FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Yihua Zhai
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.A.Sc. (Electrical Engineering)
GRADE / DEGREE

School of Information Technology and Engineering
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Implementing MSR and BSR in a Wireless Ad Hoc Testbed
TITRE DE LA THÈSE / TITLE OF THESIS

O. Yang
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

M. Bolic

C-H. Lung

Gary W. Slater
LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES /
DEAN OF THE FACULTY OF GRADUATE AND POSTDOCORAL STUDIES

Implementing MSR and BSR in a Wireless Ad Hoc Testbed

by

Yihua Zhai

A thesis submitted to the School of Graduate Studies and Research
In fulfillment of the requirements for the degree of

Master of Applied Science in Electrical Engineering

School of Information Technology and Engineering
University of Ottawa

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-494-14977-9
Our file Notre référence
ISBN: 0-494-14977-9

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

Ad hoc networks are receiving more and more attentions these years due to their flexibility of fast network organization. The challenge for such networks is the design of efficient dynamic routing protocols adaptive to the frequent topology changes. The CCNR lab has invented the MSR (Multipath Source Routing) and BSR (Backup Source Routing) as improvement over the popular DSR. MSR can support multiple paths and to effectively reduce the congestion problems in networks while BSR (Backup Source Routing) can increase the communication reliability.

In this thesis, we have setup an ad-hoc network testbed consisting of IBM laptops equipped with wireless cards. We have coded and implemented the DSR, MSR and BSR algorithms in the Linux Operating System. Systematic tests and measurements of delay, throughput, and loss rate under various indoor and outdoor scenarios (as well as static and mobile) were carried out so that we can study the tradeoffs among these protocols. Based on our experiments, we have proposed, implemented and demonstrated an improved version of the MSR. We have also implemented these protocols in Qualnet, one of the latest simulation languages built for ad-hoc networking.

Acknowledgement

I would like to thank Prof. Oliver Yang for his guidance and supervision. I also thank all those who join the testing on the testbed, especially Nadia Ramirez who helped most. The suggestions and opinions from Zhi Liao and Prof Zhang of Tianjin University are much appreciated.

I would like to thank my parents and my younger brother for their support and cares on my study and living. I also thank my lab fellow, Song Guo, and his wife and sister for the great help in my two years' study and living in Ottawa. Finally, thanks to all other CCNR members for their support.

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant No. RGPIN-2003-42878.

Table of Content

List of Figures

List of Tables

Table of Symbols and Notations

		Section of 1st Appearance
d_k:	The delay of the kth route in MSR	2.2.2
$d_{\max}$:	The maximum delay of all the routes to the same destination in MSR	2.2.2
$D_{threshold}$:	The maximum ratio of current delay to the old one which will not cause delay updates in the improved MSR	2.3
$D(\pi,\pi^*)$:	The number of sub-disjoint paths of the two paths	2.4.1
$L(\pi,\pi^*)$:	The number of overlapping links in the two paths	2.4.1
R:	A factor to control the switching frequency among routes in MSR	2.2.2
T_f:	The Maximum delay feedback time in the improved MSR	2.3
U:	A bound to limit the weight not to be too high in MSR	2.2.2
W_k:	The weight of the kth route in MSR	2.2.2
π:	The path	2.4.1
$\lvert\pi^*\rvert$:	The length of the backup path	2.4.1

Table of Acronyms

		Section of 1st Appearance
ABR:	Associativity-based Routing	1.1
ACU:	Aironet Client Utility	3.1.1
AODV:	Ad-hoc On-demand Distance Vector Routing	1.1
APE:	Ad Hoc Protocol Evaluation Testbed	1.1.3
BSR:	Backup Source Routing	1.2
CBR:	Constant Bit Rate	7.2.1
CGSR:	Clusterhead Gateway Switch Routing	1.1
DSDV:	Destination-Sequenced Distance-Vector Routing	1.1
DSR:	Dynamic Source Routing	1.1
EWANT:	Emulated Wireless Ad Hoc Network Testbed	1.1.3
FTP:	File Transfer Protocol	1.4
GPS:	Global Position System	1.1.3
GUI:	Graphic User Interface	1.4
ICMP:	Internet Control Message Protocol	3.4.1
LMR:	Lightweight Mobile Routing	1.1
MSR:	Multi-path Source Routing	1.1.2
RIP:	Routing Information Protocol	1.1
RREP:	Route Reply	1.1.2
RREQ:	Route Request	1.1.2
SSR:	Signal Stability Routing	1.1
TCP:	Transmission Control Protocol	1.4
TORA:	Temporally Ordered Routing Algorithm	1.1
WRP:	The Wireless Routing Protocol	1.1

Chapter 1

Introduction

Without the need for infrastructure such as base stations or access points, mobile Ad Hoc networks can be flexibly deployed in many situations such as academic conferences or fieldwork where it is inconvenient to construct the infrastructure for temporary data or file sharing. In ad hoc networks, each node can function as a router to forward data to other nodes as well as a host.

1.1 Literature Review

The challenge for ad hoc networks is the design of efficient dynamic routing protocols adaptive to the frequent topology changes. Routing protocols used in conventional wired networks, such as link state and distance vector (e.g. RIP [Malk93]), are not good candidates for the mobile environment due to the considerable overhead produced by periodic route update messages and the slow convergence to topological changes. Many ad hoc routing protocols have been proposed recently and they can be divided into two categories:

1. On-demand or reactive in which routes are discovered and maintained when needed. Examples are DSR (Dynamic Source Routing) [JoMa96] and AODV (Ad-hoc On-demand Distance Vector Routing) [PeRo99].
2. Table-driven or proactive protocols in which nodes periodically exchange routing information in order to maintain the routing table for the entire network. Example is DSDV (Destination-Sequenced Distance-Vector Routing) [PeBh94].

Figure 1.1 lists the categorization of some common-used ad hoc routing protocols.

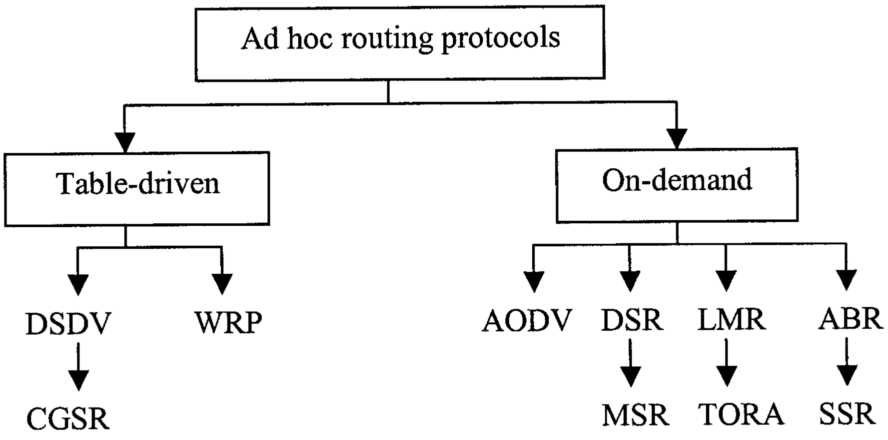


Figure 1.1: Categorization of ad hoc routing protocols

1.1.1 Table-driven Routing Protocols Review

Table-driven ad hoc routing protocols deploy traditional routing table update mechanism in a wired network to maintain up-to-date routing information. Each node periodically exchanges routing information with its neighbours so as to update its real-time routing table which stores the routing information to all the other possible destinations. Traditional Dijkstra's Algorithm [Dijk59] and Bellman-Ford Algorithm [FoFu62] are usually used in table-driven protocols.

The Destination-Sequenced Distance-Vector Routing protocol (DSDV) [PeBh94] is a table-driven algorithm based on Bellman-Ford Algorithm. Each entry in the routing table is marked with a sequence number assigned by the destination, which can distinguish old and expired routes from new ones and effectively avoid formation of loops. Routing table updates are transmitted throughout the entire network periodically in order to update and maintain the routing table. Each update contains the information of the destination, number of hops to that destination, sequence number received from the destination, and a new sequence number for the update. Only the route marked with the most recent sequence number is used for updating routing table. In order to decrease the large amount of traffic caused by frequent update, route updates can employ the incremental update strategy in which only the information that has changed since last update is transmitted.

Table-driven routing protocols try to maintain a routing table containing all the routing information to all the other nodes, regardless of whether or not it is needed. This feature causes inefficient utilization of network resources which are especially limited in wireless networks.

1.1.2 On-demand Routing Protocols Review

On-demand routing protocols, unlike table-driven routing protocols, do not need to maintain the routing table containing all the route information to the other nodes. They work in a different on-demand manner. When a node requires a route to a destination, it initiates the route discovery process to find routes to the destination. The route information is maintained by the route maintenance procedure.

Ad-hoc On-demand Distance Vector Routing (AODV) [PeRo99] is improved from DSDV and operates in an on-demand manner such that routes are discovered in case that the source needs to send data to some destination and does not have a route to the destination in its routing table. In route discovery process, a route request (RREQ) packet is broadcast by the source and flooded throughout the network. Each intermediate node which receives the RREQ records in its route table the address of the neighbour from which the first RREQ is received. Additional packets of the same RREQ are discarded in order to avoid path loop. Only the destination or those nodes which have route to the destination in their route tables can reply to the RREQ by sending back a Route REPly (RREP) to the source along the reverse path. Each node in the reverse path uses the route information recorded in RREP to add an entry in its route table. Thus the route is established between the source and destination. In a route maintenance procedure, a link failure notification message is transmitted to the source along the upstream direction of the route. The source will reinitiate route discovery for that destination if it also has data to send.

As discussed and compared in [RoTo99], on-demand routing protocols have better performance than table-driven ones due to the low amount of routing overhead in on-demand route discovery. DSR and AODV are the two most popular on-demand routing protocols and both have their own advantages in specific situations [PeRo01]. DSR uses source routing to add route information in the header of the packet so that the intermediate nodes do not need to maintain the routing table and just forward the packet according to the route information contained in the packet header. For delay and throughput performances, DSR outperforms AODV in less stressful situations such as light traffic load and low mobility. In more stressful situations such as more load and high mobility, DSR performs worse than AODV. DSR is easier to extend due to the advantage of source routing including simplicity, correctness, and flexibility [HuJo01].

MSR (Multipath Source Routing) [WaZh00] is extended from DSR to distribute the load into the multiple paths collected in the route discovery. While DSR is a single-path routing which only uses one path to forward data and causes congestion in that path, the load balancing of MSR can alleviate the congestion problem in a single path.

1.1.3 Ad Hoc Testbed Review

There are different testbeds developed for the study of different protocols such as Piconet II [Song01], APE [Nord02], EWANT [SaBr03] and Monarch DSR implementation [MaBr99].

Monarch DSR testbed [MaBr99] is designed by Carnegie Mellon University for the test of DSR. The testbed is composed of 5 mobile nodes and 2 static nodes spread over an area of 700m by 300m in a public park. The mobile nodes are represented by laptops mounted in cars with antennas on the top. A visualization tool for monitoring each link in the network is developed for the management of the nodes. GPS (Global Position System) information is used in each node to aid the position visualization. The purpose of the testbed is to study the behaviour of DSR protocol in a real testbed implementation.

Ad Hoc Protocol Evaluation Testbed (APE) [Nord02] is designed for comparative study of different routing protocols. APE deploys large-scale tests with the help of student volunteers moving with laptops and tests can be reproduced with high accuracy. A graphical animation tool, APE-view, is developed to visualize node positions and movements. APE testbed also includes a set of tools for analyzing the gathered data.

The Emulated Wireless Ad Hoc Network Testbed (EWANT) [SaBr03] is designed to provide a low-cost research environment for wireless networks. Four external antennas are connected to one PC card through demultiplexer and the transmission is switched through four antennas in order to emulate mobility.

1.2 Motivations

The main cause of large delays in Internet is the congestion problems at the links and routers, and the same is true in wireless ad hoc networks. Table-driven protocols are not suitable for ad hoc networks due to the large amount of overheads from periodic route update messages and their slow convergence to topological changes. Furthermore, most routing protocols in ad hoc networks today use single-path routing algorithms, which not only under-utilize network resources but also cannot cope with congestion and link breakage, because focusing all traffic on a single route can result in congestion and furthermore link breakage on that route. Multipath routing can overcome these shortcomings. MSR [WaZh00] is one such protocol our CCNR (Computer Communications Network Research) lab has extended based on DSR to support multiple paths and to effectively reduce the congestion problems in networks. However, it has never been implemented, and we would be interested in its real measured performance.

Another important issue relative to network performance is the transmission reliability, especially in mobile networks. The disconnection of one link can result in the failure of that route, and cause retransmission of the dropped packets through another route or route discovery if there exist no other routes in the worst case. In this way, the transmission will break until the route is re-established. Since the link disconnection occurs frequently in the network with high mobility, we need to find a way to establish a more reliable and durable connection between source and destination. As another protocol invented in our CCNR lab, BSR (Backup Source Routing) [GuYa02] uses the concept of similar paths to establish and maintain backup paths in addition to the primary path. Therefore, it would be of interest to have it implemented, and to investigate how a more durable communication is established between source and destination, while reducing the overhead induced in route reestablishment.

Although many works have been done to exploit the utilization of network resources and performance of routing protocols in Ad Hoc networks, most of them use computer simulations, which is insufficient to test and evaluate the performances of a protocol. Full-scale testbed experiments are needed. Although there have been different testbeds for different protocols as discudded in last section, none of them consider multipath routing and backup routing. Therefore, we would like to build a testbed to test our MSR and BSR protocols and measure their performances.

Finally, it should be fair to say that simulation still plays an important role in testing out a network that is too large to build. In this case, an accurate simulation model for different protocols is essential for a successful simulation.

1.3 Objectives

In this thesis, our objectives are to perform a thorough study of MSR and BSR in our testbed. Specifically, we would like to achieve the following objectives:

1. Implement MSR and BSR in our testbed;
2. Performance evaluation of MSR and BSR;
3. Conduct simulation on MSR, the improved MSR and BSR;
4. Provide simulation results as a reference for analysis.

1.4 Approaches and Methodology

To achieve the above goals, we need to implement our routing protocols as a part of operating system and inside the system kernel. Since Linux operating system is a totally open-source system and supports TCP/IP and many network applications, we choose Linux as our development platform.

There are two ways to add user-defined functions to a Linux kernel: building the function into the kernel, and implementing the function as a kernel module. If the function is built into the kernel, we need to reboot the machine every time we modify the kernel. A kernel module, on the other hand, is more flexible and can be inserted into and removed from the kernel even when the kernel is running. We finally choose to use kernel module to implement our protocols since the flexibility feature of kernel module suits for development.

We setup an ad-hoc network consisting of 8 laptops each with a wireless card. We can choose any type of high-speed and thin laptop for testing. We shall conduct our experiments for each specific-chosen topology in indoor and outdoor environments since there are different interferences between indoor and outdoor environments.

Practical testing of routing protocols is more time-consuming than simulation, but can get more useful results. In real-world tests, we evaluate our routing protocols on the performances of delay and throughput in the applications of Ping and FTP. We first evaluate the performances of MSR in terms of delay, throughput and packet loss rate, and compare them to the DSR protocol. From our study, we notice that the existing MSR need to be improved. We have therefore modified it, and then we evaluated and compared the performance with the old MSR. Finally we evaluate and compare the performances of BSR with DSR. For each performance point, typically 3 measurements are taken from which the mean and variance are obtained.

Besides real-world testbed experiments, we conduct a deeper study on our routing protocols using simulation. Since we perform our test in a real-world testbed, the results can reflect the performance of practical application, and if we get the satisfied results we can ensure that the protocols can be used in practice. The simulation, on the other hand, can assist us to analyze the results such as understanding the factors which influence the performance. We choose Qualnet as our simulation tool since Qualnet has a set of tools with all the components for custom wireless network modeling and simulation projects. Qualnet's GUI (Graphic User Interface) makes it easy to design scenarios and protocols, and the animation capability makes the simulation visual.

1.5 Contributions

The contributions of this thesis are:

1. Setting up a wireless ad hoc testbed for the implementation and testing of DSR, MSR, and BSR;
2. Improving the probing mechanism of MSR to be adaptive;
3. Evaluating the performance based on the testing results, and carrying out comparison among the protocols;
4. Developing MSR, the improved MSR, and BSR based on DSR in Qualnet;
5. Providing a simulation reference model for these protocols.

1.6 Thesis Organization

The rest of this thesis is organized as follows. Chapter 2 describes the protocols implemented in our testbed. Chapter 3 describes the testbed setup including hardware setup and code development details. The performance evaluation of MSR, the improved MSR, and BSR is given in Chapter 4, 5, and 6 respectively and the Qualnet simulation results are provided in Chapter 7. Finally in Chapter 8, we conclude this thesis, introduce some experience from the testbed experiment and list the future work.

1.7 Publications

1. Yihua Zhai, Yang, O.W.W., Wenlan Wang, Yantai Shu, "Implementing multipath source routing in a wireless ad hoc network testbed," *Communications, Computers and signal Processing, 2005. PACRIM. 2005 IEEE Pacific Rim Conference* on 24-26 Aug., 2005 Page(s):292 – 295

Chapter 2
Routing Protocols

In this chapter, we shall describe the routing protocols to be implemented in this thesis. We first introduce DSR to summarize the terminology and concept before we detail the MSR in the following sections, and finally the BSR.

2.1 DSR (Dynamic Source Routing)

DSR [JoMa96] employs source routing instead of hop-by-hop packet routing. Each data packet carries the complete path from source to destination as a sequence of IP address. The major advantage of DSR is from source routing where intermediate nodes are not required to keep route information because the path is explicitly specified. This allows loop-free packet routing and its on-demand feature eliminates the need for periodic routing information update and neighbour detection in the intermediate nodes through which packets are forwarded. DSR operates entirely on-demand, and routing packet overhead is minimized to react to changes in the current routes. The DSR protocol consists of two phases: Route Discovery and Route Maintenance.

2.1.1 Route Discovery

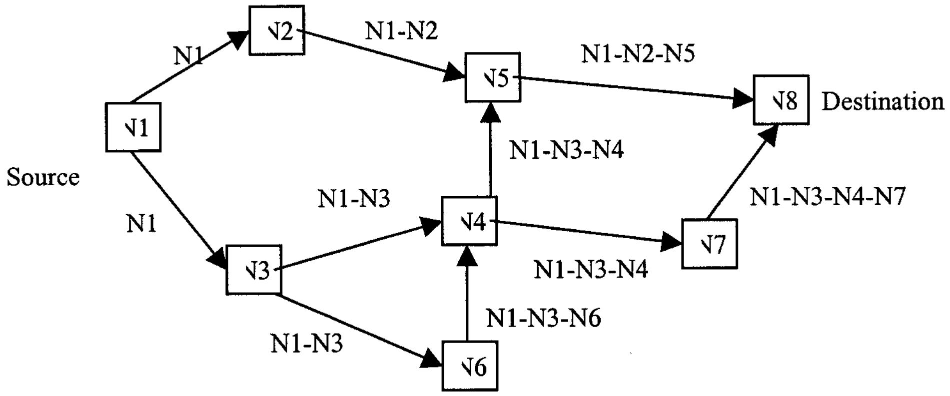


Figure 2.1: Route discovery process

Route discovery is initiated by a source whenever a source has a data packet to send but does not have any route information to the destination. To establish a route, the source floods the network with a Route REQuest (RREQ) message carrying a unique request ID. Each intermediate node, which does not have route information to the destination and is not included in the path list, will append its IP address in the path list of RREQ and rebroadcast it. Loop formation is easily avoided in this forwarding mechanism. In order to find the shortest path and reduce the route redundancy, each node only forwards the RREQ once since the second RREQ cannot form the shortest path. When the request message reaches the destination or a node that has route information to the destination, the node sends a route reply (RREP) message containing path information back to the source. In order to reduce overhead, the "route cache" at each node would record routes that a node has learned and overheard during this route discovery phase. Figure 2.1 shows one specific route discovery process. N4 forwards the RREQ from N3 and discards the one from N6 with one more hop. Finally N8 will get two routes: N1-N2-N5-N8 and N1-N3-N4-N7-N8. N8 will send back RREP via the inverse paths.

2.1.2 Route Maintenance

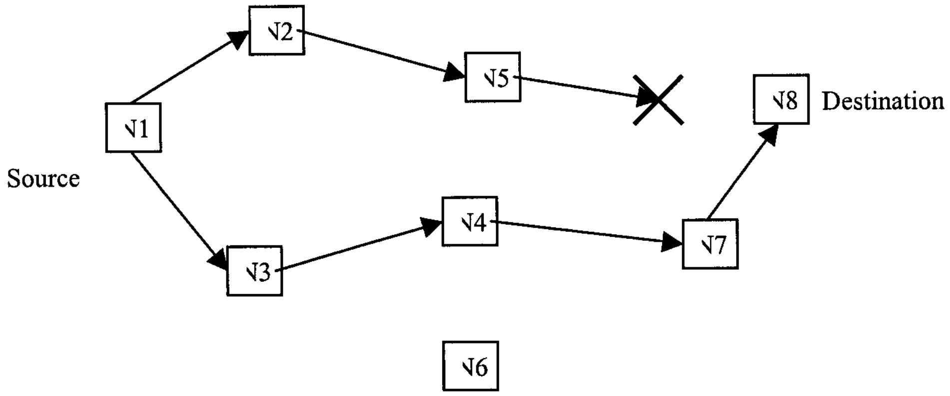


Figure 2.2: Route maintenance process

Route Maintenance is the mechanism by which a sender of a packet detects network topology changes that render useless its route to the destination if two nodes in the route have moved out of range of each other. When Route Maintenance indicates a source route is broken, source node is notified by a Route Error packet. The sender can then attempt to use any other route to destination that is already in its cache or can invoke Route Discovery again to find a new route. These key factors ensure excellent performance in multi-hop wireless ad hoc networks. Figure 2.2 shows one specific route maintenance process. The link between N5 and N8 is broken, so N5 sends a route error packet back to the source N1 to notice the link failure. After receiving the route error message, N1 will search its route cache and find another route N1-N3-N4-N7-N8. Then N1 will put its traffic on this effective route.

2.2 MSR (Multipath Source Routing)

Although DSR can find multiple paths in the Route Discovery, it only uses the shortest path as routing criterion, so it is essentially a single path protocol. To overcome the disadvantage of single path routing on underutilizing resources and leading to congestion, MSR [WaSh01] sends packets over multiple paths collected in the Route Discovery phase.

2.2.1 Path finding

MSR deploys the same route discovery mechanism as DSR. Each route found in route discovery is stored in route cache with a unique route ID, so it is easy for us to pick multiple paths from route cache. Unlike DSR, the disjoint paths are preferred in MSR in route selection because a more independent path can provide more resources between two nodes. It is not always possible to find independent paths. Fortunately in some cases, independent paths can be found effectively in route discovery by the RREQ forwarding mechanism that each intermediate node only forwards the first received RREQ while discarding all the others received later. Based on this forwarding mechanism, those paths that overlap on several hops can be greatly reduced.

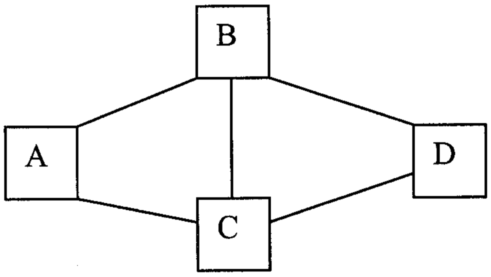


Figure 2.3: A simple example of independent path finding

Figure 2.3 shows a simple independent path finding example on the given topology. Node A initiates a route discovery to destination D. If the RREQ is forwarded by the intermediate nodes every time, four routes A-B-D, A-C-D, A-B-C-D, and A-C-B-D will be found. The routes with overlapping links are not good candidates in MSR because all the traffic will go through the overlapping link so that MSR cannot alleviate congestion in that link and the advantages of multipath routing do not exist any more. By the mechanism that each RREQ is forwarded only once by the intermediate nodes, the routes A-B-C-D and A-C-B-D will not be created since B and C only forward the first RREQ received from A. In this manner, the two independent routes A-B-D and A-C-D are finally found.

2.2.2 Load balancing

Since MSR uses source routing, intermediate nodes do nothing but forward the packet by using the full route information in its header. Thus MSR can improve the performance without adding much more processing complexity and only the source node needs to determine which route should be used. That is why source routing is easy to be extended.

In MSR, source node is responsible for load balancing. Delay is used as the metric to distribute packets over multiple routes and a periodic probing mechanism is deployed to obtain the dynamic delay information for each path in use. One proposal [ZhZh02] of an optimal weight equation for distributing loads is given as the following:

$$W_k = \min\left(\left\lceil \frac{d_{\max}}{d_k} \right\rceil, U\right) \times R, k = 1,2,\ldots,N$$

where W_k is the weight of the kth route, $d_{\max}$ is the maximum delay of all the routes to the same destination, d_k is the delay of the kth route, U is a bound to limit the weight not to be too high, and R is a factor to control the switching frequency among routes. The weighted-round-robin scheduling strategy is used in distributing the loads. For example, in the network shown in Figure 2.3, two paths exist between A and D: 1) A-B-D and 2) A-C-D. They have the same length so that they have the same weight: $W_1 = W_2 = 1$ (we choose R =1). A will send packets to D via the two paths alternatively such that A sends packet 1 through path 1, packet 2 through path 2, packet 3 through path 1, and so on.

By load balancing, MSR not only improves the network utility but also balances the energy consumption of each node so as to increase the network lifetime.

2.2.3 Probing Scheme

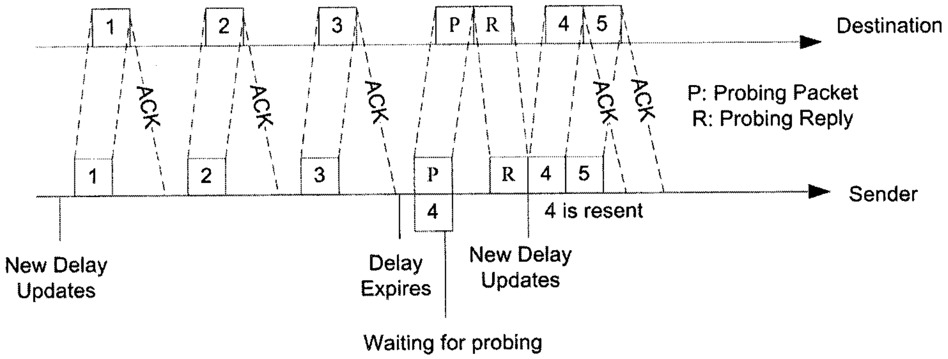


Figure 2.4: Probing Packet Operation Procedure

In order to update the delay information, MSR deploys a periodic probing scheme. A probing period is defined as the expiration time stored in a timer after which a probing packet must be sent along each path to obtain new delay to the destination.

Figure 2.4 is an example on how the probing packet is used. In this example, the sender sends 1 packet per second to the receiver, and the probing period is 3s. As seen in the diagram, after the sender sends 3 packets, the route delay timer expires. Therefore, when Packet 4 comes in, it must send a probing packet, and wait until the probing reply packet is received to update the route delay information. Thus packet 4 has a longer delay than others which do not need to wait.

2.3 The Improved MSR

Periodic probing for path delay is the best way to obtain real-time status of network and suits for the scenarios where nodes move quickly and topology changes fast, but it has the disadvantage that it cannot be adaptive to the variety of situations such as low movement scenarios where the delay varies slightly so that periodic probing obtains the same path delay every time.

Based on our measurement experience, we propose an improved scheme for adaptively probing the path delay. Upon receiving packets, the destination node can be aware of the real-time delay as long as we add the "sent-time" (i.e. the time instant the last bit of a packet is sent out) as one more parameter in the source packet header. In this way, the destination node can monitor the path delay change. Only if the delay changes greatly that we need to update the route delay information via probing packets. This can be achieved easily as long as the destination node sends back a notification packet of the new delay value to the source only if the path delay change exceeds a threshold $D_{threshold}$. This adaptive probing mechanism requires the source and destination nodes to perform a little bit more processing, but this additional complexity is negligible when compared to the total processing tasks in the destination.

In some cases where network topology changes dramatically, our adaptive probing will cause more delay notification packets than the "probing reply packets" of regular periodic probing. To overcome this shortcoming, we further modify this scheme a little bit by adding one more constraint on the frequency of sending new delay notification packet to the source node by the destination. That is, in a short period T_f after sending delay update information, the destination node would do nothing no matter how much and often the delay changes. We can choose this time period to be the probing period in normal MSR to ensure that the probing packets of the improved MSR are sent no more frequently than those of the old MSR so that the performance of the improved MSR is not much worse than the normal one.

2.4 BSR (Backup Source Routing)

BSR [GuYa02] is also an extension of DSR. BSR chooses the shortest delay path as the primary path and a most durable backup route. Like DSR, BSR contains two phases: route discovery and route maintenance.

2.4.1 Route Discovery

In route discovery, the forwarding mechanism used in DSR should be modified in BSR in order to establish the most durable backup routes because in the route discovery phase of DSR, the RREQ message by the same source is only forwarded once by each intermediate node. Instead of dropping all the duplicate RREQ messages, intermediate nodes forward RREQ if all the following requirements are met: (1) the node is not the target of RREQ packet; (2) the node is not listed in the source route; (3) the path in the duplicate packets can produce new backup routes with lower value of the heuristic cost function; (4) a candidate of backup routes cannot be obtained from the information in the cache.

After receiving the first RREQ, the destination node will choose the path π contained in the RREQ as the primary path since it is the shortest-delay path. The destination node will select the path π^* with the minimal value of the heuristic cost function $C(\pi,\pi^*) = L(\pi,\pi^*) + |\pi^*| / D(\pi,\pi^*)$ [GuYa02] where $L(\pi,\pi^*)$ is the number of overlapping links in the two paths, $D(\pi,\pi^*)$ is the number of sub-disjoint paths of the two paths, and $|\pi^*|$ is the length of the backup path. In the case that there is more than one path with the minimal cost, we select the path that has the RREQ to the destination in the earliest time.

2.4.2 Route Maintenance

In route maintenance, when a link breaks, the node detecting it will send back a ROUTE ERROR (RERR) packet along the upstream direction to the source. Upon receiving the RERR, the source will remove all the paths containing the break link and reconstruct the backup routes using the remaining information in its route cache. If there is not enough information to construct backup routes, the source will initiate a new route discovery process.

The packets which are not delivered will use the backup route in the packet header. This can effectively decrease packet drop and hence decrease the number of route discovery.

2.5 Packet Processing Procedure

In this section, we describe the detailed operation procedure for each protocol. There are two parts in each of the DSR, MSR, and BSR packet processing procedure: receiving and transmitting. We shall also introduce the operation procedure of the improved MSR after.

2.5.1 Receiving Procedure

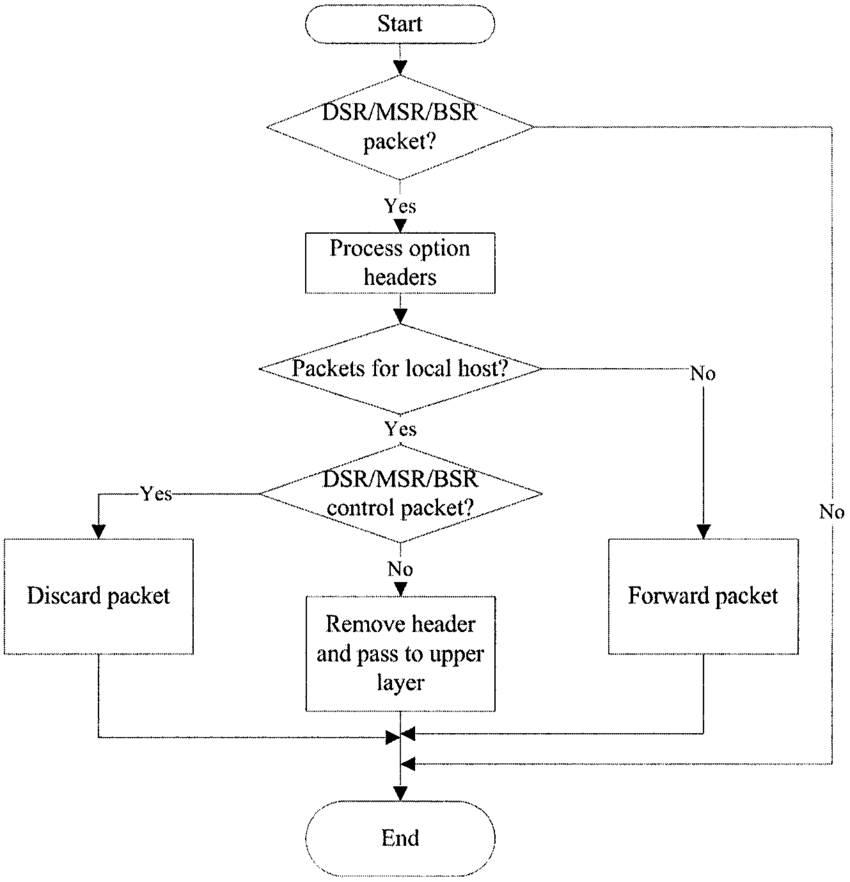


Figure 2.5: Flow Chart for Receiving Packets

All DSR, MSR, and BSR protocols we implemented have the same receiving processing procedure as shown in Figure 2.5. The only difference lies in processing the option headers since different protocol has different way in dealing with the control packet headers. After the option header is processed, the packet header will be removed. The application data packets will be passed to the upper layer and the control packets will be discarded.

Here we briefly list the processing procedure of option headers. We first list the processing procedure for the common options in our protocols.

- Route Request Option: If the node is the destination or has the route to the destination, it sends back a Route Reply packet to the source (to shorten the route search time); else the node appends its IP address in the path and floods it again.
- Route Reply Option: If the node is the destination, it inserts the path information into the Route Cache and retransmits all the packets in the send buffer using the new obtained route; else the node forwards the packet to the next hop using the source route information in the header.
- Source Route Option: If the node is the destination, it does nothing since the packet has already reached the destination; else the node forwards the packet to the next hop using the source route information in the header. (In BSR, we add a field in the header to illustrate which route in the primary path and the backup path is currently used. The next hop is determined from the currently used path.)
- Route Error Option: If the node is the destination, it removes from the route cache all the routes which have the break link; else the node forwards the packet.
- Acknowledgement Option: The node deletes the packet in the retransmission buffer according to the packet ID.

MSR needs to process two more options:

- Route Probing Option: If the node is the destination, it sends back a Probing Reply packet to the source; else the node forwards the packet.
- Probing Reply Option: If the node is the destination, it updates the route information in the probing table, recalculates the weight function and retransmit all the packets in the probing buffer; else the node forwards the packet.

2.5.2 Transmitting Procedure

The transmitting procedure is more complex than the receiving procedure. The main operation in the transmitting procedure is to add source route option to the data packet.

In DSR, when the node has data packet to send, it will search the route cache. If it cannot find any route to the destination, the node will initiate a route discovery and insert the packet into the send buffer temporarily. If there exist routes to the destination, the node will select the shortest path and insert the source route option into the packet header. Before being sent out, the packet will be inserted into the retransmission buffer waiting for the acknowledgement from next hop. Figure 2.6 shows the detailed operation process.

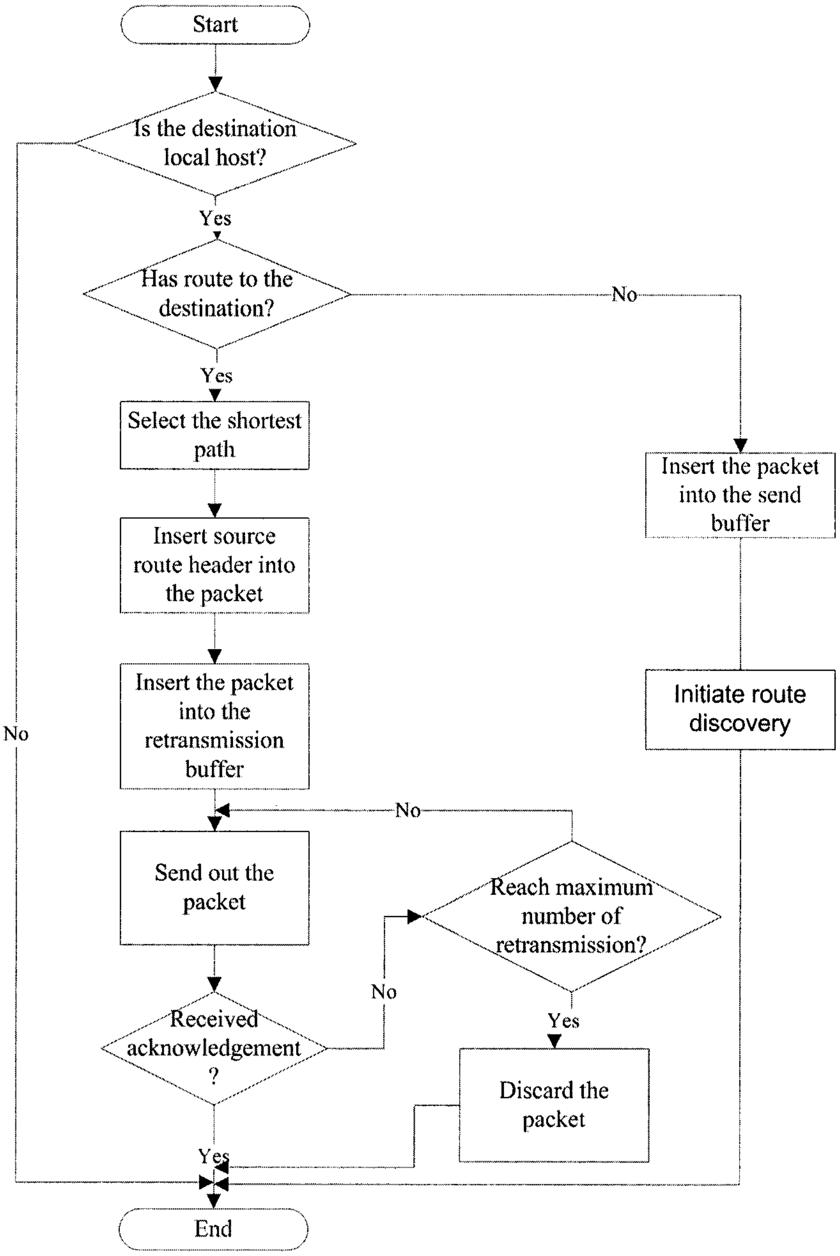


Figure 2.6: Flow Chart for Transmitting Packets in DSR

BSR has almost the same transmitting procedure as DSR except that BSR selects two paths from the route cache: the shortest path as the primary path and the most durable path as the backup path. The most durable path is selected base on the heuristic cost function $C(\pi,\pi^*) = L(\pi,\pi^*) + |\pi^*| / D(\pi,\pi^*)$. This has been discussed in Section 2.4.1. If there only exists one path to the destination, BSR will perform the same way as DSR as there is no backup path.

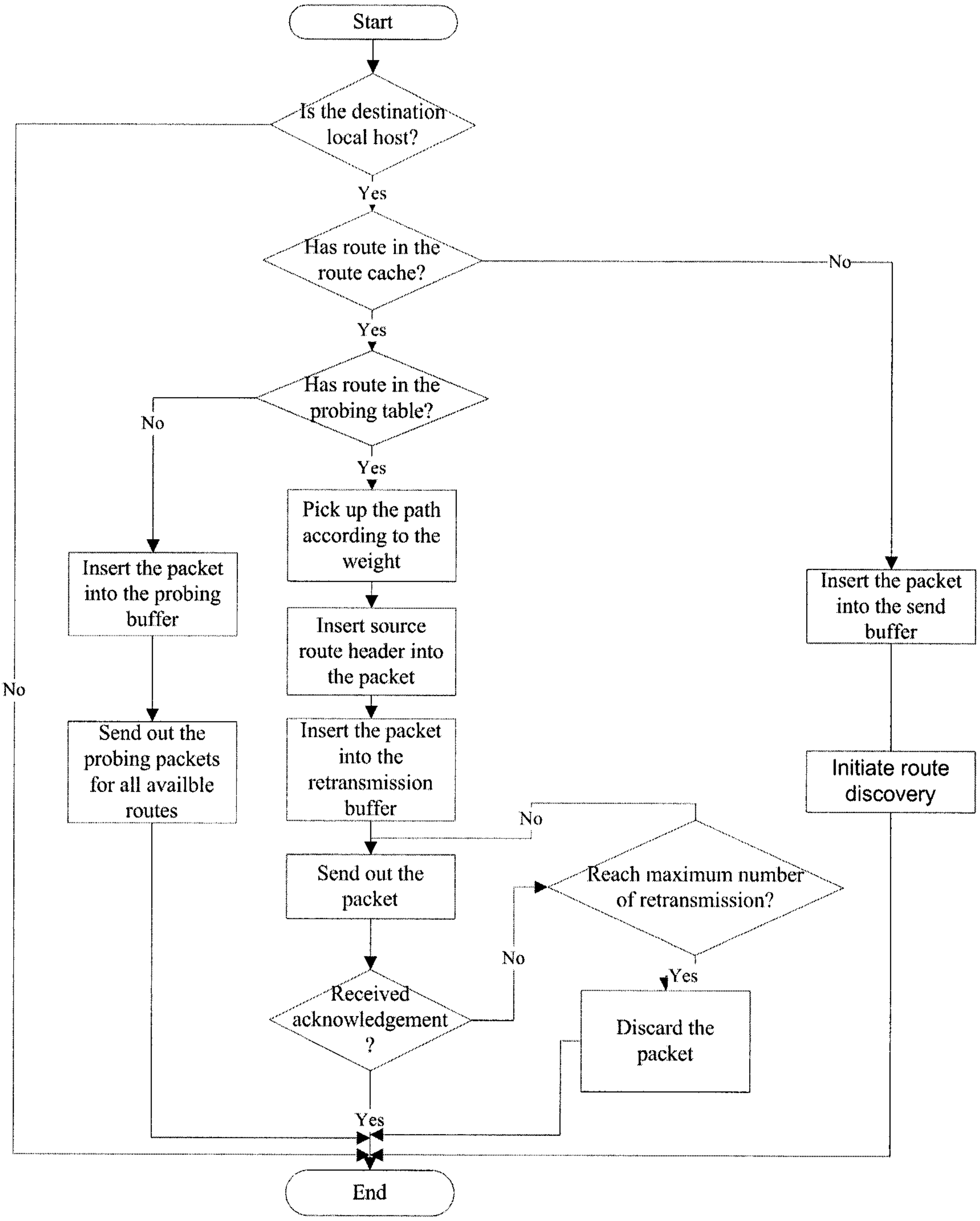


Figure 2.7: Flow Chart for Transmitting Packets in MSR

In MSR, the packet is sent out only when the node has the delay information of all the available paths. The path is selected by the weighted-round-robin scheduling strategy [Liu00] based on the weight value derived from delay. The path in the route cache does not have the status information, so after searching the route cache and getting the route information, the node has to send probing packets to every path to detect the delay status. Before getting the status details, the node inserts into the probing buffer all the packets which are waiting for being sent out. Figure 2.7 shows the detailed operation flow chart for transmitting packets in MSR.

2.5.3 The Improved MSR

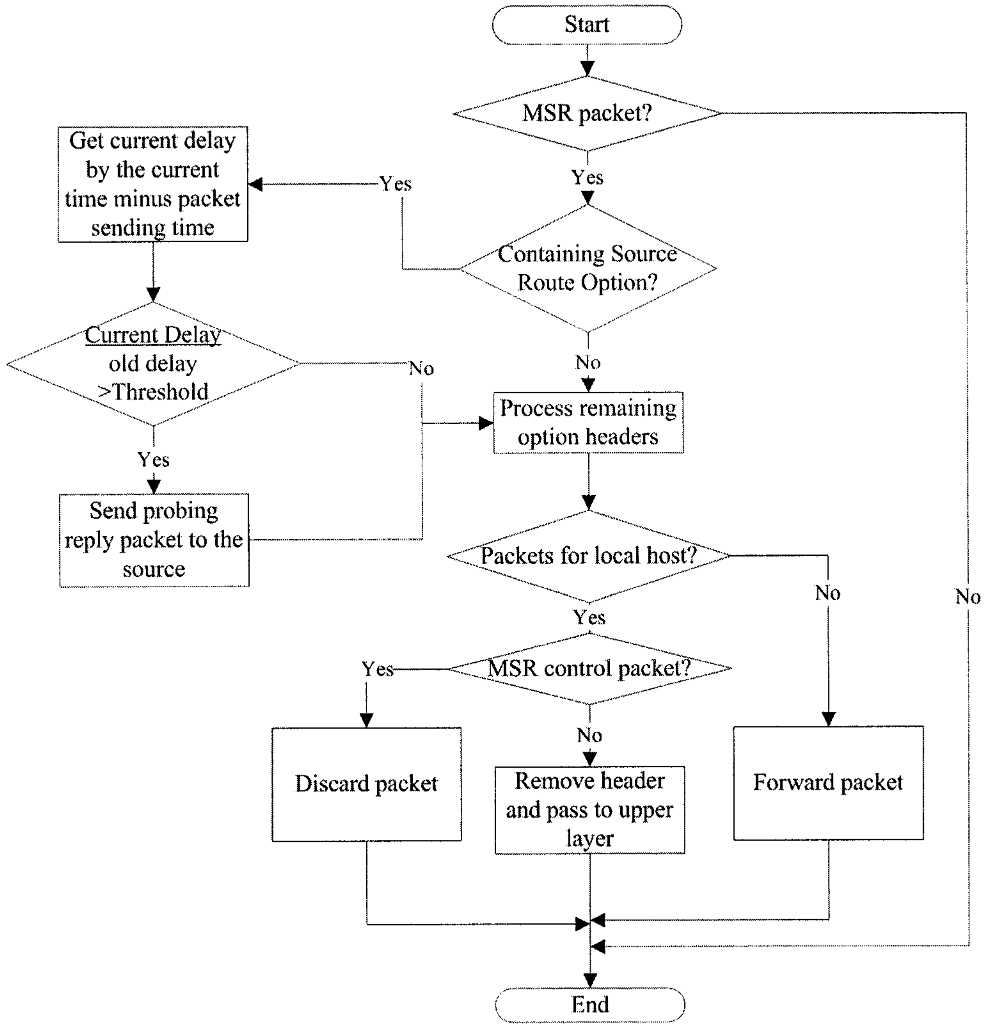


Figure 2.8: Flow Chart for Receiving Packets in the destination in the improved MSR

We deploy the adaptive probing in our improved version of MSR. In order to achieve that, we have added the "delay" and "sending time" parameters in the source route option in the MSR header. In order to minimize the work for changing the code, we only set the probing period to a large enough value so that we only use it once in our test to obtain the initial values of each path delay. It seems as if we have just disabled the periodic probing function but actually the source node only probes once to obtain the initial path delay value. When the destination receives a data packet containing source route option, it calculates the current path delay by the different between the current time and "sent time" of the packet and compares the current delay with the old one contained in the MSR header. If the difference between the two delays exceeds the threshold $D_{threshold}$, the destination will send back the new delay information to the source node. Thus the destination node will monitor the delay change in the network and feedback the new information to the source.

Figure 2.8 show the procedure in the destination for receiving packets in the improved MSR. Here we add one additional processing for the source route option in the destination. The destination calculates the ratio of the current delay to the old delay and compares it with the threshold. If the ratio exceeds the threshold, the destination sends back a probe reply to the source. The source will update the delay information in the route cache upon receiving the probing reply. The procedures of transmitting packets in all nodes and receiving packets in non-destination nodes are the same as MSR.

In order to constrain the update frequency of the delay notification, we define a maximum delay feedback time T_f. Each time when the destination detects the large change of delay and has to feedback, a time counter with the expiration time T_f will start. Before the time expires, the destination will ignore all the delay changes larger than a threshold $D_{threshold}$. We choose T_f equal to the probing period in regular MSR so that the performance of the adaptive probing is not worse than the periodic one. On the other hand, adaptive probing needs more processing on the header and may cause larger delay, but the delay performance can be more stable or have less variance and can lead to larger throughput.

2.5 Summary

MSR is based on DSR and thus inherits all the benefits of DSR. By utilizing source routing, MSR can improve performance by giving applications the freedom to use multiple paths within the same path service. On the other hand, maintaining alternative paths requires more routing table space and computation overload. Fortunately, some DSR characteristics can suppress these disadvantages. First of all, source routing is so flexible that messages can be forwarded on arbitrary paths, which makes it very easy to dispatch messages to multiple paths without any demanding path calculation at the intermediate hops. Secondly, the on-demand nature of DSR helps to reduce the routing storage and routing computation significantly. Moreover, some computation overload is not a big issue with the more powerful and fast computing devices nowadays.

The improved MSR employs adaptive probing instead of periodic probing in MSR. The delay information is added to the source route option in packet header. The destination is responsible to monitor the delay change and sends back updated delay information to the source in case delay changes a lot.

BSR takes full advantage of the many routes collected in route discovery to select one backup path in addition to the shortest path. The link reliability is improved in this manner.

We have detailed the processing procedures of each protocol with the help of flow charts. These will be implemented in the testbed to be described in the next chapter.

Chapter 3
Testbed Setup

In this chapter, we will describe our testbed for the implementation and testing of DSR, MSR, the improved MSR, and BSR. The hardware setup and software development environment of Linux are described as well as the code development.

3.1 Hardware Setup

Since Ad Hoc networks do not need an infrastructure, we simply choose several IBM laptop computers (each equipped with a wireless card) as network nodes. All the wireless cards are set to the Ad Hoc mode with the same network name and mask so that they can be interconnected in one Ad Hoc subnet[1]. Two kinds of wireless card are used: Cisco card and Demarc card.

Both wireless cards cannot transmit and receive at the same time. Therefore, when the intermediate node is transmitting packets, the previous node will not send packets to it. In this way, one more hop with the introduction of one more intermediate node will cause longer delay and smaller throughput. For example, in our testbed, the throughput with direct connection is about 250 Kbytes/sec, but the throughput for two-hop communication has dropped to about 160 Kbytes/sec.

3.1.1 Cisco Aironet 350 Series PC Card

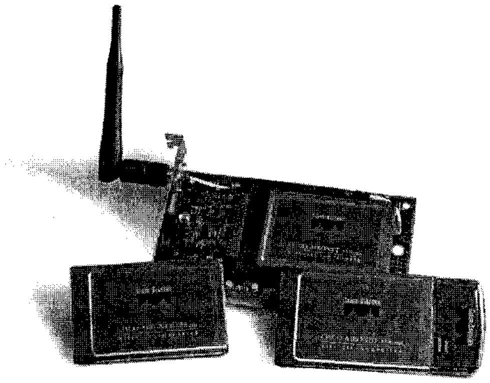


Figure 3.1: Cisco Aironet 350 Series PC card

---

[1] Each wireless card has two working modes: ad hoc and infrastructure. In the ad hoc mode, the laptops interconnect themselves without the need for the access point. Two laptops can communicate with each other as long as the signal is strong enough to form a link. In the infrastructure mode, a laptop must communicate with each other or an existing wired network via the access point. Our protocols are designed for ad hoc network which does not need access point.

We have chosen Cisco Aironet 350 Series PC card (as shown in Figure 3.1) because it comes with a Linux driver, and the transmission power can be adjusted. This card provides 6 available transmission power levels: 1mW, 5mW, 20mW, 30mW, 50mW, and 100mW. In our test, we first select the lowest transmission power level i.e. at 1mW dial setting so that we can construct different topologies in relatively small range.

3.1.2 Demarc 802.11b 100mW High Power Wireless PC Card

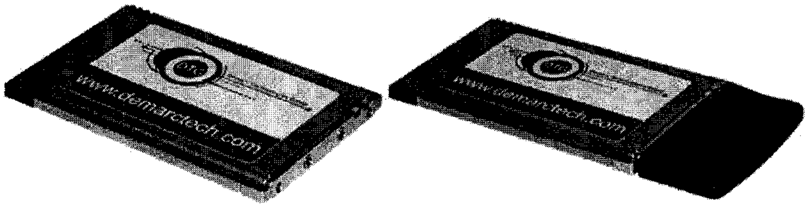


Figure 3.2: Demarc 802.11b 100mW High Power Wireless PC Card

Another wireless card we used is the Demarc 802.11b 100mW High Power Wireless PC Card shown in Figure 3.3. The most attractive feature of this kind of card is the removable antenna which is very useful especially in our debugging phase. We observed within our lab distance that the nodes equipped with the card without the antenna cannot reach each other while the node with antenna can reach the one without antenna. Based on this, we can easily construct a 2-hop scenario by removing the antennas in the source and destination so that they have to use the intermediate node as relay to communicate with each other.

3.1.3 IBM Laptops

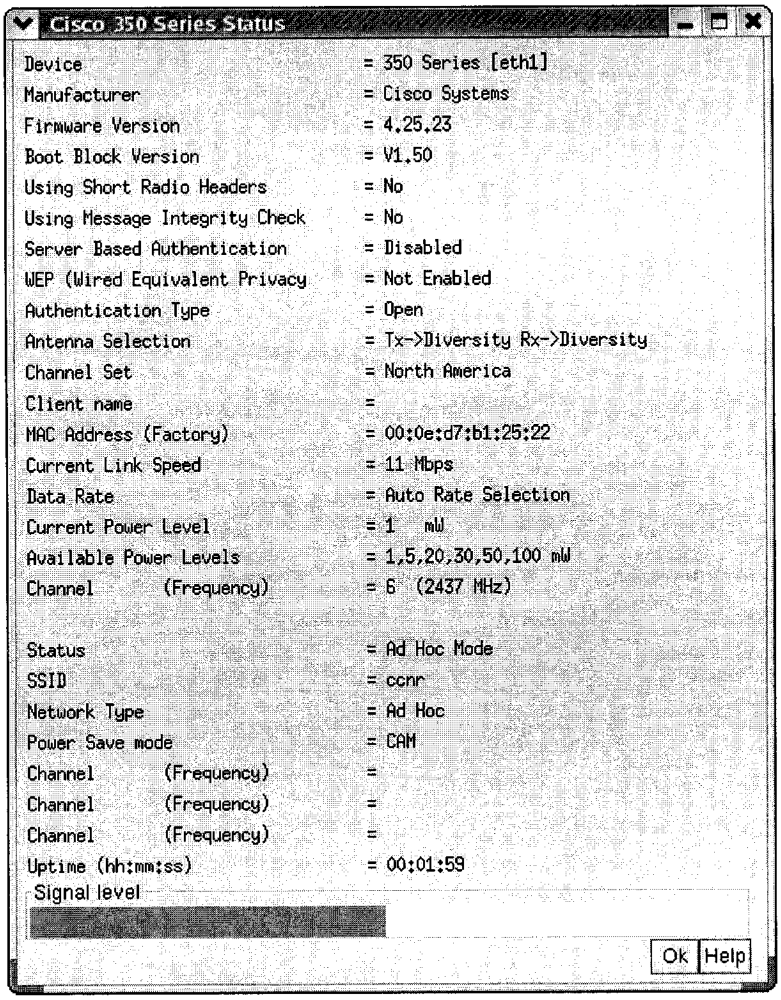
We use up to seven IBM T40 and R40 laptops in our testbed. These laptops are compact and smaller enough for us to move around to test the mobile scenario, while they are powerful enough to meet the computation and processing requirements.

3.2 Software Setup

In this section, we introduce the software setup in our testbed including wireless card drivers, development environment, and protocol implementation.

3.2.1 Linux Drivers for Wireless Cards

The Cisco wireless card provides a Linux driver with GUI tool called ACU (Aironet Client Utility) for us to configure the wireless card. The status window shown in Figure 3.3 lists all the settings for a working wireless card. The current link speed is 11 Mbps since the Cisco card works under 802.11b and the current power level is 1mW which is the lowest power. We can choose any channel as long as all the nodes are using the same channel. Here we just pick Channel 6 in our network. The status is Ad Hoc mode. The SSID (Service Set Identifier) is a sequence of characters that uniquely names a wireless network and each laptop should designate the same SSID. We use default setting for all the other parameters. The detailed description is listed in Appendix A.



```
Device                              = 350 Series [eth1]
Manufacturer                        = Cisco Systems
Firmware Version                    = 4.25.23
Boot Block Version                  = V1.50
Using Short Radio Headers           = No
Using Message Integrity Check       = No
Server Based Authentication         = Disabled
WEP (Wired Equivalent Privacy       = Not Enabled
Authentication Type                 = Open
Antenna Selection                   = Tx->Diversity Rx->Diversity
Channel Set                         = North America
Client name                         =
MAC Address (Factory)               = 00:0e:d7:b1:25:22
Current Link Speed                  = 11 Mbps
Data Rate                           = Auto Rate Selection
Current Power Level                 = 1   mW
Available Power Levels              = 1,5,20,30,50,100 mW
Channel       (Frequency)           = 6  (2437 MHz)

Status                              = Ad Hoc Mode
SSID                                = ccnr
Network Type                        = Ad Hoc
Power Save mode                     = CAM
Channel       (Frequency)           =
Channel       (Frequency)           =
Channel       (Frequency)           =
Uptime (hh:mm:ss)                   = 00:01:59
```

Figure 3.3: The status of Cisco Aironet 350 Series PC card

The Demarc wireless card provides a useful set of Linux driver called HOSTAP driver. The HOSTAP driver does not provide any graph client software to configure the wireless card, so we can only use the command "iwconfig" which is already included in Linux to configure the card. The detailed configuration using commands is list in Appendix B.

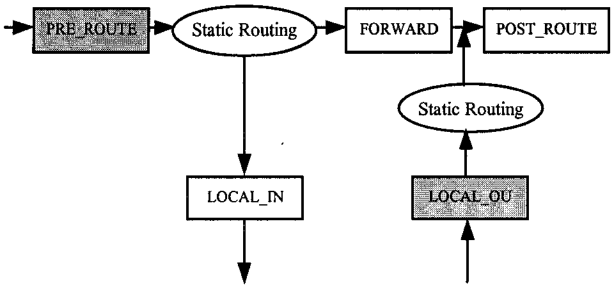


Figure 3.4: Netfilter Hook Functions for IPv4

3.2.2 Development Environment

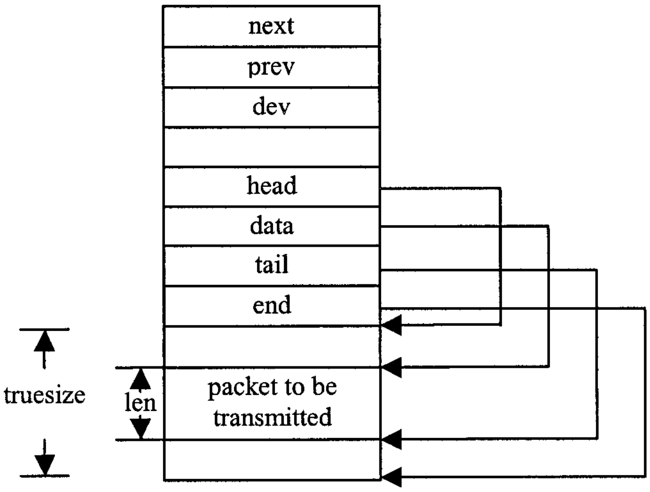


Figure 3.5: sk_buff structure

The Linux Operation System we used is Redhat 9 [Redh05] which uses 2.4 kernels. Linux 2.4 kernels provide a firewall framework, Netfileter [13], for software development. It is independent of network devices. The code implemented with Netfilter structure can be running in any type of network cards. Figure 3.4 depicts the Netfilter structure, where 5 "hook" functions are defined. A "hook" is a variable where you can store functions that can be called by system process. The PRE_ROUTE hook function is called when local machine received a non-promiscuous packet. If this packet is destined to another host, FORWARD is called and before it is finally sent out, POST_ROUTE will be called; otherwise, LOCAL_IN is called when it is destined to the local host. The LOCAL_OUT hook is called when the local host creates a packet to send. The grey boxes PRE_ROUTE and LOCAL_OUT are where we implement our protocols.

Linux uses a data structure called sk_buff in the program to pass data between the protocol layers and the network devices. This is the socket buffer containing several pointers and length fields that allow each protocol layer to process the application data via some standard functions.

As shown in Figure 3.5, each sk_buff contains three header pointers, a data block, four pointers, and two length fields. The three header poiters are listed below:

- Next: point to next buffer in list.
- Prev: point to previous buffer in list.
- Dev: the device we are leaving by.

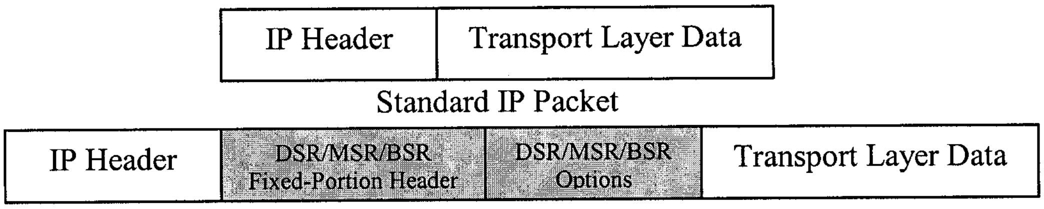
The four pointers are detailed below:

- Head: point to the starting address of the data area in the memory. The pointer is fixed after sk_buff and relevant data block are allocated.
- Data: point to the starting address of the protocol data. It varies according to the current protocol layer which processes the sk_buff.
- Tail: point to the end address of the protocol data. Like head pointer, it varies according to the current protocol layer.
- End: point to the end of the data area in the memory. It remains fixed after sk_buff is allocated.

The two length fields (len and truesize) refer to the length of protocol data packets and the total length of the data in the buffer respectively.

3.2.3 Protocol Implementation

For the implementation of DSR, MSR, and BSR, we shall focus on the details of data structures and packet processing procedure.



Standard IP Packet

IP Packet with DSR/MSR/BSR information

Figure 3.6: IP Packet Structure

3.2.3.1 Packet Formats

Our protocols make use of a special header containing control information, which can be included in any IP packet. Figure 3.6 shows the packet structure for standard IP and the modified one after inserting DSR/MSR/BSR header which contains a fixed-length portion and a sequence of options carrying option information. The IP header is added by the network layer of TCP/IP protocol and consists of 20 bytes of data. The transport layer data is transferred from the upper TCP layer.

Next Header	Reserved	Identification
Header Length		

Figure 3.7: Fixed Portion of the DSR/MSR/BSR Header

The content of the fixed portion of the header is shown in Figure 3.7. The Next Header field is defined as the protocol ID for the following header. The Reserved field is reserved for future use. The Identification field records the packet sequence number and the header length field defines the total length of the DSR/MSR/BSR header.

Since MSR and BSR are extended from DSR, they share some common options which lie in the grey box "DSR/MSR/BSR Options" in the packet. The common options used for DSR, MSR, and BSR are as follows:

- Route Request Option: used for route request packets which will be flooded in the network in the route discovery.
- Route Reply Option: used for route reply packets which contain the full path information and will be sent back to the source by the destination.
- Source Route Option: contains the full route information and used for forwarding data and other control packet. (There are two routes in BSR: one is primary path and the other is backup path.)
- Route Error Option: used for route error packets which contain the break link information and will be sent back in the upstream direction.
- Acknowledgement Option: used for the hop-by-hop acknowledgement packets which will be sent back to its previous hop to confirm the successful transmission.

There are two additional options in MSR for probing path delay:

- Route Probing Option: used for route probing packets which will be sent by the source along the path without the delay information.
- Probing Reply Option: used for probing reply packets which contain the new delay information and will be sent back to the source.

The following is a list of packet formats used in our protocols from different combination of options:

- Route Request Packet: ip header + fixed-portion header + Request Option
- Route Reply Packet: ip header + fixed-portion header + Source Route option + Route Reply Option
- Route Error Packet: ip header + fixed-portion header + Source Route Option + Route Error Option
- Acknowledgement Packet: ip header + fixed-portion header + Acknowledgement Option
- Data Packet: ip header + fixed-portion header + Source Route Option + payload
- Route Probing Packet: ip header + fixed-portion header + Source Route Option + Probing Option
- Probing Reply Packet: ip header + fixed-portion header + Probing Reply Option

3.2.3.2 Data Structures

Since MSR and BSR are extended from DSR, they have some common data structures (caches, arrays etc.) for the operation of these protocols implemented. In the code of our testbed, we pre-allocate some memory space that can be used for these data structures. We briefly introduce these data structures.

- Route Cache: All the route information acquired by a node is stored in the node's route cache. The route is inserted into the route cache when discovered and deleted when failing or expiring. The route cache is indexed by the destination address.
- Send Buffer: The send buffer of a node stores all the packets that cannot be sent due to the lack of a source route to the destination. All the packets to the destination will be resent after a successful route discovery. Each packet is associated with the time that it was inserted into the send buffer and is discarded after a period of SendBufferTimeout.
- Request Table: The request table records the route request information that have been originated or forwarded by the node. Before the entry is deleted after it times out, the node will not send or forward route request to that destination.
- Retransmission Buffer: Our protocols use hop-by-hop acknowledgement mechanism to maintain the route information. The data packet is successfully transmitted if the acknowledgement from the next hop is received. If the acknowledgement is not received

after a period of RetransmissionTimeOut, the node will retransmit the data packet or discard it after retrying it several times. The retransmission buffer records all the data packets that are waiting for the acknowledgement. It is indexed by the packet ID.

MSR employs periodic probing mechanism to get the up-to-date path delay, so it possesses some new data structures.

- Probing Buffer: It is used to record those packets that are waiting for probing reply. Different from those in the send buffer, the packets in the probing buffer have the route information to the destination, but the status or the delay of the path is unknown. Once the source node received the probing reply, it will calculate the path weight according to the returned information and send out all the packets one by one.
- Probing Table: The probing table contains the information of the paths that need to be probed in order to obtain the delay status. After the probing reply is received, the weight function is recalculated and the relevant information is updated in the probing table. Each entry in the probing table is associated with a timer, and when the timer expires, the entry will be deleted and a new probing procedure is needed. In this way, the probing table is refreshed periodically so that it keeps the most up-to-date path information. The probing table is indexed by the destination address. The probing table is the main part of the MSR protocol because each data packet needs to pick up a path from the probing table using weighted-round-robin scheduling strategy according to the weight function.

3.3 Performance Measures

There are three performance measures we used in our evaluation:

1) Delay: defined to be the time duration from the time that the first bit of the Ping Request is sent out until the last bit of the Ping Reply is received back at the sender. Ping delay is shown in Figure 3.8 which depicts a simple example for packet forwarding of Ping application in a two-hop communication. Here hop-by-hop acknowledgement is used.
2) Packet Loss Rate: defined to be the ratio of the total number of the dropped packets to the total packets sent for a given measurement period. Packet loss must be accompanied by one retransmission timer expiration.
3) Throughput: defined to be the total number of bytes of all packets sent successfully per unit time.

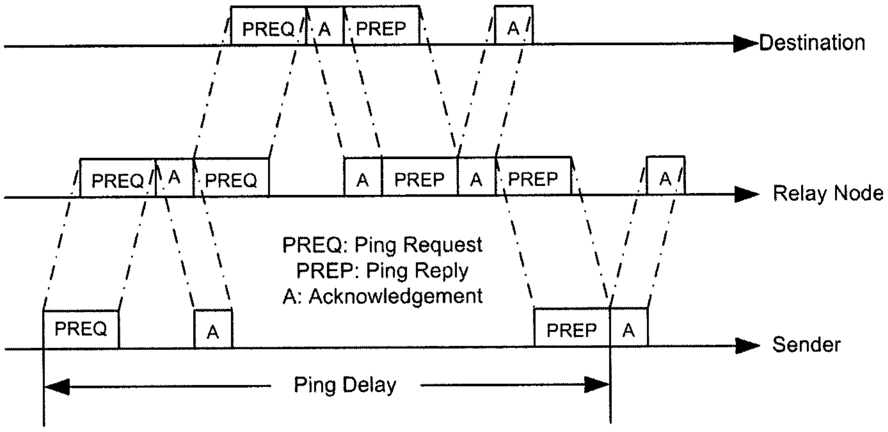


Figure 3.8: Packet Forwarding Procedure for Ping Application

We use the Ping application to obtain the delay and packet loss rate. The Ping program is a client for ICMP (Internet Control Message Protocol) which is used for network diagnoses. The Ping program will transmit a series of ICMP echo request packets from the source to the destination and the destination will send back the ICMP echo reply messages to the source. When the ICMP echo reply messages are received by the source, the Ping program will calculate the average round trip time and loss rate. In the way, we can collect the statistic results from the Ping tests.

Table 3.1: Example for Load Setting

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	Number of Packets	Offered Load (bps)	Delay (ms)	Packet Loss Rate
1	64	1	100	512	8.870	0%
2	64	0.1	1000	5120	11.435	0%
3	64	0.01	1000	51200	12.623	0%
4	512	0.01	1000	409600	16.795	0%
5	1024	0.01	1000	819200	19.570	0%
6	1024	-f	1000	1638400	20.806	0%

We adjust the load by changing the two parameters data packet size and sending interval in the ping command. The offered load of the network is measured by the number of packets sent per unit time or in bits/s. It is calculated by: (Number of Data Packets) * (Packet Size in Bytes) * (8 bits per byte) / (test period). Table 3.1 shows an example for load setting in Ping tests used throughout our experiments. We first fix the data packet size at 64 bytes and decrease the sending interval from 1 second to 0.01 second by 0.1 in each run to increase the traffic load. At a sending interval of 0.01s, we increase the data packet size from 64 bytes to 1024 bytes. In the last run, we choose the sending interval to be "-f" which means that the source node can send packets as quickly as possible (in our testbed, the sending interval in this case is about 5ms). In this way, we get different traffic loads from light traffic to heavy traffic. We set the number of packets as 1000 in all the cases except 100 for the first case to save time since the measurement results for the total number of packets 100 and 1000 are almost the same.

We use the FTP application to obtain the throughput. Linux includes a tool named "vsftpd" (Very Secure FTP Daemon) for FTP server configuration and also provides a program for FTP client. In order to connect to the FTP server, we only need to enter the command "ftp 10.0.0.2" if the FTP server has IP address 10.0.0.2. We can download files from the FTP server. Each time the file transfer finishes, the FTP client program will show the total transmission time and throughput, so we can also easily collect the statistic results.

Table 3.2: Example of throughputs for different file sizes

File size	Throughput (Kbytes/sec)			
	DSR	MSR	MSRI	BSR
100K	165	150	155	165
500K	170	145	160	170
1M	170	150	160	170
2M	160	140	160	170
3M	160	150	160	170
4M	155	150	155	160
5M	155	140	150	165
10M	160	145	160	170

Table 3.2 shows an example of the throughputs for different file sizes in FTP application. We observe that the throughputs for different file sizes are close, so in our testing we only tested five cases for file size 1M to 5M.

Note that the delay throughput performances can be much affected by the antenna's capabilities such as receiving and transmitting at the same time as discussed earlier in Section 3.1.2.

3.4 Scenarios

There are two major types of scenarios: static and mobile. We tested them in both indoor and outdoor environments with the same distance. All our outdoor work has been done on campus outside the SITE building of 6-story high and the other side is a highway and a canal with residential houses.

3.4.1 Static Topologies

We use three static topologies to test and compare our algorithm. The first one is the simplest one without background traffic while the other two are testing the interference effect from background traffic.

Scenario 1: Simple Multiple Paths

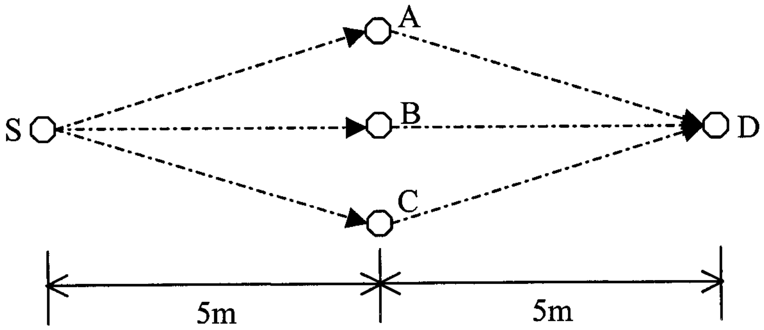


Figure 3.9: Scenario 1

In this topology, the source S and destination D are separated far (e.g. 10 meters) enough so that they cannot reach each other. We then add 3 intermediate relay nodes A, B, and C in between as the relay nodes. This would allow three independent routes S-A-D, S-B-D, and S-C-D to be found by MSR in the route discovery. Note that MSR can use all the routes simultaneously (but only send one packet on one route at a time) so as to reduce the network congestion while DSR only uses the shortest one. We shall use TCP traffic such as FTP application to verify our protocols. We also vary our traffic load to see the performance changes.

Scenario 2: Interference Test

In Figure 3.10, we use node A to introduce the background traffic by sending Ping request packets to D with the data packet size 64 bytes and the sending interval 0.1s. B is the bottleneck node and all the traffic goes through it. Three paths are found in route discovery in MSR: S-B-C-D, S-B-E-D, and S-B-F-D. DSR only uses one of them.

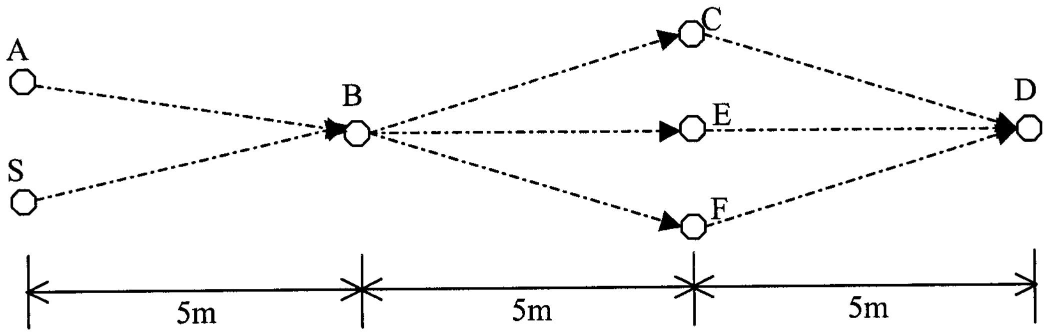


Figure 3.10: Scenario 2

Scenario 3: Bottleneck Test

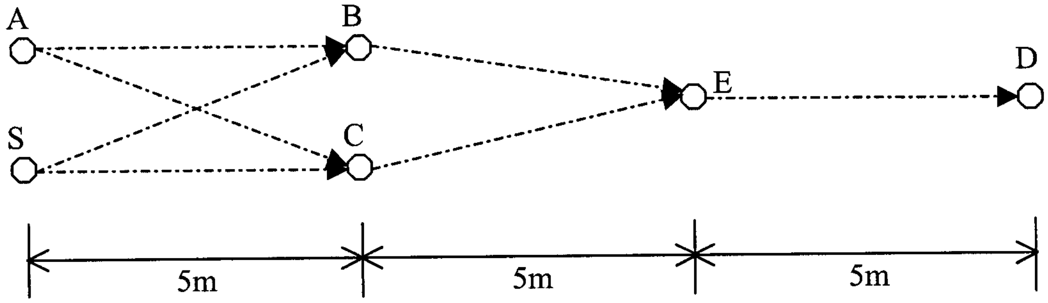


Figure 3.11: Scenario 3

In this topology, we want to investigate the effect of bottleneck location. We put the bottleneck node E at the last hop from S to D while the bottleneck node is in the first hop in scenario 2. Node A introduces the background traffic by sending Ping packets to D with a data packet size 64 bytes and a sending interval 0.1s (the same setting as Scenario 2). Two paths are found in route discovery: S-B-E-D and S-C-E-D.

3.4.2 Mobile Topologies

We try three topologies with different moving patterns. In the first one, one intermediate node is moving back and forth between the source and the destination. In the second one, the source node is moving back and forth toward the destination. In the last one, the three intermediate nodes are moving around a circle. Thus we can test different interferences by changing the mobility patterns of intermediate nodes and source node. Unlike the static topologies, no background traffic is used.

Scenario 4: Back and Forth

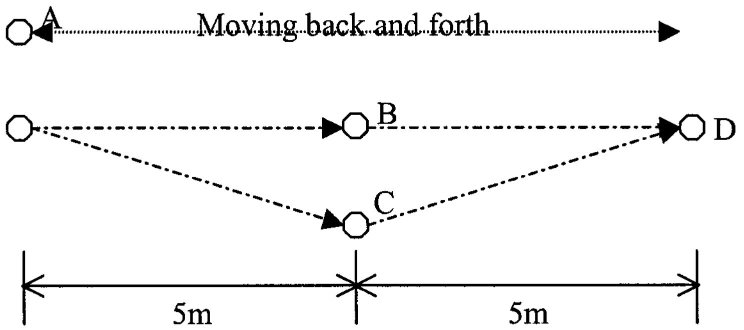


Figure 3.12: Scenario 4

In this topology, the node A is moving between the source and the destination with a walking speed 1 m/s. The link A-D is disconnected when node A moves to S and the link S-A is disconnected when node A moves to D so that the path S-A-D fails sometimes. Since DSR only uses one path in the tests, DSR may use the fixed path S-B-D or S-C-D so that the movement of A does not influence the communication. Then we check how DSR respond to route failure by excluding B and C in our tests and only putting in the moving node A as the intermediate node. One can see that a path can only be formed if A is within certain reach between S and D.

We include B and C in MSR, so the source S has three routes to the destination D: two fixed paths S-B-D and S-C-D and one unstable path S-A-D. In BSR, we also include the nodes B and C. In each test, we first examine the content of the BSR packet header without moving any node and then move the node in the primary path for data transmission. This is done to check how BSR respond to route failure.

Scenario 5: Varying S-D Distance

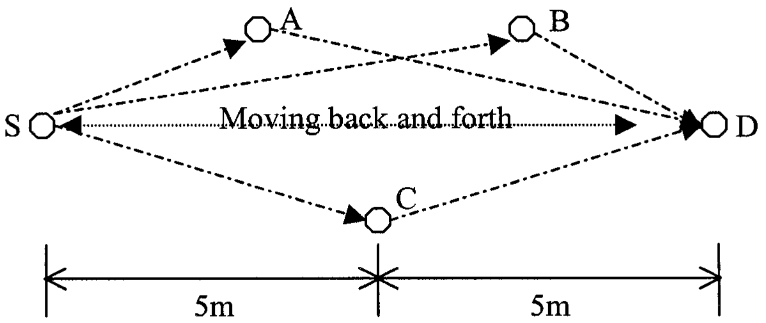


Figure 3.13: Scenario 5

In this topology, we let the source node S move back and forth toward the destination so that in some point S can reach D directly. S can reach the three intermediate nodes all the time.

Scenario 6: Circular Interference

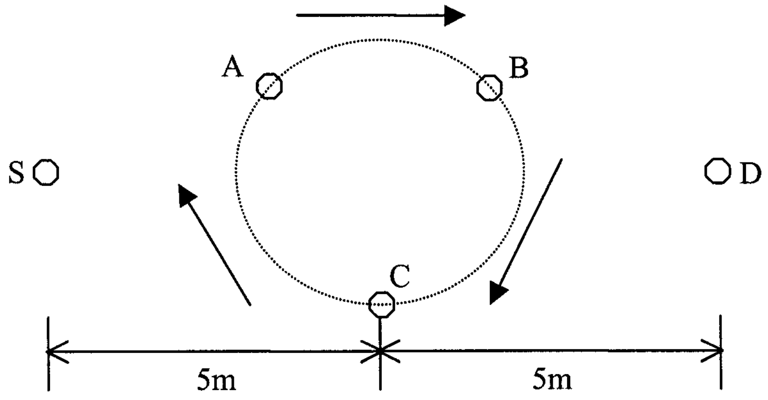


Figure 3.14: Scenario 6

This is the most challenging topology in our tests. Three people hold the three intermediate nodes A, B, and C and move around a circle. Although the links never fail in the test, the positions of the intermediate nodes change frequently which makes the delay of the links between the intermediate nodes and the source or destination change frequently. We just want to test our protocols in tougher environment.

3.5 Concluding Remark

Our testbed is composed of hardware (IBM laptops and Cisco and Demarc wireless cards) and software (Linux operation system and DSR, MSR, the improved MSR, and BSR codes). We have successfully developed our codes in Netfilter environment provided by Linux where we use the "hook" functions to link our implemented codes and system kernel.

Based on the hardware and software, we shall carry out tests such as Ping and TCP applications to obtain the delay and throughput performances. In the six scenarios described above.

Chapter 4

Performance Evaluation of MSR

In this chapter, we compare the MSR and DSR measurement results in the testbed. We investigate the effect of different static and mobile topologies in both indoor and outdoor environments. We have evaluated the performances of applications Ping and FTP. Only two tables of Scenario 1 are presented and the rest are put in the Appendix E for clarity and brevity purposes. All scenarios have been discussed in Section 3.4. Unless specified, every data point is the mean value of 3 measurements. We also provide standard deviation where appropriate. Here standard deviation is defined to be $\sigma = \sqrt{\overline{X^2} - \overline{X}^2}$. Also, in FTP, the file transmission time is defined to be the time duration from the time that the first bit of a file is sent out until the last bit of the file is received at the receiver.

4.1 Scenario 1: Simple Multiple Paths

Table 4.1: Indoor measurements of Scenario 1

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	95% Confidence Interval	Number of Packets	Offered Load (bps)	Average Delay (ms)	95% Confidence Interval
1	64	1	100	512	8.870	0.835	100	512	10.040	1.671
2	64	0.1	1000	5120	11.435	2.514	1000	5120	8.281	0.143
3	64	0.01	1000	51200	12.623	2.026	1000	51200	7.883	0.221
4	512	0.01	1000	409600	16.795	1.892	1000	409600	12.337	0.615
5	1024	0.01	1000	819200	19.570	1.346	1000	819200	15.835	0.612
6	1024	-f	1000	1638400	20.806	0.944	1000	1638400	17.699	0.422

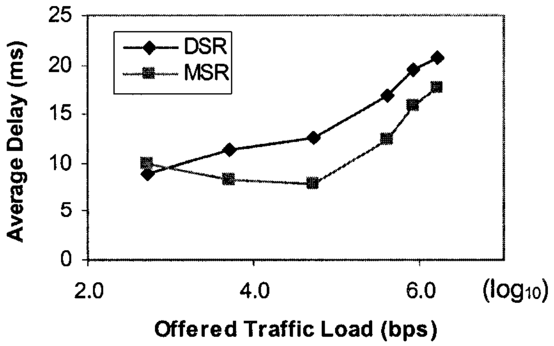


Figure 4.1: Indoor Delay Performance of Scenario 1

4.1.1 Ping Tests

Figure 4.1 shows the indoor delay performance. A $\log_{10}$ scale is used to capture a wide range of traffic load. As one can see, the delay of DSR increases with the increase of traffic load since the more traffic load can lead to more congestion in network and cause longer delay. DSR has less than 10 ms delay in the first run with sending interval 1s. With reference to Figure 3.8, we notice that the propagation delay is very small (~ μsec), while the Ping packet (ranging from 64 to 1024 bytes) transmission time (~ 0.05 msec for 64 bytes) and Ping Reply transmission time also contribute very little to the delay. The major components appear to come from the system processing time including different layer delay (e.g. MAC layer and physical layer) and DSR module processing time.

The delay of MSR first decreases with respect to offered load before increasing. Note that the probing period of MSR is 3s in all runs. In light load, there are 3 packets sent in the probing period. The percentage of probing packets (1/4=25%) is very high. Referring to Figure 2.4, the source node sends 1 probing packet every 4 data packets. One can see 3 data packets can be sent as soon as they arrive while the 4th data packet needs to wait a longer time for the probing reply. Consequently the average delay increases. With an increase of sending rate, more data packets can be sent before the expiration of delay timer and the percentage of probing packets decreases. For example, with a sending interval of 0.1s, the source node sends only 1 probing packet every 11 data packets. There are 10 data packets which do not need to wait for probing reply. Therefore, the second run has a less delay than the first run. However, when the traffic load becomes larger and larger, the network becomes more and more congested, so the delay also becomes longer later.

MSR has a longer delay in light load due to the high percentage of probing packets and less delay in all the other cases than DSR so that in our performance graph, there is a cross point between the first two runs. This is because DSR sends packets through one path, which leads to more processing loads in each intermediate node in that path and hence causes larger queuing delay. On the other hand, MSR has a better performance in delay by distributing loads among different routes. Note that for this static scenario, we did not find out any packet loss.

Both DSR and MSR have small 95% confidence intervals. MSR has smaller 95% confidence interval than DSR except the 1^{st} test run in which MSR has a large percentage of probing packets.

In our outdoor experiments, we obtained performance similar to indoor tests, which is shown in Figure 4.2. The influence by the interference from outside the network is not obvious in our tests in Scenario 1.

Table 4.2: Outdoor measurements of Scenario 1

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	95% Confidence Interval	Number of Packets	Offered Load (bps)	Average Delay (ms)	95% Confidence Interval
1	64	1	100	512	8.640	0.506	100	512	9.983	1.443
2	64	0.1	1000	5120	11.421	0.821	1000	5120	8.269	0.446
3	64	0.01	1000	51200	12.456	0.954	1000	51200	7.953	0.364
4	512	0.01	1000	409600	16.343	0.779	1000	409600	12.297	0.426
5	1024	0.01	1000	819200	19.006	0.722	1000	819200	15.749	0.648
6	1024	-f	1000	1638400	20.506	1.063	1000	1638400	17.523	0.501

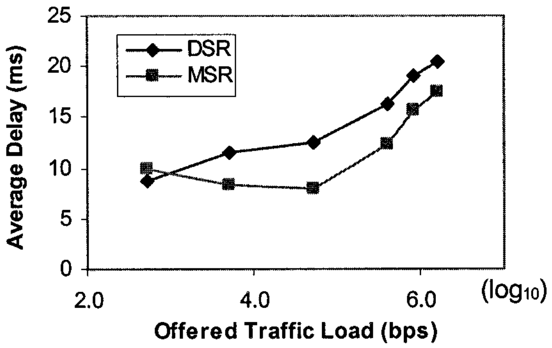
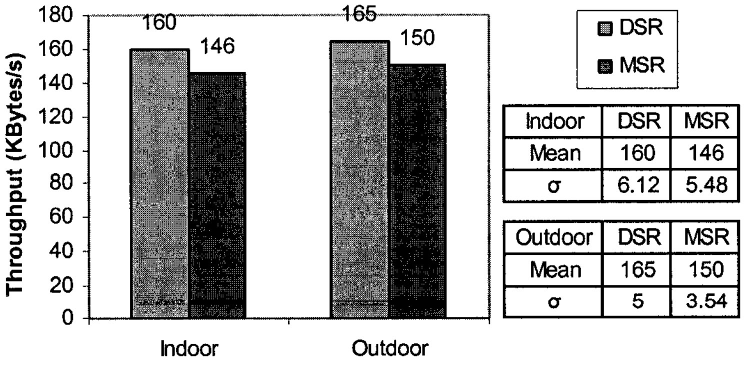


Figure 4.2: Outdoor Delay Performance of Scenario 1

4.1.2 FTP Tests



Indoor	DSR	MSR
Mean	160	146
σ	6.12	5.48

Outdoor	DSR	MSR
Mean	165	150
σ	5	3.54

Figure 4.3: FTP Performance of Scenario 1

Figure 4.3 shows the throughput performance of a destination that downloads FTP files from the source. We choose five file sizes: 1M, 2M, 3M, 4M, and 5M in our tests. We found that the throughputs for different file sizes are almost the same every time, so we only calculate the average throughput for comparison. DSR has a throughput of 160 Kbytes/s and needs about 6s to transfer 1M file. Again, this may be contributed by the processing time as follows. TCP uses 64 Kbytes windows so that 1M will be divided into 16 pieces to transfer. Our link speed is 11Mbps and only needs 0.05 sec to transfer one 64 Kbytes piece. Therefore, the total transmission time of 0.8 sec actually contributes very little to the throughput, and therefore the major component would come from the system processing time (as already explained in the performance of Ping application).

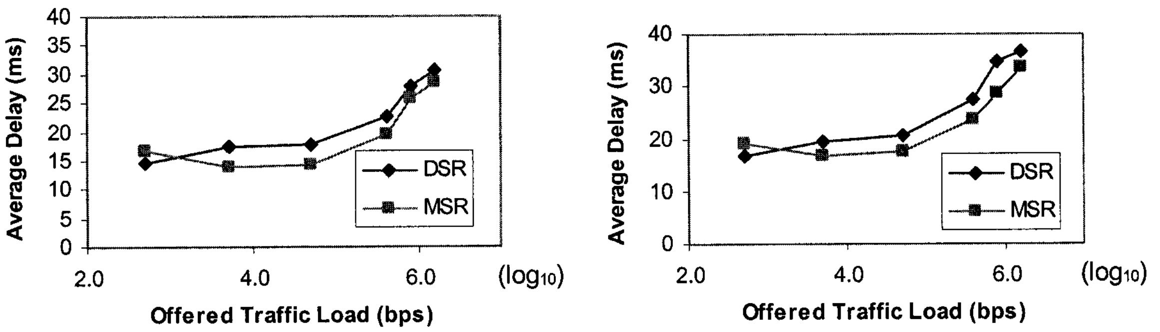
We can see that DSR has larger throughput than MSR. Although MSR can reduce congestion in one single path, the multiple paths used in MSR can cause the packets arriving at the destination out of order. The out-of-order TCP packets cause the retransmission of the missing packets. Finally we see a longer file transmission time in MSR than DSR. We can conclude that the order of the packet arrival is a leading element in determining the total performance in TCP applications. The outdoor and indoor performances are about the same (<3% difference). We also show the standard deviation of each protocol and MSR has a smaller standard deviation than DSR.

4.2 Scenario 2: Interference Test

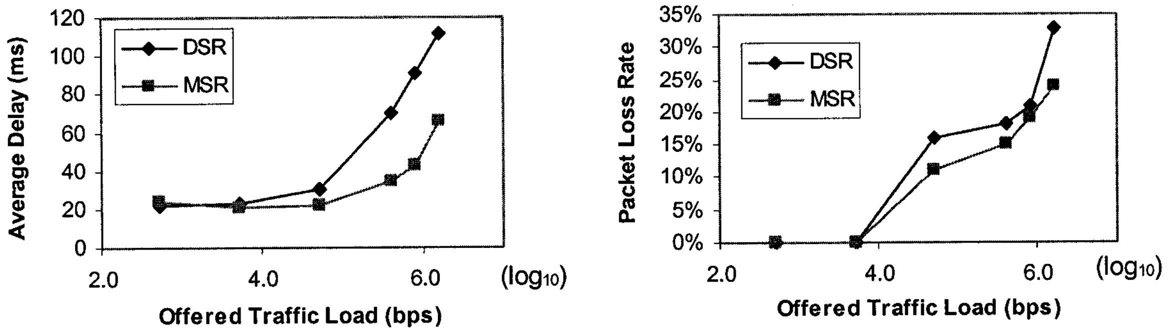
4.2.1 Ping Tests

Figure 4.4a shows the indoor delay performance without background traffic between S and D. Similar observation is made for each of the DSR and MSR delay performance curves. However, the delay is higher compared with Scenario 1 because there is one more hop with the bottleneck node in Scenario 2. The comparison between MSR and DSR is about the same as in Scenario 1. DSR sends packets through one path, which leads to more processing loads in each intermediate node in that path and hence causes larger queuing delay. MSR distributes the load into three links after the node B: B-C, B-E, and B-F and can alleviate the heavy loads more or less despite the heavy burdens in the first hop S-B.

Figure 4.4b and 4.4c depict the delay performances when A is creating background traffic by pinging D at rate of one 64 bytes packet per second, and one 64 bytes packet per 0.1 second respectively. As a result the mean delay increases further. Furthermore, the difference in performance increases as traffic load increases.

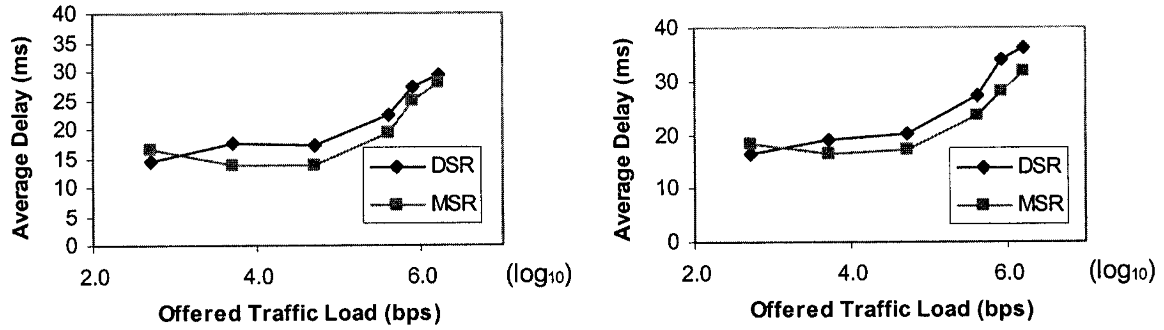


(a) No background traffic (b) With background traffic at 64 bytes/sec

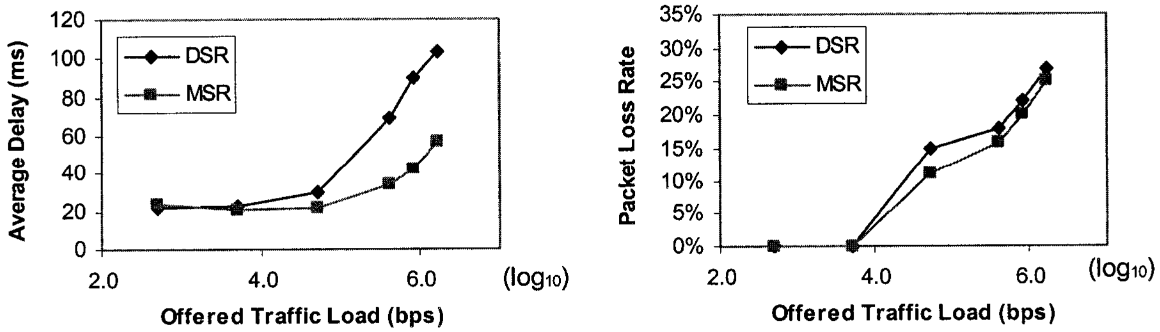


(c) With background traffic at 640 bytes/sec

Figure 4.4: Indoor Delay and Packet Loss Rate Performances of Scenario 2



(a) No background traffic (b) With background traffic at 64 bytes/sec



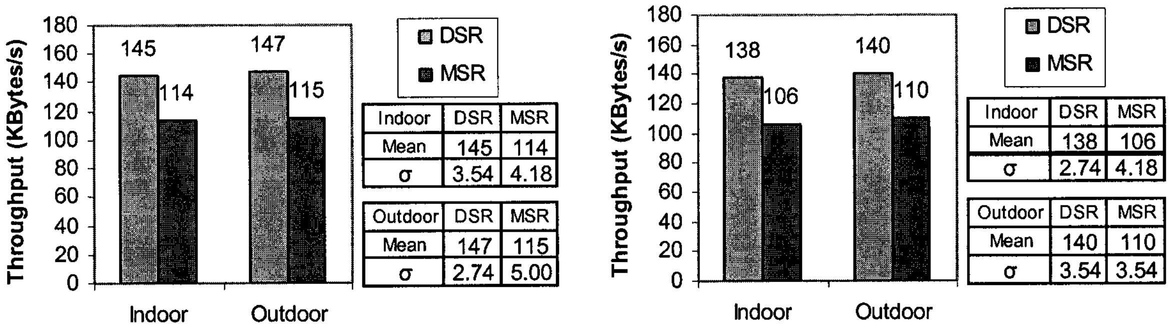
(c) With background traffic at 640 bytes/sec

Figure 4.5: Outdoor Delay and Packet Loss Rate Performances of Scenario 2

We do not observe packet loss in Figure 4.4a without background traffic and 4.4b with light background traffic. However in Figure 4.4c with heavy background traffic, we observe packet loss of DSR and MSR beyond medium load. The DSR loss rate is a lot higher than MSR because of the use of a single path which is more severe at the presence of background traffic. Notice that the out-of-order problem in FTP does not exist in MSR since Ping uses UDP protocol which does not have sequence number for each packet. Therefore, MSR can decrease the packet loss by distributing loads among multiple paths.

From Figure 4.5 in outdoor experiments, we have obtained performance similar to indoor tests. The only difference is that the packet loss rate of outdoor is slightly lower than that of indoor for each protocol because outdoor environment is not as complex as indoor environment and must have less interference.

4.2.2 FTP Tests



Indoor	DSR	MSR
Mean	145	114
σ	3.54	4.18

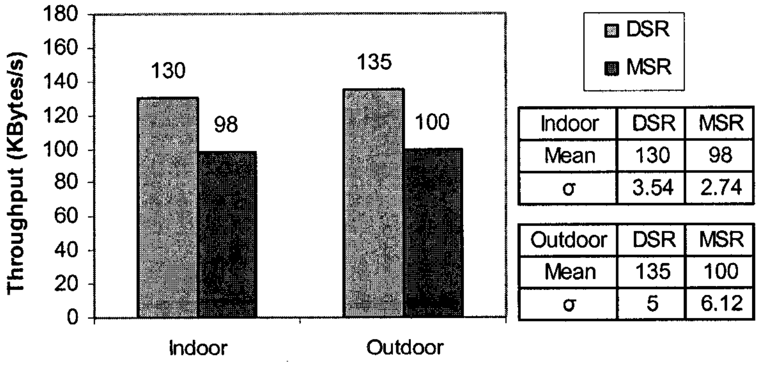
Outdoor	DSR	MSR
Mean	147	115
σ	2.74	5.00

(a) No background traffic

Indoor	DSR	MSR
Mean	138	106
σ	2.74	4.18

Outdoor	DSR	MSR
Mean	140	110
σ	3.54	3.54

(b) With background traffic at 64 bytes/sec



Indoor	DSR	MSR
Mean	130	98
σ	3.54	2.74

Outdoor	DSR	MSR
Mean	135	100
σ	5	6.12

(c) With background traffic at 640 bytes/sec

Figure 4.6: FTP Performances of Scenario 2

Figure 4.6a depicts the throughput of DSR and MSR in both indoor and outdoor environment without background traffic. As seen in Figure 4.6b and 4.6c, the throughputs of both DSR and

MSR decrease with the increase of background traffic. The throughput has gone down because of the increase of background traffic. On the other hand, the decrease of DSR (~9%) is not as significant as MSR (~22%) because the background traffic can increase the out-of-order packet arrivals in MSR. The standard deviation in throughput of DSR and MSR in both indoor and outdoor are smaller than Scenario 1. However as background traffic increases, the standard deviation can increase or decrease. So far this cannot be explained.

The performance of MSR is worse than DSR due to the out-of-order packet arrivals despite the alleviated traffic in each path. Although the total offered load of MSR in FTP application is much more than those in DSR (out-of-order packets cause retransmission which causes more duplicate packets received by the destination), a large number of packets in MSR are out-of-order and discarded by the destination, thus causing the bad performance.

We observe the outdoor performance are very similar to indoor, and can be explained similarly. Outdoor environment does not have an impact on the performance in this scenario.\

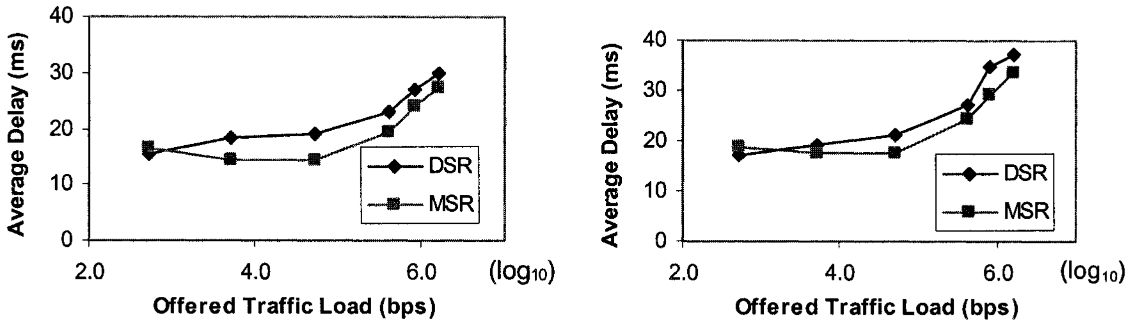
4.3 Scenario 3: Bottleneck Test

4.3.1 Ping Tests

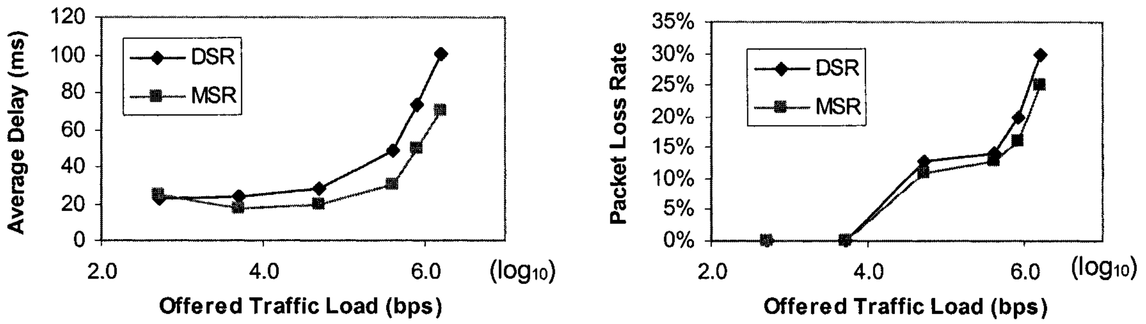
Figure 4.7a shows the indoor delay performance without background traffic while Figure 4.7b and 4.7c depict the performances with increasing background traffic. Similar observation is made for each of the DSR and MSR delay performance curves as Scenario 2 because they both have one bottleneck node and introduce background traffic. The comparison between MSR and DSR is about the same as in Scenario 2. DSR sends packets through one path, which leads to more processing loads in each intermediate node in that path and hence causes longer queuing delay. MSR distributes the load into two links before the node E: B-E and C-E and can alleviate the heavy loads more or less despite the heavy burdens in the last hop E-D.

We do not observe packet loss in Figure 4.7a without background traffic and 4.7b with light background traffic. We observe packet loss of DSR and MSR in Figure 4.7c with heavy background traffic with almost the same pattern as Scenario 2. MSR has less packet loss than DSR beyond medium load because MSR distributes traffic among multiple paths to alleviate the network congestion.

In outdoor experiments, we have obtained performance, which is shown in Figure 4.8, similar to indoor tests since we employ the same topology and the outer interference does not appear to influence the performance much in this scenario.

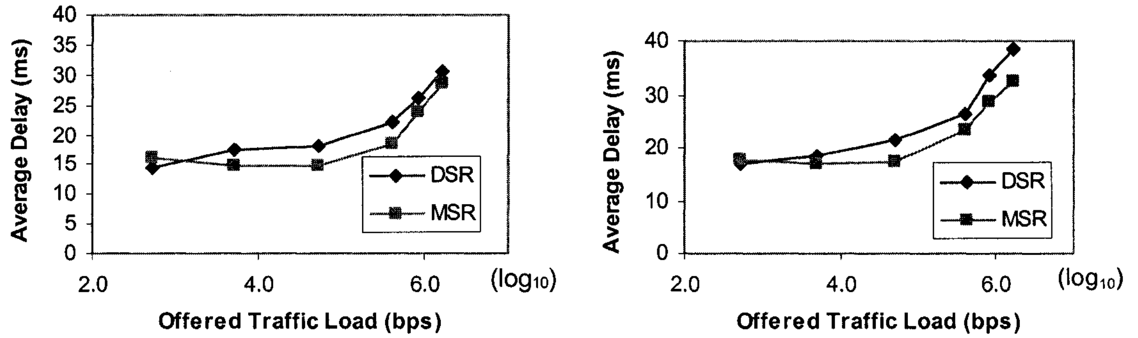


(a) No background traffic (b) With background traffic at 64 bytes/sec

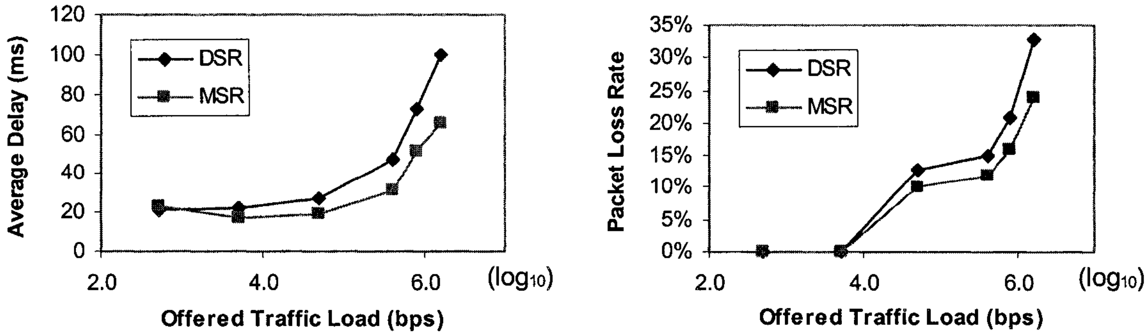


(c) With background traffic at 640 bytes/sec

Figure 4.7: Indoor Delay and Packet Loss Rate Performances of Scenario 3



(a) No background traffic (b) With background traffic at 64 bytes/sec

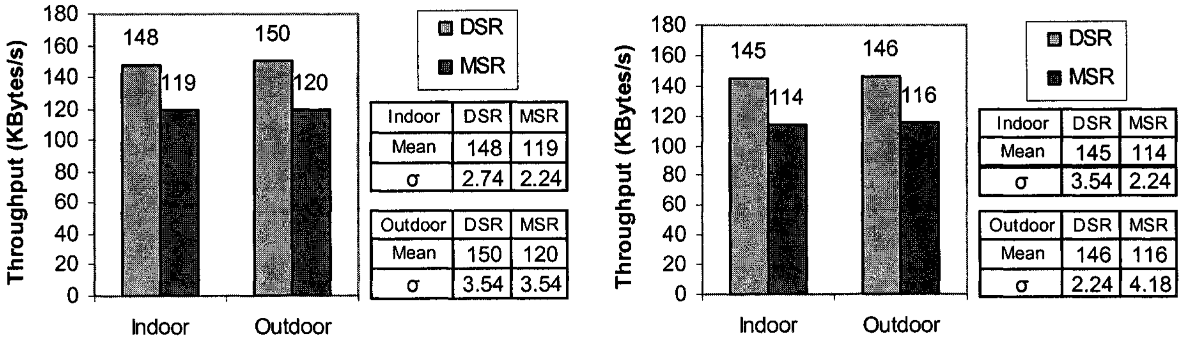


(c) With background traffic at 640 bytes/sec

Figure 4.8: Outdoor Delay and Packet Loss Rate Performances of Scenario 3

4.3.2 FTP Tests

From Figure 4.9, the throughputs of both DSR and MSR decrease with the increase of background traffic similar to Scenario 2. Due to the out-of-order packet arrivals at the destination, MSR has a lower throughput than DSR in TCP traffic in all the three situations without and with background traffic. The multiple paths have utilized the network resources more sufficient, but introduce out-of-order packet arrival to the destination. The standard deviation performance is very similar to that in Scenario 2. The outdoor and indoor performances are very similar and the outdoor environment does not have a lot impact on the performance in this scenario.



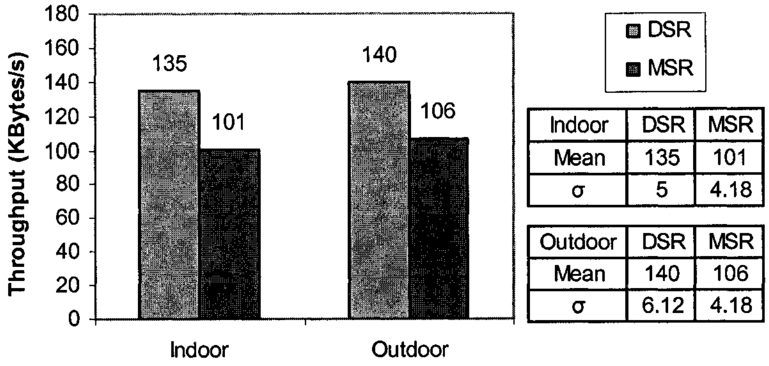
Indoor	DSR	MSR
Mean	148	119
σ	2.74	2.24

Outdoor	DSR	MSR
Mean	150	120
σ	3.54	3.54

Indoor	DSR	MSR
Mean	145	114
σ	3.54	2.24

Outdoor	DSR	MSR
Mean	146	116
σ	2.24	4.18

(a) No background traffic (b) With background traffic at 64 bytes/sec



Indoor	DSR	MSR
Mean	135	101
σ	5	4.18

Outdoor	DSR	MSR
Mean	140	106
σ	6.12	4.18

(c) With background traffic at 640 bytes/sec

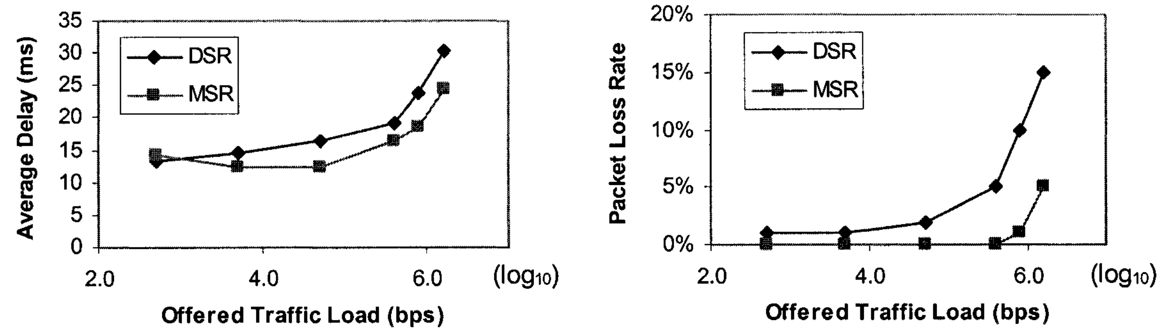
Figure 4.9: FTP Performances of Scenario 3

4.4 Scenario 4: Back and Forth

4.4.1 Ping Tests

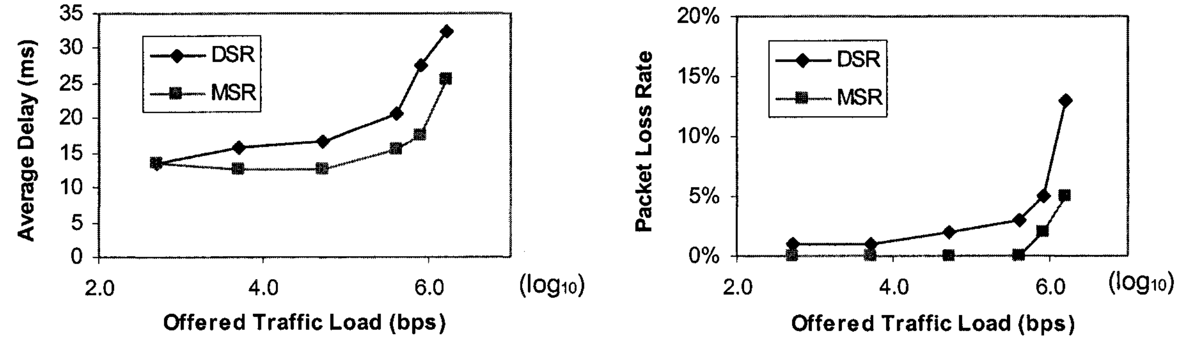
Figure 4.10a shows the indoor delay performance. Similar observation is made for each of the DSR and MSR delay performance curves as Scenario 1 except that each protocol has a higher delay than Scenario 1 due to the movement of node. MSR has shorter delay than DSR because MSR has two stable paths to use while DSR only has one unstable path to use so that MSR outperforms DSR in medium and heavy loads.

In Figure 4.10b, we now observe packet loss in DSR in all loads and packet loss in MSR in heavy load. Unlike Scenario 2 and 3, the packet loss here are due to the movement of the node A, but MSR has less packet loss than DSR because MSR uses the two stable and reliable routes in the transmission.



(a) Delay (b) Packet Loss Rate

Figure 4.10: Indoor Delay and Packet Loss Rate Performances of Scenario 4



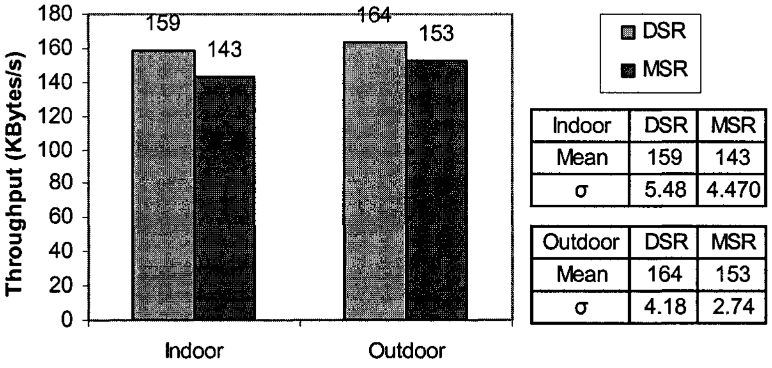
(a) Delay (b) Packet Loss Rate

Figure 4.11: Outdoor Delay and Packet Loss Rate Performances of Scenario 4

As seen from Figure 4.11 in outdoor experiments, we obtained performance similar to indoor tests. The influence by mobility of one node in indoor and outdoor environments does not appear to differ much in our tests.

4.4.2 FTP Tests

The throughput performance depicted in Figure 4.12 is very similar to Scenario 1 that MSR has smaller throughput than DSR due to the out-of-order packet arrival. Also MSR has a smaller standard deviation than DSR. The only difference is that each protocol has a smaller throughput than that in Scenario 1 due to the movement of one intermediate node. The outdoor throughput is higher than indoor since outdoor environment has less interference than indoor.



Indoor	DSR	MSR
Mean	159	143
σ	5.48	4.470

Outdoor	DSR	MSR
Mean	164	153
σ	4.18	2.74

Figure 4.12: FTP Performance of Scenario 4

4.5 Scenario 5: Varying S-D Distance

4.5.1 Ping Tests

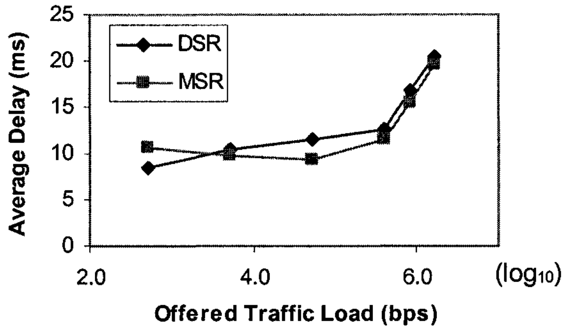


Figure 4.13: Indoor Delay Performance of Scenario 5

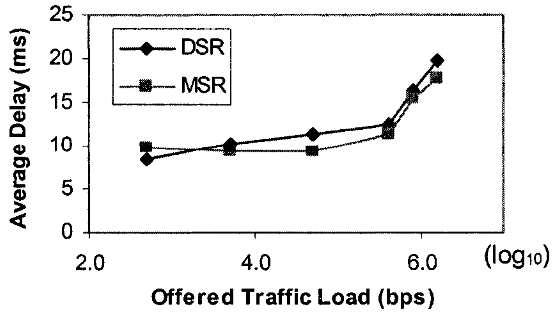
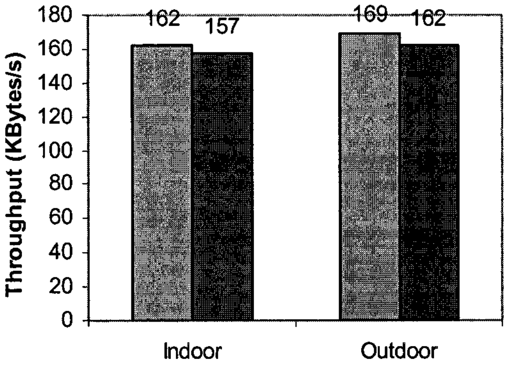


Figure 4.14: Outdoor Delay Performance of Scenario 5

Figure 4.13 shows the indoor delay performance. Similar observation to Scenario 1 is made for each of the DSR and MSR delay performance curves except that MSR has close performance to DSR in heavy load. It is because in mobility environment, each path is unstable, so in heavy load MSR does not improve the performance a lot. We did not record any packet loss.

Figure 4.14 shows the outdoor delay performance. In outdoor experiments, we have obtained performance similar to indoor tests since we employ the same topology and the outer interference does not appear to influence the performance much in this scenario.

4.5.2 FTP Tests



Indoor	DSR	MSR
Mean	162	157
σ	4.47	4.47

Outdoor	DSR	MSR
Mean	169	162
σ	4.18	2.74

Figure 4.15: FTP Performance of Scenario 5

As shown in Figure 4.15, DSR outperforms MSR in TCP transmission because of the lack of the out-of-order packets. MSR decreases network congestion via multiple paths, but introduces out-of-order packets. Although the burden of each intermediate node in MSR is smaller than that of DSR, the TCP performance of MSR is reduced compared with DSR. MSR has a smaller standard deviation than DSR. The outdoor performance is very similar to indoor, which shows the outdoor environment does not influence the performance much in this scenario.

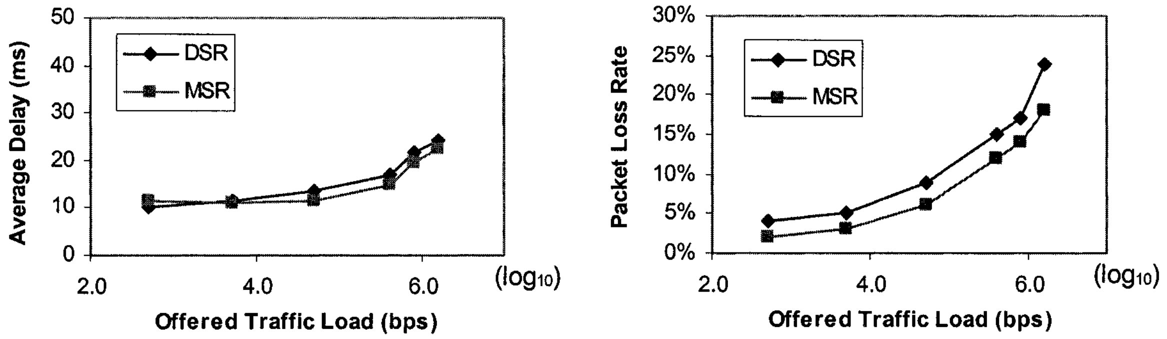
4.6 Scenario 6: Circular Interference

4.6.1 Ping Tests

Figure 4.16 shows the indoor delay and packet loss rate performances. Similar observation to Scenario 5 is made for each of the DSR and MSR delay performance curves. The comparison between MSR and DSR is about the same as in Scenario 5.

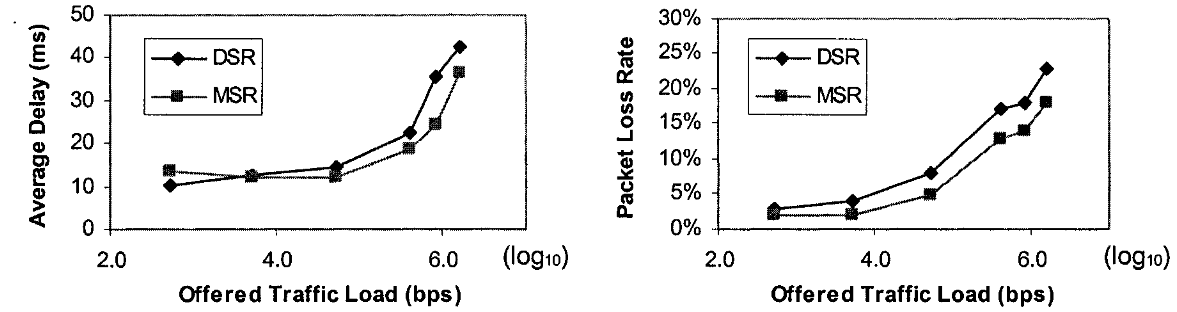
We observe packet loss for DSR and MSR in all loads due to the circular movement of all the three intermediate nodes. MSR has less packet loss than DSR (see Figure 4.16b) because

MSR distributes traffic among multiple paths to alleviate the network congestion in mobile scenarios.



(a) Delay (b) Packet Loss Rate

Figure 4.16: Indoor Delay and Packet Loss Rate Performances of Scenario 6



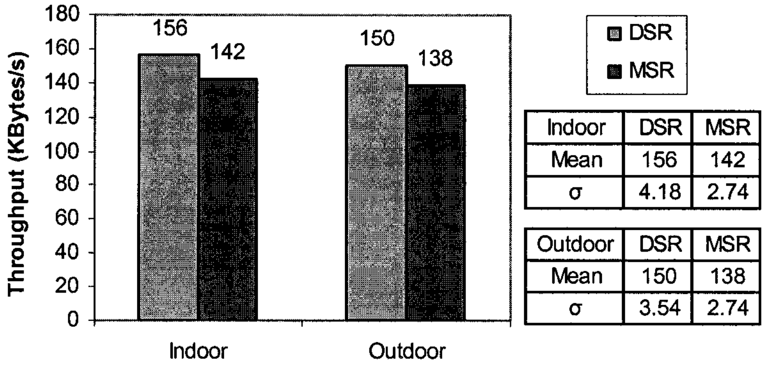
(a) Delay (b) Packet Loss Rate

Figure 4.17: Outdoor Delay and Packet Loss Rate Performances of Scenario 6

Figure 4.17 shows the outdoor measurement results. In outdoor experiments, the comparison between DSR and MSR is the same as indoor. The difference is that both protocols have longer delay in outdoor than indoor. It is because all the paths are unstable in this scenario and the performance in outdoor open area may be worse than that in the indoor environment where reflection may enhance the chance for signal receiving although it induces more interference.

4.6.2 FTP Tests

As shown in Figure 4.18, MSR has worse performance than DSR because of the out-of-order packet arrival problem. The out-of-order packet arrival problem becomes more severe in mobile environment because the delay of each path changes frequently and the packets arrive more randomly. MSR has a smaller standard deviation than DSR. The outdoor throughput performance is worse than indoor for the same reason mention in above sub-section for outdoor delay and packet loss performances.



Indoor	DSR	MSR
Mean	156	142
σ	4.18	2.74

Outdoor	DSR	MSR
Mean	150	138
σ	3.54	2.74

Figure 4.18: FTP Performance of Scenario 6

4.7 Concluding Remark

From the measurement results above in static and mobile topologies, we can see that MSR can reduce the delay in the Ping application. MSR takes advantage of multiple paths and distributes traffic loads among them to reduce congestion in one single path. MSR has fewer throughputs than DSR in FTP application due to the out-of-order packet arrival.

In static topologies, both protocols have longer delay and lower throughputs in Scenario 2 and Scenario 3 than in Scenario 1 because the path in Scenario 2 and Scenario 3 between the source and destination has one more hop and one bottleneck node and subject to background traffic.

In mobile topologies, Scenario 6 has a longer delay, more packet loss, and lower throughput than Scenarios 4 and 5 because the mobility of three nodes has much more influence on the performance than the mobility of one node. Scenario 5 has the best performance since we simply move the source node toward the destination to make the signal receiving better.

In general, mobility scenarios have worse performance than static scenarios except that the static scenarios with heavy background traffic also have bad performance. The outdoor measurements are about the same as indoor in most cases. However, the outdoor has slightly less packet loss and slightly higher throughput than indoor except in Scenario 6 where the outdoor has worse performances.

Some things are not done or tested here due to the lack of equipments or limited time such as how to measure the interference in our testbed.

Chapter 5
Performance Evaluation of the Improved MSR

In this chapter, we present our testing results on the improved MSR in the same six scenarios as in last chapter and compare the performance with MSR. The improved MSR has been discussed in detail in Section 2.3. It employs an adaptive probing scheme to reduce the probing overhead. We set the delay change threshold to be 0.1 meaning that only if the delay change exceeds 10% of the old delay value the destination node will send the new delay information back to the source. Note that we only present two tables of Scenario 1. The rest are put in the Appendix E, where different scenarios can be found in different sub-appendices. Again each value is the average of 3 measurements. The standard deviation, defined to be $\sigma = \sqrt{\overline{X^2} - \overline{X}^2}$, is also provided here.

5.1 Scenario 1: Simple Multiple Paths

Table 5.1: Indoor measurements of Scenario 1

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	95% Confidence Interval	Number of Packets	Offered Load (bps)	Average Delay (ms)	95% Confidence Interval
1	64	1	100	512	10.040	1.671	100	512	9.301	1.135
2	64	0.1	1000	5120	8.281	0.143	1000	5120	11.455	2.640
3	64	0.01	1000	51200	7.883	0.221	1000	51200	12.591	0.126
4	512	0.01	1000	409600	12.337	0.615	1000	409600	14.476	2.554
5	1024	0.01	1000	819200	15.835	0.612	1000	819200	17.749	2.008
6	1024	-f	1000	1638400	17.699	0.422	1000	1638400	19.940	0.840

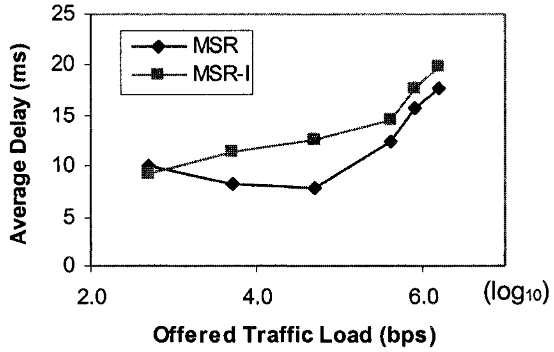


Figure 5.1: Indoor Delay Performance of Scenario 1

5.1.1 Ping Tests

Figure 5.1 shows the indoor delay performances. The performance curve of MSR is already discussed in last chapter since we use the same testing data of MSR.

The delay of the improved MSR increases with the increase of traffic load because the probing packets are greatly reduced by the adaptive probing mechanism and the more traffic load can lead to more congestion in network to cause more delay.

In the first run, MSR has a longer delay than the improved MSR due to the larger probing overhead percentage in the light traffic load. In the other cases, the improved MSR is worse than MSR in the delay performance. The percentage of the probing packets among all the packets of MSR becomes smaller and smaller with the increase of the traffic load. Meanwhile, the improved MSR requires more procession in each packet to monitor the delay changes and takes more and more time with the increase of the traffic load. Another reason is that the small change of delay in the improved MSR may not induce the delay update so that the load distribution according to the delay in the improved MSR may not be optimal in some cases. Thus the improved MSR does not improve delay performance in medium and heavy loads. We did not observe packet loss in this static topology.

Table 5.2: Outdoor measurements of Scenario 1

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	95% Confidence Interval	Number of Packets	Offered Load (bps)	Average Delay (ms)	95% Confidence Interval
1	64	1	100	512	9.983	1.443	100	512	8.940	0.316
2	64	0.1	1000	5120	8.269	0.446	1000	5120	10.350	0.474
3	64	0.01	1000	51200	7.953	0.364	1000	51200	11.825	0.800
4	512	0.01	1000	409600	12.297	0.426	1000	409600	15.257	0.464
5	1024	0.01	1000	819200	15.749	0.648	1000	819200	17.434	1.049
6	1024	-f	1000	1638400	17.523	0.501	1000	1638400	20.203	0.114

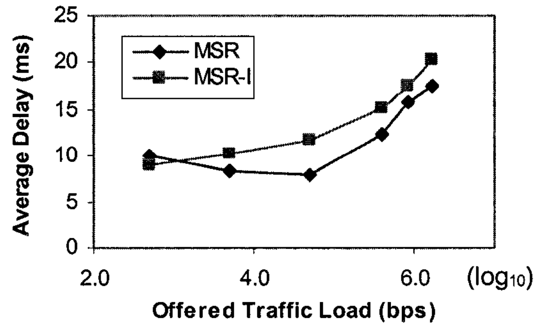
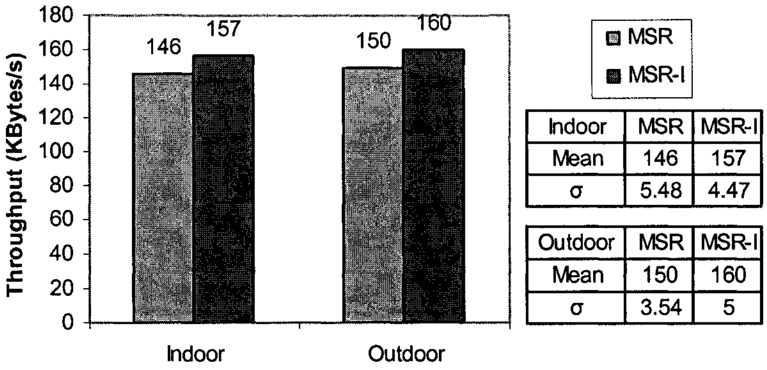


Figure 5.2: Outdoor Delay Performance of Scenario 1

Both MSR and the improved MSR have small 95% confidence intervals. MSR has smaller 95% confidence interval than the improved MSR except the 1st test run in which MSR has a large percentage of probing packets.

In outdoor experiments as shown in Figure 5.2, we have obtained performance similar to indoor tests since we employ the same topology and the outer interference does not appear to influence the performance much.

5.1.2 FTP Tests



Indoor	MSR	MSR-I
Mean	146	157
σ	5.48	4.47

Outdoor	MSR	MSR-I
Mean	150	160
σ	3.54	5

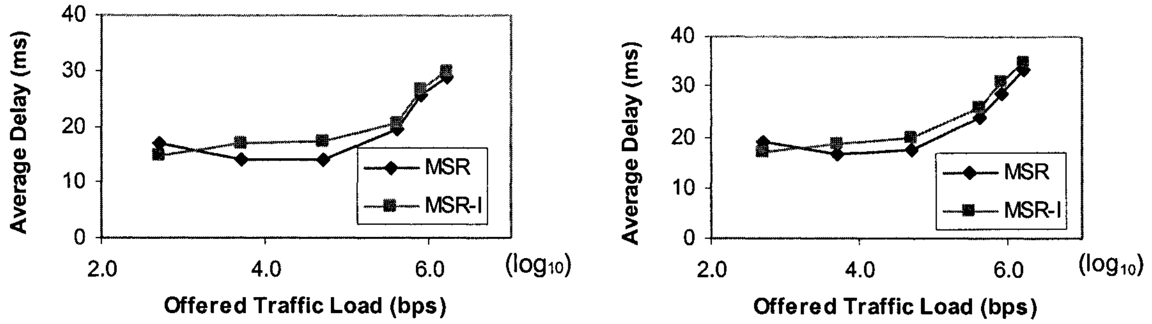
Figure 5.3: FTP Performances of Scenario 1

In the FTP application, the improved MSR has achieved a higher throughput than MSR due to the reduction of probing packets, which is shown in Figure 5.3. The FTP transferring is suspended when probing request packet is sent and started when the probing reply is received. Thus the probing packets can make the transmission discontinuous and bring more time in the transmission. Therefore, our improved MSR can improve the FTP performance or any other similar application which needs non-stopping transferring. However, the improved MSR is a little worse than DSR because we still use multiple paths here and cannot avoid the out-of-order packet arrival problems. The outdoor performance is very close to the indoor performance.

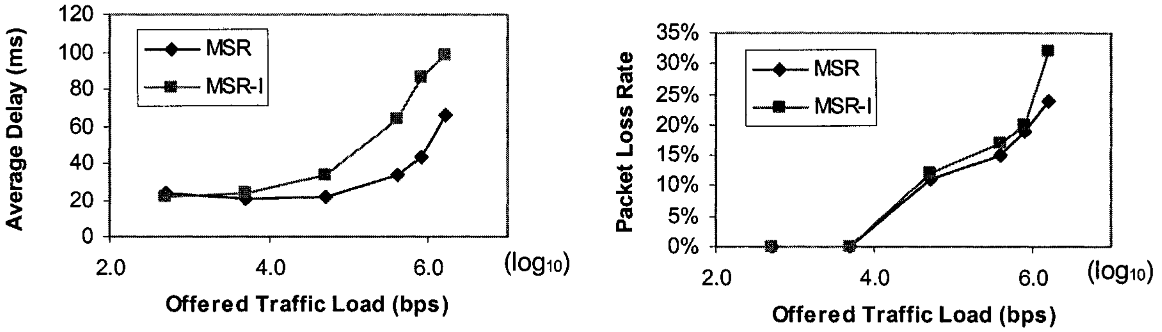
5.2 Scenario 2: Interference Test

5.2.1 Ping Tests

Figure 5.4a shows the indoor delay performance without background traffic. Similar observation is made for each of the MSR and the improved MSR delay performance curves as in Scenario1. However, the delay of each protocol in Scenario 2 is longer than that in Scenario 1 because there

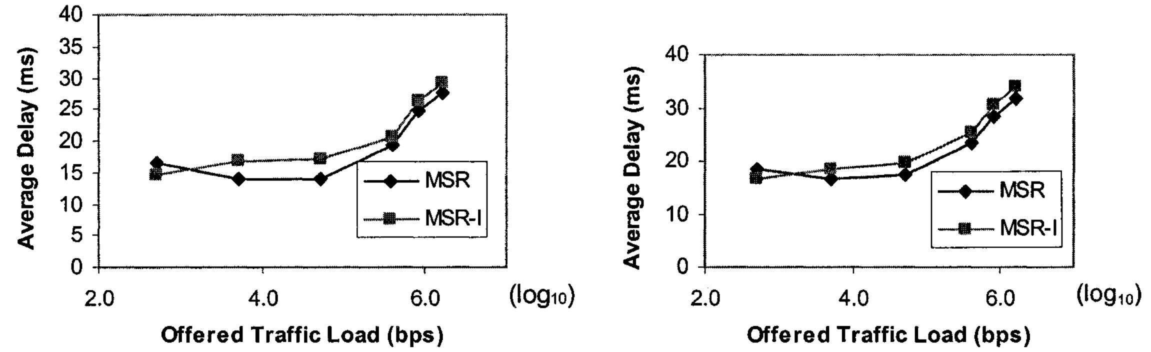


(a) No background traffic (b) With background traffic at 64 bytes/sec

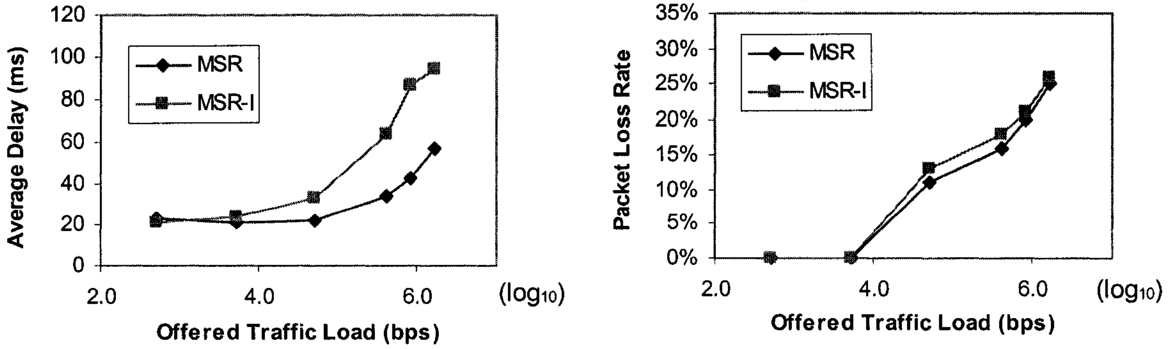


(c) With background traffic at 640 bytes/sec

Figure 5.4: Indoor Delay and Packet Loss Rate Performances of Scenario 2



(a) No background traffic (b) With background traffic at 64 bytes/sec



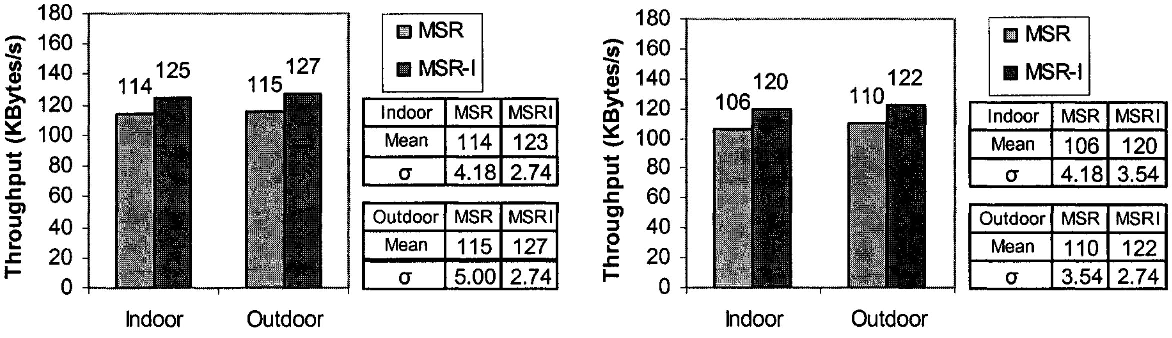
(c) With background traffic at 640 bytes/sec

Figure 5.5: Outdoor Delay and Packet Loss Rate Performances of Scenario 2

is one more hop containing the bottleneck node here. The comparison between MSR and the improved MSR is about the same as in Scenario 1.

Figure 5.4b and 5.4c show the delay and packet loss rate performances for background traffic at 64 bytes/sec and 640 bytes/sec respectively. The delay increases with the increase of background traffic, and the difference between MSR and the improved MSR increases too at high load.

We do not observe packet loss in Figure 5.4a without background traffic and 5.4b with light background traffic. However in Figure 5.4c with heavy background traffic, we observe packet loss of MSR and the improved MSR beyond medium load. The improved MSR loss rate is a little bit higher than MSR. It is because the improved MSR requires more processing in the source and destination, and the small change of delay in the improved MSR may not induce the delay update so that the load distribution according to the delay in the improved MSR may not be optimal in some cases.



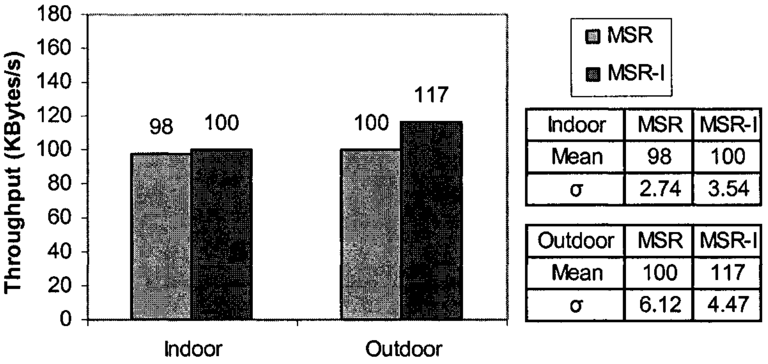
Indoor	MSR	MSRI
Mean	114	123
σ	4.18	2.74

Outdoor	MSR	MSRI
Mean	115	127
σ	5.00	2.74

Indoor	MSR	MSRI
Mean	106	120
σ	4.18	3.54

Outdoor	MSR	MSRI
Mean	110	122
σ	3.54	2.74

(a) No background traffic (b) With background traffic at 64 bytes/sec



Indoor	MSR	MSR-I
Mean	98	100
σ	2.74	3.54

Outdoor	MSR	MSR-I
Mean	100	117
σ	6.12	4.47

(c) With background traffic at 640 bytes/sec

Figure 5.6: FTP Performances of Scenario 2

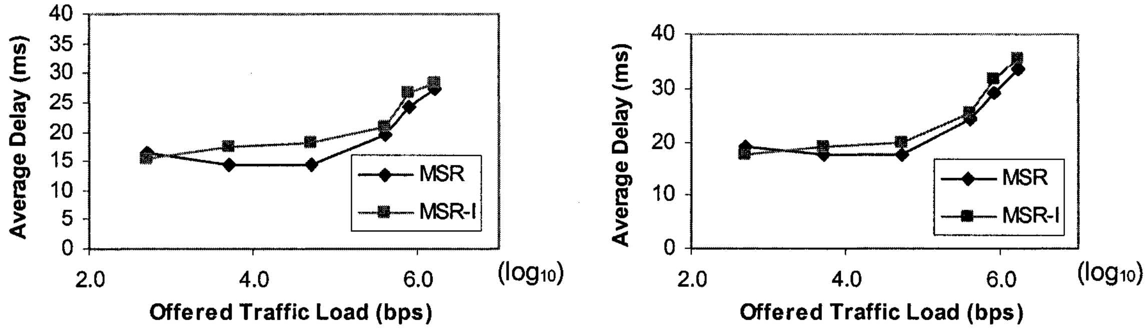
In outdoor experiments, we obtained similar performance as shown in Figure 5.5 as indoor tests except that the packet loss in outdoor is a little bit smaller than that in indoor as the outdoor environment has less interference than indoor.

5.2.2 FTP Tests

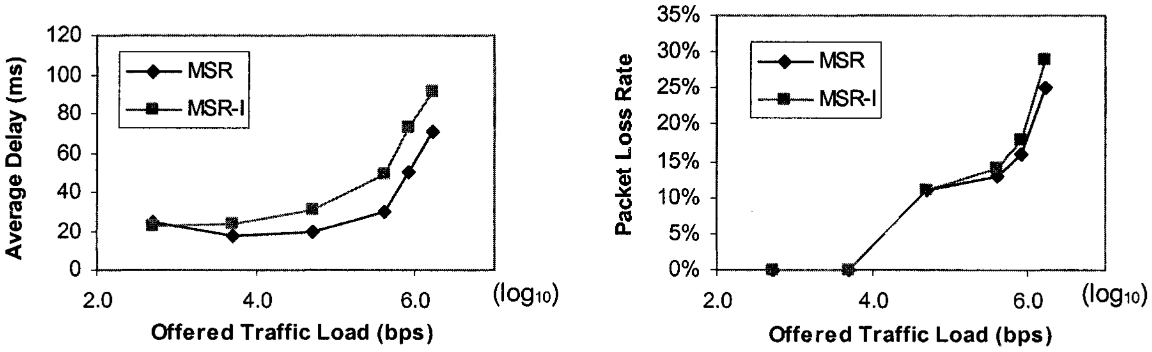
The throughputs of both MSR and the improved MSR decrease with the increase of background traffic as shown in Figure 5.6. The improved MSR outperforms MSR in FTP application. The reduction of probing packets can provide more smooth and continuous transferring and hence improve the performance of FTP file transferring. The standard deviation of the improved MSR is smaller than MSR.

5.3 Scenario 3: Bottleneck Test

5.3.1 Ping Tests



(a) No background traffic (b) With background traffic at 64 bytes/sec

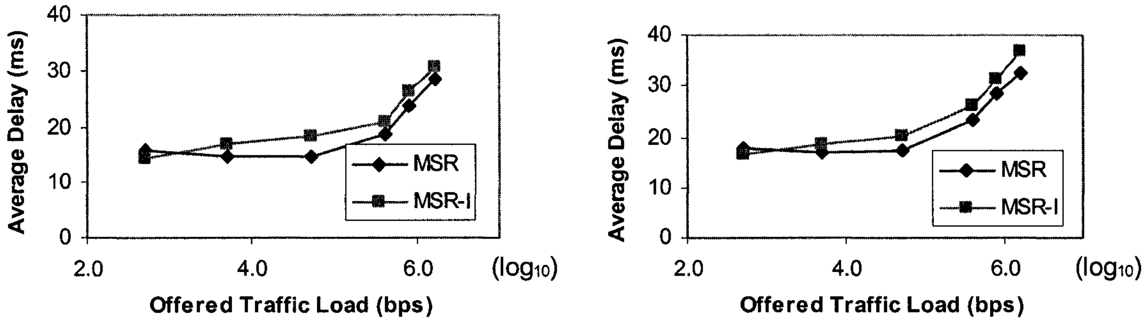


(c) With background traffic at 640 bytes/sec

Figure 5.7: Indoor Delay and Packet Loss Rate Performances of Scenario 3

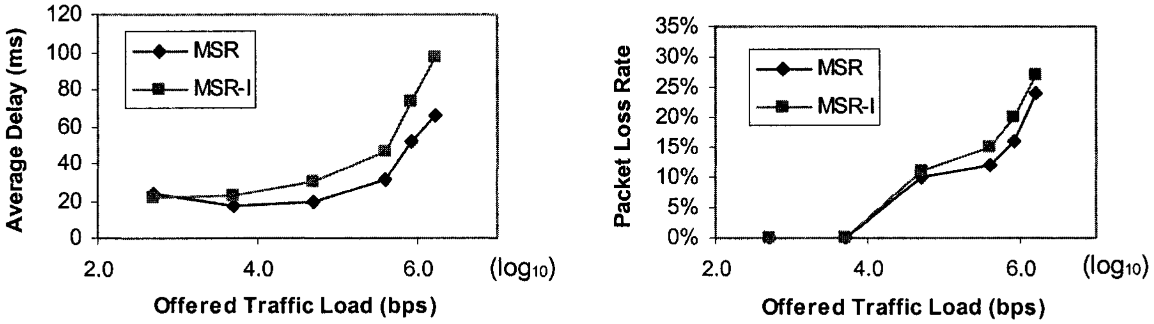
Figure 5.7 shows the indoor delay and packet loss rate performances for different background traffic. Similar observation is made for each of the MSR and the improved MSR delay performance curves as Scenario 2. The comparison between MSR and the improved MSR is also about the same as in Scenario 2.

We do not observe packet loss in Figure 5.7a without background traffic and 5.7b with light background traffic. However in Figure 5.7c with heavy background traffic, we observe packet loss of MSR and the improved MSR beyond medium load. The loss rate of the improved MSR is a little bit higher than MSR.



(a) No background traffic

(b) With background traffic at 64 bytes/sec



(c) With background traffic at 640 bytes/sec

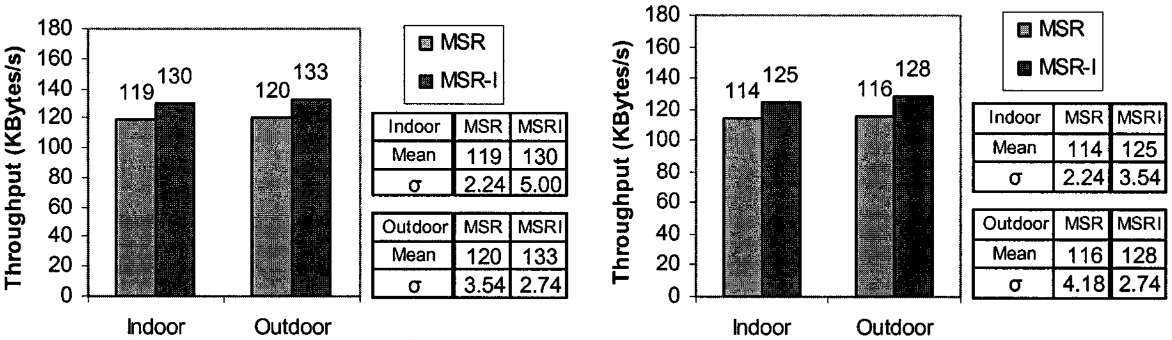
Figure 5.8: Outdoor Delay and Packet Loss Rate Performances of Scenario 3

In outdoor experiments from the Figure 5.8, we have obtained similar performance to indoor tests. The only difference is that the outdoor has slightly less packet loss than indoor for each protocol, which shows that the outdoor environment is slightly better than indoor environment for this scenario.

5.3.2 FTP Tests

The throughputs of both MSR and the improved MSR decrease with the increase of background traffic as shown in Figure 5.9. The improved MSR outperforms MSR in FTP application for the same reason as in previous scenarios. The reduction of probing packets can provide more smooth and continuous transferring and hence improve the performance of FTP file transferring. Unlike

Scenario 2, the standard deviation of the improved MSR is smaller than MSR in most of cases here.



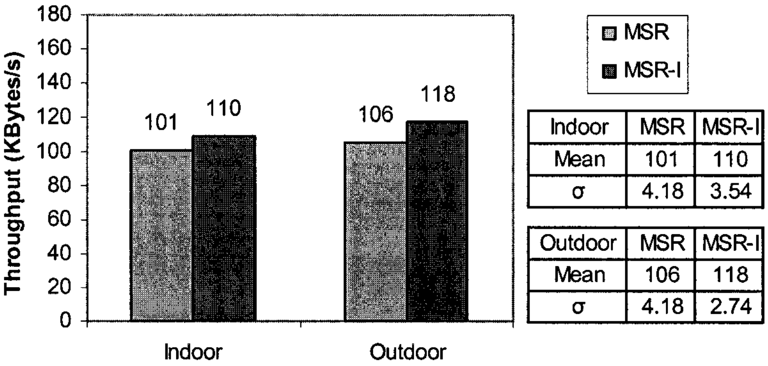
Indoor	MSR	MSRI
Mean	119	130
σ	2.24	5.00

Outdoor	MSR	MSRI
Mean	120	133
σ	3.54	2.74

Indoor	MSR	MSRI
Mean	114	125
σ	2.24	3.54

Outdoor	MSR	MSRI
Mean	116	128
σ	4.18	2.74

(a) No background traffic (b) With background traffic at 64 bytes/sec



Indoor	MSR	MSR-I
Mean	101	110
σ	4.18	3.54

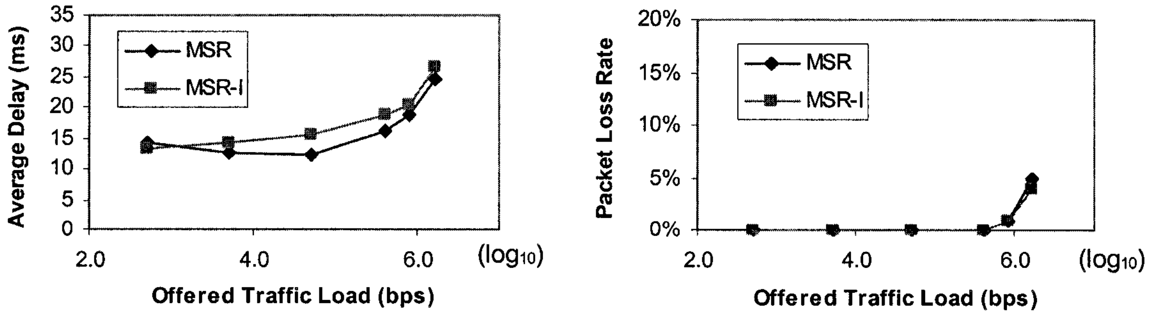
Outdoor	MSR	MSR-I
Mean	106	118
σ	4.18	2.74

(c) With background traffic at 640 bytes/sec

Figure 5.9: FTP Performances of Scenario 3

5.4 Scenario 4: Back and Forth

5.4.1 Ping Tests

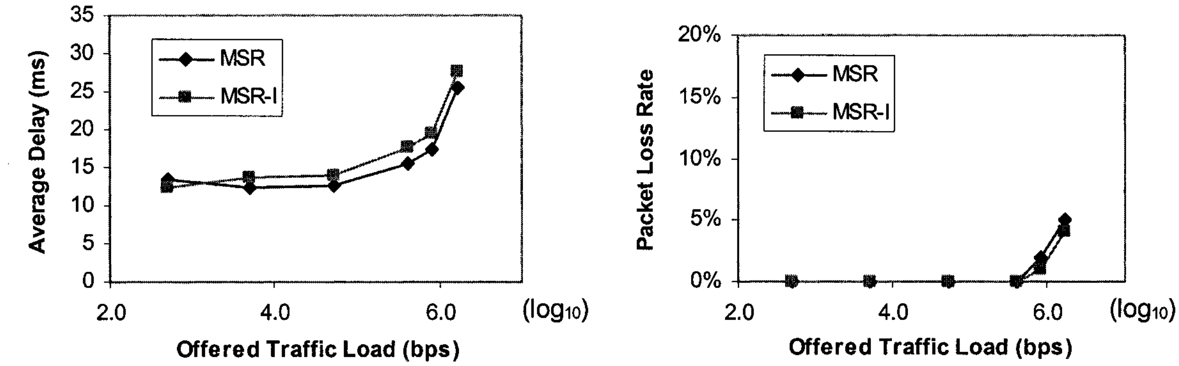


(a) Delay (b) Packet Loss Rate

Figure 5.10: Indoor Delay and Packet Loss Rate Performances of Scenario 4

Figure 5.10 shows the indoor delay and packet loss rate performances. Similar observation is made for each of the MSR and the improved MSR delay performance curves. In medium and heavy loads, the improved MSR has longer delay than MSR for the same reason in Scenario 1.

Both protocols have packet loss in heavy load. The improved MSR has a little bit less packet loss than MSR. The probing packets worsen the performance in mobile environment. The more control packets in the mobile environment, the worse the performance is. Therefore the improved MSR has less packet loss than MSR.

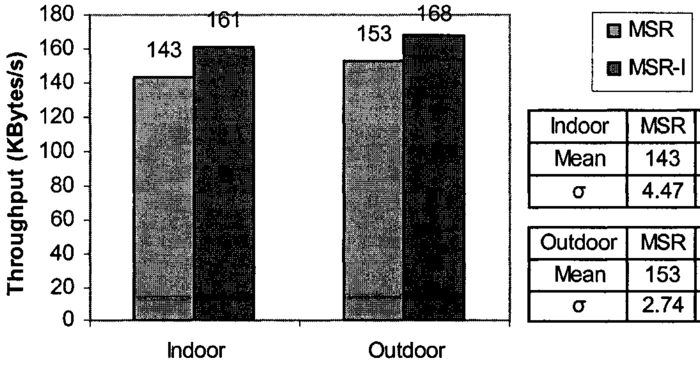


(a) Delay (b) Packet Loss Rate

Figure 5.11: Outdoor Delay and Packet Loss Rate Performances of Scenario 4

Figure 5.11 depicts the outdoor delay and packet loss performances. In outdoor experiments, we obtained performance similar to indoor tests since we employ the same topology and the outer interference does not seem to influence the performance much.

5.4.2 FTP Tests



Indoor	MSR	MSR-I
Mean	143	161
σ	4.47	4.18

Outdoor	MSR	MSR-I
Mean	153	168
σ	2.74	5.7

Figure 5.12: FTP Performance of Scenario 4

Figure 5.12 shows the throughput performances. In the mobile environment, the improved MSR outperforms MSR with larger throughput. The control packets worsen the performance a lot in the mobile environment due to the unstable links. The reduction of control packets can provide more smooth and continuous data transfer and improve the performances.

5.5 Scenario 5: Varying S-D Distance

5.5.1 Ping Tests

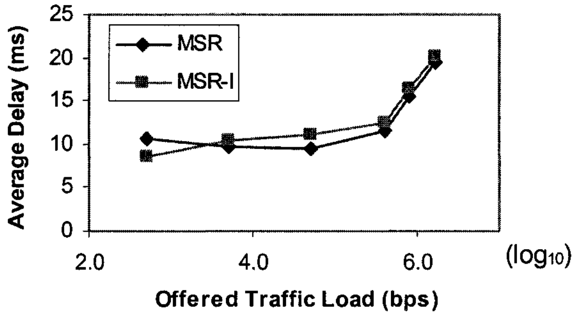


Figure 5.13: Indoor Delay Performance of Scenario 5

Figure 5.13 shows the indoor delay performances. Similar observation is made for each of the MSR and the improved MSR delay performance curves as in Scenario 4. The comparison between MSR and the improved MSR is about the same as in Scenario 4 except that they both have less delay here.

We did not find any packet loss from our measurements.

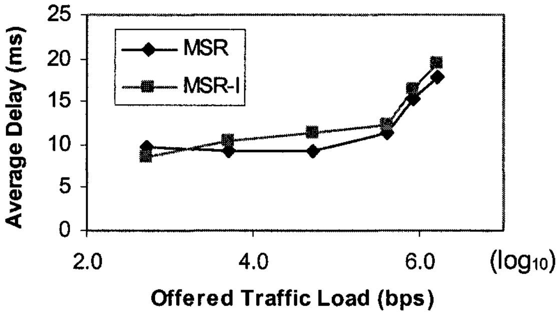
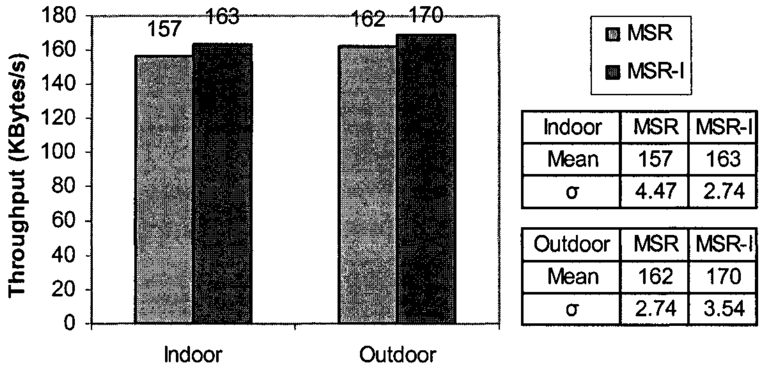


Figure 5.14: Outdoor Delay Performance of Scenario 5

In outdoor experiments, we obtained similar performance shown in Figure 5.14 as indoor tests since we employ the same topology and the outer interference does not appear to influence the performance much.

5.5.2 FTP Tests



Indoor	MSR	MSR-I
Mean	157	163
σ	4.47	2.74

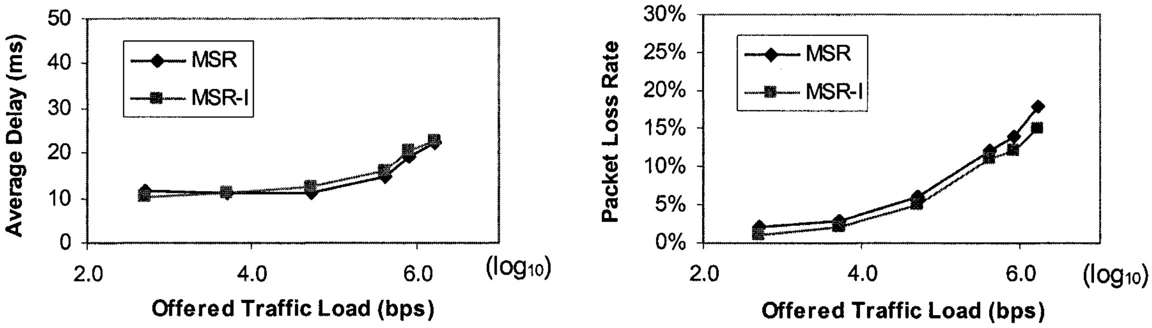
Outdoor	MSR	MSR-I
Mean	162	170
σ	2.74	3.54

Figure 5.15: FTP Performance of Scenario 5

Figure 5.15 shows the throughput performance of Scenario 5. In the mobile environment, the improved MSR outperforms MSR with larger throughput. The control packets worsen the performance a lot in the mobile environment due to the unstable links. The reduction of control packets can provide more smooth and continuous data transfers and improve the performances.

5.6 Scenario 6: Circular Interference

5.6.1 Ping Tests

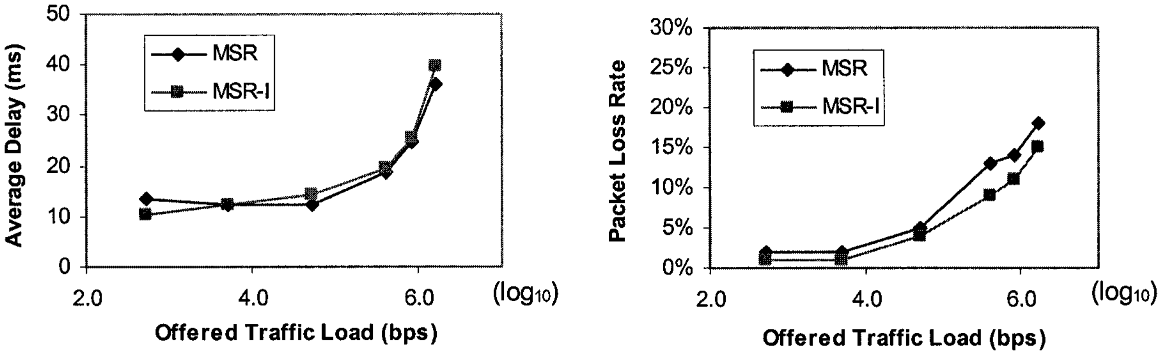


(a) Delay (b) Packet Loss Rate

Figure 5.16: Indoor Delay and Packet Loss Rate Performances of Scenario 6

Figure 5.16 shows the indoor delay and packet loss rate performances. Similar observation is made for each of the MSR and the improved MSR delay performance curves in Figure 5.16a. The comparison between MSR and the improved MSR is about the same as in Scenario 5.

Both protocols have packet loss in all loads. The improved MSR has less packet loss than MSR because it has reduced the number of probing packets responsible for poor performance due to the unstable paths.

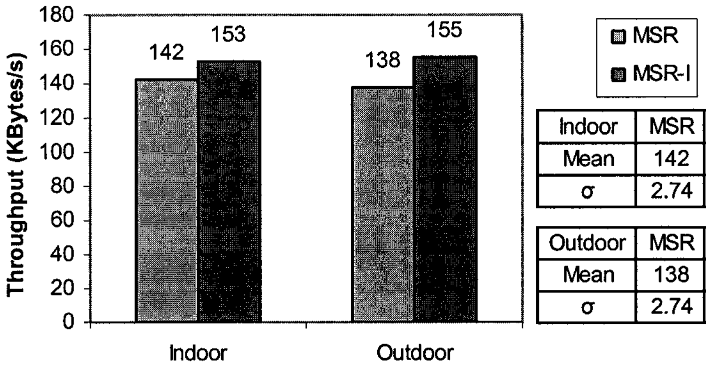


(a) Delay (b) Packet Loss Rate

Figure 5.17: Outdoor Delay and Packet Loss Rate Performances of Scenario 6

The outdoor performances are shown in Figure 5.17. In outdoor experiments, we obtained similar performance as indoor tests since we employ the same topology and the outer interference does not seem to influence the performance much.

5.6.2 FTP Tests



Indoor	MSR	MSR-I
Mean	142	153
σ	2.74	4.47

Outdoor	MSR	MSR-I
Mean	138	155
σ	2.74	3.54

Figure 5.18: FTP Performance of Scenario 6

Figure 5.18 shows the throughput performance. In the mobile environment, the improved MSR has a higher throughput than MSR. The control packets are responsible for deteriorating performance due to the unstable links. Thus the reduction of control packets can provide more smooth and continuous data transfer and improve the performances.

5.7 Concluding Remark

The improved MSR in general has a longer delay than MSR because the improved MSR requires more calculation on the delay change in the destination and the load distribution in the improved MSR may not be optimal. The improved MSR has less packet loss rate than MSR in mobile environment because the probing packets cause more packet loss in the unstable links and worsen the performance of MSR. The improved MSR has more throughput than MSR because the improved MSR can significantly reduce the number of probing packets responsible for interrupting file transfer process. The throughput of the improved MSR has been improved and is as good as DSR.

In static topologies, both protocols have longer delays and lower throughputs in Scenario 2 and Scenario 3 than in Scenario 1 because the path in Scenario 2 and Scenario 3 between the source and destination has one more hop and one bottleneck node and there exists background traffic.

In mobile topologies, Scenario 6 has a longer delay, more packet loss, and lower throughput than Scenarios 4 and 5 because the mobility of three nodes has much more influence on the performance than the mobility of one node. Scenario 5 has the best performance since we simply move the source node toward the destination to make the signal receiving better.

In general, mobility scenarios have worse performance than static scenarios except that the static scenarios with heavy background traffic also have bad performance. The outdoor measurements are about the same as indoor in most cases. However, the outdoor has slightly less packet loss and slightly higher throughput than indoor.

Chapter 6
Performance Evaluation of BSR

In this chapter, we present what we have measured for BSR in the testbed and compare the results with DSR to evaluate the performance. The details of BSR have been provided in Section 2.4. We conducted the tests in static and mobile topologies to see the effect caused by different topologies. We perform our tests in indoor and outdoor environments for each topology. The applications we test include Ping and FTP. Note that we only present two tables of Scenario 1. The rest are put in the Appendix E, where different scenarios can be found in different sub-appendices. Each value is the average of 3 measurements. The standard deviation, defined to be $\sigma = \sqrt{\overline{X^2} - \overline{X}^2}$, is also provided here.

6.1 Scenario 1: Simple Multiple Paths

Table 6.1: Indoor measurements of Scenario 1

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	95% Confidence Interval	Number of Packets	Offered Load (bps)	Average Delay (ms)	95% Confidence Interval
1	64	1	100	512	8.870	0.835	100	512	8.053	1.515
2	64	0.1	1000	5120	11.435	2.514	1000	5120	11.363	0.503
3	64	0.01	1000	51200	12.623	2.026	1000	51200	11.810	0.967
4	512	0.01	1000	409600	16.795	1.892	1000	409600	15.858	0.597
5	1024	0.01	1000	819200	19.570	1.346	1000	819200	17.667	3.013
6	1024	-f	1000	1638400	20.806	0.944	1000	1638400	19.626	0.375

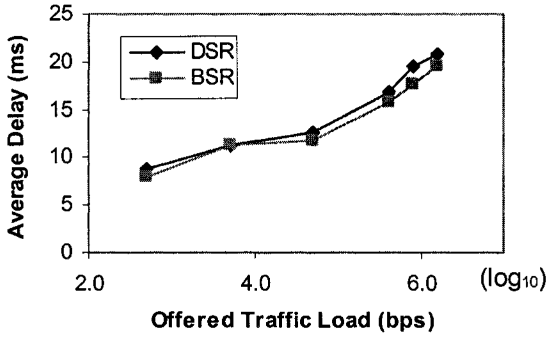


Figure 6.1: Indoor Delay Performance of Scenario 1

6.1.1 Ping Tests

Figure 6.1 shows the indoor delay performances. The performance curve of DSR is already discussed in Chapter 4 since we use the same testing data of DSR. The delay of BSR increases with the increase of traffic load for the same reason as DSR.

BSR has a little bit shorter delay than DSR because BSR can improve the performance by the use of backup path. We did not measure any packet loss in this static topology.

Both protocols have a small 95% confidence interval. BSR has a smaller 95% confidence interval than DSR in most cases except test run 1 and 5.

Table 6.2: Outdoor measurements of Scenario 1

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	95% Confidence Interval	Number of Packets	Offered Load (bps)	Average Delay (ms)	95% Confidence Interval
1	64	1	100	512	8.640	0.506	100	512	8.317	0.385
2	64	0.1	1000	5120	11.421	0.821	1000	5120	11.096	0.408
3	64	0.01	1000	51200	12.456	0.954	1000	51200	11.786	0.215
4	512	0.01	1000	409600	16.343	0.779	1000	409600	15.815	0.288
5	1024	0.01	1000	819200	19.006	0.722	1000	819200	18.003	1.400
6	1024	-f	1000	1638400	20.506	1.063	1000	1638400	20.037	0.740

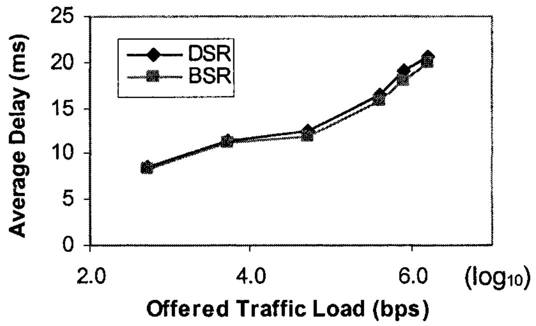
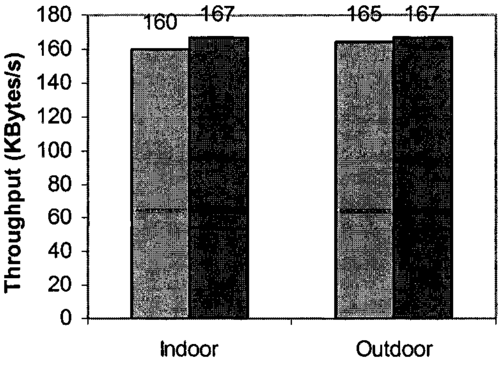


Figure 6.2: Outdoor Delay Performance of Scenario 1



▨ DSR
■ BSR

Indoor	DSR	BSR
Mean	160	167
σ	6.12	4.47

Outdoor	DSR	BSR
Mean	165	167
σ	5	4.47

Figure 6.3: FTP Performance of Scenario 1

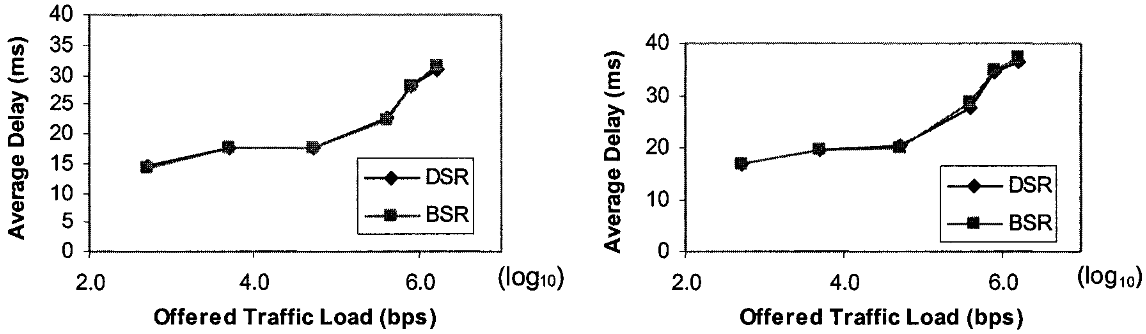
In outdoor experiments, we measured performance similar to indoor tests since we employ the same topology and the outer interference does not seem to influence the performance much.

6.1.2 FTP Tests

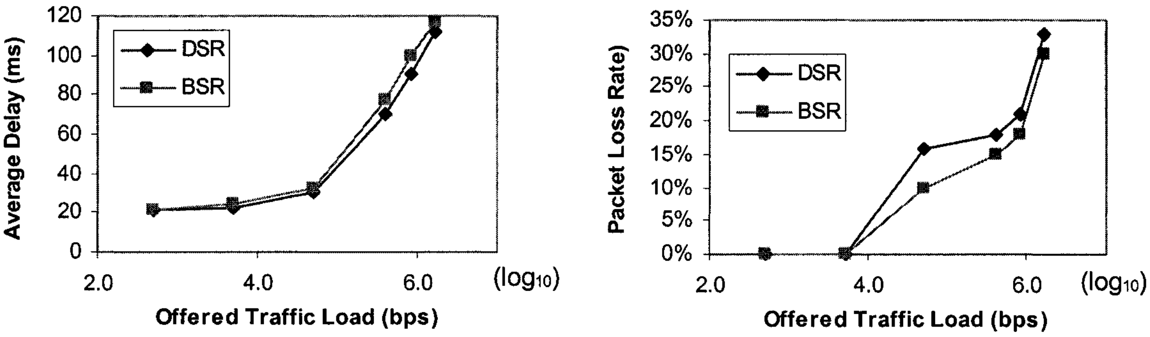
Figure 6.3 depicts the FTP performance of Scenario 1. BSR has more throughput than DSR. The packet transfer is more reliable in BSR and the FTP file transferring is more smooth and continuous which can induce improvement in throughput performance. Furthermore BSR has a smaller standard deviation than DSR. The performances of indoor and outdoor environments are very similar in this scenario.

6.2 Scenario 2: Interference Test

6.2.1 Ping Tests



(a) No background traffic (b) With background traffic at 64 bytes/sec

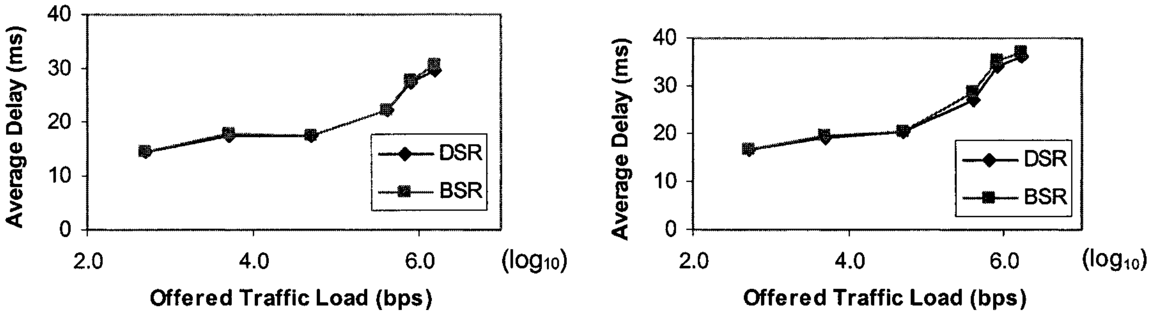


(c) With background traffic at 640 bytes/sec

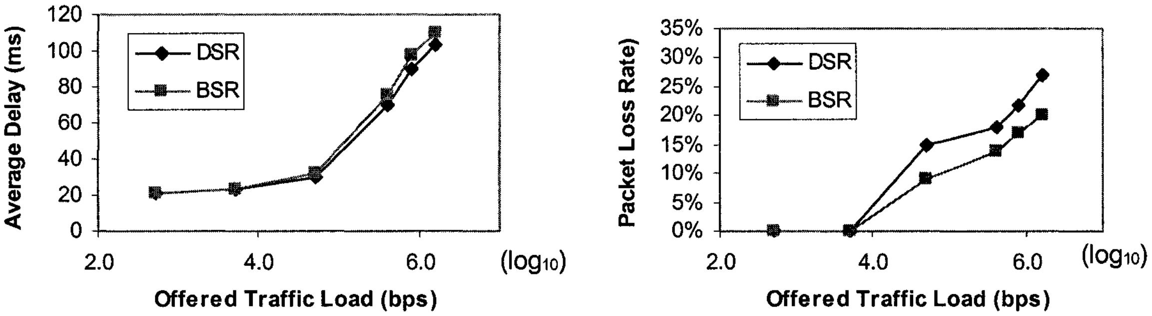
Figure 6.4: Indoor Delay and Packet Loss Rate Performances of Scenario 2

Figure 6.4a show the indoor delay performance without background traffic while Figure 6.4b and 6.4c depict the performances for background traffic 64 bytes/sec and 640 bytes/sec respectively. The delay increases for each protocol with the increasing of background traffic, so does the difference between two protocols. BSR has almost the same delay as DSR in Figure 6.4a without background traffic and 6.4b with light background traffic load which can be attributed to the bottleneck node. BSR has slightly longer delay than DSR in Figure 6.4c with heavy background traffic because BSR requires more processing on determining the backup path and the packets recovered by the backup path will enlarge the average delay.

We do not observe packet loss in Figure 6.4a without background traffic and 6.4b with light background traffic. However in Figure 6.4c with heavy background traffic, we observe packet loss of DSR and BSR beyond light load. BSR has less packet loss than DSR since the backup path increases the communication reliability.



(a) No background traffic (b) With background traffic at 64 bytes/sec



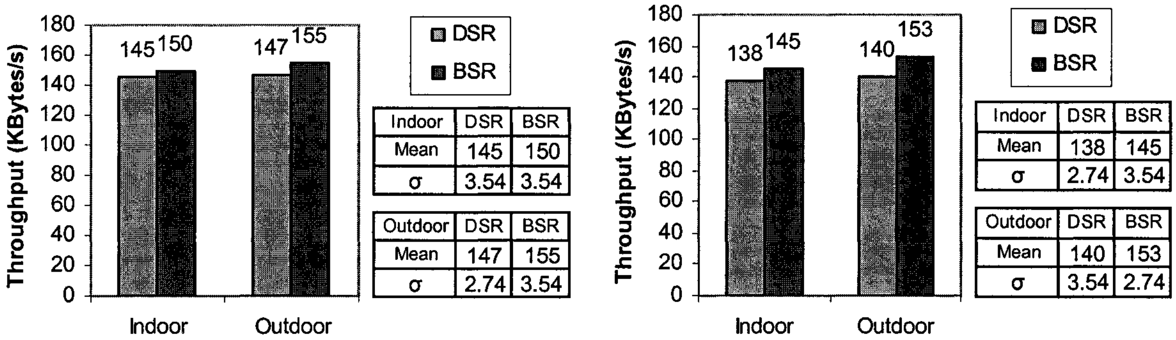
(c) With background traffic at 640 bytes/sec

Figure 6.5: Outdoor Delay and Packet Loss Rate Performances of Scenario 2

In Figure 6.5, we obtained performance similar to indoor tests except that the packet loss rate in outdoor is a little bit lower than indoor. It shows that the outdoor environment has less interference than indoor.

6.2.2 FTP Tests

As shown in Figure 6.6, the throughputs of both DSR and BSR decrease with the increase of background traffic. BSR is better than DSR for the same reason in Scenario 1.The recovery scheme through the backup path by BSR can improve the performance a lot. The outdoor throughput is a little bit larger than the indoor throughput, which shows that the outdoor environment has less interference than indoor.



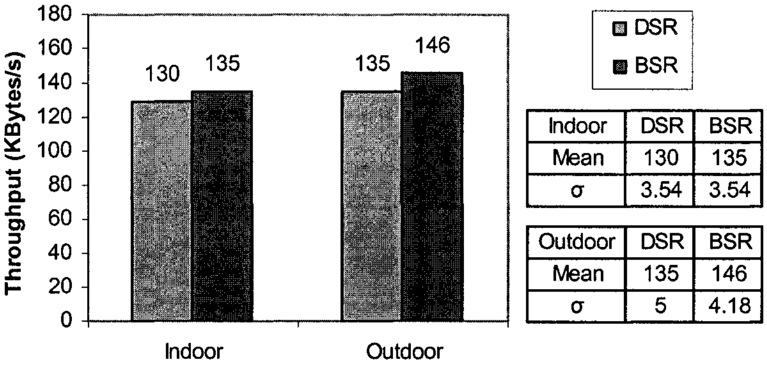
Indoor	DSR	BSR
Mean	145	150
σ	3.54	3.54

Outdoor	DSR	BSR
Mean	147	155
σ	2.74	3.54

Indoor	DSR	BSR
Mean	138	145
σ	2.74	3.54

Outdoor	DSR	BSR
Mean	140	153
σ	3.54	2.74

(a) No background traffic (b) With background traffic at 64 bytes/sec



Indoor	DSR	BSR
Mean	130	135
σ	3.54	3.54

Outdoor	DSR	BSR
Mean	135	146
σ	5	4.18

(c) With background traffic at 640 bytes/sec
Figure 6.6: FTP Performances of Scenario 2

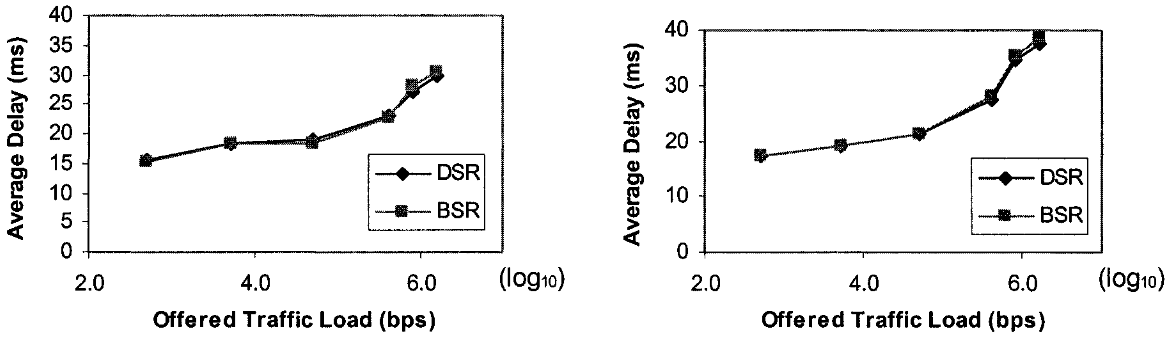
6.3 Scenario 3: Bottleneck Test

6.3.1 Ping Tests

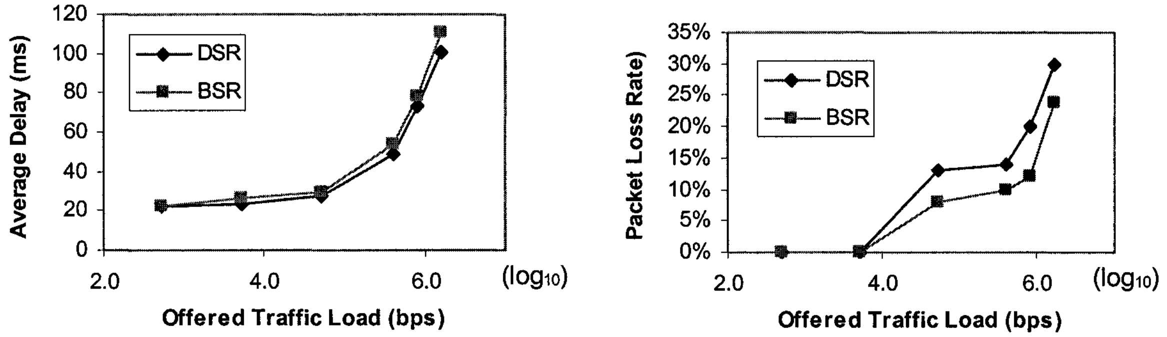
Figure 6.7 shows the indoor delay and packet loss rate performances for different background traffic. Similar observation is made for each of the DSR and BSR delay performance curves like Scenario 2. The comparison between BSR and DSR is about the same as in Scenario 2.

We do not observe packet loss in Figure 6.7a without background traffic and 6.7b with light background traffic. However in Figure 6.7c with heavy background traffic, we observe packet loss of DSR and BSR beyond light load. BSR has less packet loss than DSR like in Scenario 2.

In Figure 6.8, we obtained performance similar to indoor tests since we employ the same topology and the outer interference does not seem to influence the performance a lot.

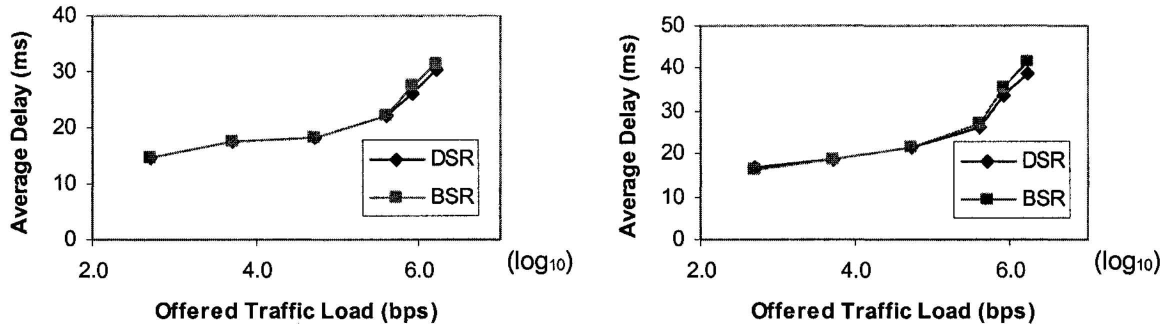


(a) No background traffic (b) With background traffic at 64 bytes/sec

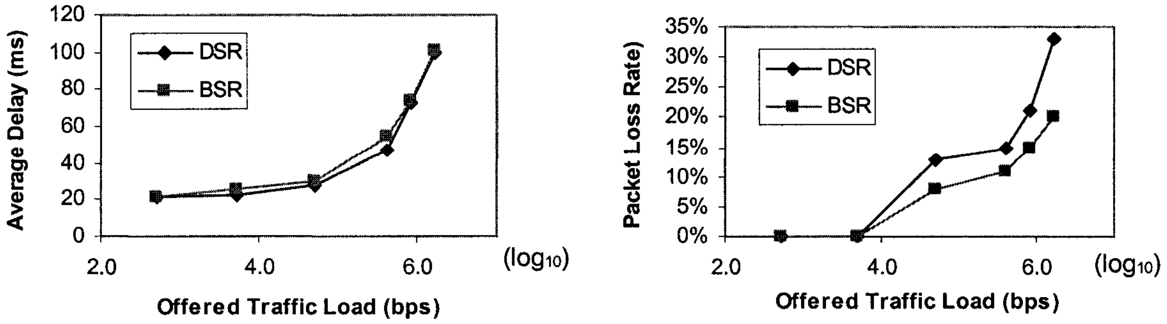


(c) With background traffic at 640 bytes/sec

Figure 6.7: Indoor Delay and Packet Loss Rate Performances of Scenario 3



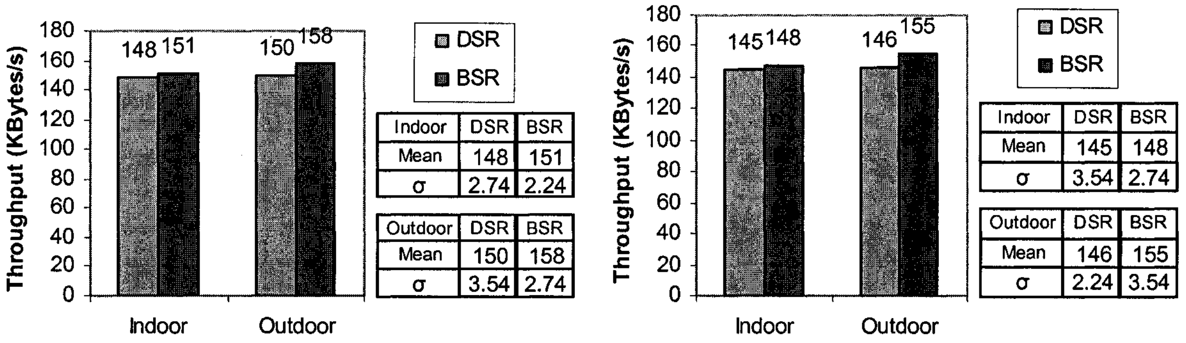
(a) No background traffic (b) With background traffic at 64 bytes/sec



(c) With background traffic at 640 bytes/sec

Figure 6.8: Outdoor Delay and Packet Loss Rate Performances of Scenario 3

6.3.2 FTP Tests



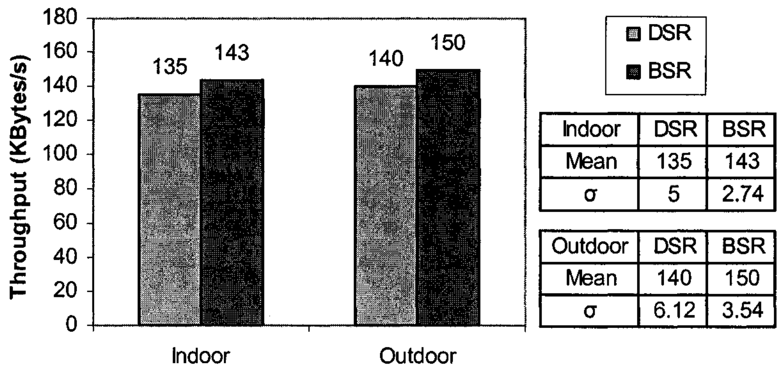
Indoor	DSR	BSR
Mean	148	151
σ	2.74	2.24

Outdoor	DSR	BSR
Mean	150	158
σ	3.54	2.74

Indoor	DSR	BSR
Mean	145	148
σ	3.54	2.74

Outdoor	DSR	BSR
Mean	146	155
σ	2.24	3.54

(a) No background traffic (b) With background traffic at 64 bytes/sec



Indoor	DSR	BSR
Mean	135	143
σ	5	2.74

Outdoor	DSR	BSR
Mean	140	150
σ	6.12	3.54

(c) With background traffic at 640 bytes/sec

Figure 6.9: FTP Performances of Scenario 3

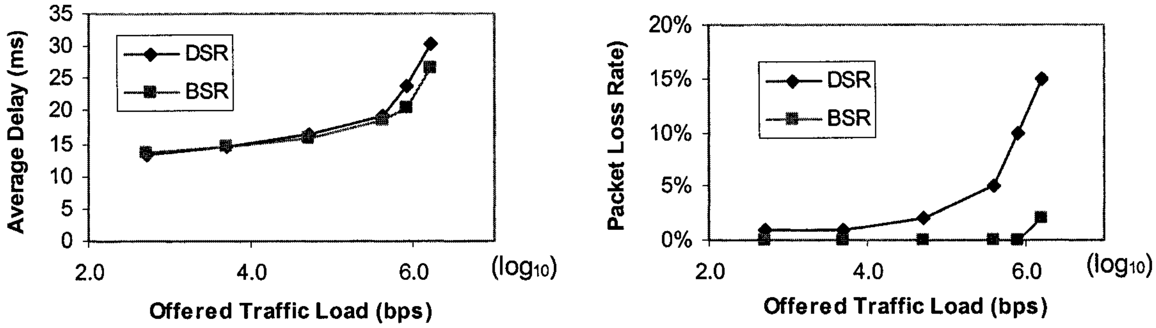
As shown in Figure 6.9, the throughputs of both DSR and BSR decrease with the increase of background traffic. Similar performances are observed here as in scenario 2. The more reliable transmission in BSR contributes a larger throughput. Furthermore, BSR has a smaller standard deviation than DSR.

6.4 Scenario 4: Back and Forth

6.4.1 Ping Tests

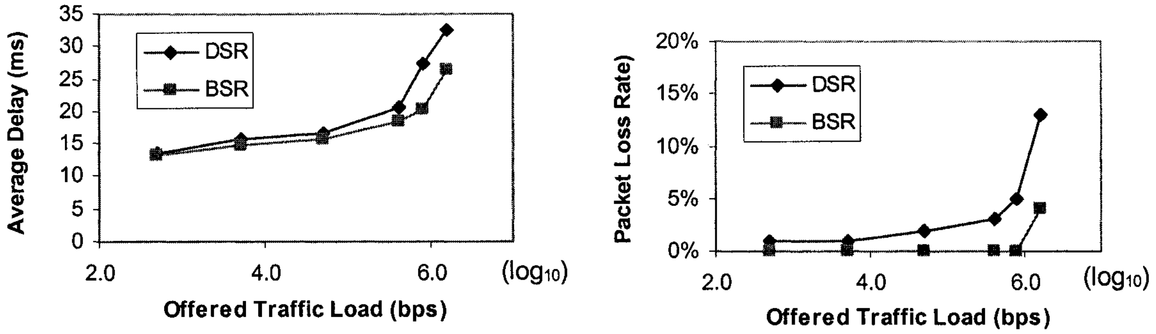
Figure 6.10 shows the indoor delay and packet loss rate performances of Scenario 4. Similar observation is made for each of the DSR and BSR delay performance curves like Scenario 1. Here BSR has a little smaller delay than DSR because BSR can switch the traffic to the backup path in case that the primary path fails.

DSR has packet loss in all loads and BSR only has packet loss in heavy load. BSR has less packet loss than DSR due to the reliable communication by using backup path.



(a) Delay (b) Packet Loss Rate

Figure 6.10: Indoor Delay and Packet Loss Rate Performances of Scenario 4

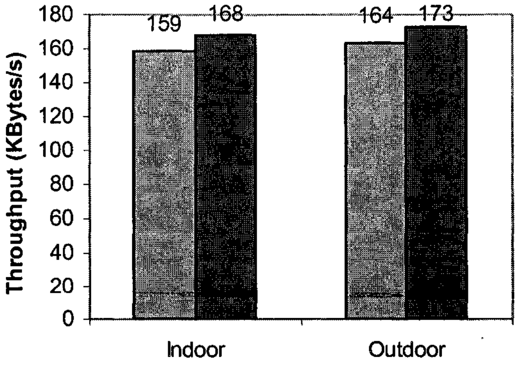


(a) Delay (b) Packet Loss Rate

Figure 6.11: Outdoor Delay and Packet Loss Rate Performances of Scenario 4

In Figure 6.11, similar performance to indoor testing is obtained. The influence by outdoor environment is not obvious in our measurements.

6.4.2 FTP Tests



Indoor	DSR	BSR
Mean	159	168
σ	5.48	2.74

Outdoor	DSR	BSR
Mean	164	173
σ	4.18	2.74

Figure 6.12: FTP Performance of Scenario 4

Figure 6.12 depicts the throughput performance for indoor and outdoor testing. BSR outperforms DSR with larger throughput because BSR may use the backup path if the primary path fails and provides more reliable communication. Furthermore, BSR has a smaller standard deviation than DSR. The outdoor performance is slightly higher than indoor performance since the outdoor has less interference than indoor.

6.5 Scenario 5: Varying S-D Distance

6.5.1 Ping Tests

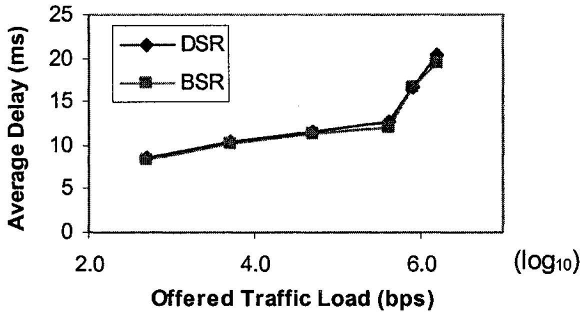


Figure 6.13: Indoor Delay Performance of Scenario 5

Figure 6.13 shows the indoor delay performance. Similar observation is made for each of the DSR and BSR delay performance curves like Scenario 4.

BSR only has slightly shorter delay than DSR. The movement of the source does not influence the performance much since the links between the source and the intermediate nodes do not fail. Thus BSR does not improve the performance much in this case.

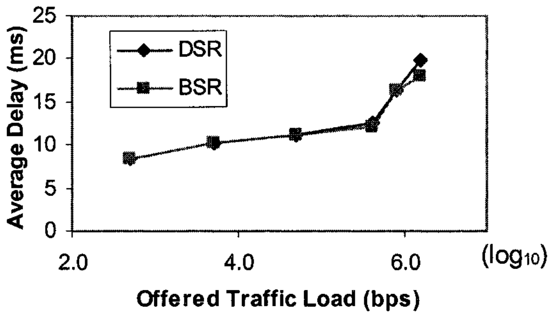
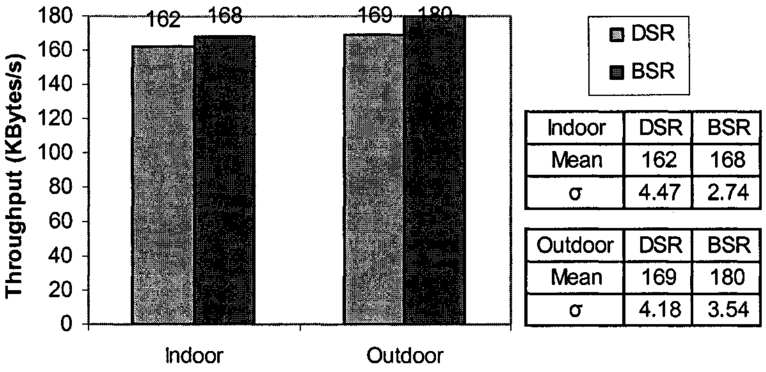


Figure 6.14: Outdoor Delay Performance of Scenario 4

The result of outdoor experiments shown in Figure 6.14 is very similar to that of indoor testing. In this scenario, we did not observe any obvious difference between indoor and outdoor environment.

6.5.2 FTP Tests



Indoor	DSR	BSR
Mean	162	168
σ	4.47	2.74

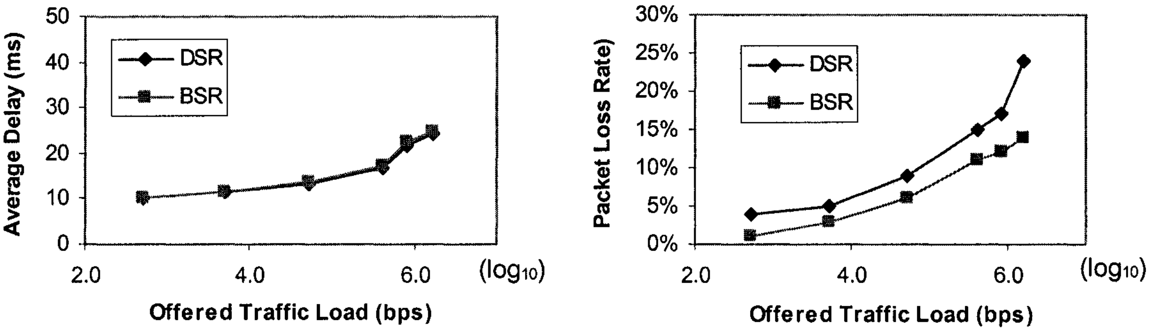
Outdoor	DSR	BSR
Mean	169	180
σ	4.18	3.54

Figure 6.15: FTP Performance of Scenario 5

As seen from Figure 6.15, BSR outperforms DSR with larger throughput because BSR may use the backup path if the primary path fails and provides more reliable communication. BSR also has a smaller standard deviation than DSR here. The outdoor performance of each protocol is higher than the indoor performance since outdoor environment has less interference than indoor.

6.6 Scenario 6: Circular Interference

6.6.1 Ping Tests

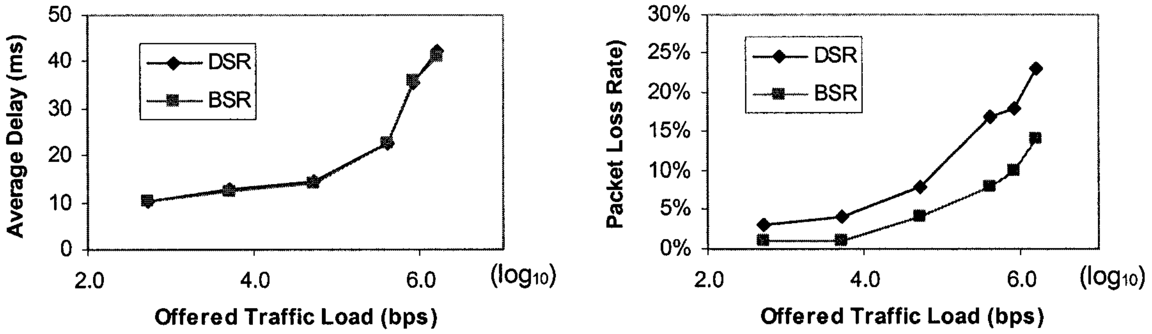


(a) Delay (b) Packet Loss Rate

Figure 6.16: Indoor Delay and Packet Loss Rate Performances of Scenario 6

Figure 6.16 shows the indoor delay and packet loss rate performances. Both protocols have increasing delay curves like before. BSR has slightly longer delay than DSR because BSR requires more processing on determining the backup path and the packets recovered by the backup path will enlarge the average delay.

Both protocols have packet loss in all loads. BSR has less packet loss than DSR due to the reliable communication between source and destination.

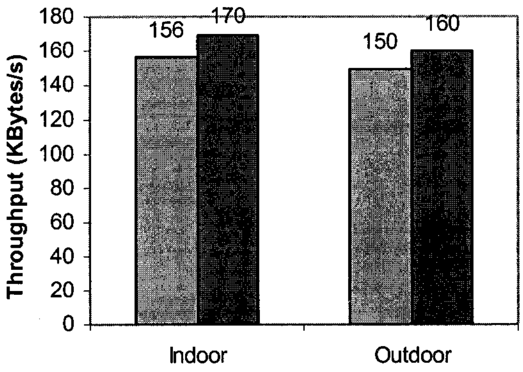


(a) Delay (b) Packet Loss Rate

Figure 6.17: Outdoor Delay and Packet Loss Rate Performances of Scenario 6

The outdoor measurement results are depicted in Figure 6.17. Both protocols have longer delay in outdoor than indoor for the same reason in Section 4.6.1.

6.6.2 FTP Tests



Indoor	DSR	BSR
Mean	156	170
σ	4.18	3.54

Outdoor	DSR	BSR
Mean	150	160
σ	3.54	3.54

Figure 6.18: FTP Performance of Scenario 6

The throughput performance is depicted in Figure 6.18. The more reliable transmission in BSR contributes the larger throughput like before. The outdoor throughput performance is worse than indoor for the same reason mention above.

6.7 Concluding Remark

BSR has slightly larger delay in Scenario 2 and 3 with bottleneck node and Scenario 6 with three intermediate nodes moving. In the severe situation, the large delay of BSR is caused by those packets which use backup path to recover from primary path failure and take a long time to reach the destination. BSR has slightly shorter delay than DSR in all the other cases by taking the advantage of backup path. However, the advantages of BSR in delay performance do not stand out as expected probably due to the simple topologies we used. Similarly BSR has less packet loss and higher throughput than DSR because BSR can provide a more reliable communication between the source and the destination.

With reference to Chapter 5, BSR has longer delay than MSR and the Improved MSR. This is because BSR is in fact a single-path routing and does not have the advantages of multiple-path routing. BSR has higher throughput than MSR and the improved MSR since single-path routing does not have the out-of-order packet arrival problems.

In static topologies, both protocols have longer delay and lower throughputs in Scenario 2 and Scenario 3 than in Scenario 1 because the path in Scenario 2 and Scenario 3 has one more hop and one bottleneck node and the worse performance is caused by the background traffic.

In mobile topologies, Scenario 6 has a longer delay, more packet loss, and lower throughput than Scenarios 4 and 5 because the mobility of three nodes has much more influence on the performance than the mobility of one node. Scenario 5 has the best performance since we simply move the source node toward the destination to make the signal receiving better.

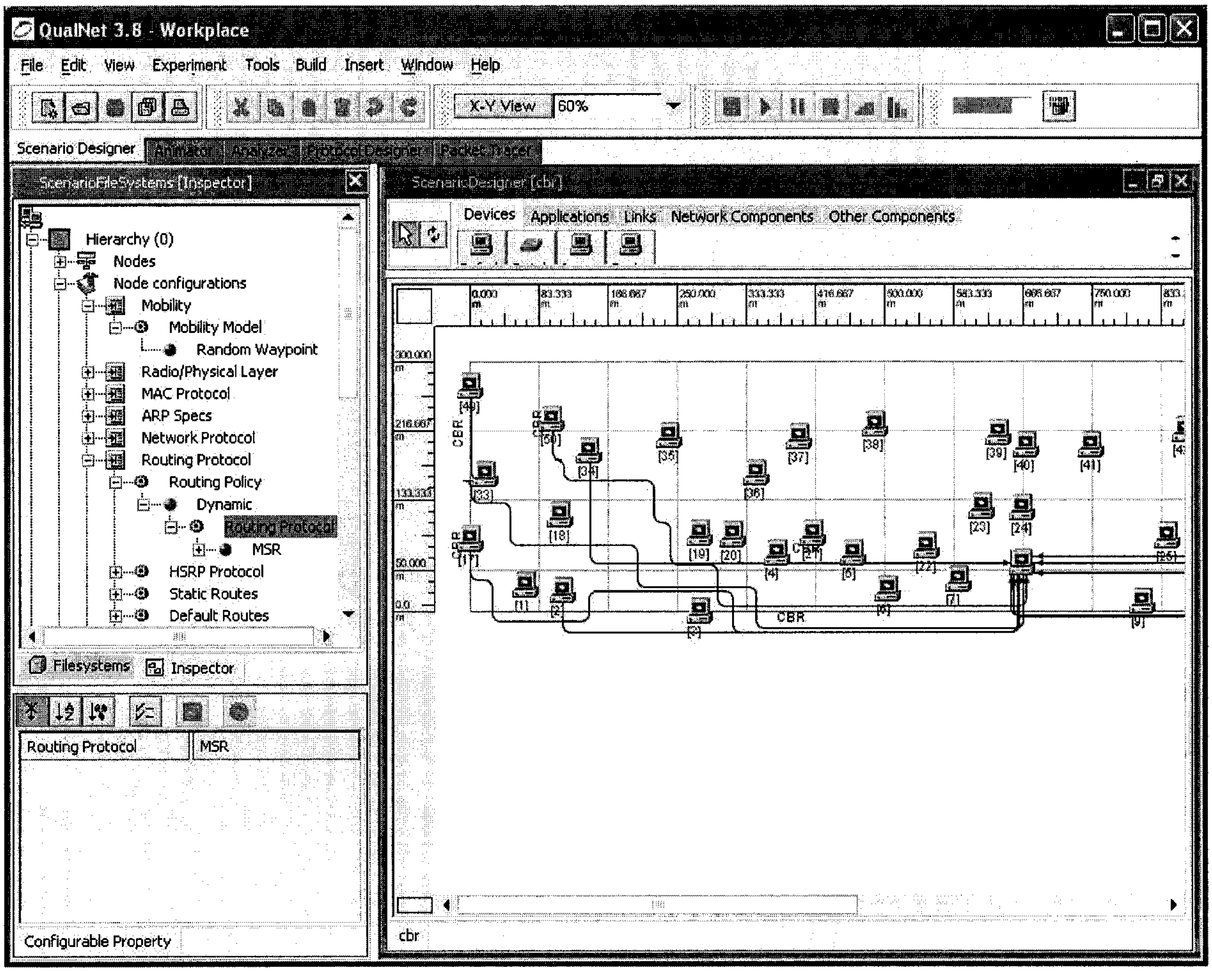
In general, mobility scenarios have worse performance than static scenarios except that the static scenarios with heavy background traffic also have bad performance. The outdoor measurements are very similar to indoor in most cases. However, the outdoor has slightly less packet loss and slightly higher throughput than indoor except in Scenario 6 where the outdoor has worse performances.

Chapter 7
Qualnet Simulation

In this chapter, we shall first introduce the simulation tool Qualnet and then discuss our simulation results.

7.1 Introduction to Qualnet

Qualnet [Qual05] is a commercial simulation tool that provides much more functions than some non-commercial tools such as NS2 and GloMoSim. We choose the Qualnet because of its two attractive features: GUI (Graphic User Interface) and the animation capability. The GUI makes it easy to design scenarios and protocols and the animation capability makes the simulation visual.



Routing Protocol | MSR

Figure 7.1: The Qualnet GUI

Figure 7.1 shows the user interface of Qualnet. Qualnet consists of five tools:

1) Scenario Designer: Used for user to design scenarios. As shown in Figure 7.1, user can put devices, applications, and other components in the graph interface conveniently. We do not need to write scenario script file which is a must in NS2.
2) Animator: Make visual the simulation process. It can show the details of each layer when simulation is running. For example, the queue length of each node can be shown dynamically with the proceeding of simulation.
3) Analyzer: A graphical statistical analyzing tool for displaying simulation results in graphs.
4) Protocol Designer: A graphical protocol designing tool.
5) Packet Tracer: A graphical packet tracing tool. We can easily see the contents of each packet generated during simulation.

In our simulation, we mainly use the Scenario Designer and the Animator to design and simulate our protocols. We did no use Protocol Designer to design our protocols because Qualnet does not provide the DSR code in Protocol Designer. However, Qualnet provides all the source codes and the documents for modifying and recompiling the source codes, so we can find the DSR source code and modify it to MSR and BSR.

7.2 Performance Evaluation by Qualnet Simulation

We shall use the same six scenarios in Section 3.4 as in the testbed. Likewise, we use the same performance measures as in the testbed, i.e. delay, packet loss rate, and throughput.

Qualnet provides an application CBR (Constant Bit Rate) which is similar to Ping application in the testbed to test the delay and packet loss rate. In CBR, the source sends packets to the destination with a constant transmission rate and does not require the acknowledgement from the destination. CBR application is a little bit different from the Ping application in the testbed. Ping application requires the destination to send back the reply message while CBR does not. That is the reason why the performance magnitudes in simulation are much different from the testbed measurements.

Qualnet also provides FTP application as in our testbed. Thus we can compare the throughput performances for our protocols.

In our simulation, we simulate 3 times by changing the seed for each point only for Scenario 6. In the other scenarios, we only simulate once.

7.2.1 Scenario 1: Simple Multiple Paths

1) CBR Traffic

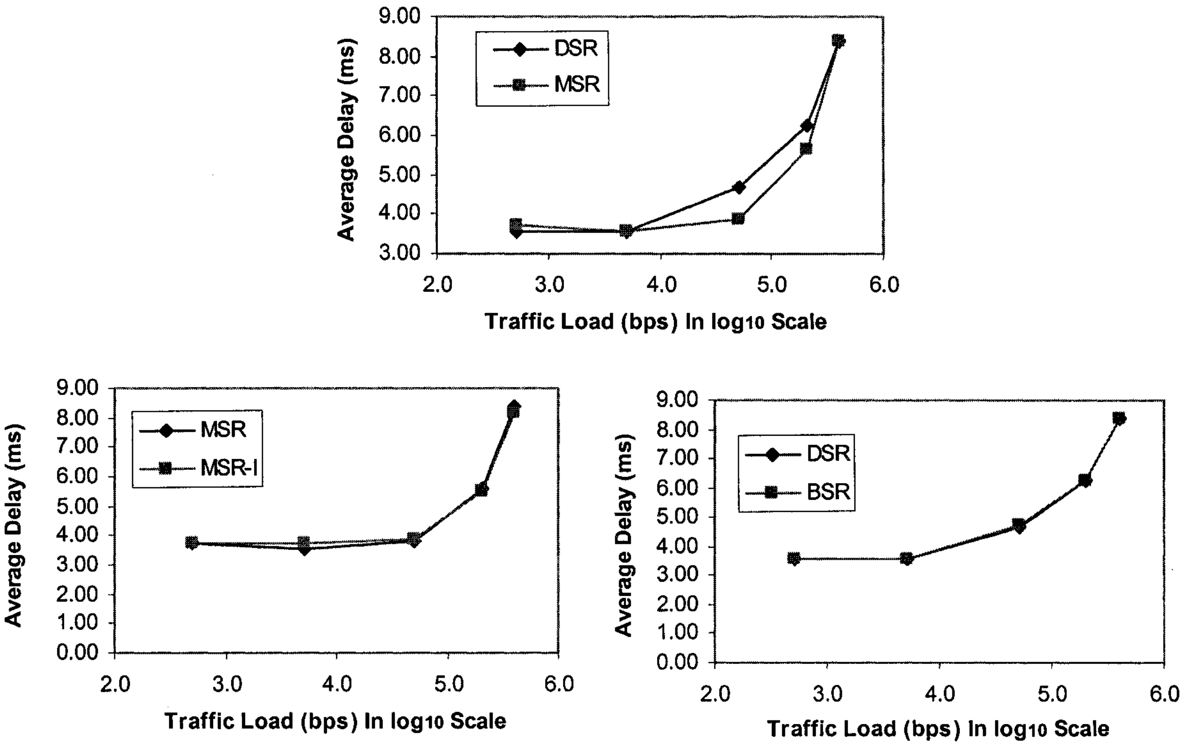


Figure 7.2: CBR Simulation Results for Scenario 1

Figure 7.2 shows the CBR simulation results of Scenario 1. The delay of MSR decreases slightly first before increasing for the same reason of probing packets as in testbed. MSR has smaller delay than DSR because MSR can distribute traffic to multiple paths to reduce congestion. MSR-I has a little more delay than MSR because MSR-I is not responsible for small delay change and the load distribution of MSR-I may be less optimal than MSR. BSR has a little more delay than DSR because the longer header and more processing. The performances are consistent with in the testbed except the smaller delay values in the simulation because CBR traffic is different from Ping application and does not need acknowledgement from the destination.

2) FTP Traffic

Figure 7.3 depicts the FTP simulation results of Scenario 1. MSR has smaller throughput than DSR due to the out-of-order packets in multiple paths. MSR-I improves the throughput from MSR a lot due to the adaptive probing. In this static topology, BSR has a little less throughput than DSR since there is no link failure in simulation and DSR can perform very well.

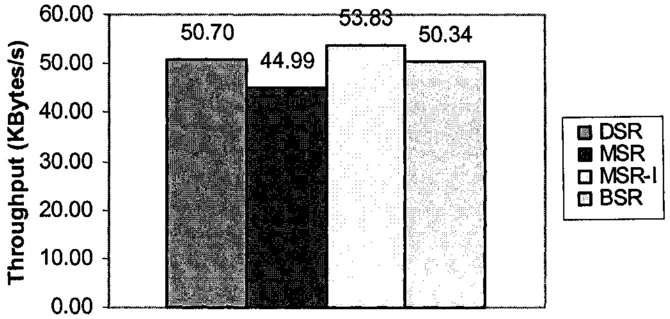


Figure 7.3: FTP Simulation Results for Scenario 1

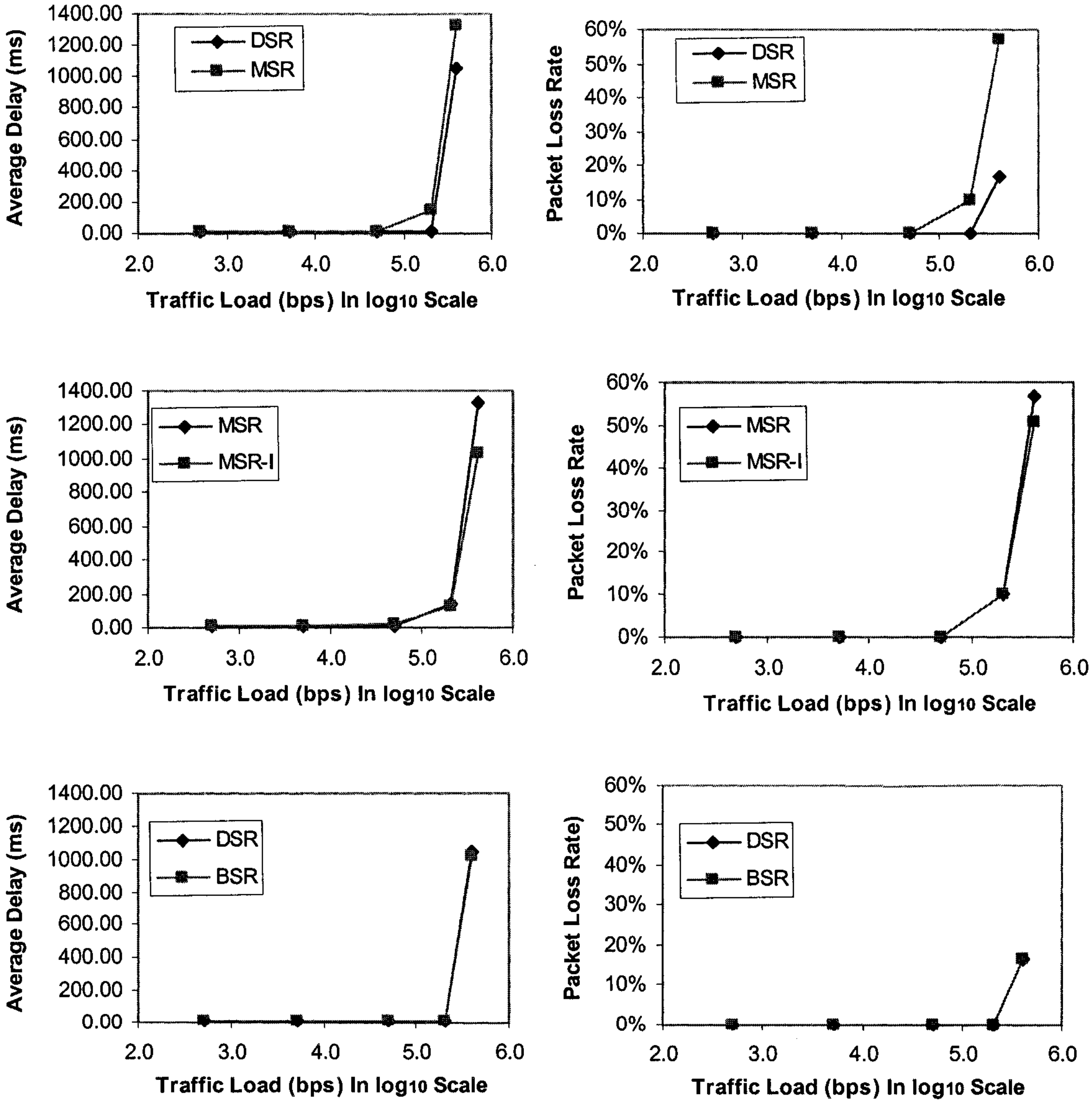


Figure 7.4: CBR Simulation Results for Scenario 2

7.2.2 Scenario 2: Interference Test

1) CBR Traffic

Figure 7.4 shows the CBR simulation results for Scenario 2. MSR has a longer delay and more packet loss than DSR because all the paths have a common first hop and MSR cannot show its advantages. MSR-I has smaller delay and less packet loss than MSR because the probing packets worsen the performance in the presence of one bottleneck node near the source. BSR has comparable delay and packet loss than DSR because the backup path via the bottleneck node will contribute little in the performance.

2) FTP Traffic

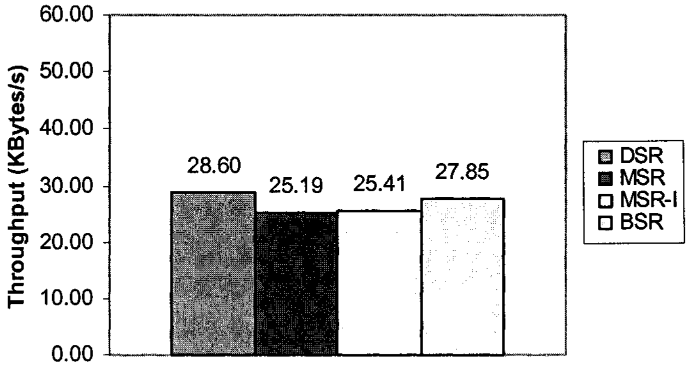


Figure 7.5: FTP Simulation Results for Scenario 2

MSR has smaller throughput than DSR due to the out-of-order packets in multiple paths. MSR-I has a little more throughput as MSR due to the adaptive probing. BSR has a little less throughput than DSR because the backup path via the bottleneck node will contribute little in the performance.

7.2.3 Scenario 3: Bottleneck Test

1) CBR Traffic

Figure 7.6 shows the CBR simulation results of Scenario 3. MSR has smaller delay and less packet loss than DSR in this topology with the bottleneck node near the destination. MSR-I has almost the same performances as MSR, and BSR has smaller delay and less packet loss than DSR.

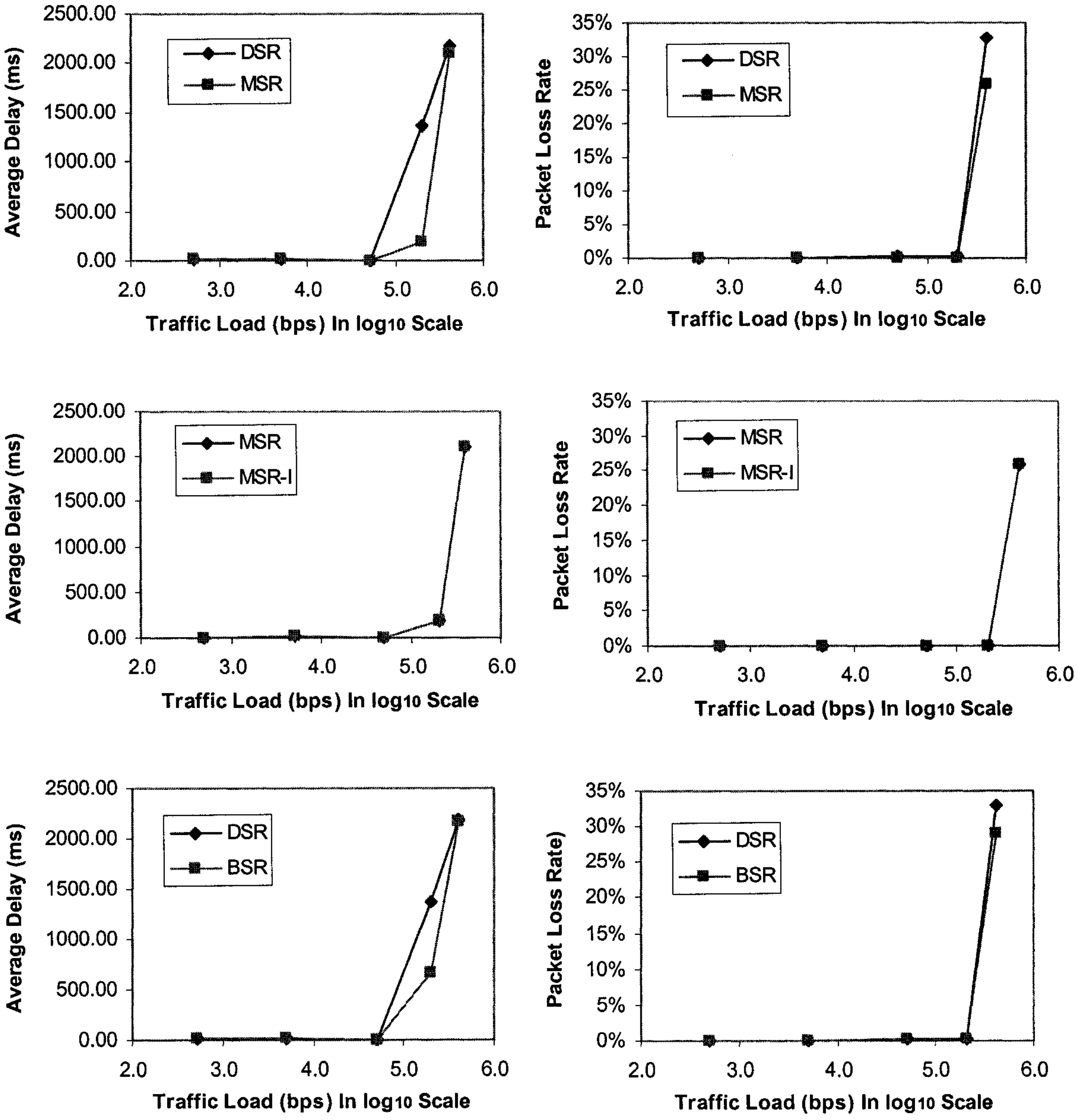


Figure 7.6: CBR Simulation Results for Scenario 3

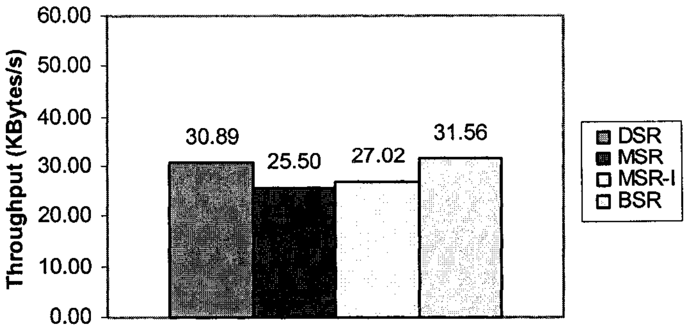


Figure 7.7: FTP Simulation Results for Scenario 3

2) FTP Traffic

Figure 7.7 shows the FTP simulation results for Scenario 3. MSR has smaller throughput than DSR due to the out-of-order packets in multiple paths. MSR-I has a little more throughput as MSR due to the adaptive probing. BSR has larger throughput than DSR because the backup path can provide more reliable communication between the source and destination.

7.2.4 Scenario 4: Back and Forth

1) CBR Traffic

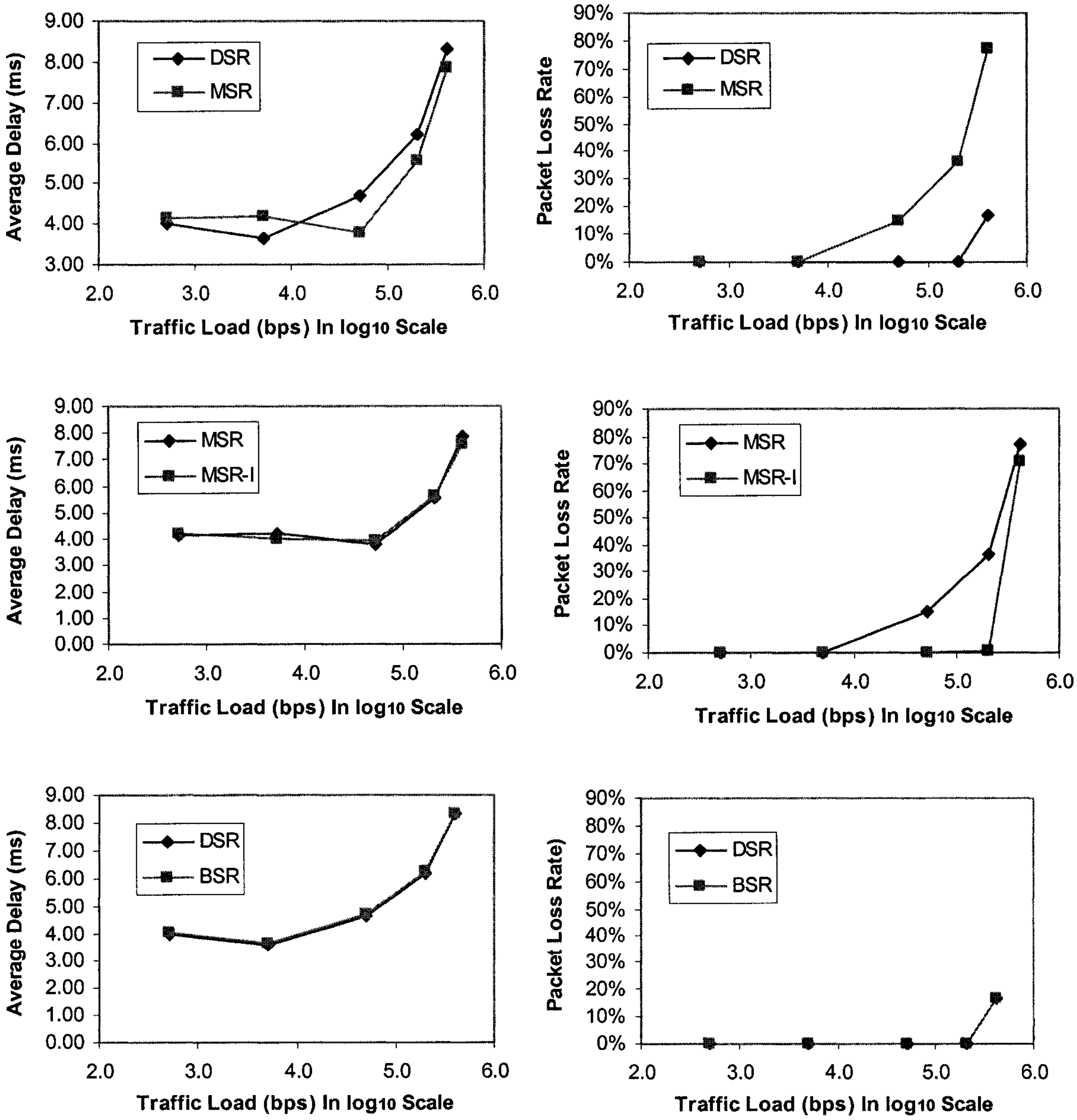


Figure 7.8: CBR Simulation Results for Scenario 4

Figure 7.8 shows the CBR simulation results for Scenario 4. In medium and heavy loads, MSR has smaller delay and more packet loss than DSR, MSR-I has smaller delay and less packet loss than MSR, and BSR has a little bit more delay than DSR. We observe that in the load about 3.7, DSR has a shorter delay than MSR since the load is not large enough to cause a lot of congestion and DSR can perform well in simulation while MSR has some percentage of probing overhead.

2) FTP Traffic

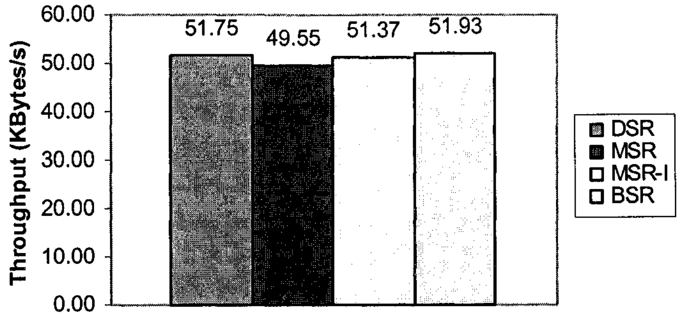


Figure 7.9: FTP Simulation Results for Scenario 4

Figure 7.9 shows the FTP simulation results for Scenario 4. MSR has smaller throughput than DSR due to the out-of-order packets in multiple paths. MSR-I has a little more throughput as MSR due to the adaptive probing. BSR has larger throughput than DSR because the backup path can provide more reliable communication between the source and destination.

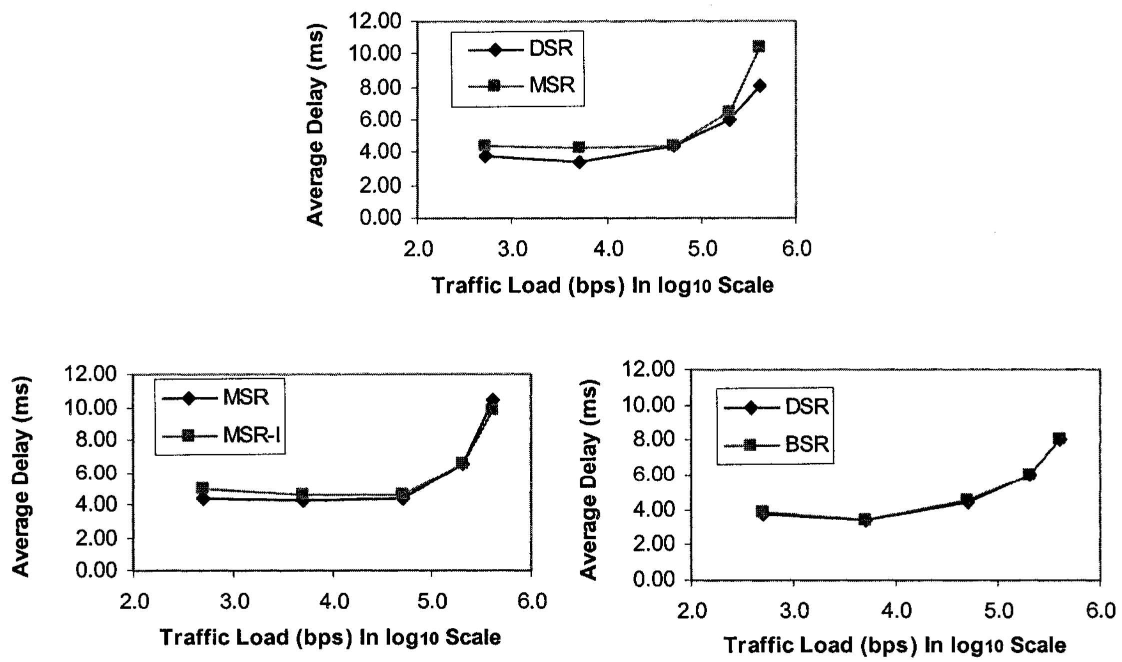


Figure 7.10: CBR Simulation Results for Scenario 5

7.2.5 Scenario 5: Varying S-D Distance

1) CBR Traffic

The CBR simulation results are shown in Figure 7.10. MSR has longer delay than DSR, MSR-I has longer delay than MSR, and BSR has a little more delay than DSR.

2) FTP Traffic

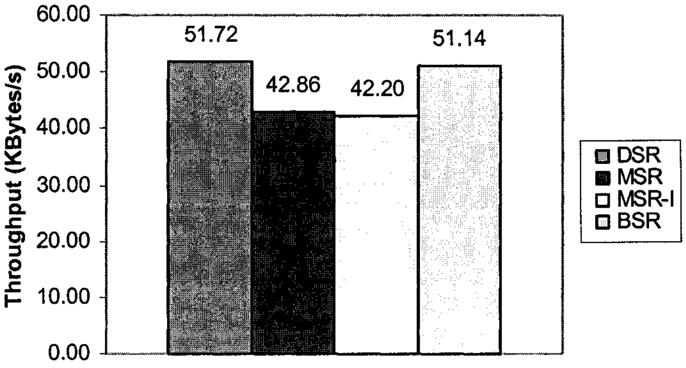


Figure 7.11: FTP Simulation Results for Scenario 5

Figure 7.11 depicts the FTP simulation results for Scenario 5. MSR has smaller throughput than DSR, MSR-I has a little smaller throughput than MSR, and BSR has a little smaller throughput than DSR.

7.2.6 Scenario 6: Circular Interference

1) CBR Traffic

Figure 7.12 shows the CBR simulation results for Scenario 6. In this scenario, each point is the average of 3 times simulation for different seeds. MSR has shorter delay and less packet loss in medium and heavy loads than DSR. The Improved MSR has a little bit longer delay and less packet loss in medium load than MSR. BSR has shorter delay and less packet loss in medium and heavy load than DSR.

2) FTP Traffic

Figure 7.13 shows the FTP simulation results for Scenario 6. MSR has larger throughput than DSR, MSR-I has smaller throughput than MSR, and BSR has a little smaller throughput than DSR.

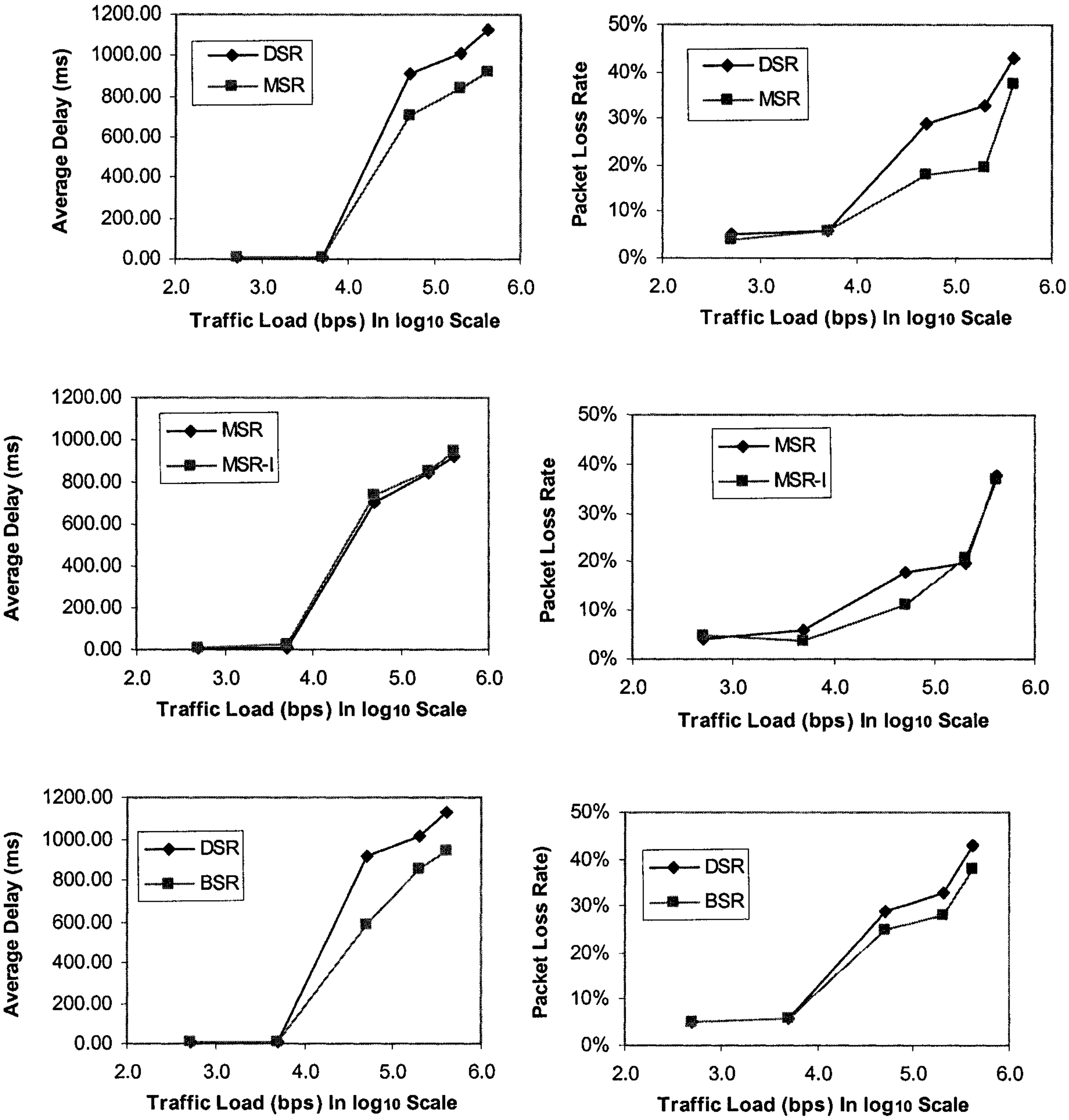


Figure 7.12: CBR Simulation Results for Scenario 6

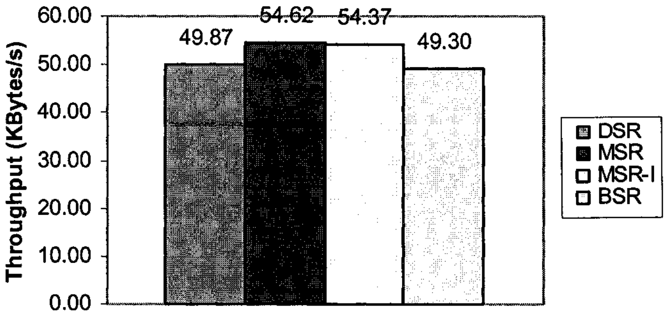


Figure 7.13: FTP Simulation Results for Scenario 6

7.3 Comparison with Testbed Measurement Results

We compare the simulation results to the testbed results here. The specific values of delay and throughput in simulation and testbed are not expected to be the same since different applications are used in simulation and tesetbed although both applications use UDP. Therefore we only compare the trends such as which protocol is better in some situation.

In static topologies, most of our simulation results are consistent with our testbed testing results. MSR has smaller delay, less packet loss, and smaller throughput than DSR, the improved MSR has a little longer delay and larger throughput than MSR, and BSR has a little bit shorter delay and larger throughput than DSR.

In mobile topologies, the simulation results from scenario 4 are consistent with testbed results, but there are some difference between the simulation results and testbed results in scenario 5 and 6. Because the node can only move along a straight line in simulation, we let the nodes move around a triangle with three lines instead of a circle in scenario 6, which causes the difference of results. In the simulation of scenario 5 and 6, the improved MSR is worse than MSR and BSR is worse than DSR in FTP application, which is different from testbed results. In scenario 5, simulation results show that MSR has a longer delay than DSR which is different from testbed results. From the different above, we need to improve the Qualnet model in future.

Chapter 8
Conclusion

In this thesis, we have setup an ad-hoc network testbed consisting of IBM laptops equipped with wireless cards. We have coded and implemented the DSR, MSR and BSR in Linux Operating System. Systematic tests and measurements of delay, throughput, and loss rate under various indoor and outdoor scenarios (as well as static and mobile) were carried out so that we can study the tradeoffs among these protocols. Based on our experiments, we have proposed, implemented and demonstrated an improved version of the MSR.

In terms of delay performance, MSR appears to have the lowest mean delay followed by the Improved MSR, BSR and then DSR. Here BSR and DSR have close delay performances. In terms of throughput performance, BSR has the highest throughput followed by DSR, the improved MSR, and MSR. Here DSR and Improved MSR have comparable throughput performances. In terms of loss rate performance, BSR has the lowest loss rate followed by the improved MSR, MSR, and DSR.

We personally feel that MSR performs better in the high traffic load and the Improved MSR would perform better in the low traffic load in UDP applications such as Ping. BSR would perform better in TCP application such FTP.

We have also implemented all protocols in Qualnet, one of the latest simulation languages built for ad-hoc networking. Initial smulation results indicate that their general behaviours are consistent with the testbed results in static scenarios but a lot different in mobility scenarios.

We have accumulated a lot of experience in our testbed experiments. Since out testbed consists of any laptops and wireless cards, the hardware setup is not so difficult as long as we follow the instruction to install Linux driver for wireless cards. The most difficult thing in our testbed is the code implementation, especially code debugging. Because our code is designed for inserting into the Linux kernel, we cannot debug the code step-by-step and a little error in code may cause the machine to be dead. In our debugging experience, most errors are caused by misuse of pointers. In testing, the first difficulty is to setup the scenarios to ensure correct connectivity. Ping appears to be the best to test link connection status. After putting the nodes, one node should ping all the other nodes to see if it can reach it or not. Node positions may be adjusted several times to meet the requirements. Another issue is the transmission range of the wireless card. Here low transmission power of the wireless card is preferred since the node distance can be significantly reduced. The analysis is not so difficult after we got the data because Microsoft Excel provides a set of powerful chart-drawing tools.

8.1 Future Work

There are many things for improvement over our present work.

- Improve the TCP performance of MSR to buffer the out-of-order packets from multiple paths although it is not easy to do in testbed since the TCP layer of Linux cannot be accessed by our code.
- Find some methods, equipments or software that can detect the EMI (Electromagnetic Interferences) at any point along the path.
- Explore whether multiple antenna can be used for simultaneously transmitting and receiving.
- Find out some convenient method to adjust the node positions quickly and correctly.
- Test the performance over a large distance using the highest possible power.
- In fact, do more outdoor tests in order to see if our observations are consistent with this thesis
- Test BSR in a larger network.
- Test more applications such as audio and video communications in our testbed.
- More accurate coding and modeling of the Qualnet scenarios.

References

[Dijk59] E. Dijkstra, "A Note on Two Problems in Connection with Graphs," *Numerical Mathematics*, October 1959

[FoFu62] L. Ford, and D. Fulkerson, *Flows in Networks*. Princeton, NJ: Princeton University Press, 1962

[GuYa02] Song Guo, and O.W. Yang, "Performance of backup source routing in mobile ad hoc networks," *Wireless Communications and Networking Conference, 2002. WCNC2002*. 2002 IEEE, Volume: 1, 17-21 Orlando, USA, March 2002

[HuJo01] Yih-Chun Hu, David B. Johnson, "Implicit source routes for on-demand ad hoc network routing," *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, Long Beach, CA, USA, 2001

[JoMa96] David B. Johnson and David A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," Chapter 5 in the book *Mobile Computing*, edited by Tomasz Imielinski and Hank Korth, Kluwer Academic Publishers, 1996.

[Liu00] Jane W.S. Liu, "Real-Time Systems," 2000, XIV + 610 pp., ISBN: 0-13-099651-3, Prentice Hall, Upper Saddle River, New Jersey 07458

[MaBr99] David A. Maltz, Josh Broch, and David B. Johnson, "Experiences Designing and Building a Multi-Hop Wireless Ad Hoc Network Testbed," *Technical Report CMU-CS-99-116*, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, March 1999

[Malk93] RIP G. Malkin -RIPversion2, Carrying Additional Information, January RFC1388, IETF 1993

[Nord02] Erik Nordström, "APE - a Large Scale Ad Hoc Network Testbed for Reproducible Performance Tests," *Master Thesis from Information Technology Department of Computer Systems*, Uppsala University, Sweden, 28th June 2002

[PeBh94] Charles E. Perkins and Pravin Bhagwat, "Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers," *Proceedings of the SIGCOMM '94 Conference on Communications Architectures, Protocols and Applications*, pages 234–244, London, UK, August 1994

[PeRo99] Charles E. Perkins and Elizabeth M. Royer, "Ad-hoc On-Demand Distance Vector Routing," *Proc. 2nd IEEE Wksp. Mobile Comp. Sys. And Apps.*, New Orleans, LA , Feb. 1999, pp. 90-100

[PeRo01] Charles E. Perkins, Elizabeth M. Royer, Samir R. Das, and Mahesh K. Marina, "Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks," *IEEE Personal Communications*, Feb. 2001, pp 16-28

[Qual05] http://www.scalable-networks.com

[Redh05] http://www.redhat.com

[RoTo99] Elizabeth M. Royer, Chai-Keong Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks," *IEEE Personal Communications*, April 1999, pp 46-55

[SaBr03] S. Sanghani, T.X. Brown, S. Bhandare, S. Doshi, "EWANT: the emulated wireless ad hoc network testbed," *Wireless Communications and Networking, 2003. WCNC 2003*. 2003 IEEE, Volume: 3, 16-20, New Orleans, Louisiana, USA, March 2003

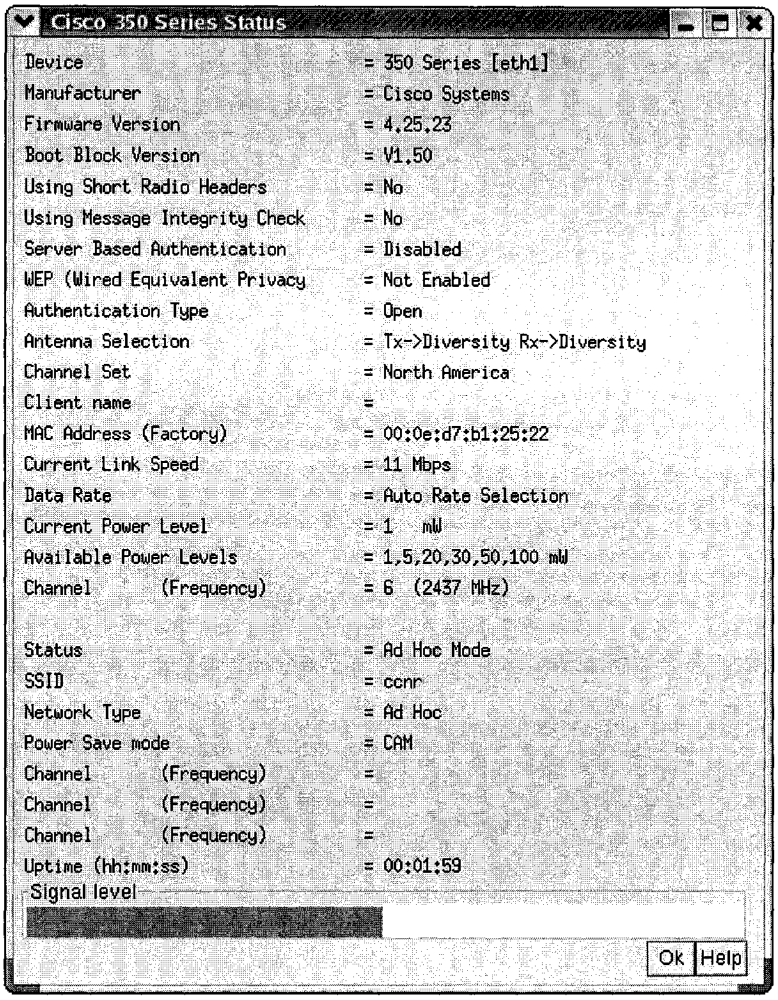
[Song01] A. Song, "Piconet II, a wireless ad hoc network for mobile handheld devices," http://piconet.sourceforge.net

[WaSh01] Lei Wang, Yantai Shu, Misao Dong, Lianfang Zhang, and Oliver W.W. Yang, "Adaptive Multipath Source Routing in Ad Hoc Networks," *IEEE ICC'2001*, Helsinki, Finland, June 2001

[WaZh00] Lei Wang, Lianfang Zhang, Yantai Shu, Miao Dong, and Oliver Yang, "Multipath Source Routing in Wireless Ad Hoc Networks," *Proceeding of 2000 Canadian Conference on Electrical and Computer Engineering,*, Volume: 1 , 7-10, Halifax, NS, March 2000

[ZhZh02] Lianfang Zhang, Zenghua Zhao, Yantai Shu, Lei Wang, and Oliver W.W. Yang, "Load Balancing of Multipath Source Routing in Ad Hoc Networks," *IEEE International Conference on Communications, 2002. ICC 2002*, Volume: 5, Budapest, Hungary, 28 April-2 May 2002

Appendix A: Status Display of Cisco 350 Wireless Card



```
Device                          = 350 Series [eth1]
Manufacturer                    = Cisco Systems
Firmware Version                = 4.25.23
Boot Block Version              = V1.50
Using Short Radio Headers       = No
Using Message Integrity Check   = No
Server Based Authentication     = Disabled
WEP (Wired Equivalent Privacy   = Not Enabled
Authentication Type             = Open
Antenna Selection               = Tx->Diversity Rx->Diversity
Channel Set                     = North America
Client name                     =
MAC Address (Factory)           = 00:0e:d7:b1:25:22
Current Link Speed              = 11 Mbps
Data Rate                       = Auto Rate Selection
Current Power Level             = 1   mW
Available Power Levels          = 1,5,20,30,50,100 mW
Channel       (Frequency)       = 6  (2437 MHz)

Status                          = Ad Hoc Mode
SSID                            = ccnr
Network Type                    = Ad Hoc
Power Save mode                 = CAM
Channel       (Frequency)       =
Channel       (Frequency)       =
Channel       (Frequency)       =
Uptime (hh:mm:ss)               = 00:01:59
```

Figure A.1: The status of Cisco Aironet 350 Series PC card

The status window in Figure A.1 is generated by the Cisco client software provided by the driver to show each parameter of working status after we configure the card.

- Device: A description of your client adapter.
- Manufacturer: The manufacturer of your client adapter.
- Firmware Version: The version of the firmware that is running on your client adapter.
- Boot Block Version: The version of the boot block firmware that is in the client adapter. The boot block firmware contains identification information for the client adapter, starts the radio, and passes control to the main firmware, which (unlike the boot block) can be modified and upgraded by the user.
- Using Short Radio Headers: Shows whether your client adapter is set up to use short radio headers.

- Using Message Integrity Check: Indicates whether your client adapter is using message integrity check (MIC) to protect packets sent to and received from the access point.
- Server Based Authentication: Shows whether LEAP is enabled for your client adapter.
- WEP (Wired Equivalent Privacy): Shows your client adapter's WEP status.
- Authentication Type: Shows whether the client adapter must share the same WEP keys as the access point in order to associate or can associate with the access point regardless of WEP settings.
- Antenna Selection: The antenna mode that your client adapter is using.
- Channel Set: The regulatory domain for which your client adapter is configured, such as North America.
- Client Name: The name your client adapter uses when it associates to an access point. Client Name is an optional setting; the adapter performs with or without a configured client name.
- MAC Address (Factory): The MAC address assigned to your client adapter at the factory

Current Link Speed: The rate at which your client adapter is transmitting data packets.

- Data Rate: The rate at which your client adapter is configured to transmit or receive data packets.
- Current Power Level: The power level at which your client adapter is transmitting.
- Available Power Levels: The power levels at which yourclient adapter is capable of transmitting.
- Channel (Frequency): The frequency that your client adapter is using as the channel for communications.
- Status: The operational mode of your client adapter.
- SSID: The SSID (the network name) that your client adapter is using.
- Network Type: The type of network in which your client adapter is being used.
- Power Save Mode: The client adapter's current power consumption setting. Value: CAM (Constantly Awake Mode), Max PSP (Power Save Polling), or Fast PSP.
- Up Time: The amount of time (in hours:minutes:seconds) that the client adapter has been receiving power.
- Signal Level: The signal strength for all received packets. The more green the bar graph shows, the stronger the signal.

Appendix B: Configuration for Demarc wireless card

Since Demarc does not provide any graph configuration tool in its Linux driver, we can only use the commands to configure it. The following are the commands we used for configuring Demarc card:

iwconfig wlan0 mode ad-hoc (set card to ad hoc mode)

iwconfig wlan0 essid "ccnr" (the network name SSID is ccnr)

iwconfig wlan0 rate 11m (link speed 11Mbps)

"wlan0" is the device name for Demarc wireless card.

Appendix C: Wireless Card Specifications

C1: Cisco Aironet 350 Series Client Adapter Specifications

- Data Rates Supported: 1, 2, 5.5, and 11 Mbps
- Network Standard: IEEE 802.11b
- System Interface: AIR-PCM35x: PC Card (PCMCIA) Type II; AIR-PCI351x: peripheral component interconnect (PCI) Bus
- Frequency Band: 2.4 to 2.4897 GHz
- Network Architecture Types: Infrastructure and ad hoc
- Wireless Medium: Direct Sequence Spread Spectrum (DSSS)
- Media Access Protocol: Carrier sense multiple access with collision avoidance (CSMA/CA)
- Modulation: DBPSK @1 Mbps; DQPSK @ 2 Mbps; CCK @ 5.5 and 11 Mbps
- Operating Channels: North America: 11; ETSI: 13; Japan: 14
- Nonoverlapping Channels: Three
- Receive Sensitivity: 1 Mbps: -94 dBm; 2 Mbps: -91 dBm; 5.5 Mbps: -89 dBm; 11 Mbps: -85 dBm
- Delay Spread: 1 Mbps: 500 ns; 2 Mbps: 400 ns; 5.5 Mbps: 300 ns; 11 Mbps: 140 ns
- Available Transmit Power Settings: 100 mW (20 dBm); 50 mW (17 dBm); 30 mW (15 dBm); 20 mW (13 dBm); 5 mW (7 dBm); 1 mW (0 dBm)

C2: Demarc Wireless Card Specifications

- Frequency Range: 2400-2484 MHz
- Radio Type: Direct Sequence Spread Spectrum (DSSS)
- Modulation: CCK (11Mb 5.5Mb); DQPSK (2Mb); DBPSK (1mb)
- Operating Sub-Channels: 11 USA
- Radio Output Power (at connector): 100 mW (20 dBm) (Spectrum Analyzer Readings) @ 420mA
- Sensitivity @FER=0.08: 11Mb < -91dBm; 5.5Mb < -93dBm; 2Mb < -95dBm; 1Mb < -96dBm
- Radio Data Rate: 11Mb, 5.5Mb, 2Mb and 1Mb with Auto Fall-Back
- Operating Voltage: 5VDC 480mA TX / 280mA RX
- Antenna: Detachable Antenna
- Antenna Connector: Two RP-MMCX (Male) for high gain antennas
- Compatibility: Fully IEEE 802.11b compliant

Appendix D. Protocol Implementation in Qualnet

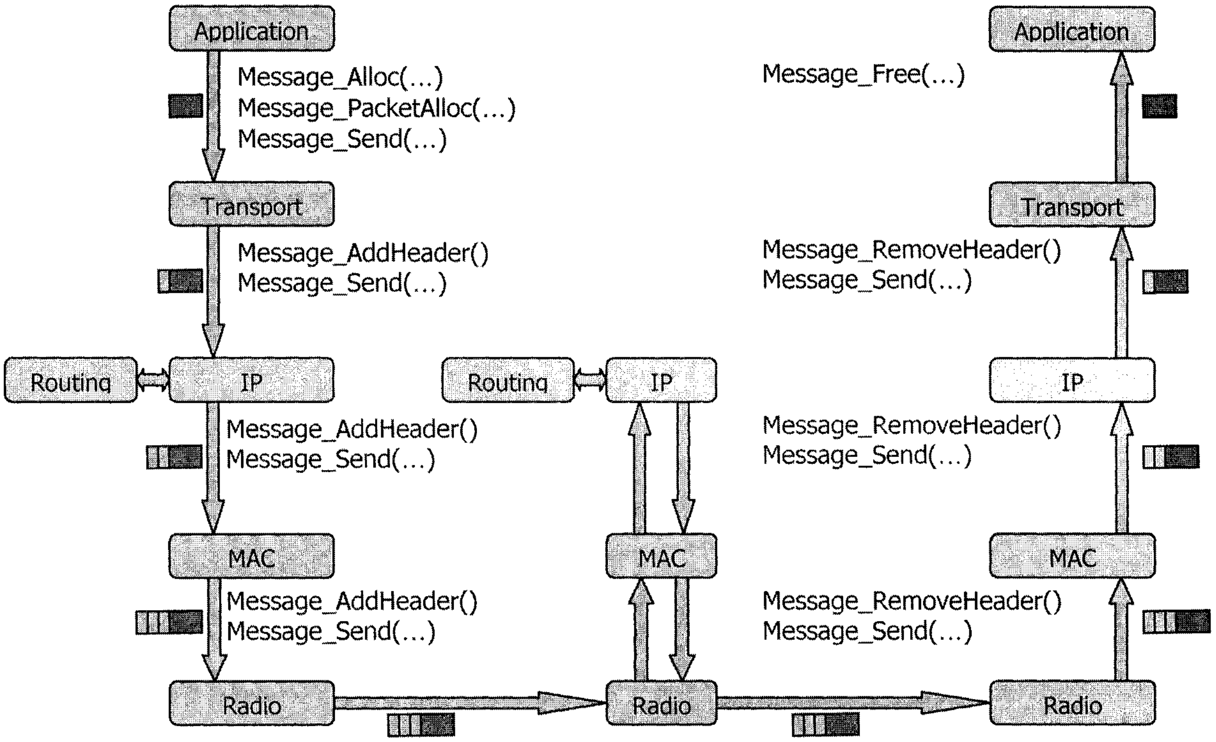


Figure D.1: Packet Life Cycle in Qualnet

Qualnet uses messages to interact between nodes while real implementation in testbed uses packets including data packets and control packets to deliver information between nodes. There are two types of messages: packets (used for communication between nodes) and timers (allow protocols to schedule future events such as the expiration of retransmission buffer). Figure A.2 shows the packet life cycle in Qualnet. The routing protocol locates in the position "Routing" box in the figure.

With the help of DSR source code, we can easily modify the code to whatever we want. There are two main functions: DsrRouterFunction() and DsrHandleProtocolPacket() in DSR. DsrRouterFunction() is represented by the "Routing" box in Figure 7.2 and used for generating packets in the source node and relaying packets in the intermediate nodes. DsrHandleProtocolPacket() is included in the right "IP" box in the destination in Figure 7.2 and is only for use in the destination node for processing received packets. Thus we implement the receiving handling functions of the destination in function DsrHandleProtocolPacket() and the receiving handling functions of the intermediate nodes in function DsrRouterFunction(). The transmitting handling functions are implemented in function DsrRouterFunction(). The detailed receiving and transmitting procedures can be referred to in Section 2.5.

Appendix E: Tables of Measurement Results in Testbed

We put all our testing data except scenario 1 here. The table of testing data for scenario 1 is put in each chapter together with the graph.

E1: Measurements of Scenario 2

Table E.1: Indoor measurements of Scenario 2

(a) No background traffic

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	14.536	0%	100	512	14.313	0%
2	64	0.1	1000	5120	17.488	0%	1000	5120	17.610	0%
3	64	0.01	1000	51200	17.578	0%	1000	51200	17.716	0%
4	512	0.01	1000	409600	22.630	0%	1000	409600	22.350	0%
5	1024	0.01	1000	819200	27.992	0%	1000	819200	28.167	0%
6	1024	-f	1000	1638400	30.713	0%	1000	1638400	31.383	0%
Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	16.823	0%	100	512	14.793	0%
2	64	0.1	1000	5120	13.952	0%	1000	5120	16.960	0%
3	64	0.01	1000	51200	14.089	0%	1000	51200	17.265	0%
4	512	0.01	1000	409600	19.638	0%	1000	409600	20.692	0%
5	1024	0.01	1000	819200	25.583	0%	1000	819200	26.507	0%
6	1024	-f	1000	1638400	28.676	0%	1000	1638400	29.788	0%

(b) With background traffic at 64 bytes/sec

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	16.773	0%	100	512	16.910	0%
2	64	0.1	1000	5120	19.585	0%	1000	5120	19.710	0%
3	64	0.01	1000	51200	20.500	0%	1000	51200	20.190	0%
4	512	0.01	1000	409600	27.553	0%	1000	409600	28.835	0%
5	1024	0.01	1000	819200	34.525	0%	1000	819200	35.010	0%
6	1024	-f	1000	1638400	36.624	0%	1000	1638400	37.418	0%
Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	18.897	0%	100	512	16.927	0%
2	64	0.1	1000	5120	16.740	0%	1000	5120	18.727	0%
3	64	0.01	1000	51200	17.417	0%	1000	51200	19.713	0%
4	512	0.01	1000	409600	23.780	0%	1000	409600	25.567	0%
5	1024	0.01	1000	819200	28.524	0%	1000	819200	30.746	0%
6	1024	-f	1000	1638400	33.451	0%	1000	1638400	34.719	0%

(c) With background traffic at 640 bytes/sec

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	21.688	0%	100	512	21.529	0%
2	64	0.1	1000	5120	22.859	0%	1000	5120	23.939	0%
3	64	0.01	1000	51200	30.295	16%	1000	51200	32.854	10%
4	512	0.01	1000	409600	70.035	18%	1000	409600	76.923	15%
5	1024	0.01	1000	819200	90.366	21%	1000	819200	99.330	18%
6	1024	-f	1000	1638400	111.404	33%	1000	1638400	116.957	30%
Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	23.809	0%	100	512	21.929	0%
2	64	0.1	1000	5120	21.025	0%	1000	5120	23.840	0%
3	64	0.01	1000	51200	21.492	11%	1000	51200	33.395	12%
4	512	0.01	1000	409600	33.932	15%	1000	409600	63.456	17%
5	1024	0.01	1000	819200	42.916	19%	1000	819200	86.474	20%
6	1024	-f	1000	1638400	65.502	24%	1000	1638400	98.582	32%

Table E.2: Outdoor measurements of Scenario 2
(a) No background traffic

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	14.331	0%	100	512	14.298	0%
2	64	0.1	1000	5120	17.445	0%	1000	5120	17.648	0%
3	64	0.01	1000	51200	17.360	0%	1000	51200	17.405	0%
4	512	0.01	1000	409600	22.333	0%	1000	409600	22.242	0%
5	1024	0.01	1000	819200	27.231	0%	1000	819200	27.708	0%
6	1024	-f	1000	1638400	29.445	0%	1000	1638400	30.573	0%
Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	16.606	0%	100	512	14.510	0%
2	64	0.1	1000	5120	13.960	0%	1000	5120	16.829	0%
3	64	0.01	1000	51200	13.944	0%	1000	51200	17.198	0%
4	512	0.01	1000	409600	19.383	0%	1000	409600	20.581	0%
5	1024	0.01	1000	819200	24.857	0%	1000	819200	26.413	0%
6	1024	-f	1000	1638400	27.764	0%	1000	1638400	29.263	0%

(b) With background traffic at 64 bytes/sec

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	16.471	0%	100	512	16.658	0%
2	64	0.1	1000	5120	19.235	0%	1000	5120	19.615	0%
3	64	0.01	1000	51200	20.371	0%	1000	51200	20.264	0%
4	512	0.01	1000	409600	27.235	0%	1000	409600	28.657	0%
5	1024	0.01	1000	819200	34.202	0%	1000	819200	35.298	0%
6	1024	-f	1000	1638400	36.238	0%	1000	1638400	37.156	0%
Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	18.402	0%	100	512	16.691	0%
2	64	0.1	1000	5120	16.431	0%	1000	5120	18.613	0%
3	64	0.01	1000	51200	17.202	0%	1000	51200	19.623	0%
4	512	0.01	1000	409600	23.498	0%	1000	409600	25.398	0%
5	1024	0.01	1000	819200	28.169	0%	1000	819200	30.398	0%
6	1024	-f	1000	1638400	31.713	0%	1000	1638400	34.110	0%

(c) With background traffic at 640 bytes/sec

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	21.378	0%	100	512	21.346	0%
2	64	0.1	1000	5120	22.915	0%	1000	5120	23.780	0%
3	64	0.01	1000	51200	29.713	15%	1000	51200	32.272	9%
4	512	0.01	1000	409600	69.519	18%	1000	409600	75.296	14%
5	1024	0.01	1000	819200	90.344	22%	1000	819200	97.522	17%
6	1024	-f	1000	1638400	103.827	27%	1000	1638400	110.421	20%
Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	23.309	0%	100	512	21.282	0%
2	64	0.1	1000	5120	20.514	0%	1000	5120	23.452	0%
3	64	0.01	1000	51200	21.496	11%	1000	51200	33.166	13%
4	512	0.01	1000	409600	33.823	16%	1000	409600	63.885	18%
5	1024	0.01	1000	819200	42.853	20%	1000	819200	86.582	21%
6	1024	-f	1000	1638400	57.116	25%	1000	1638400	94.733	26%

E2: Measurements of Scenario 3

Table E.3: Indoor measurements of Scenario 3
(a) No background traffic

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	15.448	0%	100	512	15.150	0%
2	64	0.1	1000	5120	18.284	0%	1000	5120	18.141	0%
3	64	0.01	1000	51200	18.935	0%	1000	51200	18.275	0%
4	512	0.01	1000	409600	22.927	0%	1000	409600	22.821	0%
5	1024	0.01	1000	819200	27.057	0%	1000	819200	28.301	0%
6	1024	-f	1000	1638400	29.734	0%	1000	1638400	30.406	0%
Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	16.458	0%	100	512	15.394	0%
2	64	0.1	1000	5120	14.385	0%	1000	5120	17.593	0%
3	64	0.01	1000	51200	14.324	0%	1000	51200	18.234	0%
4	512	0.01	1000	409600	19.387	0%	1000	409600	20.919	0%
5	1024	0.01	1000	819200	24.226	0%	1000	819200	26.721	0%
6	1024	-f	1000	1638400	27.440	0%	1000	1638400	28.523	0%

(b) With background traffic at 64 bytes/sec

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	17.347	0%	100	512	17.293	0%
2	64	0.1	1000	5120	19.014	0%	1000	5120	19.270	0%
3	64	0.01	1000	51200	21.271	0%	1000	51200	21.251	0%
4	512	0.01	1000	409600	27.391	0%	1000	409600	28.233	0%
5	1024	0.01	1000	819200	34.760	0%	1000	819200	35.471	0%
6	1024	-f	1000	1638400	37.351	0%	1000	1638400	38.656	0%
Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	18.947	0%	100	512	17.459	0%
2	64	0.1	1000	5120	17.510	0%	1000	5120	19.002	0%
3	64	0.01	1000	51200	17.417	0%	1000	51200	19.784	0%
4	512	0.01	1000	409600	24.251	0%	1000	409600	25.504	0%
5	1024	0.01	1000	819200	29.033	0%	1000	819200	31.927	0%
6	1024	-f	1000	1638400	33.582	0%	1000	1638400	35.490	0%

(c) With background traffic at 640 bytes/sec

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	22.238	0%	100	512	22.178	0%
2	64	0.1	1000	5120	23.319	0%	1000	5120	26.258	0%
3	64	0.01	1000	51200	27.889	13%	1000	51200	29.684	8%
4	512	0.01	1000	409600	48.855	14%	1000	409600	54.388	10%
5	1024	0.01	1000	819200	73.079	20%	1000	819200	78.510	12%
6	1024	-f	1000	1638400	100.844	30%	1000	1638400	110.386	24%
Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	24.329	0%	100	512	22.608	0%
2	64	0.1	1000	5120	17.515	0%	1000	5120	23.454	0%
3	64	0.01	1000	51200	19.369	11%	1000	51200	30.778	11%
4	512	0.01	1000	409600	30.160	13%	1000	409600	49.636	14%
5	1024	0.01	1000	819200	50.160	16%	1000	819200	72.538	18%
6	1024	-f	1000	1638400	70.426	25%	1000	1638400	91.092	29%

Table E.4: Outdoor measurements of Scenario 3
(a) No background traffic

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	14.512	0%	100	512	14.470	0%
2	64	0.1	1000	5120	17.623	0%	1000	5120	17.570	0%
3	64	0.01	1000	51200	18.268	0%	1000	51200	18.270	0%
4	512	0.01	1000	409600	22.118	0%	1000	409600	22.236	0%
5	1024	0.01	1000	819200	26.173	0%	1000	819200	27.535	0%
6	1024	-f	1000	1638400	30.486	0%	1000	1638400	31.305	0%
Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	15.998	0%	100	512	14.372	0%
2	64	0.1	1000	5120	14.737	0%	1000	5120	17.104	0%
3	64	0.01	1000	51200	14.870	0%	1000	51200	18.224	0%
4	512	0.01	1000	409600	18.604	0%	1000	409600	20.895	0%
5	1024	0.01	1000	819200	23.937	0%	1000	819200	26.315	0%
6	1024	-f	1000	1638400	28.529	0%	1000	1638400	30.659	0%

(b) With background traffic at 64 bytes/sec

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	16.830	0%	100	512	16.559	0%
2	64	0.1	1000	5120	18.591	0%	1000	5120	18.659	0%
3	64	0.01	1000	51200	21.332	0%	1000	51200	21.341	0%
4	512	0.01	1000	409600	26.361	0%	1000	409600	27.333	0%
5	1024	0.01	1000	819200	33.502	0%	1000	819200	35.558	0%
6	1024	-f	1000	1638400	38.657	0%	1000	1638400	41.819	0%
Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	17.729	0%	100	512	16.725	0%
2	64	0.1	1000	5120	17.038	0%	1000	5120	18.446	0%
3	64	0.01	1000	51200	17.338	0%	1000	51200	20.276	0%
4	512	0.01	1000	409600	23.488	0%	1000	409600	26.249	0%
5	1024	0.01	1000	819200	28.525	0%	1000	819200	31.371	0%
6	1024	-f	1000	1638400	32.439	0%	1000	1638400	36.706	0%

(c) With background traffic at 640 bytes/sec

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	21.298	0%	100	512	21.427	0%
2	64	0.1	1000	5120	22.254	0%	1000	5120	25.482	0%
3	64	0.01	1000	51200	27.621	13%	1000	51200	29.654	8%
4	512	0.01	1000	409600	47.428	15%	1000	409600	54.297	11%
5	1024	0.01	1000	819200	72.725	21%	1000	819200	74.297	15%
6	1024	-f	1000	1638400	99.884	33%	1000	1638400	101.019	20%
Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	23.490	0%	100	512	21.618	0%
2	64	0.1	1000	5120	17.596	0%	1000	5120	22.569	0%
3	64	0.01	1000	51200	19.558	10%	1000	51200	29.743	11%
4	512	0.01	1000	409600	31.357	12%	1000	409600	46.053	15%
5	1024	0.01	1000	819200	51.458	16%	1000	819200	73.312	20%
6	1024	-f	1000	1638400	65.658	24%	1000	1638400	96.974	27%

E3: Measurements of Scenario 4

Table E.5: Indoor measurements of Scenario 4

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	13.323	1%	100	512	13.582	0%
2	64	0.1	1000	5120	14.461	1%	1000	5120	14.492	0%
3	64	0.01	1000	51200	16.461	2%	1000	51200	15.714	0%
4	512	0.01	1000	409600	19.260	5%	1000	409600	18.561	0%
5	1024	0.01	1000	819200	23.722	10%	1000	819200	20.491	0%
6	1024	-f	1000	1638400	30.360	15%	1000	1638400	26.591	2%
Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	14.246	0%	100	512	13.425	0%
2	64	0.1	1000	5120	12.542	0%	1000	5120	14.395	0%
3	64	0.01	1000	51200	12.346	0%	1000	51200	15.461	0%
4	512	0.01	1000	409600	16.299	0%	1000	409600	18.645	0%
5	1024	0.01	1000	819200	18.683	1%	1000	819200	20.558	1%
6	1024	-f	1000	1638400	24.605	5%	1000	1638400	26.467	4%

Table E.6: Outdoor measurements of Scenario 4

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	13.420	1%	100	512	13.275	0%
2	64	0.1	1000	5120	15.683	1%	1000	5120	14.754	0%
3	64	0.01	1000	51200	16.675	2%	1000	51200	15.727	0%
4	512	0.01	1000	409600	20.609	3%	1000	409600	18.432	0%
5	1024	0.01	1000	819200	27.458	5%	1000	819200	20.360	0%
6	1024	-f	1000	1638400	32.474	13%	1000	1638400	26.555	4%
Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	13.424	0%	100	512	12.338	0%
2	64	0.1	1000	5120	12.491	0%	1000	5120	13.588	0%
3	64	0.01	1000	51200	12.576	0%	1000	51200	13.896	0%
4	512	0.01	1000	409600	15.539	0%	1000	409600	17.596	0%
5	1024	0.01	1000	819200	17.440	2%	1000	819200	19.396	1%
6	1024	-f	1000	1638400	25.534	5%	1000	1638400	27.554	4%

E4: Measurements of Scenario 5

Table E.7: Indoor measurements of Scenario 5

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	8.466	0%	100	512	8.419	0%
2	64	0.1	1000	5120	10.492	0%	1000	5120	10.298	0%
3	64	0.01	1000	51200	11.517	0%	1000	51200	11.309	0%
4	512	0.01	1000	409600	12.625	0%	1000	409600	12.015	0%
5	1024	0.01	1000	819200	16.722	0%	1000	819200	16.631	0%
6	1024	-f	1000	1638400	20.471	0%	1000	1638400	19.434	0%
Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	10.576	0%	100	512	8.663	0%
2	64	0.1	1000	5120	9.686	0%	1000	5120	10.433	0%
3	64	0.01	1000	51200	9.417	0%	1000	51200	11.132	0%
4	512	0.01	1000	409600	11.530	0%	1000	409600	12.479	0%
5	1024	0.01	1000	819200	15.523	0%	1000	819200	16.529	0%
6	1024	-f	1000	1638400	19.487	0%	1000	1638400	20.201	0%

Table E.8: Outdoor measurements of Scenario 5

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	8.306	0%	100	512	8.309	0%
2	64	0.1	1000	5120	10.278	0%	1000	5120	10.178	0%
3	64	0.01	1000	51200	11.329	0%	1000	51200	11.149	0%
4	512	0.01	1000	409600	12.595	0%	1000	409600	12.235	0%
5	1024	0.01	1000	819200	16.412	0%	1000	819200	16.405	0%
6	1024	-f	1000	1638400	19.749	0%	1000	1638400	18.085	0%
Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	9.797	0%	100	512	8.462	0%
2	64	0.1	1000	5120	9.247	0%	1000	5120	10.335	0%
3	64	0.01	1000	51200	9.215	0%	1000	51200	11.267	0%
4	512	0.01	1000	409600	11.313	0%	1000	409600	12.304	0%
5	1024	0.01	1000	819200	15.383	0%	1000	819200	16.331	0%
6	1024	-f	1000	1638400	17.796	0%	1000	1638400	19.374	0%

E5: Measurements of Scenario 6

Table E.9: Indoor measurements of Scenario 6

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	10.361	4%	100	512	10.373	1%
2	64	0.1	1000	5120	11.395	5%	1000	5120	11.534	3%
3	64	0.01	1000	51200	13.412	9%	1000	51200	13.821	6%
4	512	0.01	1000	409600	16.962	15%	1000	409600	17.078	11%
5	1024	0.01	1000	819200	21.492	17%	1000	819200	22.462	12%
6	1024	-f	1000	1638400	24.271	24%	1000	1638400	24.816	14%
Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	11.644	2%	100	512	10.295	1%
2	64	0.1	1000	5120	10.973	3%	1000	5120	11.206	2%
3	64	0.01	1000	51200	11.254	6%	1000	51200	12.372	5%
4	512	0.01	1000	409600	14.787	12%	1000	409600	16.202	11%
5	1024	0.01	1000	819200	19.402	14%	1000	819200	20.385	12%
6	1024	-f	1000	1638400	22.247	18%	1000	1638400	22.956	15%

Table E.10: Outdoor measurements of Scenario 6

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	10.242	3%	100	512	10.357	1%
2	64	0.1	1000	5120	12.685	4%	1000	5120	12.395	1%
3	64	0.01	1000	51200	14.374	8%	1000	51200	13.919	4%
4	512	0.01	1000	409600	22.443	17%	1000	409600	22.845	8%
5	1024	0.01	1000	819200	35.412	18%	1000	819200	35.921	10%
6	1024	-f	1000	1638400	42.442	23%	1000	1638400	40.970	14%
Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	100	512	13.456	2%	100	512	10.345	1%
2	64	0.1	1000	5120	12.221	2%	1000	5120	12.397	1%
3	64	0.01	1000	51200	12.273	5%	1000	51200	14.366	4%
4	512	0.01	1000	409600	18.511	13%	1000	409600	19.615	9%
5	1024	0.01	1000	819200	24.516	14%	1000	819200	25.202	11%
6	1024	-f	1000	1638400	36.249	18%	1000	1638400	39.657	15%

Appendix F: Tables of Simulation Results in Qualnet

We tabulate all the simulation results of Chapter 7 in this appendix.

F1: Simulation Results of Scenario 1

Table F.1: Simulation Results of Scenario 1

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	3000	512	3.56	0%	3000	512	3.59	0%
2	64	0.1	3000	5120	3.56	0%	3000	5120	3.59	0%
3	64	0.01	3000	51200	4.70	0%	3000	51200	4.73	0%
4	256	0.01	3000	204800	6.25	0%	3000	204800	6.28	0%
5	512	0.01	3000	409600	8.37	0%	3000	409600	8.41	0%
Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	3000	512	3.72	0%	3000	512	3.75	0%
2	64	0.1	3000	5120	3.58	0%	3000	5120	3.75	0%
3	64	0.01	3000	51200	3.85	0%	3000	51200	3.89	0%
4	256	0.01	3000	204800	5.62	0%	3000	204800	5.54	0%
5	512	0.01	3000	409600	8.38	0%	3000	409600	8.16	0%

F2: Simulation Results of Scenario 2

Table F.2: Simulation Results of Scenario 2

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	3000	512	9.68	0%	3000	512	10.40	0%
2	64	0.1	3000	5120	11.28	0%	3000	5120	11.59	0%
3	64	0.01	3000	51200	7.11	0%	3000	51200	7.02	0%
4	256	0.01	3000	204800	9.93	0%	3000	204800	9.70	0%
5	512	0.01	3000	409600	1049.66	17%	3000	409600	1018.64	17%
Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	3000	512	10.12	0%	3000	512	10.52	0%
2	64	0.1	3000	5120	9.94	0%	3000	5120	10.71	0%
3	64	0.01	3000	51200	16.58	0%	3000	51200	19.09	0%
4	256	0.01	3000	204800	142.86	10%	3000	204800	126.90	10%
5	512	0.01	3000	409600	1325.50	57%	3000	409600	1027.87	51%

F3: Simulation Results of Scenario 3

Table F.3: Simulation Results of Scenario 3

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	3000	512	10.18	0%	3000	512	10.16	0%
2	64	0.1	3000	5120	10.28	0%	3000	5120	10.35	0%
3	64	0.01	3000	51200	9.15	0%	3000	51200	8.88	0%
4	256	0.01	3000	204800	1367.70	0%	3000	204800	667.38	0%
5	512	0.01	3000	409600	2181.75	33%	3000	409600	2166.12	29%
Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	3000	512	10.16	0%	3000	512	10.16	0%
2	64	0.1	3000	5120	10.35	0%	3000	5120	10.35	0%
3	64	0.01	3000	51200	8.15	0%	3000	51200	8.15	0%
4	256	0.01	3000	204800	190.34	0%	3000	204800	190.34	0%
5	512	0.01	3000	409600	2106.32	26%	3000	409600	2106.32	26%

F4: Simulation Results of Scenario 4

Table F.4: Simulation Results of Scenario 4

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	3000	512	4.01	0%	3000	512	4.04	0%
2	64	0.1	3000	5120	3.63	0%	3000	5120	3.66	0%
3	64	0.01	3000	51200	4.68	0%	3000	51200	4.71	0%
4	256	0.01	3000	204800	6.23	0%	3000	204800	6.26	0%
5	512	0.01	3000	409600	8.32	17%	3000	409600	8.35	17%
Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	3000	512	4.13	0%	3000	512	4.20	0%
2	64	0.1	3000	5120	4.20	0%	3000	5120	3.98	0%
3	64	0.01	3000	51200	3.78	15%	3000	51200	3.95	0%
4	256	0.01	3000	204800	5.58	37%	3000	204800	5.63	1%
5	512	0.01	3000	409600	7.84	77%	3000	409600	7.56	71%

F5: Simulation Results of Scenario 5

Table F.5: Simulation Results of Scenario 5

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	3000	512	3.76	0%	3000	512	3.80	0%
2	64	0.1	3000	5120	3.38	0%	3000	5120	3.42	0%
3	64	0.01	3000	51200	4.42	0%	3000	51200	4.47	0%
4	256	0.01	3000	204800	5.97	0%	3000	204800	6.02	0%
5	512	0.01	3000	409600	8.04	0%	3000	409600	8.09	0%
Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	3000	512	4.40	0%	3000	512	4.96	0%
2	64	0.1	3000	5120	4.30	0%	3000	5120	4.57	0%
3	64	0.01	3000	51200	4.42	0%	3000	51200	4.56	0%
4	256	0.01	3000	204800	6.52	0%	3000	204800	6.50	0%
5	512	0.01	3000	409600	10.44	0%	3000	409600	9.80	0%

F6: Simulation Results of Scenario 6

Table F.6: Simulation Results of Scenario 6

Test Run	Data Packet Size (Bytes)	Sending Interval (s)	DSR				BSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	3000	512	9.97	5%	3000	512	12.43	5%
2	64	0.1	3000	5120	11.53	6%	3000	5120	9.27	6%
3	64	0.01	3000	51200	911.13	29%	3000	51200	583.45	25%
4	256	0.01	3000	204800	1012.18	33%	3000	204800	854.00	28%
5	512	0.01	3000	409600	1126.48	43%	3000	409600	944.37	38%
Test Run	Data Packet Size (Bytes)	Sending Interval (s)	MSR				The improved MSR			
			Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate	Number of Packets	Offered Load (bps)	Average Delay (ms)	Packet Loss Rate
1	64	1	3000	512	12.23	4%	3000	512	11.96	5%
2	64	0.1	3000	5120	10.29	6%	3000	5120	27.05	4%
3	64	0.01	3000	51200	705.34	18%	3000	51200	740.57	11%
4	256	0.01	3000	204800	845.36	20%	3000	204800	855.48	21%
5	512	0.01	3000	409600	921.31	38%	3000	409600	944.56	37%