDV.

To improve throughput, different routing protocols require different mechanisms to remedy the major causes for packet loss. Specifically, integrating congestion control techniques with DSDV will significantly improve the throughput, as shown in figure 4.3. For on-demand routing protocols like AODV, fast rediscovery of new routes will reduce the mobility-related packet loss, and gain higher throughput consequently. S.R. Das et al. proposed ad hoc on-demand multipath distance vector (AOMDV) protocol to decrease the route discovery latency [33]. Their results show that AOMDV loses 3-5% less packets than AODV. T. Goff et al. proposed preemptive routing maintenance algorithms for ad hoc networks [34]. The proactive route selection and maintenance are added to the on-demand protocols to reduce the cost in detecting the disconnection and establishing a new route.

4.5 Discussion

Figure 4.1 and 4.2 show that DSDV loses much more packets due to congestion than AODV. This difference may result from, with a very great chance, the different route maintenance schemes used by DSDV and AODV, because both protocols use distance vector to represent routing information and choose the routes based on the shortest paths. Since the per connection traffic load is much lighter (less than 8 packets/s, which is 32Kb/s) than the communication capacity of a host (2Mb/s), the occurrence of congestion indicates that

[3]The difference of the amounts of bytes sent by AODV and DSDV is smaller than 5%.

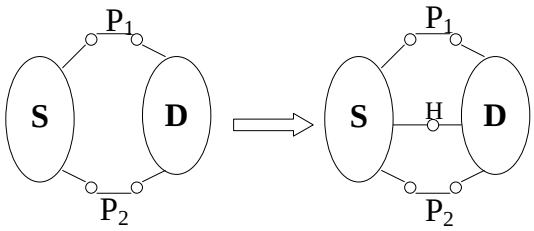


Figure 4.4. Shortest path and congestion

connections converge on heavily burdened hosts. The converged traffic load exceeds the capacity of those hosts. In a mobile ad hoc network, hosts keep moving. The shortest path between a source and a destination may change as time passes. DSDV requires periodical updates of routing information. Every host has the most recent knowledge about routes. It is likely that the path chosen to forward packets is the currently shortest one. In contrast to DSDV, AODV picks up a path (usually the shortest one) when a host initiates a route discovery. The host keeps sending packets via this path until it breaks, even if shorter paths become available after route discovery.

The difference between these two strategies can be illustrated with Figure 4.4, in which S is a set of source hosts and D is a set of destination hosts. P_1 and P_2 are two shortest paths between S and D. Originally, both DSDV and AODV send packets from S to D through these two paths. At time t, a host H moves in between S and D, and a shorter path is available. AODV still sends packets via P_1 and P_2. DSDV, however, sends all packets through the new path once it finds out the new one is shorter. Congestion may occur at host H when traffic load exceeds its capacity. This example shows that keeping sending packets through the shortest path may cause congestion.

4.6 Conclusion

To our knowledge, this work is the first attempt towards a comprehensive investigation of packet loss in mobile ad hoc networks. The contributions of congestion and mobility to the total packet loss have been examined. The impacts of host mobility, communica-

tion request, traffic load, traffic type, and AODV and DSDV routing protocols have been studied. The simulation results show:

- Mobility is the dominant cause for AODV, which is responsible for more than 60% of the total packet loss. For DSDV, more than 50% of the total packet loss is congestion-related.
- DSDV loses 10% to 20% more packets than AODV does for UDP traffic. For TCP traffic, the packet loss for DSDV is a half of that for AODV. DSDV outperforms AODV because the congestion control mechanism of TCP greatly reduces the congestion-related loss.
- Increasing the communication request or traffic load has a stronger impact on the packet loss in the less stressful situation (i.e., 10 connections at a rate of 4 packets/s).
- Host mobility decreases the packet loss for light communication request and traffic load. This confirms the argument that mobility increases the capacity of ad hoc networks [35]. For other cases, the packet loss is rather stable with host mobility.
- Always sending packets via the shortest path may cause congestion at a few heavily burdened hosts.

Inspired by this work, we are interested in investigating the relationship between shortest path and congestion. We are working on a loss sensitive routing protocol to support network layer congestion control for both UDP and TCP traffic. Our ultimate goal is to build a solid foundation for the research on routing and flow control algorithms for mobile ad hoc networks.

5 SAGA: SELF-ADJUSTING CONGESTION AVOIDANCE ROUTING PROTOCOL

5.1 Introduction

A mobile ad hoc network is a collection of mobile nodes that are deployed as a multi-hop wireless network without the aid of any preexisting infrastructure or centralized administration. The network connectivity and functionality are maintained through cooperations among nodes. Ad hoc networks use wireless links, which have significantly lower capacity than their hardwired counterparts (e.g., 54Mbps for 802.11g versus 9.952Gbps for OC192). The real throughput of a wireless link is affected by multiple access, fading, noise, and interference conditions. It is usually lower than the maximum transmission rate. The aggregated traffic demand easily reaches or exceeds the link capacity. Congestion is typically the norm rather than exception in ad hoc networks [16].

Current research efforts that address the congestion avoidance/control problem are based on the principle of conservation of packets [36]. Examples include TCP and its varieties [37–39]. The conventional TCP-type mechanisms use packet loss to infer congestion and provide per-connection congestion control. In ad hoc networks, wireless transmission loss (high bit-error rate) and route reconstruction (network partition) are significant causes for packet loss. They degrade the effectiveness of congestion inference mechanisms [30, 40]. Mechanisms have been proposed to improve TCP's performance over wireless and ad hoc networks [41–46]. The essence of TCP congestion control algorithms is to reduce the sending rate of traffic upon the occurrence of a congestion.

Two characteristics of ad hoc networks are the existence of multiple routes and the node-based routing. Routing protocols can make use of them to reduce network congestion with little sacrifice in the sending rate of traffic. Routing with load balancing has been investigated in [47–49]. The idea is to provide extra information, such as a secondary metric based on the current load on each node, to help distribute traffic load. It prevents

a single node from being overwhelmed. In an ad hoc network, the wireless channel is shared by multiple nodes. They contend for the channel not only for sending but also for receiving packets because of the hidden terminal problem [50]. The experimental study in [51] shows that contention for the channel is the primary reason for network congestion. The impact of the channel contention should be taken into account in the congestion reduction. For example, if the contention is already intense among a node's neighbors, it should not be chosen to forward packets even if there is no load on the node itself.

The main thrust of our work is to reduce network congestion by minimizing channel contentions. The objective is to avoid the *hot* spots where multiple nodes are in contention with each other. The global coupling effect of wireless channel access in ad hoc networks poses a challenge in determining the contentions locally. In addition, traffic load on a node must be taken into account, as the store-and-forward process may also cause congestion when the capacity of a node is exceeded. The shorter routes should be given higher priority because they are less likely to be involved in contentions with other nodes.

Our approach for reducing contention is as follows: (1) A single server queueing system is used to model nodes. The impact of channel contention is quantified using the service time (the time to successfully transmit a packet over the channel). The routing cost at each node is computed as the estimated delay. It reflects the effects of channel contention, current load, and expected load in the immediate future. (2) When a node has recent traffic, statistical methods are used to evaluate the mean of the delay. When no recent traffic exists, the underlying MAC protocol is analyzed and probability methods are applied to compute the expectation of delay. (3) The intermediate delay (IMD) routing metric is proposed to measure the communication delay introduced by the nodes connecting the source and destination. The route with the least intermediate delay will likely be involved in the least channel contention. (4) The self-adjusting congestion avoidance (SAGA) routing protocol is designed to reduce network congestion. Lazy route query operation that is presented in Section 5.4 is used by SAGA to accelerate the establishment of needed routes. Experimental studies are conducted to evaluate the performance of SAGA and compare it with AODV [2], DSR [3], and DSDV [1] protocols.

This research is conducted in the framework of CSMA/CA (carrier sense multiple access with collision avoidance) paradigm, which is adopted by the widely used IEEE 802.11 standard [52]. For unicast packets, CSMA/CA requires the sender and receiver to exchange the request-to-send/clear-to-send (RTS/CTS) frames prior to the transmission of the actual data frame. Broadcast packets are sent out without RTS/CTS. In this chapter, packets refer to unicast packets unless otherwise stated.

The rest of this chapter is organized as follows. Section 5.2 introduces contention-based access to shared media, channel spatial reuse, and the idea of ad hoc routing based on intermediate delay to reduce congestion. Two methods are presented in Section 5.3 to estimate delay locally. Section 5.4 presents the detail of SAGA protocol. In Section 5.5, the performance of the proposed protocol is evaluated and compared with AODV, DSR, and DSDV. The related work is discussed in Section 5.6. Section 5.7 gives analysis and guidelines resulting from this research.

5.2 Contention-based media access and congestion avoidance

5.2.1 Characteristics of contention-based access to wireless channels

When transmitting a packet through a wireless channel, the nodes within the transmission range of the sender, called neighbors, will receive it. If a neighboring node is sending or receiving a packet simultaneously, a collision occurs. The open channels and the use of CSMA/CA make the contention in ad hoc networks different from that in wired networks. Figure 5.1 illustrates the difference using a six-node network. A line between two nodes denotes that they are neighbors. In ad hoc networks, they are within each other's transmission range; in wired networks, they are attached to the same physical link. Figure 5.1 shows three transmissions T_1, T_2, and T_3. In the wired network, T_1, T_2, and T_3 can start simultaneously without collision. In the ad hoc network, T_2 will contend with T_1 because the receivers B and D are neighbors. At any time, only one transmission is allowed to use the channel shared by B and D. T_1 and T_3 can start concurrently as they are not contending with each other. The locality of contentions enables *channel spatial reuse* [53], i.e. the

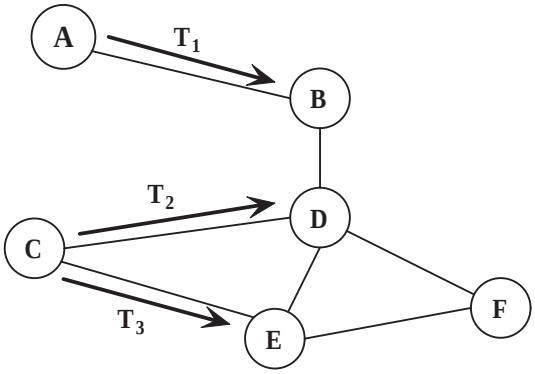


Figure 5.1. Network topology and flows

same channel in terms of frequency can be used by multiple transmissions at the same time.

Channel spatial reuse and the multi-hop routing provide a way to reduce contentions. For instance, if C wants to establish a connection session with F, selecting the route C→E→F instead of C→D→F will avoid contention between nodes B and D.

5.2.2 Ad hoc routing based on intermediate delay

The following examples illustrate the use of the intermediate delay in ad hoc routing. For the purpose of demonstration, the following simplification is used to compute the delay.

- If the capacity of the wireless channel is C, the size of a packet is P, the delay for sending a packet is P/C. The MAC layer control messages and the queueing delays are ignored.

- If n nodes are in contention for a channel, each node gets C/n share of the channel capacity. The delay for sending a packet is nP/C.

More precise estimates are proposed for SAGA protocol in section 5.3.

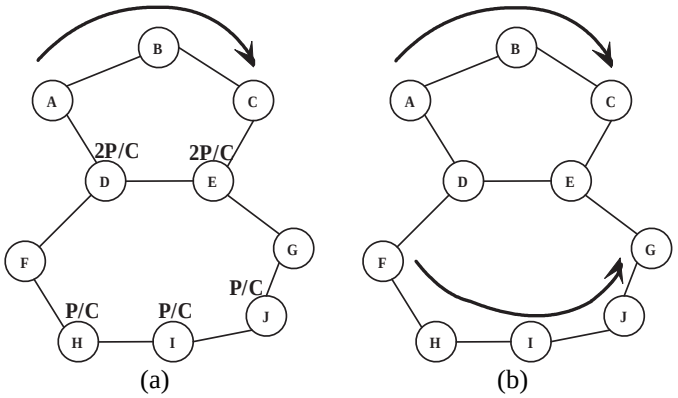


Figure 5.2. Select a route with presence of other connections

Figure 5.2, 5.3, and 5.4 illustrate route selection, adaption to traffic changes, and adaption to network topology changes. In each figure, we use a ten-node ad hoc network. The line with an arrow head represents a connection session. In these examples, a connection between nodes F and G is to be established.

Figure 5.2 illustrates the route selection process in the presence of other connection sessions. As shown in figure 5.2a, there is an active connection session between A and C when F wants to establish a connection with G. D is aware of the contention with A and computes the delay to be $2P/C$. Similarly, E's delay is $2P/C$. The delay computed by nodes H, I, and J is P/C. The IMD of the route F→D→E→G is $4P/C$, while that of the route F→H→I→J→G is $3P/C$. The later route is chosen even though it is one hop longer (Figure 5.2b). This route is better in terms of channel reuse and congestion avoidance. It introduces a lower end-to-end delay.

Figure 5.3 illustrates the adaption to changes in traffic. At the beginning, there is no traffic in the network. Every node can make full use of the channel and introduce a communication delay of P/C. The shortest route in terms of hop count is chosen to establish the connection (figure 5.3a), since it introduces the least intermediate delay. After the establishment of the connection, a new connection session from A to C is established. This connection will follow its best route A→B→C. The new connection causes channel

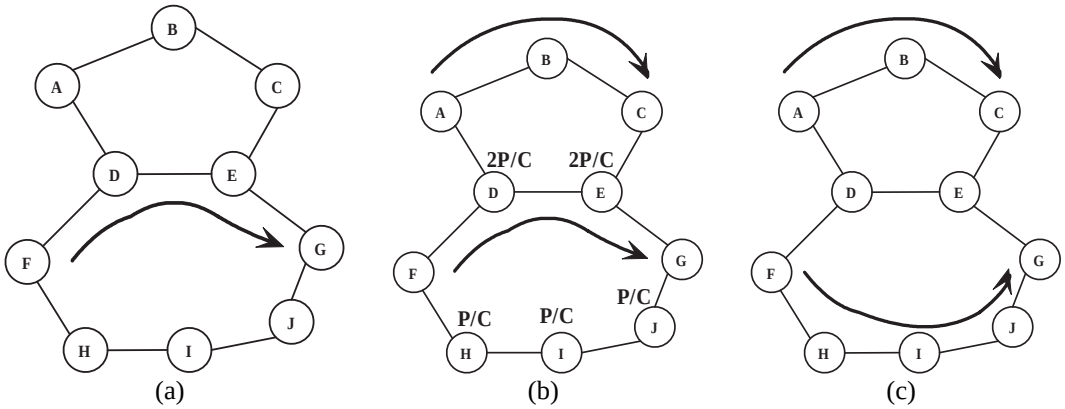


Figure 5.3. Adapt to changes in traffic

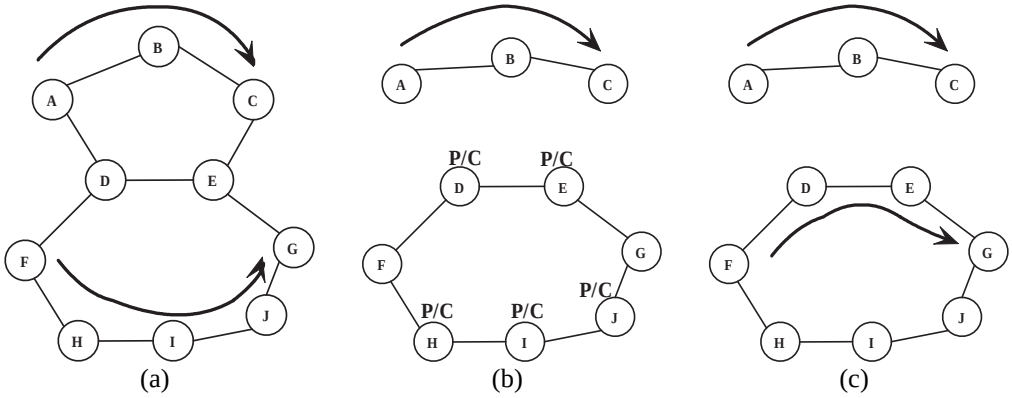


Figure 5.4. Adapt to changes in network topology

contention between A and D as well as C and E. The new delays at D and E are $2P/C$. The IMD of the route F→D→E→G is $4P/C$. The delay at nodes H, I, and J is still P/C. The route F→H→I→J→G has become a better choice, as shown in Figure 5.3b. Node F re-establishes the connection via the new route. Figure 5.3c shows the result after adapting to the new connection session.

Figure 5.4 illustrates the adaption to changes in network topology. The first two steps are same as in the example of Figure 5.2. The route F→H→I→J→G is chosen to avoid congestion (Figure 5.4a). Suppose nodes A and C have moved and are no longer contend-

ing with D and E. F will observe that the route F→D→E→G has become better since its IMD is $2P/C$. The connection is re-established as shown in Figure 5.4c.

These examples demonstrate the essential idea of congestion avoidance by using IMD. For the design of a practical routing protocol, we must consider the following: (1) At the time a node computes the delay, it may not know the number of neighbors who are contending with it. (2) Due to the locality of contention, access to a wireless channel creates global coupling effects in the entire network [53]. Even if the number of contending nodes is known, the share of capacity cannot be predetermined. (3) The successive links of a route may interfere with each other.

5.3 Delay estimation

Estimating the delay for sending a packet is critical in SAGA protocol. It is impractical to compute the accurate delay due to the dynamics and complexity of the network. Furthermore, an accurate value is not required because the delay is transient. The proposed methods calculate an approximation of the delay.

5.3.1 The model

A node can be modeled as a single server queueing system [54]. The following assumptions are made for delay estimation.

- The incoming traffic is localized with respect to time, i.e., in a short period of time, it obeys approximately the same distribution.
- The channel access is localized with respect to both time and location. If a node finds that the channel is busy, so do its neighbors.
- A node has a queue of sufficient size.
- The incoming traffic and outgoing traffic are Poisson processes.

- The incoming traffic rate and outgoing traffic rate are independent. This assumption is reasonable because (a) the complexity of channel contentions washes out the dependency and (b) the incoming traffic includes packets coming from other mobile nodes as well as from upper layer applications.

The assumptions reduce the complexity of the computation, yet result in a reasonably good estimate of the real delay. The simulation results presented in Section 5.5 show that SAGA protocol significantly improves the performance of routing by using the proposed delay estimation methods.

The following notations are used to describe the parameters of the queueing system.

λ**:** The arriving rate of packets. It is estimated using $\frac{N_A}{\Delta t}$. N_A is the number of packets arrived within the time interval Δt.

μ**:** The service rate, i.e., the number of packets transmitted over the wireless channel per second. The capacity of the channel and the contention with neighbors determine this parameter.

T_Q**:** The wait in the queue before a packet is transmitted.

T_S**:** The average service time for transmitting a packet ($T_S = \frac{1}{\mu}$).

T_D**:** The total delay at a mobile node ($T_D = T_Q + T_S$).

L**:** The current length of the queue.

If $\lambda \geq \mu$, the maximum allowed value of the delay is assigned to T_D since the wait in queue T_Q may be arbitrarily large [54]. Otherwise, T_Q can be evaluated using equation 5.1 by applying the Little's law [54]. The equation holds for general distributions of λ and μ.

$$T_Q = \frac{\lambda}{\mu(\mu - \lambda)} + T_S L \tag{5.1}$$

The delay T_D is calculated as follows.

$$\begin{aligned} T_D &= T_Q + T_S \\ &= \frac{T_S(L+1) - \frac{N_A}{\Delta t}(T_S)^2 L}{1 - \frac{N_A}{\Delta t} T_S} \end{aligned} \tag{5.2}$$

L, N_A, and Δt can be easily computed. Two cases are considered in the estimation of the service time T_S: a node with recent traffic (i.e., it recently sends out unicast packets over the wireless channel) and a node without recent traffic.

5.3.2 Node with recent traffic

If a node has transmitted packets recently, the mean value of the service time can be obtained using the statistical method. Let N_S be the number of packets and T_B be the time that the node spent on transmitting packets. T_B is less than or equal to Δt because the node may not be transmitting packets all the time.

$$T_S = \frac{T_B}{N_S} \tag{5.3}$$

The delay T_D is computed using equations 5.2 and 5.3 as follows.

$$\begin{aligned} T_D &= \frac{(L+1)\frac{T_B}{N_S} - \frac{N_A}{\Delta t} L (\frac{T_B}{N_S})^2}{1 - \frac{N_A}{\Delta t}\frac{T_B}{N_S}} \\ &= \frac{(L+1) - L\frac{N_A}{\Delta t}\frac{T_B}{N_S}}{\frac{N_S}{T_B} - \frac{N_A}{\Delta t}} \end{aligned} \tag{5.4}$$

To estimate T_D, we only need to know the number of incoming and outgoing packets, current queue length, and the time during which the node is sending packets.

5.3.3 Node without recent traffic

No recent traffic on a node does not imply that a packet can be sent out with the smallest delay. This is because the neighbors may be using the channel. The expectation of the service time can be determined by using probability methods to study the procedure of packet transmission. We analyze the IEEE 802.11 distributed coordination function (DCF) [52]. Such determination is applicable to other MAC protocols.

Figure 5.5 illustrates the procedure of transmitting a unicast packet using RTS/CTS. The corresponding state transition is shown in Figure 5.6. We briefly review it for the purpose of evaluating the expectation of transmission time. The detailed description of the process is available in [52].

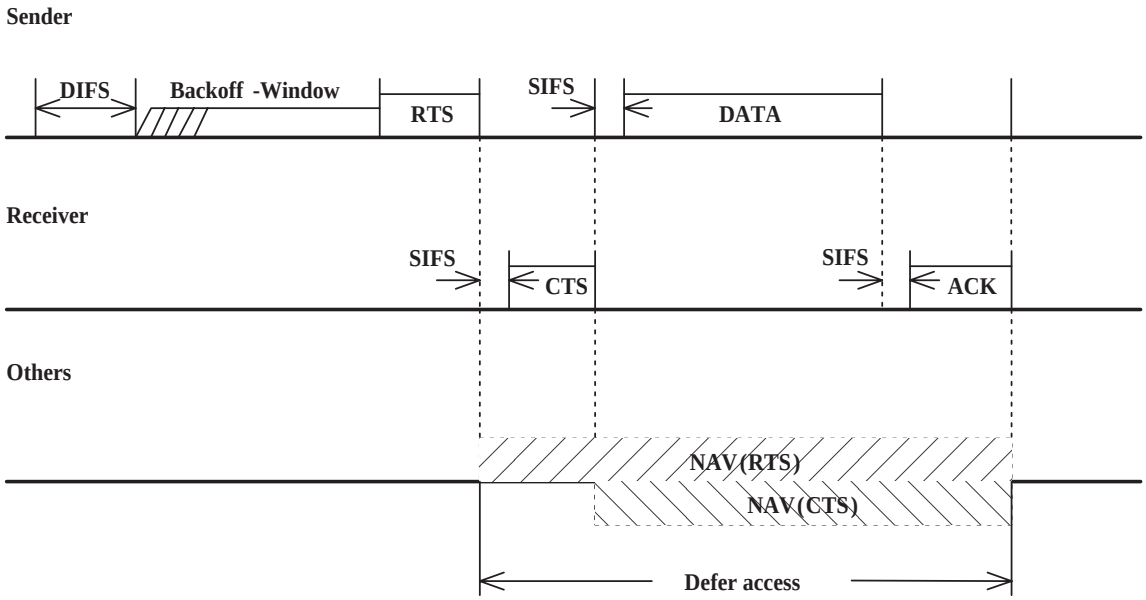


Figure 5.5. Transmission of a unicast packet using RTS/CTS in the IEEE 802.11 standard

When a packet is ready to transmit, the sender picks up a random backoff time of $b \times T_{slot}$ after observing an idle channel for the time period T_{DIFS}. b is a random number uniformly distributed over $[0, CW]$. T_{slot} and T_{DIFS} are values specified by the physical layer. The sender starts to transmit the RTS frame when the backoff time reaches zero. The receiver transmits a CTS frame after time T_{SIFS} upon receiving the RTS frame, if the media is idle. The neighbors of the sender and receiver set the network allocation vector (NAV) correspondingly to indicate that the media is reserved. The sender waits for time T_{SIFS} after receiving the CTS frame and then transmits the data. The receiver waits for time T_{SIFS} after receiving the data and replies with an acknowledge (ACK) frame. The expectation of the transmission time for a successful attempt is

$$E[T_{succ}] = T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK} + 3T_{SIFS} + E[T_{backoff}] \qquad (5.5)$$

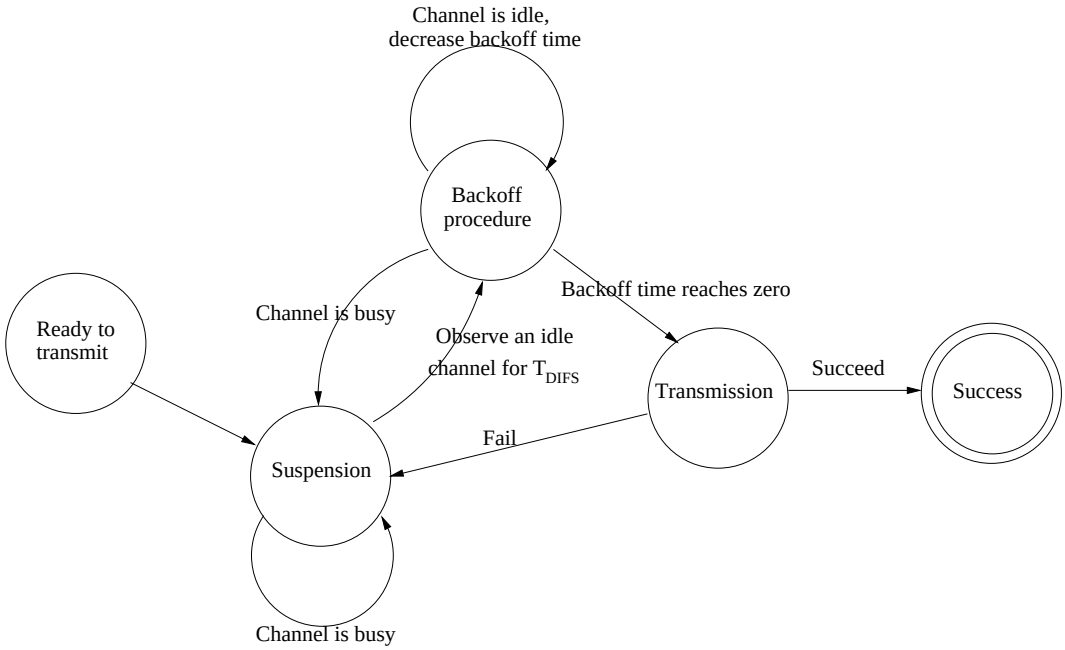


Figure 5.6. State transition of transmission procedure

where T_{DATA}, T_{RTS}, T_{CTS} and T_{ACK} are, respectively, time for transmitting a data packet, a RTS frame, a CTS frame, and an ACK frame. T_{DIFS} and T_{SIFS} are DCF interframe and short interframe time, and $T_{backoff}$ is the time spent on the backoff procedure.

The attempt fails if a CTS frame has not been received at the end of $T_{timeout}$ period following the transmission of the RTS frame. The sender then restarts this process. The expected time spent on a failed attempt is

$$E[T_{fail}] = T_{RTS} + T_{timeout} + E[T_{backoff}] \tag{5.6}$$

Now we compute the expected time spent on the backoff procedure $E[T_{backoff}]$. According to the assumption 2 in section 5.3.1, the probability that a channel is busy in a period of a unit time (i.e., the smallest time unit in the MAC specification) will not change during the transmission period. It is denoted as p. Observing an idle channel for time t is a Bernoulli trial [55]. It stops if the channel has been idle for a continuous time t. Let $\mathcal{T}_{idle}(t)$ be the

time needed for this trial. The expectation of $\mathcal{T}_{idle}(t)$ is computed recurrently as follows for a given p.

$$E[\mathcal{T}_{idle}(t)] = E[\mathcal{T}_{idle}(t-1)] + (1-p) * 1 + p * (1 + E[\mathcal{T}_{idle}(t)]) \tag{5.7}$$

Equation 5.7 aggregates two cases, assuming that the channel has been idle for a continuous time $t-1$ in the trial. (1) The channel is idle in the next unit of time and the trial stops. The probability for this case is $1-p$. (2) Otherwise, the trial still needs time $\mathcal{T}_{idle}(t)$ to stop. Simplifying equation 5.7 results in

$$E[\mathcal{T}_{idle}(t)] = \frac{E[\mathcal{T}_{idle}(t-1)] + 1}{1-p}$$

Solving the recurrence formula with the initial condition $E[\mathcal{T}_{idle}(1)] = \frac{1}{1-p}$ yields

$$E[\mathcal{T}_{idle}(t)] = \frac{1 - (\frac{1}{1-p})^{t+1}}{1 - \frac{1}{1-p}} = \frac{(\frac{1}{1-p})^{t} - 1}{p} + 1$$

Let $\mathcal{T}_{backoff}(b)$ be the time needed for a backoff procedure reaching 0 from $b \times T_{slot}$. It is

$$\mathcal{T}_{backoff}(b) = \mathcal{T}_{idle}(T_{DIFS}) + \sum_{i=1}^{b} t_0, \quad t_0 = T_{slot} \text{ or } T_{slot} + \mathcal{T}_{idle}(T_{DIFS} + T_{slot})$$

t_0 is the time needed to decrease the backoff time by T_{slot}. There are two cases.

1. During the backoff procedure, if no channel activity is detected for the duration of a particular backoff slot, the backoff time is decreased by T_{slot}. The probability is $(1-p)^{T_{slot}}$. In this case, $t_0 = T_{slot}$.

2. Otherwise, the procedure is suspended without decreasing the backoff time. It resumes after observing an idle channel for time T_{DIFS}. To decrease the backoff time, the channel must be idle for a continuous time $T_{DIFS} + T_{slot}$. $t_0 = T_{slot} + \mathcal{T}_{idle}(T_{DIFS} + T_{slot})$.

The expectation of t_0 is

$$E[t_0] = (1-p)^{T_{slot}} T_{slot} + (1 - (1-p)^{T_{slot}})(T_{slot} + E[\mathcal{T}_{idle}(T_{DIFS} + T_{slot})])$$

For a given random number b, the expectation of time spent on the backoff procedure is

$$E[\mathcal{T}_{backoff}(b)] = E[\mathcal{T}_{idle}(T_{DIFS})] + \sum_{i=1}^{b} E[t_0]$$
$$= E[\mathcal{T}_{idle}(T_{DIFS})] + bE[t_0]$$

The expected time for the backoff procedure during the transmission attempt is

$$\begin{aligned} E[T_{backoff}] &= E[E[\mathcal{T}_{backoff}(b)]] \\ &= E[\mathcal{T}_{idle}(T_{DIFS})] + \frac{CW}{2}E[t_0] \end{aligned} \tag{5.8}$$

The parameters CW (contention window), $aCWmin$, and $aCWmax$ are defined in the IEEE 802.11 standard [52]. CW takes an initial value of $aCWmin$ for the first attempt. Every time an attempt fails, CW takes the next value in the series, until it reaches $aCWmax$. A successful attempt resets CW to $aCWmin$. The CW values are powers of 2 minus 1, sequentially ascending from $aCWmin$ to $aCWmax$. They are specific to the physical layer. For example, direct sequence spread spectrum (DSSS) physical layer management information base (MIB) sets $aCWmin$ to 31 and $aCWmax$ to 1023. In the rest of this section, we assume DSSS is used as the physical layer. Let CW^n be the CW of the n-th attempt to transmit, then

$$CW^n = \begin{cases} 2^{n+4} - 1, & 1 \leq \text{n} \leq 6; \\ 2^{10} - 1, & \text{n} > 6. \end{cases}$$

Let T^n_{succ} and T^n_{fail} be the time spent on a successful transmission and a failed transmission for the n-th attempt, respectively. From equations 5.5, 5.6, and 5.8, we have

$$E[T^n_{succ}] = T_{DATA} + T_{RTS} + T_{CTS} + T_{ACK} + 3T_{SIFS} + E[\mathcal{T}_{idle}(T_{DIFS})] + \frac{CW^n}{2}E[t_0]$$

$$E[T^n_{fail}] = T_{RTS} + T_{timeout} + E[\mathcal{T}_{idle}(T_{DIFS})] + \frac{CW^n}{2}E[t_0]$$

The receiver gets the RTS frame if there is no collision during the transmission of the frame. It will transmit a CTS frame after time T_{SIFS} if the NAV indicates that the channel is idle. Otherwise, the receiver will not respond to the RTS frame. The channel must be idle in this duration of $T_{RTS} + T_{SIFS}$ for a successful RTS/CTS exchange. Since

channel access has locality characteristic, the possibility of a success P_s is approximately $(1-p)^{T_{RTS}+T_{SIFS}}$.

The expected transmission time makes sense only for successfully delivered data packets. We assume that there is no limit on retry and the sender will keep trying until the packet is delivered. The expected transmission time is

$$E[T_{trans}] = P_s E[T^1_{succ}] + \sum_{i=1}^{\infty}((1-P_s)^i P_s(E[T^{i+1}_{succ}] + \sum_{j=1}^{i} E[T^j_{fail}])) \tag{5.9}$$

CW^n is fixed when $n \geq 6$, so are $E[T^n_{succ}]$ and $E[T^n_{fail}]$. Solving equation 5.9 yields

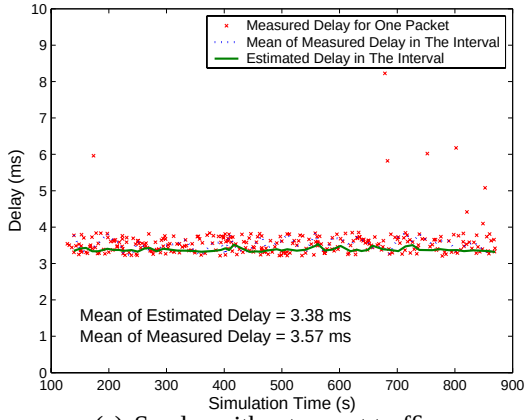
$$\begin{aligned} E[T_{trans}] &= E[T^1_{succ}] + \sum_{i=1}^{5}((1-P_s)^i(E[T^{i+1}_{succ}] - E[T^i_{succ}] + E[T^i_{fail}])) \\ &\quad + \sum_{i=6}^{\infty}((1-P_s)^i E[T^6_{fail}]) \\ &= E[T^1_{succ}] + \sum_{i=1}^{5}((1-P_s)^i(E[T^{i+1}_{succ}] - E[T^i_{succ}] + E[T^i_{fail}])) \\ &\quad + (1-P_s)^6 \frac{1}{P_s} E[T^6_{fail}] \end{aligned} \tag{5.10}$$

Once the physical layer parameters are determined, the delay of transmitting a packet T_S is characterized by $E[T_{trans}]$, which can be estimated by using equation 5.10 with the given possibility that the channel is busy. The total delay is calculated by applying the value of T_S to equation 5.2. This computation is done in constant time.

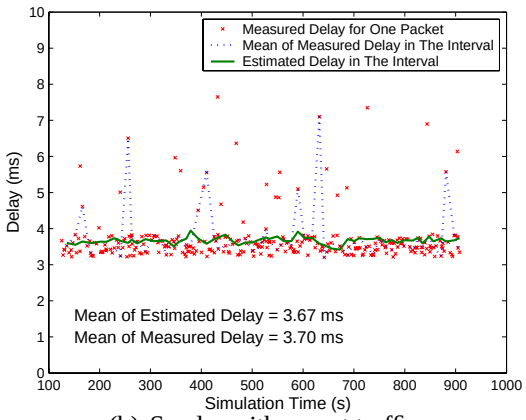
The IMD metric of a route is obtained by aggregating the delays from each intermediate nodes along the route. In the proposed delay estimations, the impact of active traffic in the neighborhood is reflected by the service time or the probability of a busy channel. The estimation of delay can be done without exchanging information with neighbors.

5.3.4 Accuracy of delay estimation

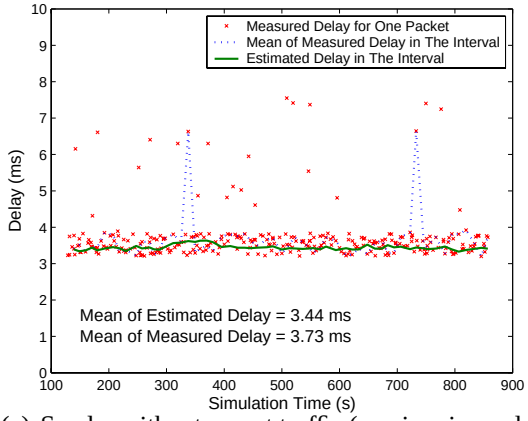
To evaluate the accuracy of the proposed delay estimation methods, two nodes A and B are put in an ad hoc network that has active traffic. Node A randomly sends dummy packets, which only have the delay information, to node B. The rate is 4 packets per 10 seconds, so that the evaluation has little impact on the real traffic. The delay for sending



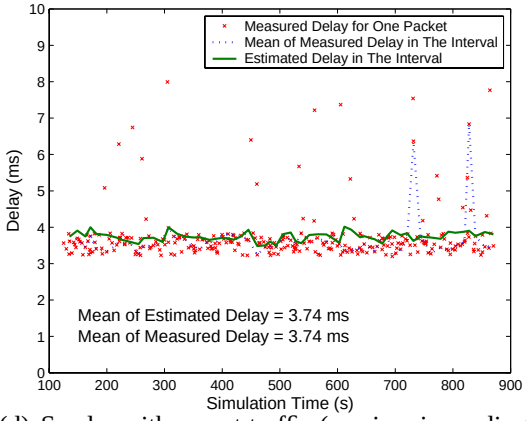
(a) Sender without recent traffic



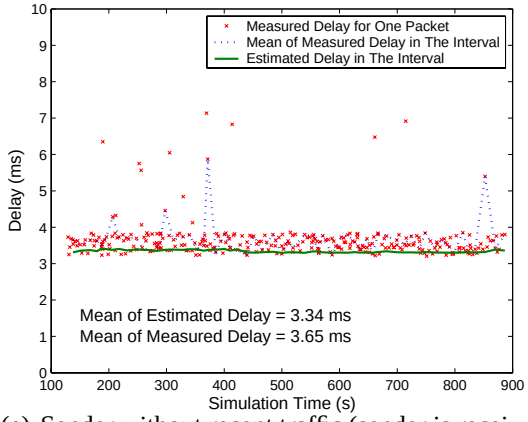
(b) Sender with recent traffic



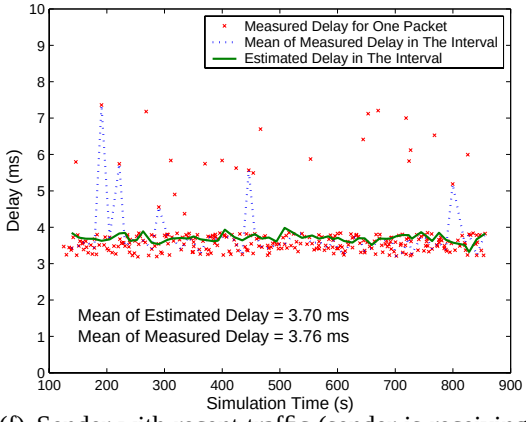
(c) Sender without recent traffic (receiver is sending packets to other nodes)



(d) Sender with recent traffic (receiver is sending packets to other nodes)



(e) Sender without recent traffic (sender is receiving packets from other nodes)



(f) Sender with recent traffic (sender is receiving packets from other nodes)

Figure 5.7. Comparison of estimated delay and measured delay

a dummy packet is not used in the statistics based delay estimation. In the experiments with recent traffic, data packets are generated from node A to node B besides the dummy packets.

Experiments have been conducted in six situations. The results are shown in Figure 5.7. It can be observed that for the statistics based delay estimation, the relative error between the estimated value and the measured value is less than 1.5%. The relative error for the probability based estimation is less than 8.5%. Both methods produce an estimate that is smaller than the real delay because of the assumptions we made about the packet arrival and sending processes.

5.4 Self-adjusting congestion avoidance routing protocol

5.4.1 Introduction

The self-adjusting congestion avoidance (SAGA) routing protocol is designed based on the ideas presented in sections 5.2 and 5.3. SAGA is a distance vector routing protocol. One of the major differences between SAGA and other distance vector based routing protocols is that SAGA uses IMD instead of hop count as the distance. It gives SAGA the capability of balancing traffic load and dealing with congestion.

To send packets, every node maintains a routing table that contains entries to all known nodes in the network. The data structure of the routing entry is shown in Figure 5.8. seq is a field of a routing entry that stores the sequence number representing the "freshness" of a route as in DSDV and AODV. It is maintained by the destination. Routes with more recent sequence numbers are always preferred for routing decisions. For routes with identical sequence number, the one with the smallest IMD is chosen.

SAGA is a proactive protocol like DSDV, which requires every node to periodically advertise the routing table to its neighbors. Significant new information such as a new route or a broken route may also trigger an advertisement. The estimated delay at this node is included in advertisement packets. A broken or unavailable route is assigned a

```
class RTEntry {
    RTEntry();
    addr_t dst;          // destination
    addr_t next_hop;     // next hop
    float imd;           // routing metric (IMD)
    uint seq;            // sequence number
    // minimum value of imd in all advertisements
    // since the last update of seq
    float min_advertised_imd;
    // when is ok to advertise this route?
    time advertise_ok_at;
    // do we need to advertise this route?
    bool need_advertise;
    // number of MAC callbacks
    uint MAC_callback_cnt;
    // event indicating this route breaks
    Event* trigger_event;
    // event indicating the next hop is not available
    Event* timeout_event;
    PacketQueue* q;      // packet queue
};
```

Figure 5.8. Data structure of the routing entry

Table 5.1
Major constants of SAGA protocol

Variable	Meaning	Value
MIN_INTERVAL	minimum time between two advertisements	1 seconds
MAC_CALLBACK	how many callbacks indicate a broken link	2
STARTUP_ADVERTISE	advertisements sent during startup	5
PERIOD_ADVERTISE	time between two full advertisements	15 seconds
DEFAULT_TTL	default value of TTL	30

delay of ∞, which is a value greater than the maximum allowed value of the delay. A route with ∞ delay is considered as invalid and is usually not included in advertisements.

Table 5.1 shows the major constants of SAGA and their values that are used in the simulation study.

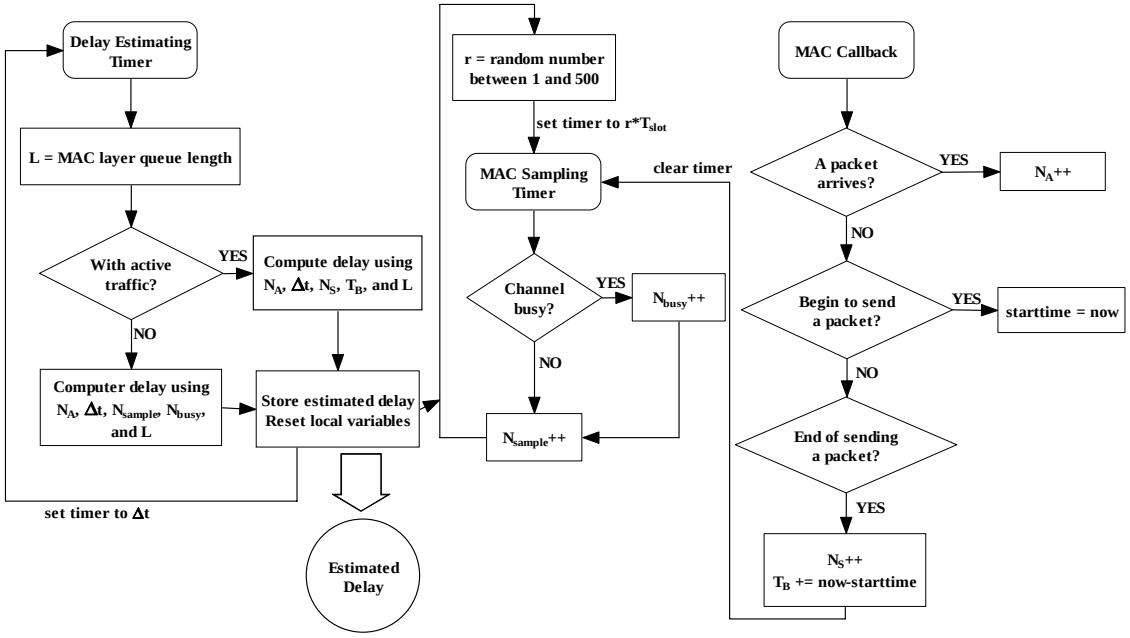


Figure 5.9. Delay estimate

5.4.2 Operations

SAGA protocol uses the following operations to estimate the delay, advertise and maintain routes, and handle broken links. The packet forwarding procedure is similar to that in DSDV. It is not discussed in this chapter.

Delay estimation

Every node estimates the delay using the methods presented in Section 5.3. The number of arrived packets N_A, the number of sent packets N_B, the time during which the node is transmitting packets T_B, the current length of the queue L, and the probability of a busy channel p are needed for estimation. The duration Δt determines how frequently the delay is estimated. It is set to the time interval between two full advertisements. N_A, N_B, and T_B are counted using a MAC callback function. This function is invoked when a packet arrives at MAC, when a packet is ready to transmit, and after a packet is transmitted. The

probability p is determined by using randomly sampling. $p = \frac{N_{busy}}{N_{sample}}$, where N_{sample} is the number of samples and N_{busy} is the number of samples that indicate a busy channel. Each node maintains a sampling timer, which will be randomly triggered about 200 times per second. When the timer is triggered, SAGA checks state of the channel. This timer is set when a new estimate process begins. It is cleared if active traffic is detected.

Route advertisement

SAGA uses route advertisements to disseminate information throughout the network. Two types of advertisements are defined in SAGA protocol. One, called "full advertisement", carries all the available routing information. The other, called "partial advertisement", carries only information changed since the last advertisement. Full advertisements are generated relatively infrequently. If a partial advertisement contains most of routing entries, it is upgraded to a full advertisement so that the next partial advertisement will be smaller. Two events will trigger an advertisement. The first one triggers a full advertisement, which is scheduled PERIOD_ADVERTISE seconds after the previous full advertisement. In the bootstrapping phase, a node may schedule full advertisements more frequently. The other event triggers a partial advertisement upon receiving significant new routing information, including: (1) a more recent sequence number, which helps SAGA to adapt in circumstances similar to the example shown in Figure 5.3; (2) a broken link, as discussed in [56], propagating bad news quickly will improve system performance.

Each routing entry is associated with two flags: *need_advertise* and *advertise_ok_at*. A partial advertisement only contains those entries whose *need_advertise* is set. A full advertisement includes all entries whose *advertisement_ok_at* is earlier than the current time. In both cases, the estimated delay that the node may introduce and the sequence number are included in the advertisement packets. Before each full advertisement, the sequence number is incremented by 2 so that the sequence number maintained by the destination is always an even integer. The pseudo code for making an advertisement packet is shown in Figure 5.10.

```
MakeAdv(periodic) {
  for each routing entry rte
    entry_count++;
    if rte.need_advertise == TRUE
      count++;
    if rte.advertise_ok_at > now
      unadvertiseable++;
    if count >= entry_count*2/3
      periodic = TRUE;
  make an advertisement packet p;
  add estimated delay to p;
  if periodic == TRUE
    increment sequence number by 2;
    add sequence number to p;
    add entry_count - unadvertiseable to p;
    for each routing entry rte
      if rte.advertise_ok_at <= now
        add rte to p;
        rte.need_advertise = FALSE;
  else
    add sequence number to p;
    add count to p;
    for each routing entry rte
      if rte.need_advertise == TRUE
        add rte to p;
        rte.need_advertise = FALSE;
}
```

Figure 5.10. Algorithm for making an advertisement packet

Route maintenance

Each routing entry in SAGA has two fields associated with the distance: (a) imd stores the intermediate delay and is used for route advertisement; and (b) $min\_advertised\_imd$ stores the minimum value of imd in all advertisements since the last update of seq. Assume that a node i receives an advertisement of a route to a node x from a neighbor j. The route has a sequence number seq_j^x and an intermediate delay imd_j^x. Node i updates its routing table if and only if one of the following four conditions is true.

1. Node i does not have a valid route to the destination x.
2. Node j is the next hop of the current route.
3. The new route contains a fresher (valid) sequence number ($seq_j^x > seq_i^x$) and $imd_j^x < \infty$.
4. $seq_j^x = seq_i^x$ and $min\_advertised\_imd_i^x > imd_j^x$.

These constraints guarantee that SAGA will not introduce loops in routes. In the third condition, as proved in [1], a loop cannot be created if nodes use fresher sequence numbers to pick routes. The loop-free property holds in the fourth condition due to the theorem proved in [57], which states that distance vector algorithms always maintain loop-free routes in presence of static or decreasing link weights.

Some distance vector routing protocols use a single field of distance for a routing entry. This field is used for both route advertisement and routing decision. Because imd reflects the extent of congestion along a route, its value may change even if the route is static. As imd is not static or decreasing, the loop-free property may not hold if it is used to make routing decisions as in the fourth condition. The use of $min\_advertised\_imd$ assures loop-free routes.

Adaptive routing metrics such as imd sometimes suffer from oscillation. After choosing a route and beginning to send packets, other routes become attractive. The tendency of the routing decision to switch excessively from one choice to alternates makes the routes unstable. This increases routing overhead and decreases performance. The oscillation problem was stated in [58]. Using $min\_advertised\_imd$ in route decision prevents a node from switching back and forth among alternative routes and helps in reducing the oscillation of the imd value. The associated cost is a possible delay in adopting a better route whose intermediate delay is lower than imd but higher than $min\_advertised\_imd$.

Figure 5.11 demonstrates the procedure of route maintenance. An extra functionality of route maintenance is shown in function $ProcessAdvEntry$. The reception of an advertisement entry with an older sequence number will trigger a partial advertisement to help the neighbor to obtain the up-to-date route.

Handling broken links

A link to a neighbor is considered as broken if no advertisement is received from it for a certain period of time. To detect broken links, each neighbor is associated with a timeout

```
ProcessAdv(pkt) {
  sender = source address of pkt;
  delay = estimate delay in pkt;
  seq = sequence number in pkt;
  rte = routing entry to sender;
  if rte does not exist
    add a routing entry rte;
    rte.dst = rte.next_hop = sender;
    rte.imd = 0;
    rte.seq = seq;
    rte.need_advertise = TRUE;
    trigger advertisement for rte;
  else
    rte.next_hop = sender;
    rte.imd = 0;
    if (rte.seq < seq)
      rte.seq = seq;
      rte.need_advertise = TRUE;
      trigger advertisement for rte;
  for each advertisement entry adv
    if adv.dst == my_address
      if adv.imd != 0
        schedule a full advertisement;
    else
      adv.imd = adv.imd + delay;
      adv.next_hop = sender;
      ProcessAdvEntry(adv);
}

ProcessAdvEntry(adv) {
  rte = routing entry to adv.dst;
  if rte does not exist
    add a routing entry rte;
    rte.dst = adv.dst;
    rte.next_hop = adv.next_hop;
    rte.imd = adv.imd;
    rte.seq = adv.seq;
    rte.need_advertise = TRUE;
    trigger advertisement for rte;
  else if rte.seq == adv.seq
    if rte.min_advertised_imd > adv.imd
      UpdateRoute(rte, adv);
  else if rte.seq < adv.seq
    if adv.imd < INFINITY or rte.next_hop == adv.next_hop
      UpdateRoute(rte, adv);
      rte.need_advertise = TRUE;
      trigger advertisement for rte;
  else if rte.seq > adv.seq
    if rte.imd < INFINITY and adv.imd == INFINITY
      rte.need_advertise = TRUE;
      trigger advertisement for rte;
}
```

Figure 5.11. Algorithm for route maintenance

```
MACCallback(pkt) {
  next_hop = next hop of pkt;
  drop pkt for MAC callback;
  rte = routing entry to next_hop;
  rte.MAC_callback_cnt++;
  if rte.MAC_callback_cnt > MAC_CALLBACK
    if rte.timeout_event exists
      cancel rte.timeout_event;
    HandleTimeout(rte);
}

HandleTimeout(rte) {
  for each routing entry rte2
    if rte2.next_hop == rte.dst and rte2.imd < INFINITY
      rte2.imd = INFINITY;
      rte2.seq++;
      rte2.need_advertise = TRUE;
      trigger advertisement for rte2;
}
```

Figure 5.12. Algorithm for handling broken links

event that will be triggered after $2 \times PERIOD\_ADVERTISE$ seconds. This event is reset every time an advertisement is received from the neighbor.

The MAC callback is another mechanism to detect broken links, because CSMA/CA will report an error when it fails to transmit a packet. Continuous occurrence of failure indicates that either the neighbor is not available or the contention is too intense. In both cases, this neighbor should not be picked as the next hop. If the number of continuous callbacks exceeds the preset threshold MAC_CALLBACK, the timeout event is triggered.

Figure 5.12 shows how MAC callback triggers the event and how SAGA handles a broken link. When a link to a neighbor is indicated broken, any route through that neighbor is immediately assigned ∞ to IMD and the sequence number is incremented by 1. Thus a broken route is always associated with an odd sequence number while the valid one is associated with an even sequence number.

Lazy route query

SAGA does not provide a dedicated route query operation as in the on-demand protocols. When a node wants to send packets to a destination but does not have a valid route,

it uses a technique called *lazy route query*. Usually, a route with ∞ delay in an advertisement packet is used to report a broken link. In this case, seq is an odd number. A route with ∞ delay and an even number of seq is treated as a query instead of an advertisement. It indicates that this node needs a route to the destination. The route's seq must be greater than the one in the query. Neighbors who have a valid route will include it in the next advertisement packet as a response to the query. Lazy route query works well with the proactive approach, because (1) each node periodically advertises its routing table, it is likely that one of the neighbors has already had a valid route; and (2) multiple routes may be queried in one advertisement packet.

These operations enable SAGA protocol to handle the dynamic and unpredictable changes in the network topology and traffic load, and to deliver packets through routes with less congestion. They are the basis of a complete implementation for experimental studies. Please refer to [59] for the details of SAGA protocol.

5.5 Experimental evaluation

The objective of the experiments is to study the performance of routing protocols under congestion. SAGA is compared with AODV, DSR, and DSDV protocols, which have received wide attention in the literature [8, 9, 25]. The use of intermediate delay in SAGA is contrasted against the use of hop count in AODV, DSR, and DSDV through the measurements obtained from the experiments.

The study is done through simulation using the network simulator ns2 [18]. The implementations of AODV, DSR and DSDV protocols are provided by ns2. All optimizations for AODV and DSR are enabled in the simulation for the comparisons. The implementation of SAGA is based on the operations presented in Section 5.4. The values of the parameters for SAGA are given in Table 5.1 (Major constants of SAGA protocol).

The wireless interface simulates the 914 MHz Lucent WaveLAN direct-sequence spread-spectrum (DSSS) radio interface [7]. The IEEE 802.11 distributed coordination function (DCF) with CSMA/CA is used as the MAC layer protocol. The random waypoint

model [9] is used to generate movements for mobile nodes. These settings are commonly used in studies reported in the literature.

Five independent scenarios are generated for each experiment. The average values are used for analysis.

5.5.1 Performance metrics

The following metrics are used to evaluate the routing protocols. They are based on a list of quantitative metrics suggested by the RFC 2501 [16].

- *Delivery Ratio*: The ratio of the data delivered to the destinations (i.e., throughput) to the data sent out by the sources. The throughput is also studied in the experiments.
- *Protocol Overhead*: The ratio of the routing load to the data successfully delivered to the destination. The routing load is measured as the number of bytes of protocol messages transmitted hop-wise. The transmission on each hop is counted once.
- *Average End-to-end Delay*: The average time it takes for a packet to reach the destination. It includes all possible delays in the source and each intermediate node. It can be caused by routing discovery, queueing at the interface queue, transmission at the MAC layer, etc. Only successfully delivered packets are counted.

5.5.2 Simulation and input parameters

UDP connections are used in most of the experiments so that there is no congestion control at the transport layer. Each connection is specified as a randomly chosen source-destination (S-D) pair. Every connection starts at a time uniformly distributed over 0 to 100 seconds, so that the proactive protocols have sufficient to warm up. The size of packets is 512 bytes. Two types of traffic are considered in the study.

- *Constant Bit Rate (CBR) traffic:* It is generated at a deterministic rate [18]. This type of traffic is widely used in the study of ad hoc network routing protocols and provides a good basis for evaluating SAGA protocol.

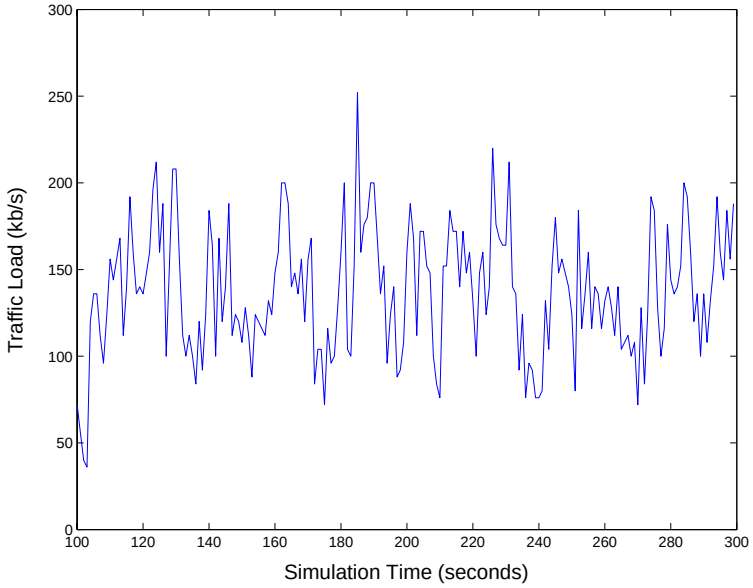


Figure 5.13. POO traffic

- *Pareto On/Off (POO) traffic:* It is generated according to a pareto on/off distribution [18]. Packets are sent at a fixed rate during on periods, and no packets are sent during off periods. Both on and off periods are drawn from a pareto distribution. POO traffic exhibits long range dependency. It closely matches with the empirically measured network traffic [60]. Figure 5.13 shows an example of the aggregated POO traffic used in the simulation.

To highlight the impact of congestion on the routing performance, the offered traffic load is taken as the input parameter. The aggregated traffic injected into the network ranges from 100 kb/s to 600 kb/s, which puts much stress on the routing protocols.

Six experiments for UDP connections and two experiments fot TCP connections have been conducted connections by varying the maximum speed of the movement of nodes and the number of connections. The maximum speeds of 4m/s and 20m/s are considered as low and high mobility respectively. The first four experiments use CBR traffic and the last two use POO traffic.The values of parameters used in the simulation are given in Table 5.2 (Simulation and input parameters).

Table 5.2
Simulation and input parameters

Simulation time	1000 seconds
Independent runs	5
Mobility model	random waypoint
Simulation area	1000m $\times$ 1000m
Maximum speed	4 m/s, 20 m/s
Pause time	10 seconds
Wireless transmission range	250m
Channel capacity	2 Mb/s
Number of mobile nodes	50
Number of connections	10, 30
Packet size	512 bytes
POO on time	500 ms
POO off time	1000 ms
POO shape	1.5

5.5.3 Measurements and observations

Experiment 1

The experiment studies the routing performance in low mobility environment. Twenty percent of the nodes are generating CBR traffic. Data in Figure 5.14 shows that the performance of the routing protocols decreases with the increase of the offered traffic load.

Figure 5.14b shows that the increase in traffic has more impact on AODV and DSR than on SAGA and DSDV. When the offered traffic load increases from 100 kb/s to 500 kb/s, the delivery ratios of AODV and DSDV drop from 96% to 56% and from 92% to 47% respectively. The delivery ratios of SAGA and DSDV stay stable when the offered traffic load is less than 300 kb/s. SAGA delivers around 95% of the packets, while DSDV

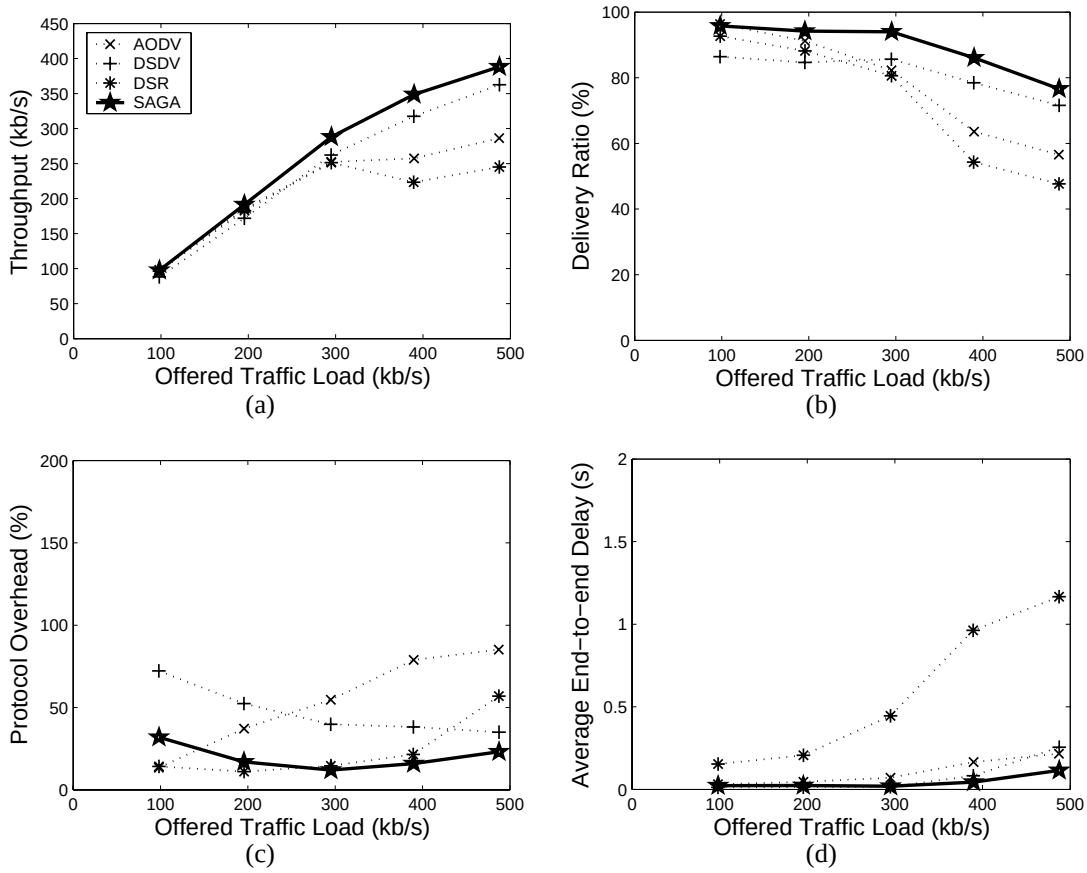


Figure 5.14. 10 CBR connections, low mobility

delivers 85%. The delivery ratios drop to 77% and 71% respectively, when the offered traffic load reaches 500 kb/s.

Figure 5.14c shows that the overhead of AODV and DSR increases with the offered traffic load, from 13% to 85% and from 14% to 57%, while that of DSDV decreases from 72% to 35%. The overhead of SAGA drops from 32% to 12% when the offered traffic load increases from 100 kb/s to 300 kb/s. It then increases to 23%.

The average end-to-end delay of DSR increases significantly from 0.15 to 1.16 seconds. The delay of AODV gradually increases from 0.03 to 0.2 seconds. SAGA and DSDV have almost the same delay of 0.02 seconds when the offered traffic load is less

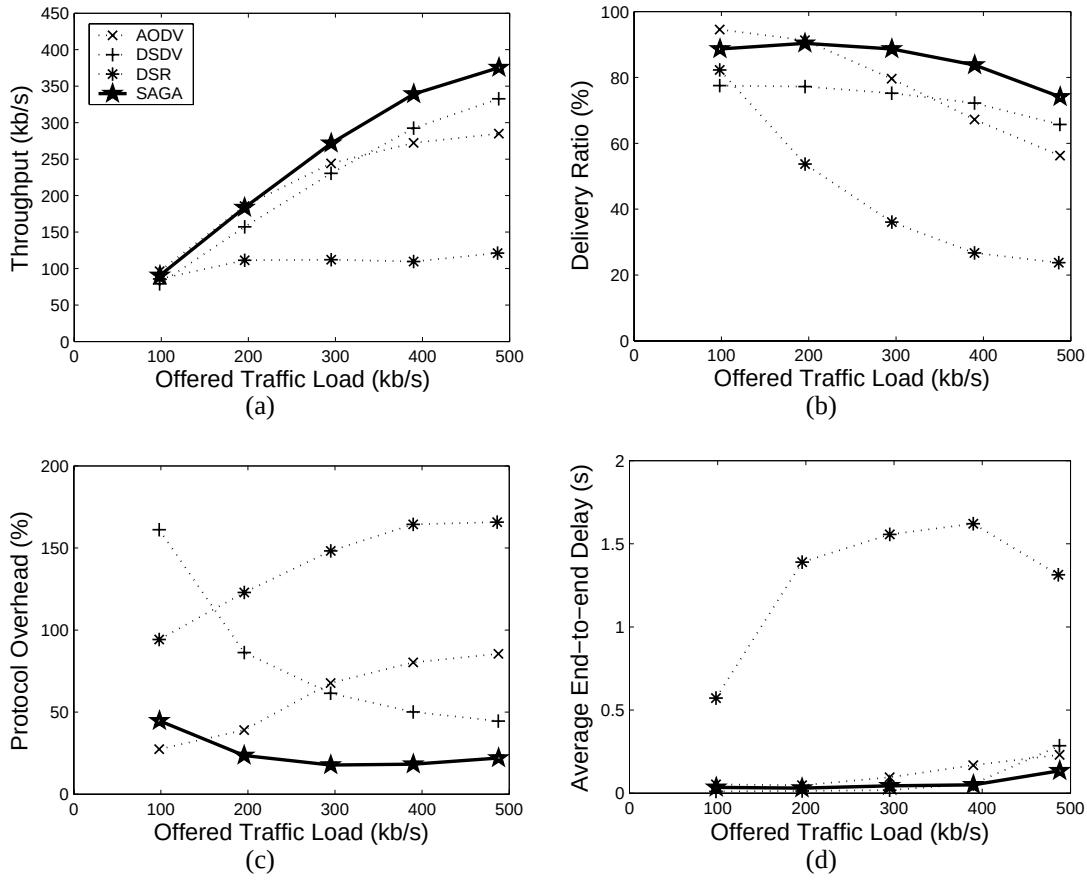


Figure 5.15. 10 CBR connections, high mobility

than 300 kb/s. The delay of DSDV then increases to 0.25 seconds, while that of SAGA increases to 0.11 seconds (Figure 5.14d).

Experiment 2

In this experiment, all the parameters are the same as in the previous experiment, except that the maximum moving speed is changed from 4 m/s to 20 m/s.

The delivery ratio of DSR drops quickly with the increase of the offered traffic load (Figure 5.15b), because its throughput stays at around 100 kb/s as shown in figure 5.15a.

The overhead of DSR increases sharply compared with results of the low mobility experiment, by 2 to 5 times depending on the offered traffic load (Figure 5.15c and 5.14c).

With the offered traffic load increasing from 100 to 500 kb/s, the delivery ratios of SAGA and DSDV drop from 89% to 74% and from 77% to 65% as shown in Figure 5.15b. When mobility changes from low to high, the overhead of DSDV is almost doubled, and that of SAGA slightly increases by 5%.

Comparing Figure 5.15 with figure 5.14, we can tell that mobility greatly affects the performance of DSR. For SAGA and DSDV, the increase of mobility has a greater impact when the offered traffic load is lighter. Mobility does not have much impact on the performance of AODV.

Experiments 3 and 4

These two experiments illustrate the performance of routing protocols when the number of connections is 30. In the simulation, 60% of the mobile nodes are generating traffic. The aggregated traffic load is the same. The results of the low mobility experiment are shown in Figure 5.16 and those of the high mobility experiment are shown in figure 5.17.

Comparing Figure 5.16 with figure 5.14, and Figure 5.17 with figure 5.15, we can conclude that routing performance decreases with the number of connections, which has a greater impact on AODV and DSR, the on-demand protocols, than on SAGA and DSDV. In the low mobility experiment, the delivery ratio of each protocol is almost unchanged with the increase of the number of connection when the offered traffic load is light (less than 200 kb/s). It decreases by less than 10% with 500 kb/s traffic. Unlike the other three protocols, the overhead of AODV increases significantly. The average delay increases for each protocol, but AODV has the greatest growth.

In the high mobility experiment, the throughput of DSR is saturated at 100 kb/s, almost the same as in the experiment with 10 connections. The throughput of AODV is saturated at about 200 kb/s, while the saturation is not obvious in the corresponding 10-connection experiment (Figure 5.17a and 5.15a).

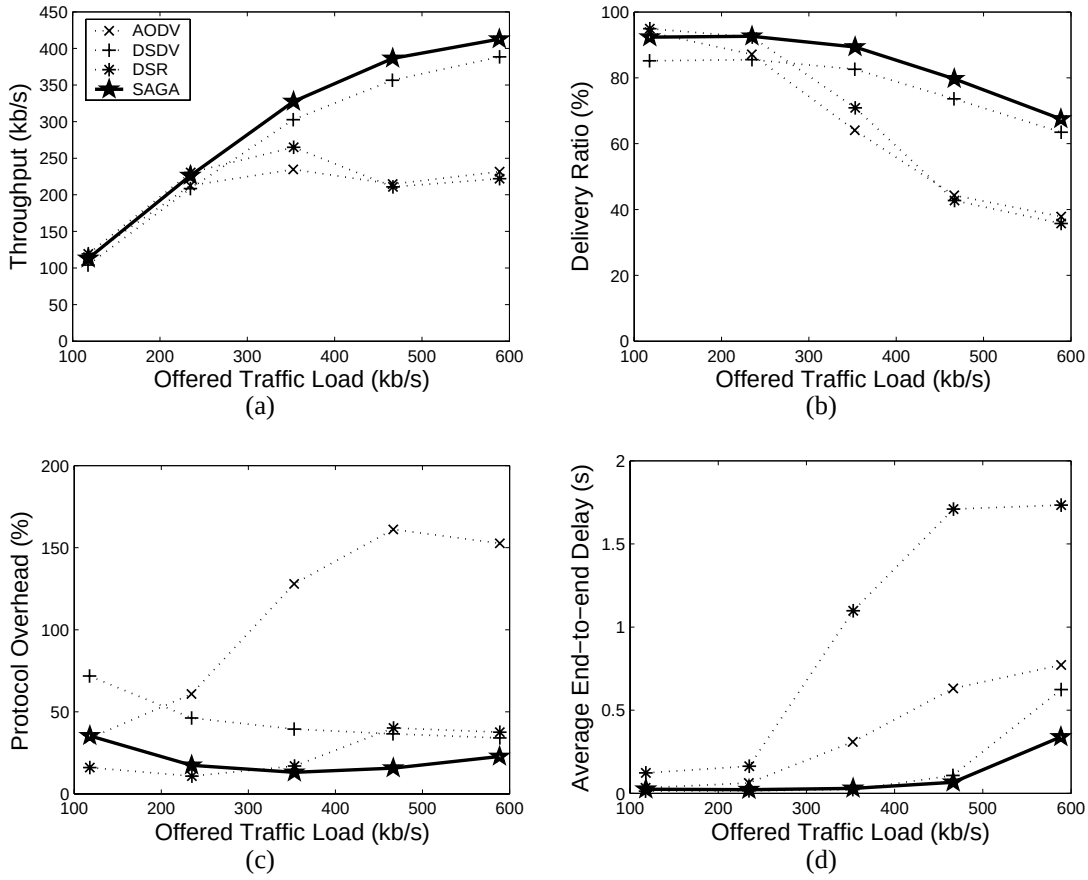


Figure 5.16. 30 CBR connections, low mobility

Experiments 5 and 6

In addition to CBR traffic, experiments have been conducted to study the performance of routing protocols using POO traffic. The long range dependency of the aggregated POO traffic closely matches with the actual network traffic. The study provides a better understanding on the performance when the routing protocols are implemented for ad hoc networks in practice. The simulation parameters in these experiments are the same as in the 10-connection experiments, except that every source of a connection generates POO traffic instead of CBR traffic. As shown in Figure 5.13, although the average traffic load is about 133 kb/s, the real-time load often approaches or exceeds 200 kb/s. The fluctuation

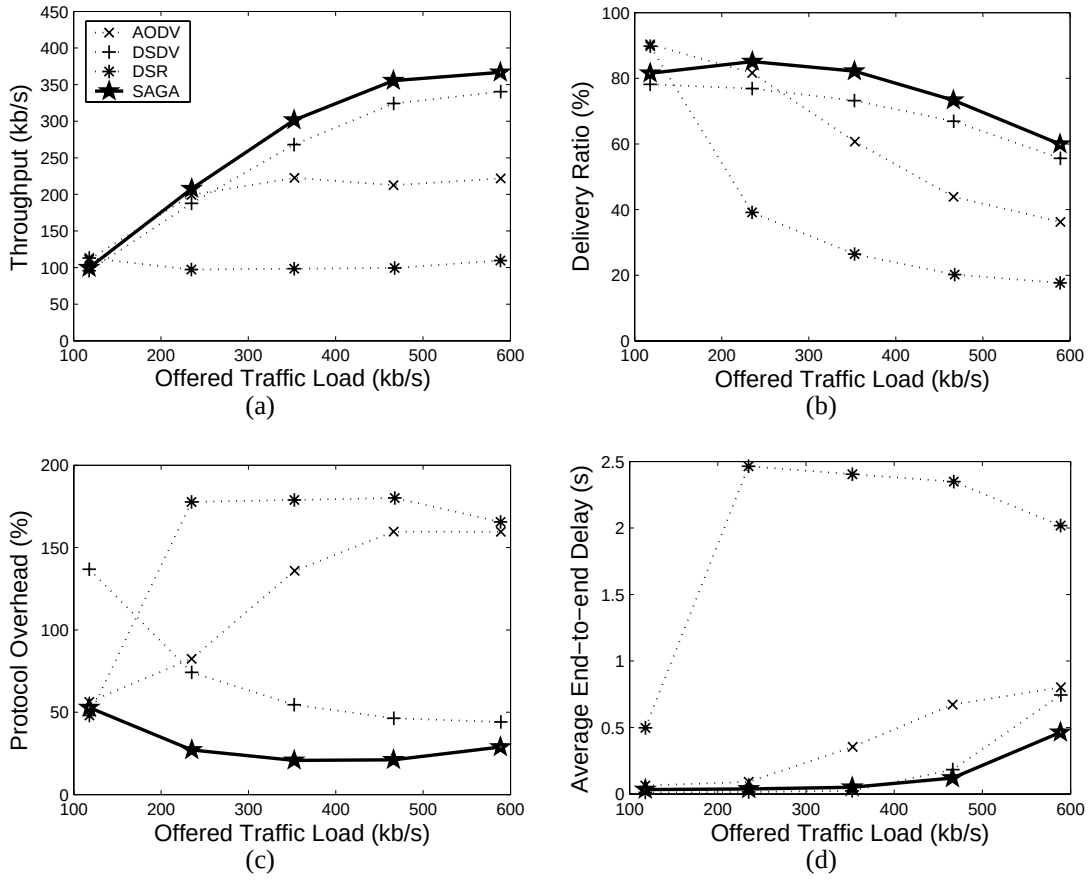


Figure 5.17. 30 CBR connections, high mobility

of traffic load poses a challenge that requires a quick response to traffic dynamics. The results are shown in Figure 5.18 and figure 5.19.

The performance of DSDV and DSR is almost the same as in the 10-connection experiments. SAGA performs even better when the offered traffic load is in the range of 100 kb/s to 300 kb/s. In terms of delivery ratio, SAGA outperforms all evaluated protocols in all cases except for DSR in the 67 kb/s traffic and high mobility (Figure 5.18b and 5.19b).

AODV delivers less than 40% of packets in the low mobility experiment and about 20% of packets in the high mobility experiment. Figure 5.18c and 5.19c show that the overhead of AODV is less than 10%, which is much lower compared with the results of

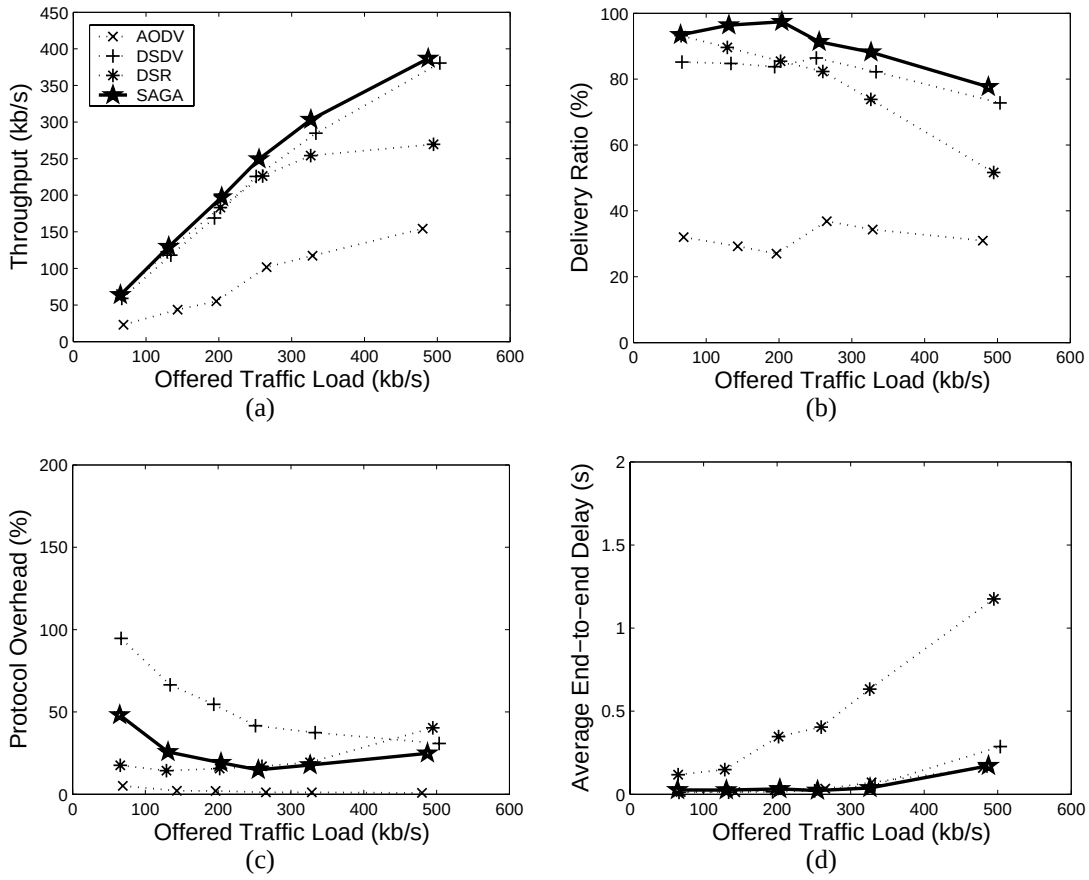


Figure 5.18. POO traffic, low mobility

the CBR traffic experiments. It indicates that AODV does not exchange much routing information when traffic bursts. Many packets are dropped due to congestion.

Experiments for TCP traffic

Two experiments have been conducted to evaluate the performance of SAGA with TCP traffic in low and high mobility scenarios. The results are shown in Figure 5.20. All evaluated routing protocols except for DSR have almost the same end-to-end delay regardless the number of connections as shown in Figure 5.20c and 5.20d. The proactive protocols achieve higher throughput than the on-demand ones. This is consistent with the

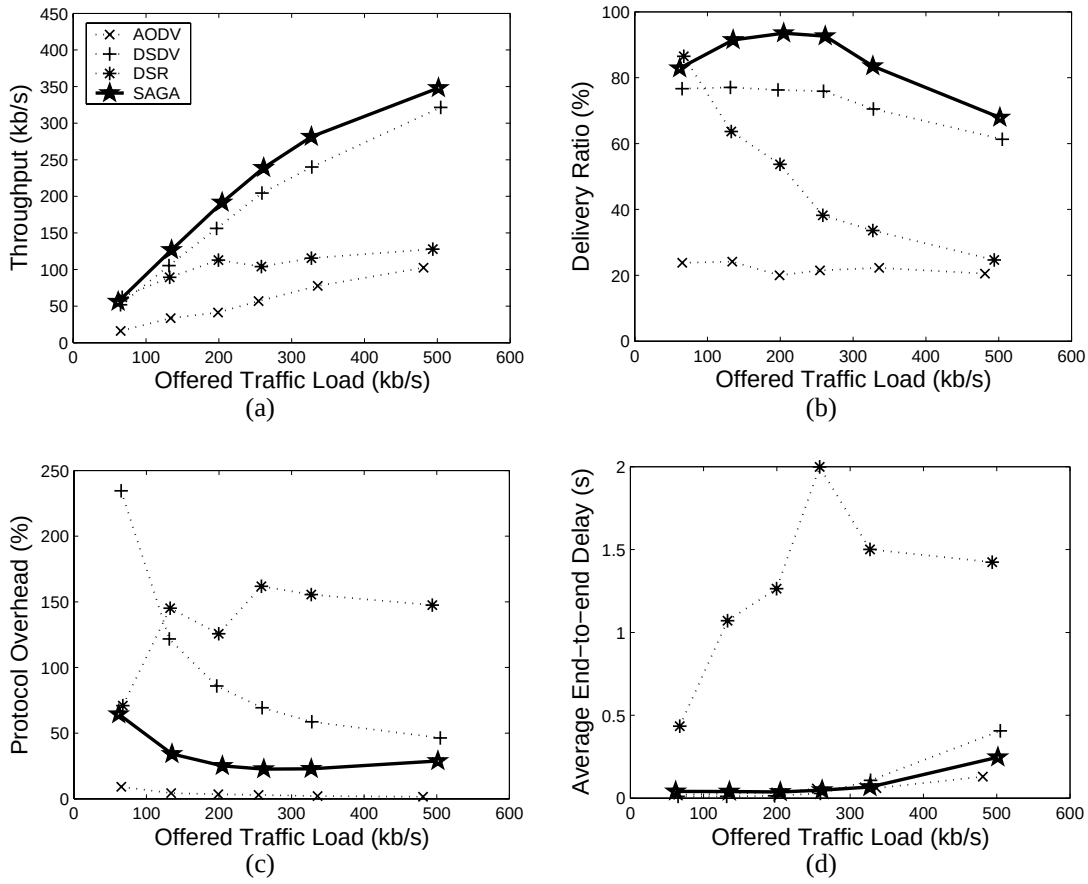
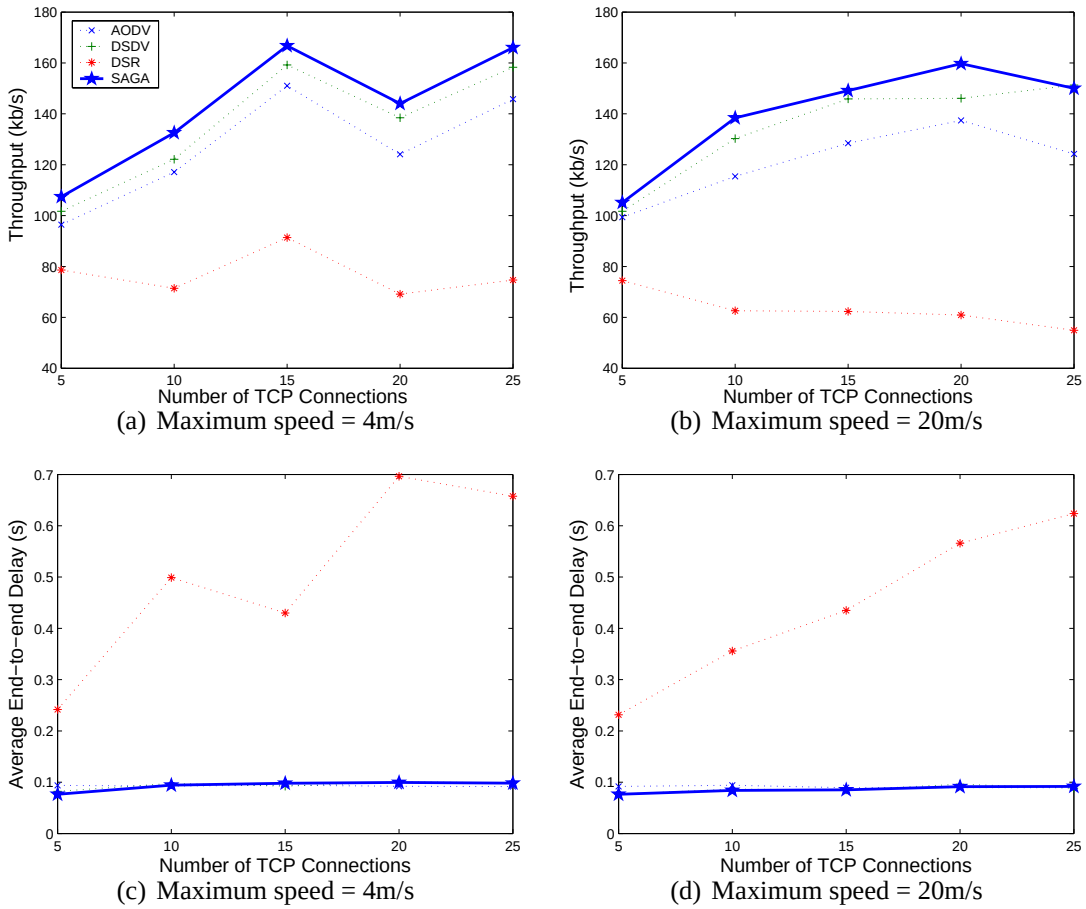


Figure 5.19. POO traffic, high mobility

results obtained from the study of packet loss. SAGA still performs better than DSDV, but not much. This is because TCP also tries to control congestion, thus diminishes the advantage of SAGA in terms of congestion avoidance.

5.5.4 Analysis and discussion

We classify the traffic load offered by CBR connections into low, moderate, and high based on whether it is less than 200 kb/s, between 200 and 400 kb/s, or greater than 400 kb/s. For the traffic load offered by POO connections, the two classifying values are 132 kb/s and 330 kb/s.



(a) Maximum speed = 4m/s (b) Maximum speed = 20m/s

(c) Maximum speed = 4m/s (d) Maximum speed = 20m/s

Figure 5.20. TCP traffic

SAGA versus on-demand protocols

- *Throughput:* SAGA is able to sustain heavier traffic load and offers higher peak throughput than AODV and DSR. Since SAGA can balance traffic load and avoid congestion, it enables better utilization of the aggregated network capacity. SAGA provides a peak throughput of 400 kb/s while AODV and DSR saturate at around 250 kb/s, when mobility is low (Figure 5.14a and 5.16a). In high mobility scenarios, DSR saturates at 120 kb/s. The peak throughput of AODV is 220 to 280 kb/s. It decreases as the number of connections increases. SAGA can reach about 370 kb/s regardless of the number of connections (Figure 5.15a and 5.17a). POO traffic

does not have much impact on the peak throughput of SAGA and DSR. It causes the peak throughput of AODV to drop to 150 kb/s and 100 kb/s in low and high mobility scenarios respectively (Figure 5.18a and 5.19a). In summary, SAGA can consistently offer a peak throughput of 370 to 400 kb/s in all cases, which is 1.5 to 3.5 times of the peak throughput achieved by the on-demand protocols.

- *Delivery ratio:* SAGA does not achieve high delivery ratio in high mobility and low traffic load (Figure 5.15b, 5.17b, and 5.19b). More than 95% of the dropped packets are caused by broken routes, because the routes obtained from advertisements may be stale by the time they are used. In the implementation of SAGA, a link is considered broken if two consecutive packets to the same neighbor are dropped. This increases the accuracy of broken link detection, at the cost of more dropped packets. When mobility is high and traffic load is low, one packet might be enough to infer a broken link since the probability of dropping packets due to congestion is low. In this case, SAGA does not deliver as many packets as AODV and DSR. When the offered traffic load increases from low to moderate, the delivery ratio of SAGA increases because of the accuracy of broken link detection.

  SAGA performs as well as the on-demand protocols in low traffic load and low mobility. It outperforms them when the offered traffic load is moderate or high.

- *Protocol overhead:* The protocol overhead of SAGA is in the range of 15% to 50% of the total delivered data. Because SAGA uses one-hop broadcast and requires the interval between two consecutive advertisements to be at least one second, its overhead is not affected much by traffic load, mobility, and the number of connections.

  The overhead of AODV increases rapidly with the offered traffic load. The POO traffic experiments are exceptions where AODV fails to deliver most of the packets. AODV uses network-wide broadcast to re-discover a route when a packet is dropped and the route is considered as broken. With the increase of the offered traffic load, a larger number of packets are dropped due to congestion. This causes AODV to initiate additional route re-discoveries. AODV introduces less overhead than SAGA

only in low traffic load. In the worst case, the overhead of AODV is as ten times as that of SAGA (Figure 5.16c).

The overhead of DSR is affected by mobility. It is almost tripled when the maximum moving speed of nodes changes from 4 m/s to 20 m/s. DSR uses route cache and snooping that are not effective in highly dynamic networks. Only when mobility and the offered traffic load are low, DSR can outperform SAGA in terms of protocol overhead. Otherwise, it introduces up to 8 times overhead as SAGA does (Figure 5.17c).

- *End-to-end delay:* SAGA offers lower average end-to-end delay than AODV and DSR, because it uses the intermediate delay instead of the hop count as the routing metric. The advantage of using the new metric is significant when the offered traffic load is high. In those cases, the delay of SAGA is 50% less than that of AODV and 80% less than that of DSR (Figure 5.14d, 5.15d, 5.16d, and 5.17d).

SAGA versus DSDV

SAGA outperforms DSDV in the measured metrics at the conducted experiments. It delivers 10% more packets than DSDV with less than half of the protocol overhead. The average end-to-end delay of SAGA is almost the same as that of DSDV when the offered traffic load is less than 300 kb/s. It is around 50% to 70% of the delay of DSDV with 500 kb/s traffic, depending on mobility and the number of connections. DSDV fails to provide high delivery ratio in low traffic load. It delivers about 85% of the packets while the other protocols can deliver 95% (Figure 5.14b and 5.16b). In addition, it introduces 1 to 2 times more overhead than other protocols in high mobility and low traffic load (Figure 5.15c and 5.17c).

Additional experiments have been done with various maximum speeds ranging from 4 m/s to 24 m/s and numbers of connections ranging from 10 to 50. They lead to the similar conclusions.

5.6 Related work

Associativity-based routing (ABR) [47] is one of the first protocols that consider load as a part of the routing metric. The load is based on the number of routes in which a node is involved. Load balancing routing protocols [48, 49] use a similar idea as ABR but different methods to compute load. Various traffic loads on different routes have not been considered.

Multipath routing protocols [33, 61, 62] can be adjusted for load balancing by allowing sources to deliver packets through different paths. The source-based load balancing may still cause congestion. Even though every single source evenly distributes load over multiple paths, nodes that are involved in several paths can be overloaded.

A. Boukerche and S.K. Das present a new approach to control congestion in wireless ad hoc networks in [63, 64]. They propose a randomized version of the DSDV routing protocol called R-DSDV. R-DSDV propagates the routing messages according to a routing probability distribution rather than on a periodic basis. It controls congestion in the store-and-forward procedure. If the current queue size is over the congestion level, a newly arrived packet is dropped or queued according to a probability. The data packets have higher priorities than the advertisement packets.

The experimental evidence from two empirical wireless test-beds presented by D.S.J. Couto, D. Aguayo, B.A. Chambers, and R. Morris in [65] shows that the minimum-hop-count routing often chooses routes that have significantly less capacity than the best paths in a multi-hop wireless network. A new metric, the expected transmission count (ETX), is designed for routing protocols to find high-throughput paths [58]. The expected number of transmission is determined by the forward and reverse delivery ratios of a link, which are measured using dedicated link probe packets. The ETX metric incorporates the effect of link loss ratio and the interference among the successive links of a path. It does not account for mobility and does not route around congested links. It is complementary to the IMD metric proposed in this chapter.

C. Cordeiro, S.R. Das, and D.P. Agrawal propose contention-based path selection (COPAS) for TCP over multi-hop wireless networks [66]. COPAS monitors the MAC layer contention and accordingly changes the forward and reverse paths for a TCP connection. It enhances the performance of TCP by minimizing the likelihood of the capture problem [67]. The number of backoffs is used to measure contention. Because the number of backoffs is closely related to the number of packets that are sent during the measured time, research is needed for a more precise indication of channel contention. Intermediate nodes continuously piggyback their contention information on packets that pass through them. If the number of backoffs exceeds a predefined threshold, the route is reconstructed. In a network with heavy traffic or lossy links that result in a large number of backoffs, unnecessary route reconstructions can be caused.

MR^2RP is a delay-oriented multi-rate/multi-range routing protocol for IEEE 802.11 ad hoc networks [68]. It is designed to maximize the channel utilization and minimize the network transfer delay. The medium access control (MAC) protocol is analyzed to predict the transfer delay of a routing path. The authors assume: (a) the packet arrival process is a Poisson process, (b) all nodes have the same packet arrival rate, (c) each node knows the buffer information of every other node, (d) every node knows the connectivity matrix of the network so that the Dijkstra algorithm can be employed to find the shortest path. SAGA is based on weaker assumptions as discussed in section 5.3.1. It will be preferable if the delay is estimated locally without exchanging information among neighbors.

Quality-of-Service (QoS) routing protocols for ad hoc networks select routes with sufficient resources to satisfy certain requirements such as delay or bandwidth [69–71]. They work on a per-connection basis. The QoS routing requires the underlying MAC protocol to support and guarantee resource reservation as well as provide information and constraints about delay and bandwidth, etc. If QoS support is not available, SAGA's delay estimation methods can be extended for contention-based media access protocols to provide this information to the upper layer protocols and applications.

5.7 Conclusion

Congestion control can be a problem in ad hoc networks. Compared to the traditional solutions at the transport layer, SAGA routing protocol is implemented at the network layer. SAGA protocol integrates the channel spatial reuse with the multi-hop nature of ad hoc routing to reduce congestion. SAGA is a distance vector routing protocol that uses intermediate delay (IMD) instead of hop count as the distance. The use of IMD enables routing protocols to select routes that bypass hot spots where contention is intense. The lazy route query operation in SAGA protocol uses a special route advertisement for route discovery. Multiple queries can be included in one advertisement packet to accelerate the establishment of needed routes. SAGA provides an approach to reduce the oscillation of the value of IMD and makes the routes stable.

The use of IMD in routing decisions can enhance the performance of many routing protocols. It is especially of benefit to networks where topology changes are much less frequent than traffic changes. The lazy route query can be applied to other proactive routing protocols that do not have a dedicated route discovery operation. SAGA protocol reduces congestion at every intermediate node. It can be used as a complementary scheme to the end-to-end congestion control/avoidance mechanisms. The proposed delay estimation methods can be extended for contention-based media access protocols to provide quality of service (QoS) information to upper layer protocols and applications. The intermediate delay obtained from SAGA protocol can be used to improve the accuracy of round-trip-time (RTT) estimation for TCP connections.

This research provides methods to estimate the delay at a node using only local information. When a node has recent traffic, statistical methods are used to evaluate the mean of the delay. Otherwise, the underlying MAC protocol is analyzed and probability methods are applied to compute the expectation of the delay. We analyze the packet transmission procedure of the distributed coordination function in the IEEE 802.11 standard as a case of the practical study. These methods are applicable to other contention-based media access protocols.

A series of experiments have been conducted to study the performance of routing protocols under congestion. Two types of UDP traffic as well as the TCP traffic are considered and the offered traffic load is taken as the input parameter. The maximum moving speed of nodes and the number of connections are varied. SAGA performs better than DSDV in all our measurements. A summary of comparison of SAGA with AODV and DSR for throughput, overhead, and end-to-end delay is as follows.

- SAGA is able to deliver around 90% of the data packets with an offered traffic load up to 300 kb/s. It can offer a peak throughput of 370 to 400 kb/s in all experiments. This is 1.5 to 3.5 times as compared to the throughput of AODV and DSR.

- Overhead is measured as the ratio of the routing load to the data successfully delivered to the destination. The overhead of SAGA remains in a range of 15% to 50%. In similar cases, the overhead of AODV and DSR varies widely and increases fast as the offered traffic load goes high. The overhead of SAGA is as low as 10% of that of AODV and 12% of that of DSR in high traffic load.

- For low traffic load, the average end-to-end delay of SAGA is the same as that of AODV and DSR. When traffic reaches 500 kb/s, the delay of SAGA is 50% less than that of AODV and 80% less than that of DSR.

Evaluating SAGA protocol in an emulation instead of simulation environment is preferable for its success in real world use. In the future, we plan to use the mobile ad hoc emulator MobiEmu [72] to conduct experimental studies. The impact of the accuracy of delay estimation on the performance of SAGA protocol will be investigated. The results of the research on the lifetime of routes in mobile ad hoc networks [73] will be adopted to improve the accuracy of delay estimation. Research will be conducted to integrate SAGA's congestion reduction mechanism with the TCP congestion control algorithms. The idea of randomization [64] may be adjusted for SAGA protocol to decrease routing overhead and provide better congestion reduction.

6 HIERARCHICAL ARCHITECTURE FOR SUPPORTING MOVABLE BASE STATIONS IN WIRELESS NETWORKS

6.1 Introduction

Wireless LAN is significant for people to keep connected on the move. Stationary sites (i.e., base stations) provide high-speed network connections for mobile hosts. For instance, IEEE 802.11a supports up to 54 Mbit/s communication capacity [74]. The fixed infrastructure makes it easy to manage the network, enforce security policies, and extend the system. It, however, limits the deployment of the network in environments where wireless access to a wired backbone is either inefficient or impossible. For tactical military networks, the fixed base stations are attractive targets, therefore highly vulnerable.

Most limitations of wireless LAN, such as inflexibility and vulnerability, can be eliminated by letting base stations move. Base on this idea, we propose a new type of wireless networks called *wireless network with movable base stations* (WNMBS). WNMBS is comprised of mobile hosts and *movable base stations* (MBS). It can be rapidly deployed without any preexisting infrastructure. Flexibility can be achieved without losing much scalability. Supporting movable base stations in wireless networks introduces a lot of challenging research questions. One fundamental problem that requires investigation is how to organize MBS and effectively maintain the dynamic network topology. Because all base stations and mobile hosts are moving, the location of a host is not determinable by its network address. Traditional routing protocols for wireless LAN are not suitable in this circumstance. The ad hoc routing protocols do not scale well, as indicated in [75]. They do not take advantages of movable base stations either. Thus, the design of a new routing protocol is mandatory.

We propose a hierarchical structure to support movable base stations in wireless networks and address the issues of network maintenance and routing. This architecture is

called hierarchical mobile wireless network (HMWN). The rest of this chapter is organized as follows. Section 6.2 discusses the design considerations. The network architecture and four basic operations are described in Section 6.3. Section 6.4 presents the detail of an efficient membership management protocol. The segmented membership-base group routing protocol is proposed in section 6.5. In Section 6.6, a simulation evaluation and its result are discussed. Section 6.7 discusses related work. Section 6.8 concludes the chapter.

6.2 Design considerations

WNMBS has its unique characteristics that need to be considered in the design of the network architecture. The following issues have been taken into account.

6.2.1 Asymmetric capacity and asymmetric responsibility

Most mobile hosts are portable computing facilities such as PDA, GPS, notebook computer, etc., with portable wireless communication devices. These facilities have limited system resources and low computing capabilities. Lightweight batteries may power these facilities along with their communication devices. The weak power and the limited battery life will impose restrictions on the transmission range, communication activity, and computational power of the communication devices. Such mobile hosts can hardly afford the overheads of providing network services. On the other hand, movable base stations (e.g., workstations mounted on vehicles) are powered by heavy-duty batteries, equipped with high-speed communication devices. They are capable of providing reliable network services. The design of the network architecture should fully utilize the capacity of movable base stations and minimize computation and communication overheads for less powerful mobile hosts. For instance, computation-complex and resource-consuming operations, such as routing maintenance and authentication, are done at MBS.

6.2.2 Coordinated movement

The random way-point mobility model [9] is commonly used to generate the movement of mobile hosts in the study of ad hoc networks. According to this model, individuals move independently. The speed and direction of the motion in the new time interval have no relation to those of the motion in the previous time interval. In reality, the members belonging to a group tend to coordinate their movements. The reference point group mobility (RPGM) model [76] describes this kind of movement. RPGM partitions the network into several groups. Each group has a logical center. The center's motion defines the motion of the entire group. Each member in a group has independent random motion with respect to the logical center in addition to the group's motion.

6.2.3 Localized traffic

The reality of network traffic is that a small percentage of hosts in a domain are communicating outside of the domain at any given time. Many (if not most) hosts never communicate outside of their domain [77]. For example, it is much more likely that communication will take place between two soldiers in the same battalion, rather than between two soldiers in two different brigades. To take advantage of this kind of traffic pattern, the design of networks should give priority to intra-domain communications.

6.2.4 Heterogeneous wireless networks

In large scale applications, incompatible wireless networks, such as bluetooth networks, waveLAN networks, or satellite networks, may coexist. A desirable feature of the network architecture is the capability of accommodating heterogeneous wireless networks and providing simultaneous and seamless support for different MAC protocols. MBS that are equipped with multiple wireless network interfaces are needed to forward packets between two groups that use incompatible protocols (like routers in wired networks).

6.3 Network architecture

Based on the considerations discussed in the previous section, *hierarchical mobile wireless network* (HMWN) is designed to support WNMBS. It can be applied to ad hoc networks as well to build a virtual hierarchy. To broaden its application, HMWN is presented in the following sections in a generic way, in which movable base stations are treated as a special type of mobile hosts.

6.3.1 Definitions

The following is a set of definitions that will be used in the rest of the chapter.

Definition 1: A group is a set of mobile hosts. Each group has one representative (i.e., agent). A group is denoted as $group(M)$, where M is the agent. A host can be an agent for at most one group. The home group (HG) is where the mobile host registers its membership. A foreign group (FG) is a group other than the HG. The current group (CG) is the one to which the host currently attached. The corresponding group agents are called home group agent (HGA), foreign group agent (FGA), and current group agent (CGA), respectively. Usually, movable base stations are chosen to be agents.

For every mobile host, its HG is assigned by the "Grouping" operation.This relationship keeps unchanged during the life-time of the network. A mobile host's CG is changed when the "Migration" operation completes.

Definition 2: The groups in a HMWN system form a group hierarchy. The level of a group G, which is denoted as $lv(G)$, represents how close it is to the root of the hierarchy. The lower the level is, the closer the group is to the root. The level of the root group is 0.

Any mobile host can be a member of two different groups, in one of which it is the agent, in the other one it is a non-agent member. Suppose the agent of group G_1 is a non-agent member of group G_2, then $lv(G_1) = lv(G_2) + 1$. If a mobile host MH is a member of group G, the level of MH is

$$lv(MH) = \begin{cases} lv(G), & MH \text{ is the agent of group } G; \\ lv(G) + 1, & \text{otherwise.} \end{cases} \tag{6.1}$$

Definition 3: A group G_1 is a subgroup of group G_2 if and only if

1. the agent of G_1 is a non-agent member of G_2
2. or the agent of G_1 is a non-agent member of one of G_2's subgroups.

G_2 is called a supergroup of G_1. Operators $sub(G_1, G_2)$ and $sup(G_2, G_1)$ are used to denote that G_1 is a subgroup of G_2 and G_2 is a supergroup of G_1, respectively. In HMWN, sub and sup are partial orders.

Definition 4: A domain derived from a group G consists of and only consists of G and all its subgroups, denoted as $domain(G)$. The group agent of G is also the domain agent of $domain(G)$. Derived domains have the following property.

$$domain(G_1) \subseteq domain(G_2) \Longleftrightarrow sub(G_1, G_2) \tag{6.2}$$

Definition 5: A closure domain of two groups G_1 and G_2, denoted as $closure(G_1, G_2)$, is the smallest derived domain that contains G_1 and G_2. Formally, $closure(G_1, G_2) = domain(G)$ if and only if

1. $G_1 \subseteq domain(G)$ and $G_2 \subseteq domain(G)$
2. For any derived $domain(G')$, $G_1 \subseteq domain(G')$ and $G_2 \subseteq domain(G') \Longrightarrow domain(G) \subseteq domain(G')$

6.3.2 An example

Figure 6.1 is an example of the HMWN system. Every small square represents a mobile host and the dark ones are group agents. A solid line between two mobile hosts represents the wireless link. The dashed line circles represent groups and the solid line circles represent derived domains. The root group only contains three members $\{A, B, C\}$, where A is the agent. There are two level 1 groups, $\{B, D, E\}$ and $\{C, F, G\}$. B and C are group agents, respectively. D, E, F, and G are agents for level 2 groups. Figure 6.2 shows an alternate representation of the group hierarchy, where every group is represented by its

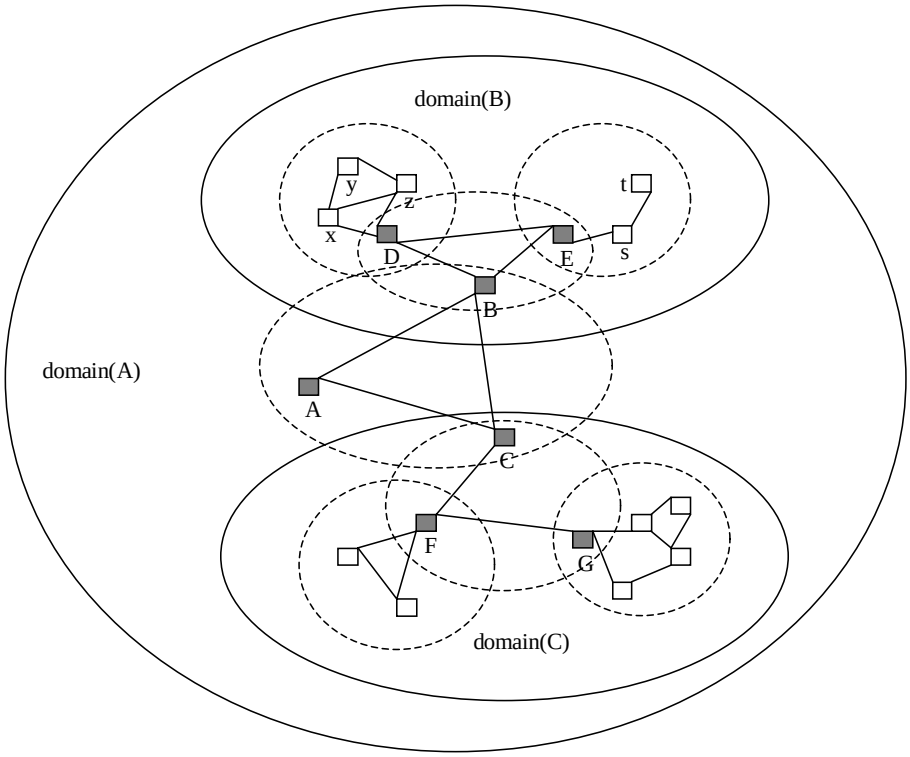


Figure 6.1. Hierarchical mobile wireless network

agent at a lower level. In this network, the $domain(A)$ contains 7 groups and all hosts in the system. The $domain(B)$ consists of 3 groups and mobile hosts $\{B, D, E, s, t, x, y, z\}$.

In HMWN, mobile hosts that belong to the same group use a multi-hop ad hoc routing protocol to communicate. Communication with a host outside the group is accomplished by the segmented membership-based group routing protocol presented in Section 6.5.

6.3.3 Basic operations

The following four basic operations are defined for setting up and maintaining a HMWN system.

1) Grouping is the operation used to set up the static membership in a HMWN system. It is only performed at the bootstrapping phase. "Grouping" is accomplished in two steps.

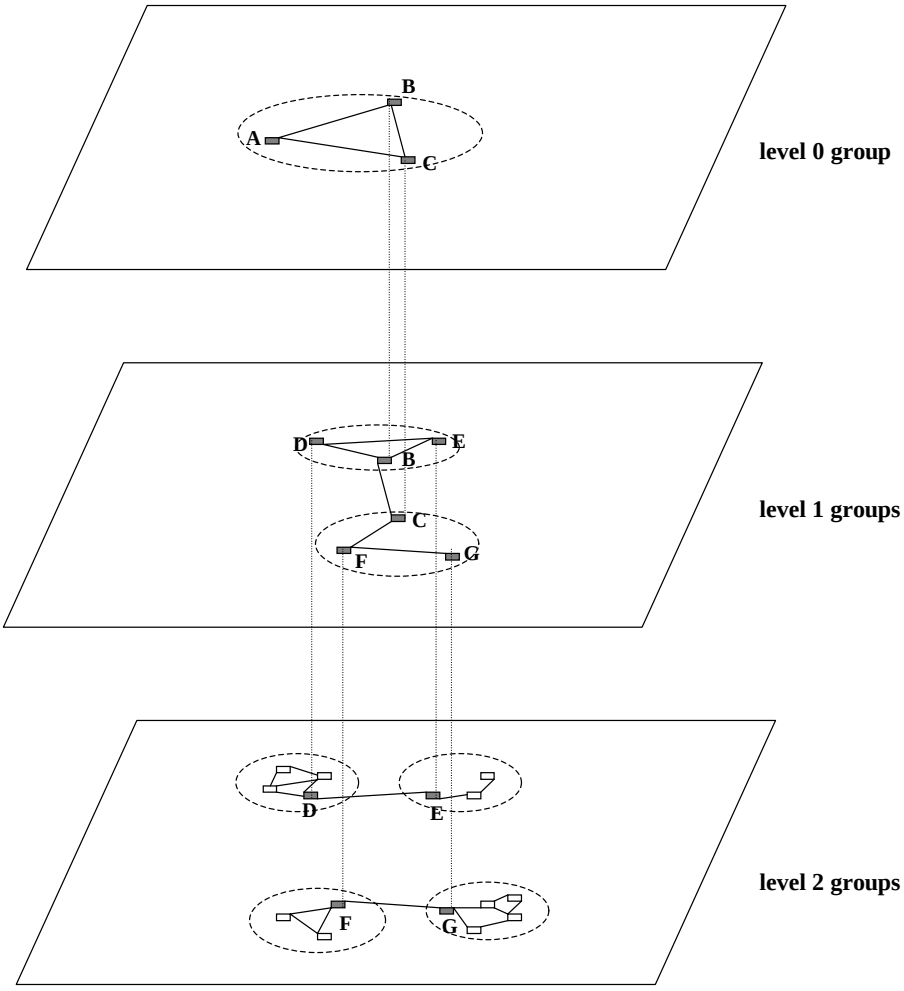


Figure 6.2. Hierarchy of groups

The first is to organize mobile hosts into groups (i.e., assign HG for each mobile host). The second is to determine group agents (HGA). The criteria for "Grouping" include

- Mobility: If a set of mobile hosts are going to coordinate their movements, they may form a group.
- Organization: If all mobile hosts belong to a organization that has a well established hierarchy, the hosts can be grouped based on this hierarchy.
- Wireless MAC protocol: If multiple wireless MAC protocols are used in the network, the mobile hosts that support compatible protocols may be grouped together.
- Capacity: Capacity is used to determine group agents. The higher the capacity is, the greater the chance is that the mobile host will be chosen as an agent. Several factors are taken into consideration when the capacity of a mobile host is evaluated, e.g., the computation capability, system resource, power level, communication bandwidth and range, the number of wireless network interfaces.

This operation can be done in a distributed or centralized way.

- Mobile hosts may autonomously organize themselves into groups, then supergroups. In the autonomous procedure, each agent will exchange the organization, the MAC protocol, and capacity information with its neighbors to determine the static membership relationship. This process is accomplished in a distributed way. It is hard to obtain the optimal result.
- A trusted authority may take charge of the operation. Every mobile host reports its information to the authority. The authority employs some global optimization algorithm to establish the hierarchy and distributes the result to all participated hosts.

The first scheme is also suitable for self-organizing ad hoc networks, in which mobile hosts have no prior knowledge about the network. In practice, a mobile host is usually assigned a home agent before joining the network, or knows some information that is

helpful for grouping. Automatically grouping in a distributed fashion itself is a non-trivial problem. We do not address it in this dissertation.

2) Registration is the operation that a mobile host must complete before it can connect to the network. "Grouping" only determines the static membership. "Registration", along with "Leaving" and "Migration", maintains the dynamic topology of the network (e.g., CG for a mobile host). Registration takes place between a mobile host MH and its HGA. One-hop registration is recommended to reduce the possibility of denial-of-service and man-in-the-middle attacks.

This operation begins with MH broadcasting the"Registration" request. If the HGA is within the neighborhood, the operation continues with an identity verification process. Upon successfully registered, MH will obtain the group information such as group ID, group shared secrets, etc. from the HGA, and set the HGA to be its CGA. In case that MH itself is an agent of another group, all hosts in the derived $domain(MH)$ implicitly become members of the network. MH keeps moving and sending out the request periodically if it cannot reach the HGA directly. Other hosts may provide aid to locate the HGA so that MH can adjust its movement.

If connectivity rather than security is preferred, remote registration (i.e., MH registers itself to the HGA via intermediate hosts) will be allowed.

3) Leaving operation is completed by group agents. It may be triggered by two events.

- When a mobile host MH decides to leave the network (along with all hosts in the derived $domain(MH)$), it sends a "leave group" message to its CGA.
- When the agent finds out that the route to a mobile host MH is broken, it starts a *Leaving Timer*. If a route to MH cannot be reestablished or a "Migration"message has not received within the *Leaving Interval Time* as described in equation 6.3, the agent starts the "Leaving" operation.

Leaving Interval Time

$$= Robustness \times Ad\_Interval \times (Max\_Hop + 1) \tag{6.3}$$

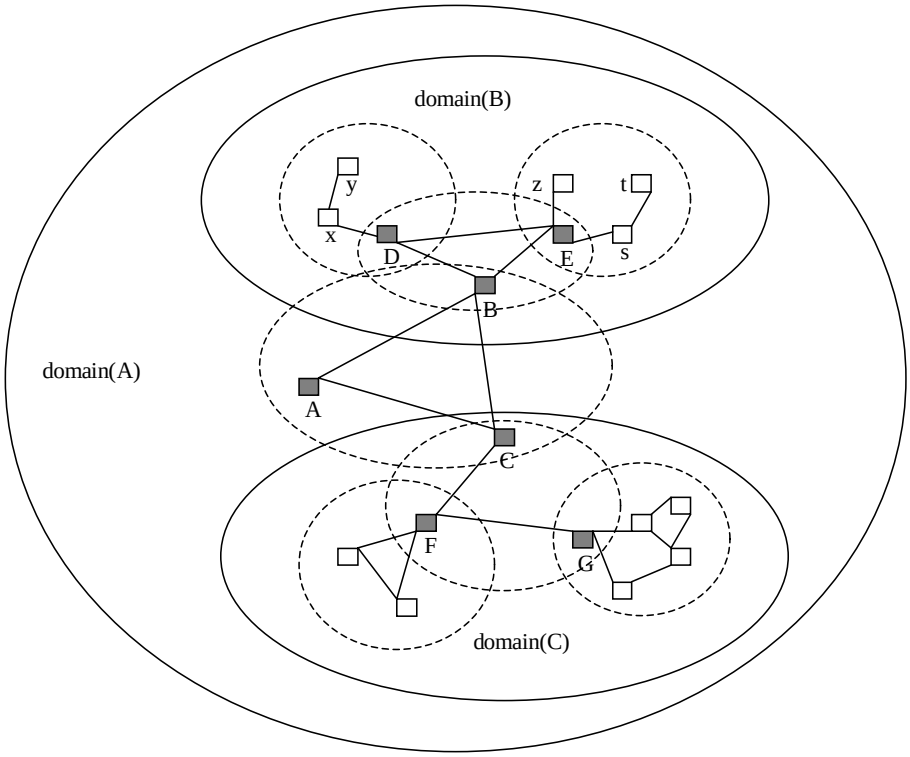


Figure 6.3. After migration

The *Ad_Interval* is the time interval between the route advertisements sent out by a host. The *Max_Hop* is the hop number of the longest route in the agent's routing table. *Leaving Interval Time* is the maximum time it will take to get MH's routing information if MH is still a member of the group. The *Robustness* allows tuning for the expected packet loss on wireless links. The "Leaving" operation is able to tolerate ($Robustness - 1$) failures. Thus *Robustness* must be greater than 1. If the system is expected to be lossy, the *Robustness* may have a larger value.

After the CGA of MH updates the membership information, it will forward the "leave group" message to its own CGA.

4) Migration operation is initiated by a mobile host that decides to leave its current group and join a foreign group. Usually, when a host MH realizes that the CGA is no longer reachable, it starts this operation by sending out a "Migration" request. Foreign

agents that are in the neighborhood reply this request based on the MAC protocol compatibility and capacity, and the security policy that determines whether or not to provide the migration support. MH chooses the FGA whose reply comes first, sets it to be the CGA, and invokes the hand-off procedure. Every agent that replies the request will start a timer. When the timer expires, the agent will cancel the operation.

Figure 6.3 illustrates the topology of the example HMWN system shown in Figure 6.1 after mobile host z migrated from $group(D)$ to $group(E)$.

6.4 Membership management

Maintaining the network topology in an efficient way is significant in a HMWN system. Essentially, it is a membership management problem because the mobile hosts are organized as hierarchical groups. The following subsections present the membership management protocol.

6.4.1 Data structure

The membership information is mainly used for two purposes. The first is to verify the identity of a host (i.e., the static membership). The second is to help routing protocols to choose the proper route to forward packets (i.e., the dynamic membership). Each agent G maintains two separate tables.

Static_Member_Table contains the identification information of mobile hosts whose HGA is G. This table is mainly used by security protocols such as authentication and identity verification. The table has an entry for every potential member, which is a 3-tuple {*ID*, *shared_secret*, *public_key*}. Initially, an entry only contains the *ID* and the *shared_secret*. After registration, the public key of the member will be recorded in the entry.

Current_Member_Table contains the information of all the mobile hosts that currently belong to the domain $domain(G)$. The entry of the table is a 3-tuple {*ID*, *intermediate_host*, *home_agent*}. The *intermediate_host* is the non-agent member in this group

whose *Current_Member_Table* also contains the mobile host (i.e., the mobile host is in the domain derived from the *intermediate_host*). The *home_agent* is the HGA of the mobile host. This table is used by the routing protocol to locate mobile hosts.

Depend on the size of the tables and the available memory, these two tables can be maintained using a hash table, a ordered list, or a trie to accelerate the searching process. “Registration”,“Leaving”, and “Migration” will operate on these two tables.

6.4.2 Registration

Upon successful registration, a host will get the group information from the agent. The host sets the agent to be its CGA. In case that security protocols are deployed, a mutual challenge-and-response process will be initiated to verify the identity of the host and the agent. If verification succeeds, the agent will record the host’s public key in the corresponding entry of *Static_Member_Table*, the host will get the group key, the agent’s public key, and other information required by the security protocols such as a certificate.

The host will send a list of all members in its *Current_Member_Table* to the agent so that all members in its derived domain will be implicitly registered. This list will be forwarded via the path from the agent to the root of the hierarchy. Every agent on the path will add the members to its own *Current_Member_Table*.

6.4.3 Leaving

When a host leaves a group, all members in its derived domain also leave the group implicitly. The host sends a list of all members in its *Current_Member_Table* to the agent. This list will be forwarded via the path from the agent to the root of the hierarchy. Every agent on the path will remove the members from its own *Current_Member_Table*.

6.4.4 Migration

When a mobile host MH is leaving the current group G_1 and joining another group G_2, both the CGA and the FGA will update their *Current_Member_Table*. If MH is an agent, all mobile hosts in $domain(MH)$ also implicitly leave $domain(G_1)$ and join the $domain(G_2)$. After joining the foreign group, MH will send messages to the FGA and the CGA to help them update the membership.

At the FGA side, MH sends the following message to the foreign agent.

$$[ADD, ID, previous\_agent, member\_list]$$

where *ID* is the identification of MH, *previous_agent* is MH's CGA before joining the group, *member_list* is MH's *Current_Member_Table*.

For each host in the *member_list*, the FGA adds it to the *Current_Member_Table* if it does not exist already, and sets the *intermediate_host* to the MH that sent the message. If *previous_agent* is not a member in the *Current_Member_Table*, the FGA sends the same message to its own CGA. Every agent that receives the message will update the membership as well.

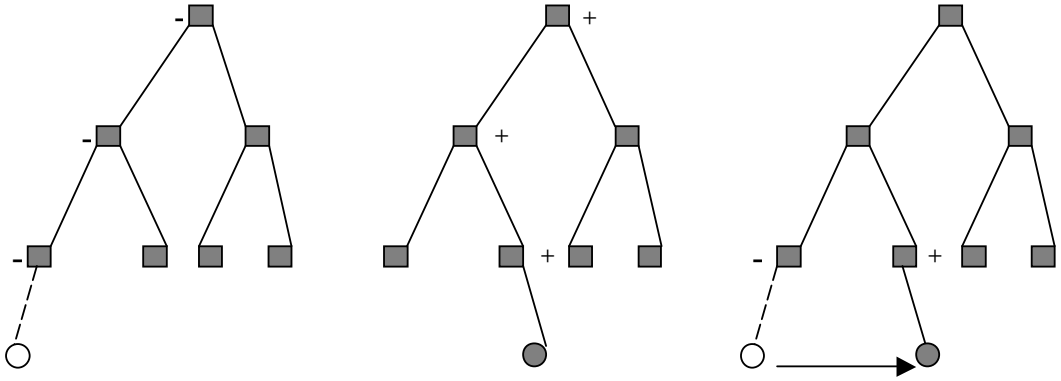
At the CGA side, MH sends the following message to the current agent.

$$[REMOVE, ID, foreign\_agent, member\_list]$$

where *ID* is the identification of MH, *foreign_agent* is the agent of the foreign group, *member_list* is MH's *Current_Member_Table*.

If the *foreign_agent* is also a member in the *Current_Member_Table*, which means the MH moves from one sub-group to another, then the CGA does nothing. Otherwise, it removes every host in the *member_list* from the *Current_Member_Table* and forwards the message to its own CGA. Every agent that receives the message will update the membership as well.

Figure 6.4 shows the difference between "Registration", "Leaving", and "Migration" operations with respect to the modification of *Current_Member_Table*. The small circles represent the mobile host. For "Registration" and "Leaving", the effect will be propagated to the root of the hierarchy. Thus $lv(A) + 1$ unicast are required, where A is



\- Remove members from Current_Member_Table

\+ Add members to Current_Member_Table

Figure 6.4. Membership modification

the agent. For "Migration", the effect is only propagated to the agent of the domain $closure(previous\_agent, foreign\_agent)$. The number of required unicast is

$$lv(previous\_agent) + lv(foreign\_agent) \\ -2 * lv(closure(previous\_agent, foreign\_agent)) \quad (6.4)$$

6.5 Segmented membership-based group routing

Segmented membership-based group routing (SMGR) protocol is proposed for the HMWN system to take advantage of the hierarchical group structure and available membership information.

6.5.1 Data structure

SMGR protocol requires two tables. One is the routing table, in which each entry is a 4-tuple <*destination, next_hop, distance, sequence_number*>. The *sequence_number* represents the freshness of the route. Each host maintains a *sequence_number* for itself.

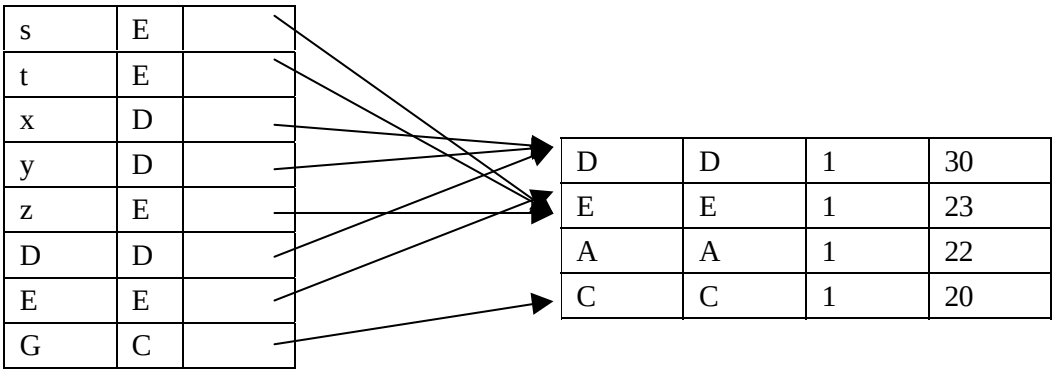


Figure 6.5. Membership table and routing table

This number is monotonically increasing. Only routes to the group-mates are maintained in the routing table. These routes are updated using DSDV [1] protocol.

The other is the membership table, in which every entry is a 3-tuple <*final_destination*, *intermediate_host*, *routing_entry*>. *routing_entry* is a pointer to the entry in the routing table that specifies the route to the *intermediate_host*. Every entry in *Current_Member_Table* has a corresponding entry in this table.

Take host B in Figure 6.3 as an example, Figure 6.5 shows the routing table, the membership table, and the pointers maintained by B.

The size of the routing table is bounded by the size of the group, which is nearly a constant.

SMGR protocol will add a header, which is a 4-tuple <*source*, *final_destination*, *intermediate_host*, *next_hop*>, to each packet. The header is used to route the packet.

6.5.2 Routing

When a host receives a data packet, either from another host or from a application running on itself, it takes different actions to forward the packet, based on whether it is the *intermediate_host* or not. Here we assume that the routing table is up-to-date.

If the host is not the *intermediate_host*, it simply forwards the packet based on the available routing information. Otherwise, it is responsible for locating the next *intermediate_host* (or the final destination) from its membership table. The packet is forwarded to the next *intermediate_host* if it is located, otherwise, the packet is forwarded to the CGA. Since the root group agent can locate any mobile host, the packet will eventually reach the destination. In the routing process, "membership expires" or "redirect" message may be sent out to update the membership information.

A host will removes the corresponding entry from the membership table when it receives a "membership expires" message. When a host receives a "redirect" message, it adds an entry in the membership table, set *intermediate_host* to be the redirected host.

Figure 6.6 shows the pseudo code of the SMGR routing algorithm.

6.6 Evaluation

A simplified version of SMGR has been implemented in the network simulator ns2 [18]. In this version, the membership modification is completed through broadcast instead of unicast. It is predictable that more protocol overhead will be introduced by the simplification. We have also implemented the computation delay component to simulate different computation capacities. The purpose of the experiment is to evaluate the scalability of HMWN in terms of protocol overhead. Because there is no other routing protocol designed for WNMBS, we apply HMWN to ad hoc networks for comparison purpose. Since SMGR utilizes distance vector, we compare it with two distance vector based ad hoc routing protocols, DSDV and AODV.

In this experimental study, we take the protocol overhead (protocol load divided by throughput) [8] as the metric to evaluate the scalability of routing protocols. The experiments simulate a 1000m x 1000m area. Random way-point mobility model is used to generate movement for mobile hosts, the maximum speed is $5\mathrm{m/s}$, the pause time is 3 seconds. The number of end-to-end connections is equal to the number of hosts. The source-destination (S-D) pair of each connection is randomly chosen. Constant bit rate

```
if it is the final_destination
  send the packet to the corresponding application;
else if it is the next_hop
  find out the route to the intermediate_host;
  change next_hop and send out the packet;
else if it is the intermediate_host
  search the Current_Member_Table;
  if an entry e exists for the final_destination
    set the intermediate_host to e.intermediate_host;
    get the routing table entry re;
    set the next_hop to re.next_hop;
    send out the packet;
    if the packet comes from a host which is in the
    same group of e.intermediate_host
      send a "redirect" message to the host;
  else
    if the packet comes from a host of which it is
    the agent
      set the intermediate_host to CGA;
      send the packet to CGA;
    else
      send out a "membership expires" message to
      the source;
else if it is the source
  search the Current_Member_Table;
  if an entry e exists for the final_destination
    set the intermediate_host to e.intermediate_host;
    get the routing table entry re;
    set the next_hop to re.next_hop;
    send out the packet;
  else if it is not the root of the hierarchy
    set the intermediate_host to CGA;
    send the packet to CGA;
  else
    drop the packet and notify the application;
else
  drop the packet silently;
```

Figure 6.6. SMGR routing algorithm

(CBR) traffic is generated for all connections. The number of hosts ranges over {20, 30, 40, 50, 60}. For each value, five scenarios are created. Individual simulation runs 1000 seconds. The protocol overhead is computed from the traffic trace file.

The result of the experiment is shown in Figure 6.7. The curves present the mean value of the protocol overhead for each protocol. When the number of hosts is less than 40, three protocols have similar performance, with AODV being outperformed a little bit. When the number of hosts reaches 60, the overhead of DSDV is about 50% higher than

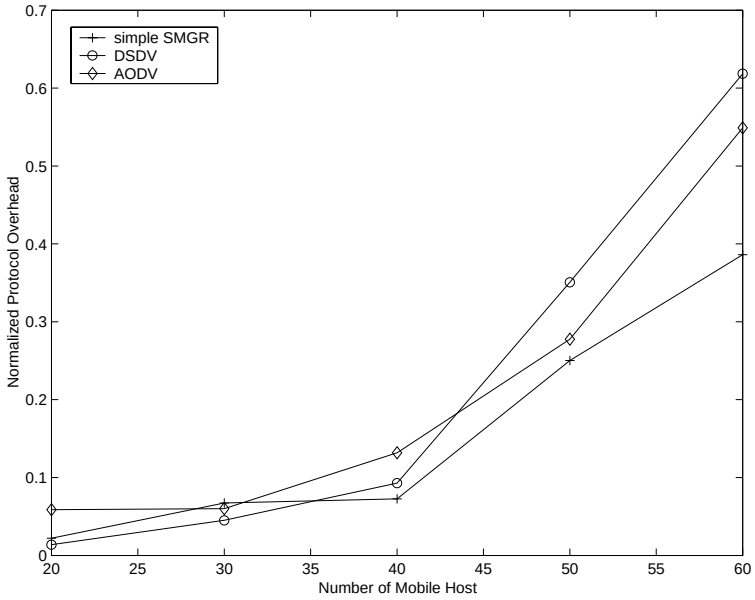


Figure 6.7. Protocol overhead versus number of mobile hosts

that of the simple SMGR, while the overhead of AODV is about 38% higher. The result shows that the simple SMGR is more scalable in terms of protocol overhead.

Considering the random way-point mobility model and the random traffic pattern that are used for the experiments favor ad hoc networks, and the simple SMGR introduces extra protocol overhead because of unnecessary broadcast, we may expect a HMWN system supported by SMGR protocol to be more scalable with the presence of movable base stations.

6.7 Related work

Examples of integrated heterogeneous wireless networks include the integrated ad hoc and cellular networks. X. Wu et al. proposed mobile-assisted connection-admission (MACA) channel allocation scheme to achieve load balancing in a cellular network [6]. In MACA, some special channels are used to connect mobile units from different cells.

When a mobile unit cannot connect to its own base station due to heavy load, it may be able to get connected to its neighboring cell's base station through a two-hop link.

A similar approach, integrated cellular and ad hoc relaying systems (iCAR), is proposed by H. Wu et al. in [5]. It addresses the congestion problem due to unbalanced traffic in a cellular system and provides interoperability for heterogeneous networks. The basic idea is to place a number of ad hoc repaying stations at strategic locations, which can be used to relay signals between mobile hosts and base stations.

Multihop cellular networks (MCN) is presented by Y.-D. Lin and Y.-C. Hsu in [78]. MCN allows wireless transmission to go through mobile stations in multiple hops in the cellular networks. It reduces the number of required base stations and improves the throughput, while limiting path vulnerability encountered in ad hoc networks.

H. Luo et al. proposed the unified cellular and ad-hoc network (UCAN) to enhance cell throughput and maintain fairness in the third generation (3G) data networks [79]. The scheduling algorithm for the 3G base station is refined so that the throughput gains of active clients are distributed proportional to their average channel rate. A secure crediting mechanism is developed to motivate users to participate in relaying packets for others.

In [80], S. Nesargi and R. Prakash present a distributed, dynamic channel allocation (DCA) algorithm for virtual cellular networks where the fixed base stations are replaced by mobile base stations. Principles of mutual exclusion pertaining to distributed computing systems are employed in the development of the algorithm. This work to some extent provides the physical layer support to our research in WNMBS.

R. Ramanathan and M. Steenstrup proposed a MMWN system, an acronym for multimedia support for mobile wireless networks [81]. A MMWN system consists of switches and endpoints. While both can be sources of or destinations for packets, only switches can route packets. The switches and endpoints are organized using hierarchical clustering to provide support for quality of service. Routing information is distributed in the form of link states, which contains connectivity and service information pertaining to clusters at all levels within the hierarchical control structure. The quality-of- service routing is realized by establishing and maintaining a virtual-circuit between the source and destination.

S. Banerjee and S. Khuller present the design and implementation of a clustering scheme for hierarchical control in multi-hop wireless networks in [82]. The clustering problem is defined in a graph theoretic framework. The properties of the underlying communication graphs of wireless network are exploited to achieve desired solutions, which satisfy the requirements such as cluster connectivity, upper and lower bounds on cluster size, low overlap between two clusters, etc.

Many research efforts [83–85] introduce structures on ad hoc networks to provide scalable solutions for routing, location management, and resource allocation. Most schemes assume that ad hoc networks are self-organized to discover and maintain the structure. It requires extra message exchanges that may consume a large portion of the limited bandwidth.

6.8 Conclusion

We present a hierarchical structure to support movable base stations in wireless networks. In a HMWN system, mobile hosts form hierarchical groups. Group agents (usually movable base stations) take major responsibilities for managing membership and routing packets. HMWN integrates the routing protocol with membership management to reduce overhead. It is capable of accommodating incompatible wireless MAC protocols and managing heterogenous wireless networks in a unified way. Four basic operations that are used to set up and maintain the hierarchy have been discussed. The detail of an efficient membership management protocol is presented. The segmented membership-base group routing protocol for HMWN is proposed. An experimental study is carried out to compare the scalability of SMGR with AODV and DSDV ad hoc routing protocols in terms of protocol overhead. The SMGR outperforms these two protocols for about 50% when the number of hosts reaches 60.

This work is only the first step in the research on wireless networks with movable base stations. We are developing multiple MAC protocols and supporting modules in ns2 to carry out experimental studies on HMWN and SMGR protocol with respect to

other performance metrics. Automatically grouping in a distributed way and introducing security mechanisms are the next steps. We hope this work will help to build a foundation for the research of flexible, scalable, and secure wireless networks.

7 SECURING WIRELESS NETWORKS WITH MOVABLE BASE STATIONS

7.1 Introduction

7.1.1 Wireless network with movable base stations

Wireless communication technology is significant in networking infrastructure. Mobile ad hoc networks and wireless LAN are two typical packet-switching wireless networks [1].

A mobile ad hoc network consists of mobile hosts that communicate with each other over multi-hop wireless links in a collaborative way [86]. There is no fixed infrastructure or stationary base station to coordinate communications. These characteristics provide users with maximum flexibility, at the cost of limitations on scalability. The scalability problem is analytically studied in [75]. The result shows that even the most scalable routing protocol introduces a total overhead of $O(N^{1.5})$, where N is the number of hosts. The experimental study also shows that the increase of the number of hosts is the dominant cause for performance degradation [56].

In a wireless LAN, stationary sites (i.e., base stations) provide high-speed network connections for mobile hosts. The fixed infrastructure makes it easy to manage the network, enforce security policies, and extend the system. It, however, limits the deployment of the network in environments where wireless access to a wired backbone is either inefficient or impossible. For tactical military networks, the fixed base stations are attractive targets, therefore, highly vulnerable.

Most limitations of wireless LAN, such as inflexibility and vulnerability, can be eliminated by letting base stations move. We deviate from the conventional wireless networks

[1] Sensor network is a new class of wireless networks that has become an attractive research area. A sensor network is essentially an ad hoc network that consists of a large number of tiny disposable and low-power devices. These devices are immobile, or have low mobility as compared with hosts in mobile ad hoc networks.

and propose wireless network with movable base stations (WNMBS). WNMBS is comprised of mobile hosts and movable base stations. The movable base stations typically are mounted on vehicles such as tanks and trucks and form a mobile backbone. They have more resources than mobile hosts in terms of memory, computation capability, transmission power, energy supply, etc. Neighboring base stations use wireless links to communicate. Because all base stations and mobile hosts are moving, the location of a node is not determinable by its network address. The traditional network architecture and routing protocols for wireless LAN are not suitable in this circumstance. We develop hierarchical mobile wireless network (HMWN) to support WNMBS. The details of HMWN, including the network maintenance mechanism, the routing protocol, and control overhead, are presented in the previous chapter.

7.1.2 Security issues in WNMBS

Achieving security in a wireless network is challenging because of:

- The use of wireless channels that are susceptible to link attacks [87];
- Roaming in a hostile environment with relatively poor physical protection that makes a mobile host vulnerable;
- Dynamic network topology and memberships.

Security mechanisms have been proposed for protecting a single wireless link, such as secure protocols for wireless LAN [88, 89]. The use of cryptography to secure ad hoc routing protocols has been investigated in [90–93]. A scalable security solution for mobile ad hoc networks is proposed in [94]. The idea of threshold secret sharing and secret share updates is used to tolerate intrusions. Ariadne is an on-demand ad hoc routing protocol that provides security against one compromised node and arbitrary active attackers [95]. Ariadne relies only on symmetric cryptography, thus it does not require a trusted hardware or powerful processors. These research efforts require mobile hosts to be able to identify

each other based on some priori knowledge. The following mechanisms are usually used for identification. They have deficiencies when being applied to wireless networks.

- All hosts share a secret key so that everyone can prove its membership by showing the knowledge of this secret key. This scheme is relatively insecure. If one host is compromised, the whole system is compromised.

- Every host knows the public keys of all other hosts so that it can identify a host by using public-key cryptography. This scheme is not scalable. It requires all hosts to be known before the network is set up. If a host wants to change its public/private key pair, it has to inform all others in the system.

- There exists a centralized trusted entity, such as a key distribution center (KDC) or a trusted third party (TTP), which knows the public key of every host. Two hosts can use some authentication protocol, such as Yahalom, DASS, Woo-Lam, etc. [96], to authenticate each other. In this scheme, the centralized entity is the bottleneck of a system that will decrease the effectiveness of security solutions. It is prone to denial-of-service attacks and may become the single point of failure.

In a WNMBS, the mobile backbone (i.e., base stations) is typically maintained by system administrators (e.g., service providers) and provides network services to mobile users. The base stations, with appropriate security enhancements, naturally form a distributed trusted entity that is capable of balancing service load and tolerating site failures. To utilize movable base stations as a distributed trusted entity, research questions, such as how to organize base stations, how to distribute keys, and how to authenticate mobile hosts, need investigation.

We present mechanisms integrated with HMWN to secure WNMBS. The protection of network infrastructure, authentication and key distribution, and secure roaming support are addressed. The rest of the chapter is organized as follows. Section 7.2 discusses the security objective and assumptions. Secure packet forwarding mechanism that protects the network infrastructure is proposed in section 7.3. Section 7.4 presents the authentication protocol. Section 7.5 discusses the secure roaming support. The computation overhead of

the security mechanisms is numerically investigated in Section 7.6. Section 7.7 concludes the chapter.

7.2 Security objective and assumptions

We focus on protecting the network infrastructure against both passive and active attacks, such as insertion, modification or replay of control messages, and traffic analysis. End-to-end data communications are protected from unauthorized access by higher layer protocols. As long as the network infrastructure is available and secure, the two ends of a communication can always set up a symmetric secret key by using some key-exchange algorithm such as Diffie-Hellman or COMSET [96]. The data packets can be encrypted by using the secret key to ensure confidentiality and integrity.

The objective is achieved by deploying secure packet forwarding and authentication protocols that are presented in the following sections. These security mechanisms are based upon the following assumptions:

- The wireless communication is robust with respect to attacks against the physical layer. These layers are well protected by lower-layer mechanisms, such as anti-jamming techniques [97, 98].
- The underlying cryptography primitives, such as digital signature and encryption, are practically secure (i.e., they are unbreakable with current computation power).
- All base stations know each other's public key (For instance, if each group has 50 members, a 5000-node networks requires about 100 base stations to maintains about 150 public keys, instead of 5000 nodes, most of which are resource-poor mobile hosts, to maintain 5000 public keys.).

7.3 Protection of network infrastructure

Unlike a wired network where the infrastructure is protected by physically securing the cables, the infrastructure of a wireless network is protected by ensuring that every mobile

host has correct knowledge about the current network topology and the memberships. A mobile host obtains this knowledge by securely exchanging control information, such as neighbors, routes, etc., with other trustworthy hosts. An adversary should not be able to eavesdrop, insert, or modify the information. It is guaranteed by using unforgeable encryptions.

In addition to routing and control messages, packet headers need to be encrypted. Although encryption hides the content of a message, the packet header that contains the source, the destination, and the next hop will expose the relationships among the involved hosts. This is a reason why eavesdropping technology such as Carnivore is useful even in the presence of unbreakable communication [99]. Preferred targets can be identified in this way and attacks can be concentrated on the nerve centers. Encrypting packet headers will effectively obfuscate relationships among hosts.

We assume that each mobile host in a HMWN system has a public/private key pair and group members know the public key of the group agent. Each group agent maintains a potential member list (defined by the "Grouping" operation), which contains the public keys of mobile hosts that might be a member of that group.

The secure packet forwarding algorithm is designed for the protection of the network infrastructure. To use a symmetric cipher, each group has a group-shared secret key. This key is maintained and distributed by the group agent. It is renewed periodically, when a mobile host joins or leaves the group, or at the time a compromised host is discovered.

When a mobile host X registers to a group, it authenticates itself with the group agent and gets the group shared key K by invoking the protocol presented in Section 7.4. X uses K to communicate with other group members confidentially. A group agent may know two groups' shared keys.

The pseudo-code in Figure 7.1 shows how X handles (sends, receives, and forwards) packets after joining the group. This algorithm integrates with the routing protocol to realize secure packet forwarding.

Encrypting and checking headers when sending, receiving, or forwarding packets serve the following purposes.

Part I: sending a packet P:

```
1.  X uses K to encrypt the header
2.  if P is a routing or control packet
3.      it uses K to encrypt the body of P
4.  X transmits encrypted packet P
```

Part II: receiving a packet P:

```
1.  X decrypts and checks the header
2.  if X itself is the destination and P is a control packet
3.      it decrypts the body
4.  else
5.      X makes any necessary modifications to the header
6.      if X is a group agent AND P is sent from one group to another
7.          it encrypts the header with the destination group's key K'
8.          if P is a routing or control packet
9.              it decrypts the body with K and re-encrypts it with K'
10.     else
11.         X encrypts the header with K
12.     X forwards P to the next hop
```

Figure 7.1. Secure packet forwarding algorithm

1. The correctly encrypted header testifies that a packet is sent by a member of the group. Adversaries cannot produce such a header because they do not know the secret key. It prevents the network from being flooded with false control and data packets generated by malicious hosts.

2. The encrypted header ensures that routing and location information, which is valuable to attackers, will not be disclosed. For example, if an adversary captures a packet and knows the next hop is host X, he can tell that X is within the radio range of the sender and initiates attacks against X.

7.4 Authentication and key exchange

The capability of a mobile host to authenticate itself and obtain the group-shared key is the basis of secure packet forwarding. In this section, we discuss the authentication and key exchange protocol.

7.4.1 Notations and protocol

We introduce the following notations.

- X, Y: mobile hosts
- G: group agent
- gid: group ID
- R: request. It could be a request for joining a group or a request for secure roaming support.
- T: time stamp
- K: shared secret key
- K_X: public key of host X
- M: message
- E_X(M): encrypting message M with host X's public key so that only X can read M
- S_X(M): signing message M with X's private key so that every host that knows X's public key can verify that M is signed by X
- V_X(M): verifying message M with X's public key
- E_K(M): encrypting message M with secret key K
- D_K(M): decrypting message M with secret key K

1. X→G: <gid, X, R, S_X(gid, X, R)>
2. G: V_X(gid, X, R)
3. G→X: <gid, G, X, R, E_X(gid, G, X, R, K, S_G(gid, G, X, R, K))>
4. X: V_G(gid, G, X, R, K)
5. X→G: <X, G, E_K(X, G, R)>

Figure 7.2. Authentication and key exchange protocol

The protocol shown in Figure 7.2 illustrates the process invoked by the "Registration" operation when host X joins a group whose ID is "gid". This protocol does not use a time stamp to guarantee the freshness of the request because a mobile host only registers once in the network. The agent can tell if the request is new by examining the membership information it maintains.

7.4.2 Correctness

The correctness of the protocol can be proven by adopting the logic of authentication [100]. The following terms are used.

- X *believes* P: host X thinks that a statement P is true.
- X *sees* P: host X receives a statement P.
- X *controls* P: host X is trusted in the matter of the statement P.
- *fresh(P)*: P is a fresh statement.

The following three deductions are used in the proof.

1. X sees S_Y(P) and X believes fresh(P) ⇒ X believes *Y believes P*.
2. X believes *Y believes P* and X believes *Y controls P* ⇒ X believes P.
3. X sees E_K(P) and X believes *X and Y share K* and X believes *Y controls P* ⇒ Y believes K.

Theorem: The authentication and key exchange protocol shown in Figure 7.2 authenticates X and G and establishes a shared key between X and G.

Proof: The following believes are held before the protocol starts.

1. G believes *X controls R* (because G knows that X initiates the request)
2. X believes *G controls K* (because X knows that G generates the shared key).
3. Both X and G believe fresh(R) (because X is yet a member of the group)

After step 2:

- G believes *X believes R*
- G believes R

After step 4:

- X believes fresh(K)
- X believes *G believes K*
- X believes K

After step 5:

- G believes *X believes K*

At the end of the protocol:

- X believes K
- X believes *G believes K*
- G believes K
- G believes *X believes K*

A detailed derivation of the proof is presented in [101].

7.4.3 Security discussion

A security protocol should be robust against malicious attacks. The authentication and key exchange protocol is immunized to the “man-in-the-middle” attack. An adversary can not modify the request or response because of the use of asymmetric cryptography. The “replay” attack will not work either since this protocol is invoked only once for each mobile host. Both X and G are capable of telling whether the request is brand new with respect to X.

The most severe threat to this protocol is that an attacker could use it to initiate denial-of-service (DoS) attacks against group agents. Because the mobile host does not know the shared key and can not encrypt the packet header at this time, an attacker can discover the identity of a group agent and locate its position by eavesdropping these requests and analyzing the packet headers. This threat may be avoided by encrypting the packet header of the request with the agent’s public key and the packet header of the response with the mobile host’s public key. An attacker could not distinguish the authentication protocol packets with other control or data packets. Furthermore, the movement of a group agent makes it complicated for an attacker to launch continuous DoS attacks.

7.5 Secure roaming support

A mobile network allows mobile hosts to roam within the network. In wired environments, Mobile IP is the most widely used protocol to support roaming. Mobile IP is not an ideal solution for HMWN, because (a) it establishes a “tunnel” between the home agent and foreign agent, which consumes wireless bandwidth; (b) it does not support “group roaming” (i.e., a whole group moves from one place to another). The essence of roaming support is relocating a mobile host. SMGR protocol naturally supports roaming as it dynamically locates the destination when forwarding a packet.

In case secure packet forwarding is required by the foreign group, the mobile host must authenticate itself to the foreign group agent and obtain the shared key before it can communicate with other hosts in the foreign group. This process is call secure roaming.

Mobile host:

```
1.  if homeless
2.     broadcasts a "join a group temporarily" request
3.     if a response from a FGA is received
4.        invokes the authentication process with that agent
5.        if authenticated
6.           changes the group ID and the shared key along with the CG and CGA
```

Group agent:

```
1.  if a "join temporarily" request is received
2.     if the security policy allows hosting
3.        sends a response to the mobile host
4.        invokes the authentication process
5.        if authentication succeeds
6.           issues a new shared key
7.           distributes the new key to the current group members
8.           sends the group information (gid, key) to the mobile host
```

Figure 7.3. Secure roaming support algorithm

1. X→FGA: $<X, FGA, HGA, R, T, S_X(X, FGA, HGA, R, T)>$
2. FGA→HGA: $<X, FGA, HGA, R, T, S_X(X, FGA, HGA, R, T)>$
3. HGA→FGA: $<S_{HGA}(X, K_X, R, T), S_{HGA}(FGA, K_{FGA}, R, T)>$
4. FGA→X: $<S_{HGA}(FGA, K_{FGA}, R, T), E_X(FGA, X, R, T, K, S_{FGA}(FGA, X, R, T, K))>$
5. X→FGA: $<X, FGA, T, E_K(X, FGA, T)>$

Figure 7.4. Mutual authentication protocol

7.5.1 Secure roaming support algorithm

The pseudo-code in Figure 7.3 shows the sketch of the secure roaming support algorithm. This algorithm is a part of the "Migration" operation. Its purpose is to verify

the identity of the mobile host and distribute the shared key safely. Other issues related to “Migration” are discussed in the previous chapter, including when to initiates the operation, how to choose a foreign group to join, how to update membership, and how to maintain routing table.

7.5.2 Mutual authentication between a mobile host and a FGA

Mutual authentication is required by secure roaming support algorithm to protect the foreign group as well as the mobile host. Figure 7.4 shows the mutual authentication protocol. Only messages exchanges are presented. The verifications at X, HGA, and FGA are omitted without losing the essence of the protocol. Through this protocol, X and FGA can get each other’s public key, which is signed by the HGA. FGA can verify that the request is initiated by X. The fourth step ensures that only X can get K. X must verify that K is generated by FGA using FGA’s public key. Because roaming support may be required by the same mobile host multiple times, a time stamp is associated with each request to prove its freshness. The use of time stamp may avoid the “replay” attack. It requires a loose synchronization among all mobile hosts.

The correctness of the mutual authentication protocol can be proven using the logic of authentication similarly to the proof presented in the previous section.

7.5.3 Fault-tolerant authentication

In a WNMBS, group agents are also moving. When the mutual authentication protocol is taking place, the HGA of X may be temporarily or permanently unavailable because of movement or failure. In this case, X’s request for the temporary membership in the foreign group will be denied. Mobile hosts will be detached from the system if their HGAs are no longer available. To make HMWN networks survivable from such kind of unavailability, a fault-tolerant authentication scheme is proposed in [102].

In a HMWN system. A group agent itself may be a member of another group and has its own HGA, unless it's the root of the hierarchy. We define mobile host X's Intention Agent (IA) as follows:

Mobile host Y is X's IA if and only if Y is the HGA of X's HGA or Y is the HGA of one of X's IAs.

For example, in Figure 6.1, agents A and B are IAs of mobile host x. In the proposed fault-tolerant scheme, not only its HGA, but also all its IAs know the public key of a mobile host. A mobile host also knows all its IAs' public keys. Each IA has a priority based on several factors [103]. When the mutual authentication protocol fails due to the unavailability of the HGA, the mobile host will choose the IA with the highest priority and retry the authentication process until it is authenticated or no IA is available. With this improvement, a mobile host at level n can tolerate n agent failures.

7.6 Computation overhead

The majority of computation overhead introduced by the security mechanisms comes from two sources: the secure packet forwarding and the secure roaming support. We numerically investigate the overhead by conducting a series of experiments and simulations.

The test-bed is a PC running Linux kernel 2.4.2. It has an Intel Celeron 700MHz CPU, 128M memory, and a 10G hard disk. Currently, even a low-end notebook computer has better configuration than the test-bed machine in terms of computation power.

The cryptography implementations used in the experimental study are provided by the GNU Crypto package. The testing programs are written in Java and compiled using JDK 1.3.1.

7.6.1 Overhead of secure packet forwarding

For any host that forwards a packet, it will decrypt and encrypt the packet header once using some symmetric cryptographic algorithm. We denote B as the bandwidth available

Table 7.1
Encryption/decryption speed of block ciphers

Cipher	Encryption Speed (KB/s)	Decryption Speed (KB/s)	CPU Usage
DES	4035	4061	3%
Triple-DES	1338	1323	9.8%
Twofish	1284	1277	10%
Rijndael	8185	8134	1.6%

to the mobile host, L_p as the average packet length, and L_h as the length of the packet header. We let S_e be the encryption speed and S_d be the decryption speed. The equation

$$\frac{L_h}{L_p}B/S_e + \frac{L_h}{L_p}B/S_d \tag{7.1}$$

estimates the maximum computation time required to encrypt and decrypt the data going through the host in one second.

We take the IEEE 802.11b standard as an example, which supports up to 11Mbps wireless bandwidth (i.e., B=11Mbps). Suppose only the IP header is encrypted (i.e., L_h = 20 bytes). Based on the study of IP packet length distribution [104], we let L_p = 420 bytes, the mean of IP packet length obtained from more than 200 million packets. The computation time can be derived as follows based on equation 7.1.

$$\begin{aligned} & \frac{L_h}{L_p}B/S_e + \frac{L_h}{L_p}B/S_d \\ = \; & \frac{20}{420} \times 11Mbps/S_e + \frac{20}{420} \times 11Mbps/S_d \\ \approx \; & 0.0655MBps/S_e + 0.0655MBps/S_d \end{aligned} \tag{7.2}$$

Four block ciphers are studied. They are DES (Data Encryption Standard), Triple-DES, Twofish (a 128-bit block cipher that accepts variable-length key up to 256 bits [105]), and Rijndael (Advanced Encryption Standard [106]). Table 7.1 shows the results obtained from processing 1,000,000 blocks. The encryption/decryption speeds (column 2 and 3 in Table 7.1) are obtained by using the GNU CipherSpeed tool. The CPU usage is computed based on equation 7.2.

The results demonstrate that secure packet forwarding is quite feasible in wireless networks as the appropriate cipher only uses about 1.6% of a mobile host's CPU time.

7.6.2 Overhead of secure roaming support

The computation overhead of the secure roaming support is introduced by the mutual authentication protocol. The time consumed by different cryptography operations using the RSA algorithm are shown in Table 7.2. They are obtained by operating 1,000 64-byte blocks with different keys whose length is 1024 bits. The computation time in one roaming request can be estimated as follows according to the mutual authentication protocol.

- *Mobile host:* one signing, one asymmetric decryption, two verifying, and one symmetric encryption (whose computation time can be ignored) operations are required. The computation time is about 90ms.
- *Foreign agent:* one verifying, one asymmetric encryption, and one signing operations are required. The computation time is about 50ms.
- *Home agent:* one verifying and two signing operations are required. The computation time is about 90ms.

Since roaming is caused by the relative motion between a mobile host and its group agent, for demonstration purpose, only hosts are moving in the simulations. Figure 7.5 shows the topology of a typical WNMBS. Mobile hosts move in a square area that is fully covered by 13 base stations. The movement is determined by the random way-point mobility [56] model. The pause time is 0 second. The maximum speed ranges from 2m/s,

Table 7.2
Speed of RSA

Operation	Signing	Verifying	Encryption	Decryption
Time (ms)	40.73	2.38	2.29	40.66

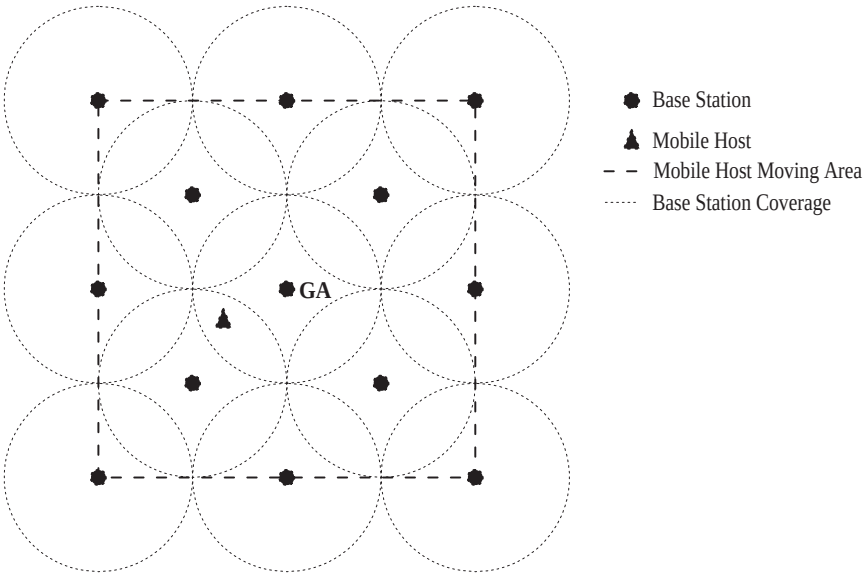
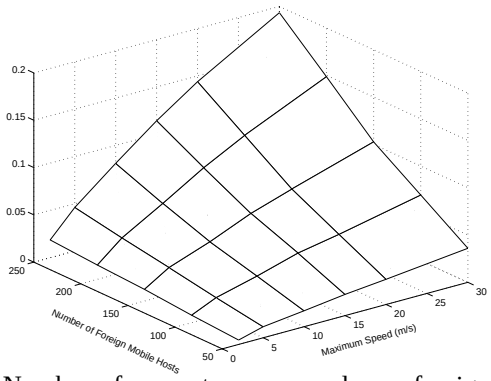


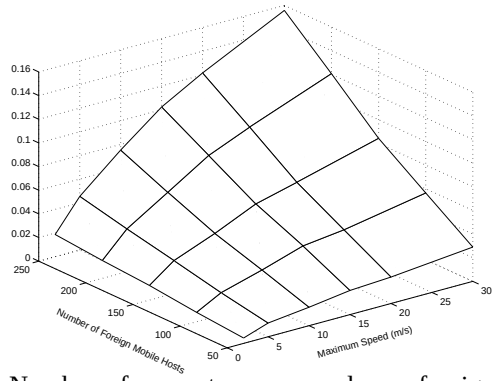
Figure 7.5. Topology of a WNMBS

the jogging speed of a person, to 30m/s, the speed of a running vehicle. The radius of every circle is 250m. Each simulation runs for 5000 seconds. For a mobile host, the mean interval between two consecutive requests is 416.38 and 56.49 seconds, respectively, when the maximum speed is 2m/s and 30 m/s.

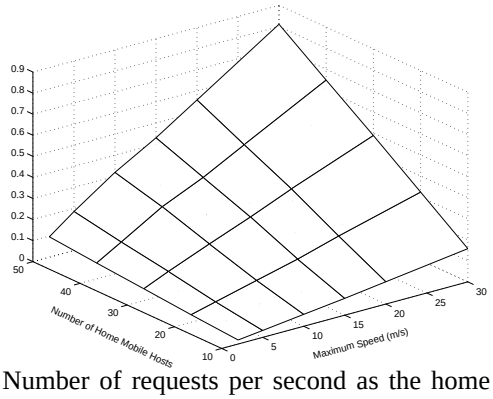
The rest experiments study the requests related to the group agent GA. Figure 7.6a shows the frequency of requests as a function of the number of foreign hosts in the area and their maximum speed, when GA acts as a foreign agent. For 50 foreign hosts, the



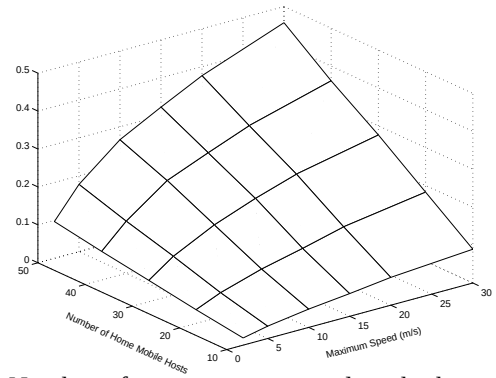
(a) Number of requests per second as a foreign agent (do not cache keys)



(b) Number of requests per second as a foreign agent (cache keys)



(c) Number of requests per second as the home agent (do not cache keys)



(d) Number of requests per second as the home agent (cache keys)

Figure 7.6. Frequency of roaming requests

number of requests per second increases from 0.005 to 0.04 with the maximum speed increasing from 2m/s to 30m/s. Even with 250 foreign hosts and 30m/s maximum speed, there are less than 0.2 requests per second. In this set of experiments, the computation overhead on GA of being a foreign agent is always less than 1% CPU time.

The overhead on GA of being the home agent is determined by the number of hosts whose home agent is GA and their mobility. Figure 7.6c shows the frequency of requests as a function of the number of home hosts in the area and the maximum speed. For 50 home hosts and 30m/s maximum speed, the frequency is as high as 0.8 requests per

second, because the home agent is involved in every roaming request. In this case, the computation overhead is about 7.2% CPU time.

The number of requests can be reduced if foreign agents cache the public key of a mobile host for a period of time. Figure 7.6d shows the results of the experiments in which foreign agents cache public keys for 200 seconds. The highest frequency is 0.45 requests per second, about a half of that in the previous experiment. The corresponding computation overhead is about 4% CPU time. The total computation overhead on GA ranges from 0.2% to 5% CPU time in the experimental study depending on the number of hosts and their mobility.

7.7 Conclusion

This chapter presents security mechanisms for HMWN to support wireless networks with movable base stations. The base stations (group agents) serve as a distributed trust entity. A secure packet forwarding algorithm is designed to protect the network infrastructure. A protocol is developed to authenticate a mobile host and distribute the group-shared key. An algorithm is designed to support mobile hosts roaming within the network. To secure both the foreign group and the mobile host, they mutually authenticate each other with the help from the home group agent. Experiments have been conducted on a low-end 700MHz PC. The results justify the feasibility of the proposed security mechanisms. The computation overhead of secure packet forwarding is less than 2% CPU time, and that of secure roaming support ranges from 0.2% to 5% CPU time depending on the number of hosts and their motion.

8 CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

8.1.1 Study of ad hoc routing protocols

Studying different approaches instead of individual protocols will be of great benefit to the design and improvement of ad hoc routing protocols. We choose AODV and DSDV as the representatives of on-demand and proactive approaches. Both protocols utilize distance vector coupled with destination sequence number, and choose routes in the same manner. They are differentiated by the way in which they operate (i.e., proactive versus on-demand). We investigate the performance of DSDV and AODV in terms of packet delivery ratio, average end-to-end delay, normalized protocol overhead, and normalized power consumption, under a wide range of network contexts with varied network size, mobility, and traffic load.

The major observations in this study include:

- Both proactive and on-demand approaches handle topology changes appropriately as the increase of mobility does not affect much the performance.
- The on-demand approach outperforms the proactive approach in less stressful situations (i.e., traffic load is light). The proactive approach is more scalable with respect to traffic load.
- The on-demand approach consumes less power, because it propagates the link break information faster, thus it avoids sending packets that are dropped eventually.

Although the published results [9, 25] showed that on-demand protocols outperform proactive protocols and are better suited for mobile ad hoc networks, the proactive protocols provide better support for quality of service (QoS) and anomaly detection. We iden-

tify that network congestion is the major reason for performance degradation. Congestion-aware distance vector (CADV) routing protocol is proposed to address the congestion issue.

In CADV, each routing entry is associated with an *expected delay*, which measures congestion at the next hop. Every host estimates the expected delay based on the mean of delay for all data packets sent in a past short period of time. When a host broadcasts an update to neighbors, it specifies the delay it may introduce. A routing decision is made based on the distance to the destination as well as the expected delay at the next hop. CADV tries to balance traffic and avoid congestion by giving priority to a route having low expected delay.

The preliminary study shows CADV outperforms AODV by about 5% in terms of packet delivery ratio with less protocol overhead.

8.1.2 Study of packet loss in ad hoc networks

Throughput is generally accepted as one of the most important metrics to evaluate the performance of a routing protocol. It is determined by how many packets have been sent and how many packets have lost. Studying when and why a packet is dropped will provide insights in the design of routing and flow control algorithms and the dimensioning of buffers. We concentrate on congestion-related and mobility-related packet loss.

- Congestion in a network occurs whenever the demands exceed the maximum capacity of a communication link, especially when multiple hosts try to access a shared media simultaneously.

- Mobility may cause packet loss in different ways. A packet may be dropped at the source if a route to the destination is not available, or the buffer that stores pending packets is full. It may also be dropped at an intermediate host if the link to the next hop has broken.

We study the percentages of packet loss due to congestion and mobility in various network contexts. AODV and DSDV are chosen as representatives of on-demand and proactive routing protocols respectively. We observe from the experiment results:

- Mobility is the dominant cause for packet loss in AODV, which is responsible for more than 60% of total packet loss. For DSDV, more than 50% of total packet loss is congestion-related.

- DSDV loses 10% to 20% more packets than AODV does for UDP traffic. For TCP traffic, the packet loss for DSDV is a half of that for AODV, because TCP greatly reduces congestion-related loss.

- Increasing traffic load has a strong impact on packet loss. Mobility decreases packet loss with light traffic load.

8.1.3 Congestion avoidance routing protocol for ad hoc networks

Congestion control is a problem in ad hoc networks. Compared to the traditional solutions at the transport layer, self-adjusting congestion avoidance (SAGA) routing protocol is implemented at the network layer. SAGA integrates the channel spatial reuse with the multi-hop routing to reduce congestion. SAGA is a distance vector routing protocol that uses intermediate delay (IMD) instead of hop count as the distance. The use of IMD enables routing protocols to select routes that bypass hot spots where contention is intense, thus enhance the routing performance. It is especially of benefit to networks where topology changes are much less frequent than traffic changes. The lazy route query operation in SAGA uses a special route advertisement for route discovery. Multiple queries can be included in one advertisement packet to accelerate the establishment of needed routes. The lazy route query can be applied to other proactive routing protocols that do not have a dedicated route discovery operation. An approach is provided in SAGA to reduce the oscillation of the value of IMD and makes the routes stable. SAGA protocol reduces congestion at every intermediate node. It can be used as a complementary scheme to the

end-to-end congestion control/avoidance mechanisms. The intermediate delay obtained from SAGA can be used to improve the accuracy of round-trip-time (RTT) estimation for TCP connections.

This research provides methods to estimate the delay at a node using only local information. When a node has recent traffic, statistical methods are used to evaluate the mean of the delay. Otherwise, the underlying MAC protocol is analyzed and probability methods are applied to compute the expectation of the delay. We analyze the packet transmission procedure of the distributed coordination function in the IEEE 802.11 standard as a case of the practical study. These methods are applicable to other contention-based media access protocols. They can be extended to provide quality of service (QoS) information to upper layer protocols and applications.

A series of experiments have been conducted to study the performance of routing protocols under congestion. Two types of UDP traffic as well as the TCP traffic are considered and the offered traffic load is taken as the input parameter. The maximum moving speed of nodes and the number of connections are varied. SAGA performs better than DSDV in all our measurements. A summary of comparison of SAGA with AODV and DSR for throughput, overhead, and end-to-end delay is as follows.

- SAGA is able to deliver around 90% of the data packets with an offered traffic load up to 300 kb/s. Its peak throughput is 1.5 to 3.5 times as compared to that of AODV and DSR.

- Overhead is measured as the ratio of the routing load to the data successfully delivered to the destination. The overhead of SAGA remains in a range of 15% to 50%. In similar cases, the overhead of AODV and DSR varies widely and increases fast as the offered traffic load goes high. The overhead of SAGA is as low as 10% of that of AODV and 12% of that of DSR in high traffic load.

- For low traffic load, the average end-to-end delay of SAGA is the same as that of AODV and DSR. When traffic reaches 500 kb/s, the delay of SAGA is 50% less than that of AODV and 80% less than that of DSR.

8.1.4 Wireless networks with movable base stations

The *hierarchical mobile wireless network* (HMWN) is proposed to support movable base stations in wireless networks. In a HMWN, mobile hosts are partitioned into groups. Each group can be viewed as an ad hoc network. It consists of some members and a group agent that may be a member of another group. The group agent is the representative of a group. The agent-member relationship forms a hierarchy. A group agent (i.e., a movable base station) acts as a gateway that connects these two groups. Mobile hosts belonging to the same group rely on multi-hop routing to communicate with each other. Communication with a host outside the group is accomplished by the proposed inter-group routing protocol.

Unlike in the fixed networks, where the location of a host is determined by its network address, in a mobile network, hosts can move to anywhere without changing the addresses. In HMWN, the location of a host is the group to which it belongs. The hierarchical membership management scheme serves two purposes: (a) verifying the identity of a mobile host for authentication, (b) locating a mobile host for routing protocols. Two kinds of memberships are maintained by group agents.

- Permanent membership. This is the registration information of a mobile host, such as public key, billing information, etc. It is established in the bootstrapping phase and determines if a host can join the system.

- Current membership. This is the location information for a mobile host. The management requires efficient update schemes to dynamically update it when a mobile hosts joins, leaves, or roam from one group to another.

As a HMWN is comprised of autonomous groups, the routing protocol must be capable of accommodating various intra-group routing protocols with least extra overhead. The approach is to localize the more frequently changing information while disseminating the less dynamic one. The proposed segmented membership-based group routing (SMGR) protocol has the following features:

- *Segmented:* Each group routes packets autonomously using its own protocol. When destination of a packet is outside the group, the packet is sent to the appropriate *exit host*.

- *Distributed membership-based locating:* The exit host of a packet in the group is identified by querying the membership of the destination.

- *Packet encapsulation:* The exit host encapsulates the packet to hide the differences among the routing protocols adopted by different groups. It is also used by security algorithms.

In SMGR, the topology change (the more dynamic information) is captured and propagated locally within a group by the routing protocol, while the membership change (the less dynamic information) is distributed to agents following the hierarchy by membership management. Simulation-based experiments demonstrate the scalability of SMGR in terms of protocol overhead.

8.1.5 Securing wireless networks with movable base stations

In a wireless system, the network infrastructure is protected by ensuring the routing information will not be forged, modified, or disclosed to an adversary. The packet header needs to be encrypted as it contains the source, destination, and next hop, which will expose the relationship among the involved hosts. To reduce encryption/decryption overhead and the impact of compromised hosts, we propose the security scheme that uses group shared symmetric key. The scheme consists of two parts:

- A protocol that authenticates a mobile host with its home agent to establish a shared key.

- An algorithm that cooperates with the routing algorithm to encrypt/decrypt packet headers. The content of control packets is protected using the same cryptographic technique. The content of data packets is protected by the upper applications themselves.

The computation overhead of different cryptographic algorithms is studied through experiments. The result justifies the feasibility of the proposed mechanisms.

In wired environments, Mobile IP is the most widely used protocol to support roaming. Mobile IP is not an ideal solution for HMWN, because (a) it establishes a “tunnel” between the home agent and foreign agent, which consumes wireless bandwidth; (2) it does not support ”group roaming” (i.e., a whole group moves from one place to another). The essence of roaming support is relocating a mobile host. SMGR protocol naturally supports roaming as it dynamically locates the destination when forwarding a packet.

Secure roaming support is mandatory for protecting a mobile system. A mutual authentication protocol is developed to authenticate a mobile host with a foreign agent and establish the shared key. The protocol provides protection to both the foreign group and the roaming host. The home agent of the host acts as the trust third party in this protocol.

The authentication protocol will fail if the home agent is not available because of movement or failure. The host will be detached from the system. To make HMWN survivable from such a single point of failure, the hierarchical fault-tolerant authentication scheme is applied. A host shares a secret with each of the agents on the path from itself to the root of the hierarchy. Any of those agents may be the trust third party in the authentication protocol. The scheme tolerates $L - 1$ agent failures for a mobile host, where L is the height of the hierarchy.

8.2 Future work

The research in this dissertation can be extended in a number of directions. The following summaries some of these directions.

8.2.1 Congestion control in ad hoc networks

Congestion in ad hoc networks can greatly degrade the performance. The set of TCP congestion control algorithms are based on the principle of conservation of packets. In ad hoc networks, the existence of multiple routes between two nodes provides an opportunity

for the routing protocols to select appropriate ones. A cross-layer design integrating the MAC, network, and transport layers will provide solutions to the congestion control problem. Based upon the research on the congestion avoidance routing protocol in this dissertation, the following research questions need investigation: How routing protocols can use the directional transmission provide by smart antennae to reduce channel interference and contention? How to utilize the multi-rate, multi-range, and multi-channel supports from IEEE 802.11 standard to minimize congestion? What is the tradeoff between connectivity and congestion avoidance? What are the advantages and disadvantages of different error-detection strategies, e.g. network detection and end-node detection, when they are used to infer congestion? This research will contribute to the development of adaptive protocol suites for ad hoc networks. It will be of benefit in advancing perversive computing and communication.

8.2.2 Trusted communication

Secure and trusted collaboration over worldwide computer networks will enable the formation of trusted global partnership in education, research, and with applications to business, military, security of the nation. Trusted communication is a necessity for trusted collaboration. Although security mechanisms can be applied to protect the communication between two participants, the collaboration is vulnerable to untrustworthy behaviors. In the collaborative network, every collaborator participates with others to deliver information. The safety of a communication solely depends on a proper choice of a sequence of collaborators to reach the destination. Sending information through a path that only involves trustworthy participants will decrease the probability of malicious attacks and information leakage. To investigate the problem of using trust to estimate the risk of selecting a collaborator, we need to (a) formalize trust for communication by building a model that quantifies the trustworthiness of a collaborator based on its behaviors, reliability, and security; (b) develop algorithms to assess the trustworthiness of a path based on information of collaborators; (c) design protocols that propagate trust information and

discover paths according to specific requirements; (d) experimentally study the integration of security mechanisms such as authentication, encryption/decryption, and filtering to defend against malicious attacks.

8.2.3 Privacy-preserved communication

The increasing amount of data sharing and collaboration calls for privacy-preserving mechanisms. Existing research efforts have studied the anonymous communication problem by hiding the identity of the subject in a group of participants. The proposed schemes ensure that the source of a communication is unknown, but everybody may know the content. Another approach for the privacy preservation problem is to remove the association between the content of the communication and the identity of the source. Somebody may know the source while others may know the content, but nobody knows both. The approaches will use trusted proxies to protect privacy in a dynamic communication environment. Research questions include: How to establish and maintain the trust relationships? How to measure the level of privacy that a specific approach can achieve? What are the tradeoffs for achieving a certain level privacy? What are the possible attacks and security threats?

LIST OF REFERENCES

LIST OF REFERENCES

[1] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proceedings of the ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM)*, (London, United Kingdom), pp. 234–244, August 1994.

[2] C. E. Perkins, E. M. Belding-Royer, and S. R. Das, "Ad hoc on-demand distance vector (AODV) routing," *RFC 3561, http://www.ietf.org/rfc/rfc3561.txt*, July 2003.

[3] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing* (T. Imielinski and H. Korth, eds.), ch. 5, pp. 153–181, Kluwer Academic Publishers, 1996.

[4] H.-Y. Hsieh and R. Sivakumar, "On using the ad-hoc network model in cellular packet data networks," in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc)*, (Lausanne, Switzerland), pp. 36–47, June 2002.

[5] H. Wu, C. Qiao, S. De, and O. Tonguz, "Integrated cellular and ad hoc relaying system: iCAR," *IEEE Journal on Selected Area in Communications*, vol. 19, no. 10, pp. 2105–2115, 2001.

[6] X. Wu, B. Mukherjee, and G.-H. Chan, "MACA: An efficient channel allocation scheme in cellular network," in *Proceedings of GlobeCom 2000*, vol. 3, (San Francisco), pp. 1385–1389, November 2000.

[7] "WaveLAN/PCMCIA card user's guide, Lucent Technologies," October 1996.

[8] C. E. Perkins, E. M. Royer, S. R. Das, and M. K. Marine, "Performance comparison of two on-demand routing protocols for ad hoc networks," *IEEE Personal Communications*, vol. 8, no. 1, pp. 16–28, 2001.

[9] J. Broch, D. A. Maltz, D. B. Johnson, Y. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of the 4th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, (Dallas), pp. 85–97, October 1998.

[10] Y. Lu, W. Wang, Y. Zhong, and B. Bhargava, "Performance study of two distance vector routing protocols for mobile ad hoc networks," Tech. Rep. CSD 02-016, Department of Computer Sciences, Purdue University, 2002.

[11] J. J. Garcia-Luna-Aceves and M. Spohn, "Source-tree routing in wireless networks," in *Proceedings of 7th International Conference on Network Protocols (ICNP)*, pp. 273–282, November 1999.

[12] J. J. Garcia-Luna-Aceves and M. Spohn, "Transmission-efficient routing in wireless networks using link-state information," *Mobile Networks and Applications*, vol. 6, no. 3, pp. 223–238, 2001.

[13] M. R. Pearlman, Z. J. Haas, and S. I. Mir, "Using routing zones to support route maintenance in ad hoc networks," in *Proceedings of Wireless Communications and Networking Conference (WCNC)*, September 2000.

[14] Z. J. Haas, M. R. Pearlman, and P. Samar, "The zone routing protocol (ZRP) for ad hoc networks," *IETF Internet Draft. http://www.ietf.org/internet-drafts/draft-ietf-manet-zone-zrp-02.txt*, July 2002.

[15] R. Sivakumar, P. Sinha, and V. Bharghavan, "CEDAR: A core-extraction distributed ad hoc routing algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 1454–1465, 1999.

[16] S. Corson and J. Macker, "Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations," *RFC 2501, http://www.ietf.org/rfc/rfc2501.txt*, January 1999.

[17] S. R. Das, R. C. neda, and J. Yan, "Simulation-based performance evaluation of routing protocols for mobile ad hoc networks," *Mobile Networks and Applications*, vol. 5, pp. 179–189, 2000.

[18] "The network simulator – ns-2," *http://www.isi.edu/nsnam/ns/*.

[19] H. Jiang and J. J. Garcia-Luna-Aceves, "Performance comparison of three routing protocols for ad hoc networks," in *Proceedings of IEEE ICCCN 2001*, (Phoenix), pp. 547–554, October 2001.

[20] X. Zeng, R. Bagrodia, and M. Geria, "GloMoSim: A library for parallel simulation of large-scale wireless networks," in *Proceedings of the 12th Workshop on Parallel and Distributed Simulations (PADS)*, (Banff, Alberta, Canada), May 1998.

[21] T. Camp, J. Boleng, B. Williams, L. Wilcox, and W. Navidi, "Performance comparison of two location based routing protocols for ad hoc networks," in *Proceedings of IEEE INFOCOM*, vol. 3, (New York), pp. 1678–1687, June 2002.

[22] R. V. Boppana and S. P. Konduru, "An adaptive distance vector routing algorithm for mobile, ad hoc networks," in *Proceedings of IEEE INFOCOM*, vol. 3, (Anchorage, Alaska), pp. 1753–1762, April 2001.

[23] N. R. Draper and H. Smith, *Applied Regression Analysis (Third Edition)*. John Wiley & Sons, Inc., 1998.

[24] R. Kravets and P. Krishnan, "Application-driven power management for mobile communication," *Wireless Networks*, vol. 6, no. 4, pp. 263–277, 2000.

[25] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, "Scenario-based performance analysis of routing protocols for mobile ad-hoc networks," in *Proceedings of the 5th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, (Seattle), pp. 195–206, August 1999.

[26] W. Wang, Y. Lu, and B. Bhargava, "On vulnerability and protection of ad hoc on-demand distance vector protocol," in *Proceedings of International Conference on Telecommunications (ICT)*, (Papeete, French Polynesia), pp. 375–382, February 2003.

[27] W. Wang, Y. Lu, and B. Bhargava, "On security study of two distance vector routing protocols for mobile ad hoc networks," in *Proceedings of IEEE International Conference on Pervasive Computing and Communications (PerCom)*, (Texas), pp. 179–186, March 2003.

[28] I. Cidon, A. Khamisy, and M. Sidi, "Analysis of packet loss process in high-speed networks," *IEEE Transactions on Information Theory*, vol. 39, pp. 98–108, January 1993.

[29] J.-C. Bolot, "End-to-end packet delay and loss behavior in the internet," in *Proceedings of Conference on Communications Architectures, Protocols and Applications (SIGCOMM)*, (San Francisco), pp. 289–298, September 1993.

[30] S. Biaz and N. H. Vaidya, "Distinguishing congestion losses from wireless transmission losses: A negative result," in *Proceedings of IEEE ICCCN*, (New Orleans), pp. 722–731, October 1998.

[31] F. Anjum and L. Tassiulas, "On the behavior of different TCP algorithms over a wireless channel with correlated packet losses," in *Proceedings of the International Conference on Measurement and Modeling of Computer Systems*, (Atlanta), pp. 155–165, 1999.

[32] T. V. Lakshman, U. Madhow, and B. Suter, "TCP/IP performance with random loss and bidirectional congestion," *IEEE/ACM Transactions on Networking (TON)*, vol. 8, no. 5, pp. 541–555, 2000.

[33] M. K. Marina and S. R. Das, "On-demand multipath distance vector routing in ad hoc networks," in *Proceedings of the International Conference on Network Procotols (ICNP)*, (Riverside), pp. 14–23, November 2001.

[34] T. Goff, N. B. Abu-Ghazaleh, D. S. Phatak, and R. Kahvecioglu, "Preemptive routing in ad hoc networks," in *Proceedings of the 7th International Conference on Mobile Computing and Networking (MobiCom)*, July 2001.

[35] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad-hoc wireless networks," in *Proceedings of IEEE INFOCOM*, vol. 3, (Anchorage), pp. 1360–1369, April 2001.

[36] V. Jacobson, "Congestion avoidance and control," in *Proceedings of Symposium on Communications Architectures and Protocols (SIGCOMM)*, (Stanford), pp. 314–329, August 1988.

[37] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "TCP Vegas: New techniques for congestion detection and avoidance," in *Proceedings of the Conference on Communications Architectures, Protocols and Applications (SIGCOMM)*, (London, United Kingdom), pp. 24–35, August 1994.

[38] J. Mo, R. La, V. Anantharam, and J. Walrand, "Analysis and comparison of TCP Reno and Vegas," in *Proceedings of INFOCOM*, pp. 1556–1563, March 1999.

[39] J. Padhye, V. Firoiu, D. F. Towsley, and J. F. Kurose, "Modeling TCP Reno performance: A simple model and its empirical validation," *IEEE/ACM Transactions on Networking (TON)*, vol. 8, no. 2, pp. 133–145, 2000.

[40] J. Liu and S. Singh, "ATCP: TCP for mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 7, pp. 1300–1315, 2001.

[41] C. Parsa and J. J. Garcia-Luna-Aceves, "Improving TCP performance over wireless networks at the link layer," *Mobile Networks and Applications*, vol. 5, no. 1, pp. 57–71, 2000.

[42] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, "A feedback-based scheme for improving TCP performance in ad hoc wireless networks," *IEEE Personal Communications*, vol. 8, no. 1, pp. 34–39, 2001.

[43] C. Zhang and V. Tsaoussidis, "TCP-Real: Improving real-time capabilities of TCP over heterogeneous networks," in *Proceedings of the 11th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, (Port Jefferson), pp. 189–198, June 2001.

[44] F. Wang and Y. Zhang, "Improving TCP performance over mobile ad-hoc networks with out-of-order detection and response," in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc)*, (Lausanne, Switzerland), pp. 217–225, June 2002.

[45] S. Cen, P. C. Cosman, and G. M. Voelker, "End-to-end differentiation of congestion and wireless losses," *IEEE/ACM Transactions on Networking (TON)*, vol. 11, no. 5, pp. 703–717, 2003.

[46] S. Kunniyur and R. Srikant, "End-to-end congestion control schemes: Utility functions, random losses and ECN marks," *IEEE/ACM Transactions on Networking (TON)*, vol. 11, no. 5, pp. 689–702, 2003.

[47] C.-K. Toh, "Associativity-based routing for ad-hoc mobile networks," *Wireless Personal Communications Journal*, vol. 4, no. 2, pp. 103–139, 1997.

[48] H. Hassanein and A. Zhou, "Routing with load balancing in wireless ad hoc networks," in *Proceedings of the 4th ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, (Rome, Italy), pp. 89–96, July 2001.

[49] S.-J. Lee and M. Gerla, "Dynamic load-aware routing in ad hoc networks," in *Proceedings of ICC*, vol. 10, pp. 3206–3210, June 2001.

[50] A. A. Bertossi and M. A. Bonuccelli, "Code assignment for hidden terminal interference avoidance in multihop packet radio networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 3, no. 4, pp. 441–449, 1995.

[51] Y. Lu, Y. Zhong, and B. Bhargava, "Packet loss in mobile ad hoc networks," Technical Report CSD 03-009, Department of Computer Sciences, Purdue University, 2003.

[52] "ANSI/IEEE Std 802.11, 1999 Edition," *http://standards.ieee.org/getieee802/download/802.11-1999.pdf*.

[53] H. Luo, P. Medvedev, and S. Lu, "A self-coordinating approach to distributed fair queueing in ad hoc wireless networks," in *Proceedings of IEEE INFOCOM*, vol. 3, (Anchorage), pp. 1370–1379, April 2001.

[54] E. Gelenbe and G. Pujolle, *Introduction to Queueing Networks, Second Edition*. John Wiley & Sons, 1998.

[55] W. Mendenhall and R. J. Beaver, *Introduction to Probability and Statistics*. Boston: PWS-Kent Pub. Co., 8th ed., 1991.

[56] Y. Lu, W. Wang, Y. Zhong, and B. Bhargava, "Study of distance vector routing protocols for mobile ad hoc networks," in *Proceedings of IEEE International Conference on Pervasive Computing and Communications (PerCom)*, (Texas), pp. 187–194, March 2003.

[57] J. Jaffe and F. Moss, "A responsive distributed routing algorithm for computer networks," *IEEE Transactions on Communications*, vol. 30, no. 7, pp. 1758–1762, 1982.

[58] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proceedings of the 9th International Conference on Mobile Computing and Networking (MobiCom)*, (San Diego), pp. 134–146, September 2003.

[59] Y. Lu and B. Bhargava, "Self-adjusting congestion avoidance routing protocol for ad hoc networks," Technical Report CSD 03-018, Department of Computer Sciences, Purdue University, 2003.

[60] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson, "Self-similarity through high-variability: Statistical analysis of ethernet LAN traffic at the source level," *IEEE/ACM Transactions on Networking (TON)*, vol. 5, no. 1, pp. 71–86, 1997.

[61] A. Tsirigos and Z. Haas, "Multipath routing in the presence of frequent topological changes," *IEEE Communications Magazine*, vol. 39, no. 11, pp. 132–138, 2001.

[62] S. K. Das, A. Mukherjee, S. Bandyopadhyay, D. Saha, and K. Paul, "An adaptive framework for QoS routing through multiple paths in ad hoc wireless networks," *Journal of Parallel and Distributed Computing*, vol. 63, no. 2, pp. 141–153, 2003.

[63] A. Boukerche, S. K. Das, and A. Fabbri, "Message traffic control capabilities of the R-DSDV protocol in mobile ad hoc networks," in *Proceedings of the 4th ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, (Rome, Italy), pp. 105–112, 2001.

[64] A. Boukerche and S. K. Das, "Congestion control performance of R-DSDV protocol in multihop wireless ad hoc networks," *Wireless Networks*, vol. 9, no. 3, pp. 261–270, 2003.

[65] D. S. J. De Couto, D. Aguayo, B. A. Chambers, and R. Morris, "Performance of multihop wireless networks: Shortest path is not enough," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 83–88, 2003.

[66] C. Cordeiro, S. R. Das, and D. P. Agrawal, "COPAS: Dynamic contention-balancing to enhance the performance of TCP over multi-hop wireless networks," in *Proceedings of the 10th International Conference on Computer Communication and Networks (IC3N)*, (Miami), pp. 382–387, October 2002.

[67] S. Xu and T. Saadawi, "Revealing the problems with 802.11 medium access control protocol in multi-hop wireless ad hoc networks," *Journal of Computer Networks*, vol. 38, no. 4, pp. 531–548, 2002.

[68] S.-T. Sheu, Y. Tsai, and J. Chen, "MR^{2}RP: The multi-rate and multi-range routing protocol for IEEE 802.11 ad hoc wireless networks," *Wireless Networks*, vol. 9, no. 2, pp. 165–177, 2003.

[69] S. Chen and K. Nahrstedt, "Distributed quality-of-service routing in ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1488–1505, 1999.

[70] C. Lin, "On-demand QoS routing in multihop mobile networks," in *Proceedings of IEEE INFOCOM*, vol. 3, (Anchorage), pp. 1735–1744, April 2001.

[71] C. Zhu and M. Corson, "QoS routing for mobile ad hoc networks," in *Proceedings of IEEE INFOCOM*, vol. 2, (New York), pp. 958–967, June 2002.

[72] Y. Zhang and W. Li, "An integrated environment for testing mobile ad-hoc networks," in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc)*, (Lausanne, Switzerland), pp. 104–111, June 2002.

[73] Y.-C. Tseng, Y.-F. Li, and Y.-C. Chang, "On route lifetime in multihop mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 366–376, 2003.

[74] "IEEE Std 802.11a-1999 (supplement to IEEE Std 802.11-1999)," *http://standards.ieee.org/getieee802/download/802.11a-1999.pdf*.

[75] C. Santiváñez, B. McDonald, I. Stavrakakis, and R. Ramanathan, "On the scalability of ad hoc routing protocols," in *Proceedings of IEEE INFOCOM*, vol. 3, (New York), pp. 1688–1697, June 2002.

[76] X. Hong, M. Gerla, G. Pei, and C. Chiang, "A group mobility model for ad hoc wireless networks," in *Proceedings of the 2nd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, (Seattle), pp. 53–60, August 1999.

[77] K. B. Egevang, C. Communications, and P. Francis, "The IP network address translator (nat)," *RFC 1631, http://www.ietf.org/rfc/rfc1631.txt*, May 1994.

[78] Y.-D. Lin and Y.-C. Hsu, "Multihop cellular: A new architecture for wireless communications," in *Proceedings of IEEE INFOCOM*, vol. 3, (Tel-Aviv, Israel), pp. 1273–1282, March 2000.

[79] H. Luo, R. Ramjee, P. Sinha, L. E. Li, and S. Lu, "UCAN: a unified cellular and ad-hoc network architecture," in *Proceedings of the 9th ACM International Conference on Mobile Computing and Networking (MobiCom)*, (San Diego), pp. 353–367, September 2003.

[80] S. Nesargi and R. Prakash, "Distributed wireless channel allocation in networks with mobile base stations," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 6, pp. 1407–1421, 2002.

[81] R. Ramanathan and S. Steenstrup, "Hierarchically-organized, multihop mobile wireless networks for quality-of-service support," *Mobile Networks and Applications*, vol. 3, no. 1, pp. 101–119, 1998.

[82] S. Banerjee and S. Khuller, "A clustering scheme for hierarchical control in multi-hop wireless networks," in *Proceedings of IEEE INFOCOM*, vol. 2, (Anchorage), pp. 1028–1037, April 2001.

[83] B. Liang and Z. J. Haas, "Virtual backbone generation and maintenance in ad hoc network mobility management," in *Proceedings of IEEE INFOCOM*, (Tel Aviv, Israel), March 2000.

[84] Y. Chang and C. Hsu, "Routing in wireless/mobile ad-hoc networks via dynamic group construction," *Mobile Networks and Applications*, vol. 5, no. 1, pp. 27–37, 2000.

[85] I. Rubin, A. Behzad, R. Zhang, H. Luo, and E. Caballero, "TBONE: A mobile-backbone protocol for ad hoc wireless networks," in *Proceedings of IEEE Aerospace Conference*, vol. 6, (Big Sky, MT), pp. 2727–2740, March 2002.

[86] "IETF MANET working group," *http://www.ietf.org/html.charters/manet-charter.html*.

[87] N. Borisov, I. Goldberg, and D. Wagner, "Intercepting mobile communications: The insecurity of 802.11," in *Proceedings of the 7th ACM International Conference on Mobile Computing and Networking (MobiCom)*, (Rome, Italy), pp. 180–189, July 2001.

[88] V. Bharghavan, "Secure wireless LANs," in *Proceedings of the 2nd ACM Conference on Computer and Communications Security (CCS)*, (Fairfax), pp. 10–17, 1994.

[89] S. H. Park, A. Ganz, and Z. Ganz, "Security protocol for IEEE 802.11 wireless local area network," *Mobile Networks and Applications*, vol. 3, no. 3, pp. 237–246, 1998.

[90] L. Zhou and Z. J. Haas, "Securing ad hoc networks," *IEEE Network Magazine*, vol. 13, no. 6, 1999.

[91] J. Binkley and W. Trost, "Authenticated ad hoc routing at the link layer for mobile systems," *Wireless Networks*, vol. 7, no. 2, pp. 139–145, 2001.

[92] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens, "An on-demand secure routing protocol resilient to byzantine failures," in *Proceedings of Workshop on Wireless Security (WiSe)*, (Atlanta), September 2002.

[93] M. G. Zapata and N. Asokan, "Securing ad hoc routing protocols," in *Proceedings of Workshop on Wireless Security (WiSe)*, (Atlanta), September 2002.

[94] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing robust and ubiquitous security support for mobile ad-hoc networks," in *Proceedings of 9th International Conference on Network Protocols (ICNP)*, (Riverside), pp. 251–260, November 2001.

[95] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," in *Proceedings of the 8th International Conference on Mobile Computing and Networking (MobiCom)*, (Atlanta), pp. 12–23, 2002.

[96] B. Schneier, *Applied Cryptography Second Edition : Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, Inc., 1996.

[97] J. Chuprun, C. Bergstrom, and A. Guzek, "Advanced interference rejection and anti-jam methods for low power mobile battlefield communications," in *Proceedings of IEEE Military Communications Conference (MILCOM 97)*, vol. 2, pp. 841–846, 1997.

[98] W. Myrick, J. Goldstein, and M. Zoltowski, "Low complexity anti-jam space-time processing for gps," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01)*, vol. 4, (Salt Lake City), pp. 2233–2236, May 2001.

[99] F. Buchholz, T. E. Daniels, B. Kuperman, and C. Shields, "Packet tracker final report," Tech. Rep. TR 2000-23, CERIAS, Purdue University, 2000.

[100] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Transactions on Computer Systems*, vol. 8, no. 1, 1990.

[101] Y. Lu, B. Bhargava, W. Wang, Y. Zhong, and X. Wu, "Secure wireless network with movable base stations," *IEICE Transactions on Communications, IEICE/IEEE Joint Special Issue on Assurance Systems and Networks*, vol. E86-B, pp. 2922–2930, October 2003.

[102] B. Bhargava, S. Kamisetty, and S. Madria, "Fault-tolerant authentication in mobile computing," in *Proceedings of International Conference on Internet Computing (IC)*, (Las Vegas), June 2000.

[103] D. McClure and B. Bhargava, "On assigning priorities of keying parameters in a secure mobile network," in *Proceedings of IEEE Workshop on "Reliable and Secure Application in Mobile Environment"*, (New Orleans), October 2001.

[104] "Packet length distributions," *http://www.caida.org/analysis/AIX/plen_hist/*.

[105] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, and C. Hall, "Twofish: A 128-bit block cipher," *http://www.counterpane.com/twofish.pdf*, June 1998.

[106] J. Daemen and V. Rijmen, *The Design of Rijndael*. Information Security and Cryptography SE, Springer-Verlag New York, Inc., 2002.

VITA

VITA

Yi Lu received the Doctor of Philosophy degree in computer science from Purdue University, West Lafayette in August 2004. He got the master's degree from Institute of Software, Chinese Academy of Sciences in 1999, and the bachelor's degree from University of Science and Technology of China in 1996, both in computer science. His research interests include wireless network security, heterogeneous wireless networks, routing and congestion control protocols for ad hoc networks, and trust modeling for peer-to-peer applications. He is a member of IEEE and ACM, and a member of IEEE Computer Society.