

a357861

Route Discovery Optimization of Reactive Routing Protocols  
for Mobile Ad Hoc Networks

Seet Boon Chong

School of Computer Engineering

A thesis submitted to the Nanyang Technological University  
in fulfillment of the requirement for the degree of  
Doctor of Philosophy

2005



TK  
5105.55  
S453  
2005

## **Acknowledgements**

I would like to express sincere gratitude to my supervisor Dr. Lau Chiew Tong, and my co-supervisor Dr. Lee Bu Sung, both of whom have been very forthcoming in spending time to discuss with me about my research work, providing many good suggestions and encouragement during the course of my Ph.D. study.

I would also like to thank Dr. Cheng Tee Hiang, who was the first to introduce me to this exciting field of mobile ad hoc networking. His support and provision of an opportunity for me to work on a related DSTA-funded project during his term as Director of NTRC, have given me much head start in the early days of my candidature.

Finally, I am very grateful to my parents. It is their constant support and encouragement that have made it possible for me to come this far.

## Abstract

A *Mobile Ad Hoc Network* (MANET) is a radio-based wireless network operated entirely by mobile nodes that cooperate to provide a communication service with no support from wired networks required. Routing in MANET is especially challenging because of the constantly changing topology with mobility, and efficiency of routing is one important concern as wireless bandwidth is typically low.

To minimize the level of control traffic caused by routing, *reactive routing protocols* for MANET have been proposed within the Internet Engineering Task Force (IETF). These protocols react to topology changes only when there are packets to be sent, and as such, incur less routing overhead and manage mobility in a more efficient manner. However, the *route discovery process* of existing reactive protocols is still bandwidth-expensive as control packets are typically flooded across the network to discover a routing path to the packet's destination. Such network-wide floods may disturb many nodes unnecessarily. Individual nodes may expend more resources such as memory, CPU time and battery power to process these control packets. Frequent flooding potentially leads to serious bandwidth congestion, excessive packet collisions and contention in the wireless medium, all of which degrade network performance.

This thesis develops two optimization techniques for route discovery of reactive MANET routing protocols. The goal is to achieve a significant reduction of control packets while improving overall end-to-end network performance. The two optimizations developed can be applied independently, with each targeting a different type of control packet: one aims

to reduce Route Request (RREQ), while the other targets Route Reply (RREP) packets. Both control packets together constitute the overhead of route discovery.

The first optimization developed in this thesis is called the UNIQUE (UNIcast QUERY) technique. UNIQUE is designed to reduce RREQ (or *query*) packets by allowing the source to take advantage of any existing route to carry the RREQ (as a *unicast* message) closer to the destination, before a low-overhead localized flood is attempted to find the required route. By using both location and *route* information in its optimization process, UNIQUE outperforms the well-known Location Aided Routing (LAR) scheme that only uses location information, for reducing routing overhead of the Dynamic Source Routing (DSR) protocol – a well-studied reactive routing protocol. Experimental studies carried out across various scenarios with different node mobility, network size, and traffic loads also show that UNIQUE improves packet delivery and end-to-end delay performances as a result of reduced congestion caused by routing traffic.

The second optimization developed is known as the SURE (SUppress REply) technique, which is targeting RREP packets. SURE exploits a rarely noted but commonly occurring phenomenon to provide intermediate nodes with query state information that can prevent them from replying to a RREQ (even if received for the first time) if they do not offer a more effective route to the destination. Application of SURE to DSR is found to yield significant improvements in both protocol efficiency and performance.

Finally, it has been shown that the best performance can be attained if both UNIQUE and SURE techniques are jointly applied.

## Table of Contents

<b>List of Figures</b> .....	viii
<b>List of Tables</b> .....	x
<b>Chapter 1 Introduction</b> .....	1
1.1 Motivation .....	1
1.2 Contribution .....	2
1.3 Thesis Outline .....	4
<b>Chapter 2 Literature Review</b> .....	5
2.1 Overview of Mobile Ad Hoc Networks (MANET) .....	5
2.2 Classification of Routing in MANET .....	7
2.3 Reactive Routing Protocols .....	10
2.3.1 General Features .....	10
2.3.2 Dynamic Source Routing (DSR) .....	12
2.3.3 Ad-hoc On-demand Distance Vector (AODV) .....	14
2.4 DSR Optimizations to Route Discovery .....	15
2.4.1 Caching Overheard Routing Information .....	16
2.4.2 Non-Propagating Route Requests .....	16
2.4.3 Preventing Route Reply Storms .....	17
2.5 AODV Optimizations to Route Discovery .....	17
2.5.1 Expanding Ring Search .....	17
2.5.2 Local Repair .....	18

2.6	Location Aided Routing (LAR) Optimizations to Route Discovery .....	19
2.6.1	LAR Scheme 1 .....	19
2.6.2	LAR Scheme 2 .....	20
2.6.3	Further Optimizations.....	22
2.7	Other Related Work .....	23
<b>Chapter 3 UNIQUE Optimization for RREQ Reduction .....</b>		<b>27</b>
3.1	Introduction.....	27
3.1.1	Route Discovery with UNIQUE.....	27
3.1.2	UNIQUE with Query Redirection.....	30
3.2	Protocol Description .....	31
3.2.1	DSR+L .....	32
3.2.2	DSR+L+U .....	34
3.3	Simulation Model and Methodology .....	35
3.3.1	Channel and Radio Model.....	36
3.3.2	Medium Access Control (MAC) Protocol.....	36
3.3.3	Traffic and Mobility Pattern.....	37
3.3.4	Performance Metrics .....	38
3.4	Evaluation Results .....	40
3.4.1	Route Discovery Overhead .....	40
3.4.2	Total Routing and MAC Overhead .....	45
3.4.3	Route Discovery Latency .....	52
3.4.4	End-to-End Delay.....	54

3.4.5	Packet Delivery Ratio.....	57
3.4.6	Effect of Network Size and Traffic Load.....	59
3.5	Chapter Summary .....	62
<b>Chapter 4 SURE Optimization for RREP Reduction .....</b>		<b>64</b>
4.1	Introduction.....	64
4.1.1	Observation .....	64
4.1.2	Proposed Optimization.....	66
4.2	Protocol Description .....	68
4.2.1	DSR+S.....	68
4.3	Simulation Environment and Parameters .....	68
4.4	Performance Results and Analysis .....	70
4.4.1	Route Discovery Overhead .....	70
4.4.2	Total Routing and MAC Overhead .....	72
4.4.3	Route Discovery Latency .....	75
4.4.4	End-to-End Delay.....	77
4.4.5	Packet Delivery Ratio.....	79
4.4.6	Effect of Network Size and Traffic Load.....	80
4.5	Chapter Summary .....	83
<b>Chapter 5 Joint Application of UNIQUE and SURE .....</b>		<b>84</b>
5.1	Protocol Description .....	85
5.2	Simulation Environment and Parameters .....	86

5.3 Performance Results and Analysis .....	85
5.4 Chapter Summary .....	92
<b>Chapter 6 Conclusions and Future Work .....</b>	<b>93</b>
6.1 Conclusions.....	94
6.1.1 UNIQUE Optimization for RREQ Reduction.....	95
6.1.2 SURE Optimization for RREP Reduction.....	95
6.1.3 Joint Application for UNIQUE and SURE .....	97
6.2 Future Work.....	98
<b>Bibliography .....</b>	<b>102</b>
<b>Appendix A List of Publications .....</b>	<b>112</b>
A.1 For This Thesis.....	112
A.2 For Other Related Research.....	113
<b>Appendix B Acronyms.....</b>	<b>116</b>

## List of Figures

Figure 2.1	Example of a MANET .....	5
Figure 2.2	Discovery of multiple routes to a destination by DSR .....	13
Figure 2.3	LAR-1: concepts of request and expected zones.....	20
Figure 2.4	LAR-2: progressive forwarding of RREQ .....	21
Figure 3.1	UNIQUE: RREQ forwarded to target neighbor by unicast.....	28
Figure 3.2	Pseudo-code for UNIQUE.....	30
Figure 3.3	Route Discovery (RD) algorithm for DSR+L+U.....	35
Figure 3.4	Number of routing packets generated by route discoveries .....	40
Figure 3.5	Number of route discoveries performed by each protocol .....	42
Figure 3.6a	Total routing packets generated.....	46
Figure 3.6b	Total routing bytes generated .....	48
Figure 3.7	Modified algorithm for DSR+L+U (noOPT).....	50
Figure 3.8	Total routing overhead with no optimizations from DSR .....	50
Figure 3.9	Composition of routing packets .....	51
Figure 3.10	Latency in route discovery .....	53
Figure 3.11	End-to-end delay of data packets .....	56
Figure 3.12	Path optimality of data packets .....	56
Figure 3.13	Percentage of data packets delivered.....	58
Figure 4.1	Example scenarios that illustrate the effects of channel contention, path length, packet collision and mobility, which are potential factors leading to the observed phenomenon.....	65

Figure 4.2	Pseudo-code for SURE.....	67
Figure 4.3	Comparison of RREP packets between DSR and DSR+S.....	70
Figure 4.4a	Number of route discovery packets by DSR .....	71
Figure 4.4b	Number of route discovery packets by DSR+S.....	72
Figure 4.5	Total routing overhead vs. pause time.....	73
Figure 4.6	Total routing overhead with no Ring-Zero search .....	74
Figure 4.7	Composition of routing packets of each protocol.....	75
Figure 4.8	Route discovery latency vs. pause time.....	76
Figure 4.9	End-to-end delay vs. pause time.....	77
Figure 4.10	Path optimality of data packets .....	78
Figure 4.11	Packet delivery ratio vs. pause time .....	79
Figure 5.1	Route discovery overhead as a function of pause time .....	86
Figure 5.2	Total routing overhead as a function of pause time .....	87
Figure 5.3	Histogram showing the constituents of routing packets.....	88
Figure 5.4	Data packet end-to-end delay .....	89
Figure 5.5	Data packet delivery ratio.....	90
Figure 6.1	An alternative UNIQUE strategy .....	99
Figure 6.2	Suppress the forwarding of RREP.....	100

## List of Tables

Table 3.1	Data packet dropped summary for LAR with and without LS .....	33
Table 3.2	LAR performance results (with and without LS) .....	34
Table 3.3	Summary of simulation parameters .....	40
Table 3.4	Cache hit rates of protocols under constant mobility .....	42
Table 3.5	Percentage breakdown of the number of route discoveries and route discovery packets (RREQ + RREP) by individual techniques employed in each protocol .....	43
Table 3.6	Number of hops traversed by first received RREP .....	54
Table 3.7	Data packet dropped summary for the three protocols .....	58
Table 3.8	Summary of results for a 50 node model with 20 sources .....	60
Table 3.9	Effect of changing network size and traffic load .....	61
Table 3.10	Number of routing packets per route discovery .....	62
Table 4.1	Number of hops traversed by first received RREP .....	76
Table 4.2	Data packet dropped summary for the two protocols .....	80
Table 4.3	Summary of results for a 100 node model with 40 sources .....	80
Table 4.4	Effect of changing network size and traffic load .....	81
Table 4.5	Cache hit rates with 40 sources and zero pause time .....	82
Table 5.1	MAC signaling overhead under constant mobility .....	91

# Chapter 1

## Introduction

### 1.1 Motivation

Recent advances in wireless technology have spawned an increasing demand for wireless data capability in portable devices such as laptops and personal digital assistants (PDAs) which allows networked communication even while the user is on the move. Unlike the present cellular paradigm [1], where fixed wired base-stations provide mobile nodes with the wireless last-mile, a *Mobile Ad Hoc Network* (MANET) is a radio-based wireless network operated entirely by mobile nodes that cooperate to provide a communication service with no support from wired networks required. Mobile nodes within each other's radio range communicate directly via wireless links, while those that are far apart must rely on other nodes to route messages for them. A *routing protocol* is therefore required for each node to function as a router, in addition to being a host. Routing in MANET is especially challenging because of the constantly changing topology with mobility, and efficiency of routing is one important concern as wireless bandwidth is typically low.

To minimize the level of control traffic caused by routing, *reactive routing protocols* for MANET have been proposed within the Internet Engineering Task Force (IETF) [2]. These protocols react to topology changes only when there are packets to be sent, and as such, incur less routing overhead and manage mobility in a more efficient manner. However, the *route discovery process* of existing reactive routing protocols [3] is still bandwidth-expensive as control packets are typically flooded across the network to

discover a routing path to the packet's destination. Such network-wide floods may disturb many nodes unnecessarily.

Individual nodes may expend more resources such as memory, CPU time and battery power to process these control packets. Frequent flooding potentially leads to serious bandwidth congestion, excessive packet collisions and contention in wireless medium [4], all of which degrade network performance.

## 1.2 Contribution

The objective of this thesis is to develop techniques that enhance the efficiency as well as performance of this promising class of reactive, also referred to as on-demand, routing protocols. Specifically, the goal is to achieve a significant reduction of control packets while improving overall end-to-end network performance (in terms of delay and success of packet delivery). As route discovery is a major source of control overhead for reactive protocols, this thesis focuses on optimizing the efficiency of the route discovery process.

The main contributions are as follows:

- Implemented the well-known Location-Aided Routing (LAR) optimizations [5] to Dynamic Source Routing (DSR) code of VINT *ns-2* simulator [6]. Our work is one of the first to study the performance of LAR under realistic physical and link layer characteristics of 802.11 [7] wireless networks. Furthermore, we compare LAR with a natively optimized (not simple-flooding) DSR protocol, which is a well-studied reactive routing protocol. Previously, LAR was studied only with an

idealized MAC that assumes collision-free transmission, and comparison had been made with only a simple flooding protocol. This work thus contributes to a better understanding of its performance and limitations, in a realistic environment.

- From the insights gained in the above work, a new optimization technique named UNIQUE (UNICAST QUery) is proposed. UNIQUE is designed to reduce RREQ (or *query*) packets by allowing the source to take advantage of any existing route to carry the RREQ (as a *unicast* message) closer to the destination, before a low-overhead localized flood is attempted to find the required route. By using both location and *route* information in its optimization process, UNIQUE outperforms LAR that only uses location information, for reducing routing overhead of the DSR protocol. Additionally, UNIQUE improves packet delivery and end-to-end delay performances as a result of reduced congestion caused by routing traffic.
- This thesis is also the first to report and investigate an observation of a previously unreported phenomenon of *reversed query/response arriving order* that is seen to occur at some nodes during a route discovery. Our investigation attributes this rarely noted but commonly occurring phenomenon to several effects, including channel contention, path length, packet collision, and mobility, which causes a reversal of the arriving order of RREQ and RREP packets at these nodes.
- The observation inspired the proposal of a second optimization known as SURE (SUppress REply) technique, which is targeting RREP packets. SURE exploits the phenomenon to provide intermediate nodes with query state information that can prevent them from replying to a RREQ (even if received for the first time) if they

do not offer a more effective route to the destination. As in the case of UNIQUE, the application of SURE to DSR is found to yield significant improvements in both protocol efficiency and performance.

### **1.3 Thesis Outline**

The remainder of this thesis is outlined as follows. Chapter 2 provides the technical background to this thesis, including an overview of MANET, a classification of MANET routing protocols, a description of current IETF reactive MANET routing protocols, and representative works on route discovery optimization. The next two chapters present the new optimization techniques proposed in this thesis. Chapter 3 and 4 introduce the UNIQUE and SURE techniques respectively, by presenting a description of their optimization objective, design, implementation, and comparative performance with existing works. Chapter 5 investigates the possibility for further enhanced performance through joint application of the proposed techniques. Finally, Chapter 6 concludes this thesis with a summary and outlines some suggestions for future work.

## Chapter 2

### Literature Review

#### 2.1 Overview of Mobile Ad Hoc Networks (MANET)

MANET is essentially a cooperative-based network where nodes may rely on each other to route their messages. As an example, Figure 2.1 illustrates a MANET that consists of five mobile nodes A to E, each having a communication link to any other nodes that are within its radio transmission range. Hence a direct communication between nodes A and B, and between B and C is possible. However, for two nodes that are beyond the range of one another, a *multihop* path that consists of one or more other nodes to serve as relay (router) between them is necessary. For instance, although no direct link exists between nodes A and C, but A could communicate with C through B, which helps to relay traffic between A and C. Alternatively, A could communicate with C through D and E, although this path is longer and as such, might increase the time needed by a message to traverse end-to-end between A and C.

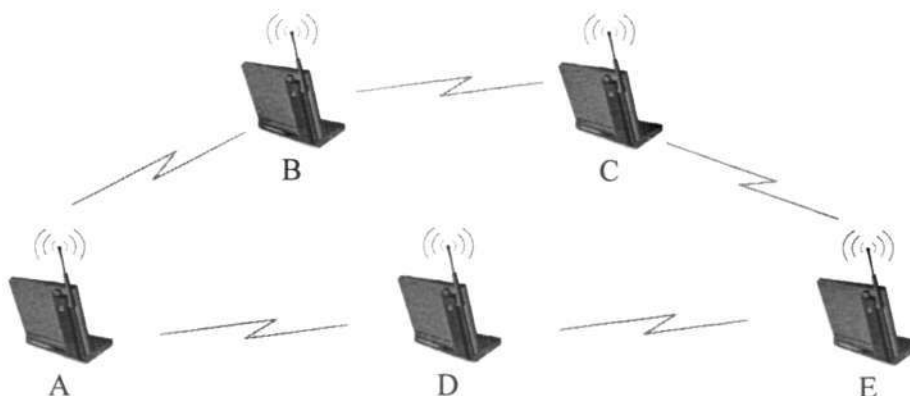


Figure 2.1: Example of a MANET

Since nodes are typically free to move about arbitrarily, the *mobility* of these nodes may cause frequent changes in the network topology (interconnection pattern between nodes). As an example, if B moves beyond the range of A, causing A to lose its link to B, then A would have to route through D and E in order to maintain its communication with C. Similarly, the state of physical links between other nodes, e.g. between C and E, or A and D, may change over time with mobility, thus requiring messages to be frequently rerouted through the other nodes. Hence, *routing* is an integral part of any MANET deployment, and minimizing the control traffic to setup and maintain routes has been one of the key research objectives in mobile ad hoc networking.

One of the original motivations for mobile ad hoc networks is found in the military need for battlefield survivability [8-10]. To survive under battlefield conditions, soldiers and their mobile platforms must be able to move swiftly and freely without any restrictions imposed by wired communication devices. Hence, the need for battlefield survivability translates into a mobile and wireless communications system that operates in a distributed manner for coordinated group actions. Some regions such as deserts and jungles have no terrestrial communications infrastructure. In other regions, the infrastructure is available but unusable due to its destruction or damage by war. A communications system such as MANET, which is both wire-free and rapidly deployable, is therefore highly desirable in military battlefield scenarios.

Another motivation stems from the need for energy-efficiency, which derives from the constraints imposed by limited battery capacity of portable mobile devices. It has long

been shown that using several short intermediate hops to send an information bit is more energy-efficient than using one longer hop [11]. The multihop approach used in MANET for transmission can thus reduce the energy consumption of nodes, thereby minimizing node failures due to battery depletion. A lower transmission power also leads to reduced interference [12], thus improving the system capacity and the efficiency of spectrum use. A third motivating factor is that a multihop MANET could provide an effective means for routing traffic around radio propagation obstacles such as tall buildings and towers, as well as for routing to destinations located in underground or indoor environments [13], which may otherwise be unreachable.

MANET is useful in applications that require mobile data exchanges in regions where there is little or no pre-existing infrastructure such as base stations available. Typical applications include military battlefield scenarios (as mentioned), emergency and rescue missions in disaster areas, or for communications in rural and remote areas where basic infrastructure is not well established. More recently, pervasive computing [14], wireless personal area networks (WPAN) [15], vehicular communications [16], and coverage and capacity enhancement of cellular networks [17], are some other applications of MANET that have attracted considerable interest.

## **2.2 Classification of Routing in MANET**

Traditional routing protocols such as Open Shortest Path First (OSPF) [18] were designed to operate in environments where network topology is relatively stable and bandwidth is generally in abundance. Each router is required to constantly maintain a complete and

accurate view of the network by continuously tracking network connectivity changes and regularly exchanging routing information with the other routers. As such, OSPF is known to not perform well in a wireless and mobile environment, where bandwidth is typically limited and easily saturated by the control traffic needed to maintain a full and up-to-date view of the network in the face of mobility. Thus, there have been many research efforts over the recent years to develop new protocols suitable for wireless routing application in MANET with dynamic topologies. Generally, the protocols proposed can be classified into one of the following categories: *proactive*, *reactive*, or *hybrid* routing.

In proactive routing, each node attempts to always know a route to all other nodes so that when a packet needs to be sent, the route is already known and can be used immediately. OSPF is an example of a proactive routing protocol for wired IP networks [19]. But as mentioned, the control traffic resulted may be too prohibiting for these protocols to work over air links of limited bandwidth. Thus, more efficient methods of routing information exchange have been proposed to enable proactive-style routing in resource-constrained MANET environment, e.g. [20, 21, 54].

By contrast, reactive routing does not require routine exchange of routing information for each node to always know a route to all other nodes. Instead, routes are established and maintained only if they are needed for communications, thus following an “on-demand” philosophy. Consequently, less control traffic is created, but the route acquisition time may be more variable than the constant table-lookup time associated with proactive protocols.

Dynamic Source Routing (DSR) [22] and Ad-hoc On-demand Distance Vector (AODV) [23] are well-known examples of reactive routing protocols.

Hybrid routing combines proactive and reactive routing approaches. For example in a suite of protocols [24-27] proposed for hybrid routing, each node *proactively* maintains routes to nodes within its local zone defined in terms of hop-radius, and *reactively* establishes routes to nodes beyond its zone if they are needed for communications. The rationale is that routes to the nearby nodes are expected to be more often utilized, e.g. for routing messages to distant nodes, thus justifying the need for them to be proactively maintained. Both proactive and reactive elements in hybrid routing may come from any proactive and reactive protocols. Hybrid routing is thus not a replacement but an approach for these protocols to be used in conjunction with each other.

In addition to the above so-called *topology-based routing* protocols, because they utilize physical link (or topology) information to make their routing decisions, there exists a separate class of *position-based routing* protocols that utilize only physical node location (or position) information for forwarding of data packets, e.g. [28, 29, 30, 31, 53].

Nodes using position-based routing do not acquire or maintain routing information. Instead they acquire and maintain *node location information*. The prerequisite for position-based routing is that each node must know the location of: i) *its own*, e.g. through the Global Positioning Systems (GPS) [32]; ii) *its one-hop neighbors*, e.g. through periodic HELLO exchanges; and iii) *the packet's destination*, e.g. through a flood-based search or querying

an external location service [33]. Using this information, a node with a given data packet typically selects and forwards the packet to a one-hop neighbor that would make the most progress towards the packet's destination.

## **2.3 Reactive Routing Protocols**

In this section, we present a further description of the reactive-class of routing protocols since it is the protocol class that we are primarily interested in, in this thesis. Among the reactive routing protocols, we pay particular attention to those chosen by the IETF MANET Working Group [34] for further development into Internet standards, namely DSR and AODV. We begin this section with an introduction to the general features of reactive routing common to both protocols, followed by a description of the distinctive features of each that differentiate one from the other.

### **2.3.1 General Features**

Reactive routing is based on the on-demand philosophy that protocol should react only if needed for communications (as mentioned in Section 2.2), and reactive routing protocols typically consists of two basic mechanisms:

- Route discovery; and
- Route maintenance;

Route discovery is performed whenever a source has a packet to send, but does not know the route to the packet's destination. The source initially broadcasts a Route Request (RREQ) message to all its immediate neighbors. Each neighbor on receiving the RREQ, checks whether it is the destination, or knows a route to the destination. If so, the node

unicasts a Route Reply (RREP) message back to the source, informing the source of the route to the destination. The RREP follows a path that is typically the reverse of the path followed by the RREQ. Otherwise, the node rebroadcasts the RREQ to its own neighbors, which in turn process it in the same manner. The RREQ thus propagates from one node to another and may, over time, be propagated over the entire network. Such a network-wide broadcast operation is typically known as *flooding*, which is not only resource exhaustive but also may disturb many nodes unnecessarily.

Individual nodes may expend more resources such as memory, CPU time and battery power to process these control packets. Frequent flooding potentially leads to serious bandwidth congestion, excessive packet collisions and contention in wireless medium [4], all of which degrade network performance.

Each node only processes a given RREQ once and discards duplicates of the same RREQ received from its neighbors due to the omni-directional nature of broadcast transmission. Nodes detect duplicate RREQ by tracking and comparing packet ID and source address of each received RREQ. To prevent endless rebroadcasting of RREQ, the number of hops traversable by a RREQ is also typically limited to the maximum expected route length. A RREQ having reached its hop-limit upon arrival at a node will be automatically dropped.

Once the source receives a RREP, it sends the packet immediately to the destination using the route obtained. Route maintenance is then carried out to detect any link breaks, e.g. due to node mobility, in the route in use. Each node on the current route is responsible for

sensing whether the link to its next-hop is broken. If so, the node unicasts a Route Error (RERR) message back to the source, informing the source of the link in error. The source immediately stops sending any more packets using the faulty route and may attempt a route discovery to obtain a new route to the destination.

### 2.3.2 Dynamic Source Routing (DSR)

As mentioned, DSR [22] is a candidate reactive routing protocol targeted by the IETF for standardization. The protocol discovers and maintains routes as described in Section 2.3.1 based on the concept of *source routing* [35] – a technique whereby each packet that is sent by the source carries the complete route needed to reach the destination. The route contains a sequenced list of nodes to be followed from source to destination.

The RREQ that is sent by the source carries a *route record* that initially contains only the source address, but builds up the route to the destination as it propagates. An intermediate node upon receiving the RREQ, checks whether it knows a route to the destination. If so, the route is appended to the route record (thus giving the full path), which is then carried back to the source by the RREP. Otherwise, the node appends its own address to the route record before rebroadcasting the RREQ to its neighbors.

A key advantage of source routing is that all routes are easily guaranteed to be loop-free, i.e. no duplicate hops, since the entire sequence of nodes from source to destination is available. Its main disadvantage is that packet size, i.e. overhead bytes, can grow due to the inclusion of source routes. However, on the other hand, the carrying of source routes

allows the intermediate nodes to learn not only a route to the destination, but also a route to other nodes listed in the source route. This routing information can be saved for future use by the nodes, thus avoiding the need to perform some route discoveries, which in turn reduces the amount of overhead packets, e.g. RREQ and RREP.

In DSR, each node maintains a *route cache* to store the source routes that it has learned. A further important feature of DSR is the support for *multiple routes* to any destination, through two provisions. First, the destination of the route discovery is allowed to reply to multiple copies of the RREQ that arrived via different routes, unlike intermediate nodes that only reply to the first arrived RREQ. This is to allow the source to acquire multiple routes to the destination, from the destination itself, when no intermediate nodes reply.

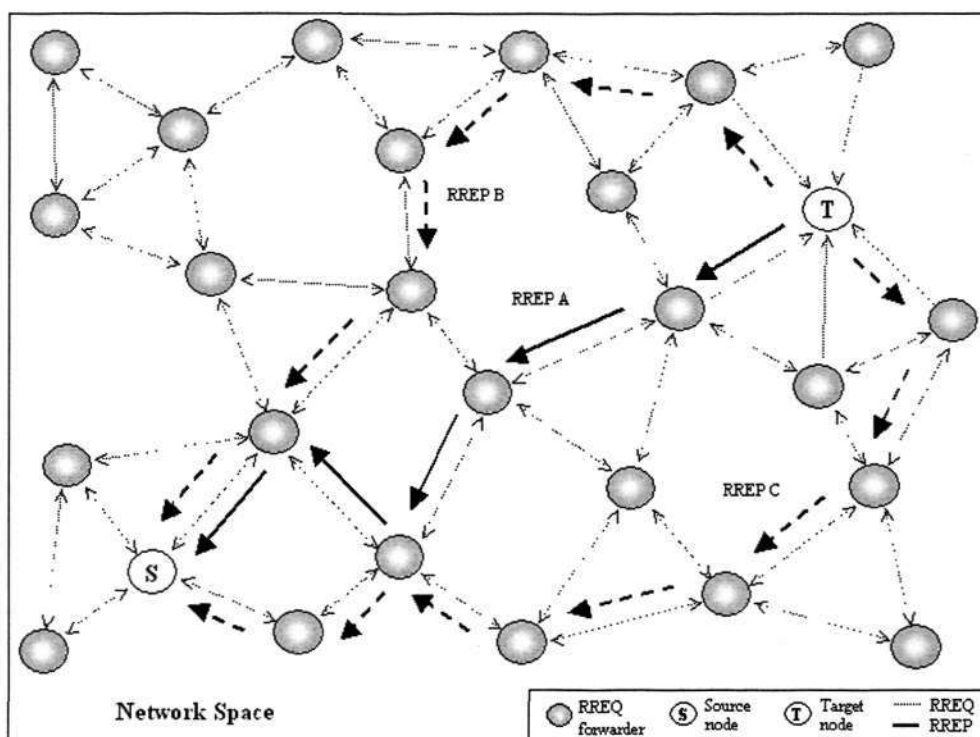


Figure 2.2: Discovery of multiple routes to a destination by DSR

Second, the source is allowed to maintain multiple route cache entries per destination, with each entry storing a route carried back by the RREP. An example in which a source  $S$  obtains multiple RREP from destination target  $T$ , each going through (and carrying) a different route, is shown in Figure 2.2. In DSR, with multiple routes, the source typically selects the shortest route (with least hop-count) to send its data packets to the destination. This support for multiple routes can allow a node to react much more rapidly to routing changes by immediately switching to using a different cached route if the one it has been using breaks. Furthermore, overhead packets can be saved by not having to perform a new route discovery every time a route in use fails.

Some other distinctive features of DSR include: i) *packet salvaging*: a process for data packets that cannot be forwarded over a broken link to be reroute (rather than dropped) using an alternate path from the forwarding node; and ii) *automatic route shortening*: a feature for source routes in use to be shortened when nodes move such that one or more intermediate hops are no longer necessary.

### **2.3.3 Ad-hoc On-demand Distance Vector (AODV)**

AODV [23] shares many features with DSR, e.g. similarly using RREQ, RREP and RERR messages for route discovery and maintenance, but uses a different approach to managing routing information. Unlike DSR that is based on source routing, AODV uses traditional *hop-by-hop routing* [36], where each node with a packet to send looks up its *routing table* for the next-hop to which the packet is to be forwarded in order to reach the destination. The routing table maintains one entry per destination, meaning each node would have at

most a *single route* to any destination. Each entry contains a distance vector composing of next-hop and hop-distance to the destination.

Each intermediate node that is forwarding a RREQ creates a *reverse route* for itself back to the source. The destination does not need to reply to all but the first arrived RREQ, since the source only maintains one route for each destination. When the RREQ reaches the destination or an intermediate node with a route to the destination, it unicasts a RREP to the source using the reverse route created. Each node that forwards this RREP to the source then creates a *forward route* to the destination.

In AODV, each node maintains a *sequence number* for itself which the node increments and attaches to each RREQ and RREP it originates. Each node also maintains information about the sequence number of other nodes in its routing table. Upon receiving a routing packet, the node compares the packet's sequence number with that from its table entry for the packet sender. The routing table is only updated if the packet's sequence number is: i) higher; or ii) equal to that in the table, but the new route in the packet, i.e. RREP, is shorter. This prevents updating of routing table with stale or unuseful information and the formation of loops that cause packets to be circulated indefinitely.

## 2.4 DSR Optimizations to Route Discovery

As previously noted (in Section 2.3.1), the basic reactive route discovery process is based on a simple but resource-exhaustive flooding mechanism. This section overviews three

techniques proposed by the DSR's authors for optimizing DSR's route discovery process. In this thesis, we call them the DSR's *native* optimization techniques:

### 2.4.1 Caching Overheard Routing Information

Recall that carrying of source routes in the packets may allow a node *on the route* to cache routing information to other nodes that lie on the same route. However, it is also possible for nodes *not on the route*, but lie close to those that are, to overhear and cache this routing information. This is achieved if nodes can operate their network interfaces in promiscuous mode, which allows them to acquire the routes from the headers of packets addressed to their neighbors. For example, suppose source  $S$  sends a packet to destination  $D$ . The packet carries the source route  $S-B-C-D$ . A node  $E$ , which is a neighbor of  $B$ , then overhears  $B$ 's transmission to  $C$ . If  $E$  knows it has a forward link to  $B$  (i.e. link between  $E$  and  $B$  is bi-directional),  $E$  can cache a route to  $D$ , e.g.  $E-B-C-D$ . Hence, this increases the amount of routing information a node can learn and save for its future use, and (as mentioned in Section 2.3.2) further avoiding the need to perform some route discoveries, which in turn further reduces the amount of overhead packets, e.g. RREQ and RREP.

### 2.4.2 Non-Propagating Route Requests

The "Time-To-Live" (TTL) field in the IP header is proposed to limit the number of hops for which a RREQ is allowed to traverse from the source. DSR uses this RREQ hop-limit to implement a two-phase route discovery. In the initial phase, the source conducts what is called a *Ring Zero* search, in which a "non-propagating" RREQ whose TTL = 1 (one that does not traverse beyond 1 hop from the source) is broadcast to query only its

immediate neighbors for the destination, or a route to the destination. If no RREP is returned within the nominal 1-hop Round Trip Time (RTT), the source proceeds to broadcast a “propagating” RREQ (whose TTL is typically set to the maximum expected route length) that floods the entire network. The prior use of a “non-propagating” RREQ potentially incurs a slight cost (small delay of 1-hop RTT) in exchange for a large saving on overhead packets by avoiding the need to flood in every route discovery attempt.

### **2.4.3 Preventing Route Reply Storms**

Due to the broadcast nature of wireless transmission, many immediate neighbors of a broadcasting node may receive the RREQ and send RREPs simultaneously. This may result in what is dubbed a RREP “storm”, in which a great number of intermediate nodes attempt to reply from their caches simultaneously, causing local congestion and excessive packet collisions. To relieve this problem, DSR introduces a delay time for each replying node to send its RREP. The delay time is specified to be:  $d = Hx(h - 1 + r)$  where  $h$  is the hop-length of the returned source route,  $r$  is a random number between 0 and 1, and  $H$  is a small constant delay to be introduced per hop. Using this expression, a replying node with a shorter route (smaller  $h$ ) shall send its RREP earlier. If during the delay period, this node receives a data packet (of the same source-destination pair) that uses a source route of length  $\geq h$ , inferring the source has already received a RREP giving an equally good or better route, this node can abort sending its RREP for this route discovery.

## **2.5 AODV Optimizations to Route Discovery**

### **2.5.1 Expanding Ring Search**

Recall that DSR uses the RREQ hop-limit to implement a two-phase route discovery. Similarly, AODV uses this hop-limit to implement a *multi-phase* expanding ring search. Concisely put, expanding ring search is a technique in which the source searches in increasingly larger neighborhood, i.e. sending out successive RREQ, each with a TTL larger than the one used in the previous attempt. The source initially searches only the region within some limited hops from itself. If no RREP is returned after timeout, the source attempts another search, but with a wider search scope by increasing the TTL of the RREQ. This process continues until the TTL reaches a maximum threshold, after which the RREQ is flooded throughout the network. Expanding ring search is beneficial if the desired route can be found within some initial attempts. Otherwise, the resulting overhead and latency of route discovery can be even higher than that incurred by flooding on the first attempt, since multiple route discovery attempts and timeouts may be required before a route to the destination can be found.

### **2.5.2 Local Repair**

In the event of a break in a currently used route, the intermediate node upstream of the link in failure may attempt to “locally” repair the route, instead of relying on the source node to rediscover a route to the destination. This process is known as “local repair” and is carried out by the intermediate node, which sends a RREQ whose TTL is set to the remaining hop-distance to the destination, added with an increment value. Data packets are buffered at this node during the local repair and sent once a RREP containing a new route from itself to the destination is returned. However, if no RREP is returned after a timeout, the buffered data packets are dropped, and a RERR is sent to the source, which

then attempts a source-initiated route discovery. Local repair is useful if the link failure occurs closer to the destination than to the source. By performing route discovery from the “point-of-failure” instead of from the source, the amount of control traffic created can be reduced. More data packets can also be delivered since they are not dropped due to the link failure. However, frequent repairs can cause a route to be prolonged, leading to sub-optimal performance, e.g. in packet end-to-end delay.

## 2.6 Location Aided Optimizations (LAR) to Route Discovery

LAR [5] is a technique that uses the location of nodes to restrict the area of flooding RREQ to a smaller region. The prerequisites for employing LAR in route discovery are: i) each node must know its own location; and ii) the source must know the destination’s location. GPS [32] is assumed to provide each node with its own location. It is further assumed that the source initially does not know the destination’s location and thus uses a simple flooding-based route discovery to obtain a route to the destination. However, when the destination replies to the source, it includes within the RREP its own location, which is stored by the source for future route discovery using LAR. Two variants of LAR are proposed: Scheme 1 (LAR-1) and Scheme 2 (LAR-2):

### 2.6.1 LAR Scheme 1

Figure 2.3 illustrates the LAR-1 scheme. From knowing location  $L$  and average speed  $v$  of the destination target at time  $t_0$ , the source can define at time  $t_1$ , a circular region of radius  $v(t_1 - t_0)$  known as the “expected zone”, which is an area where the destination is expected to be found. The source next defines the smallest rectangle that includes both

the expected zone and itself as the “request zone”, whose coordinates (defining the four corners of the rectangle) are sent along with the RREQ. Only nodes inside the request zone may forward the RREQ. Others discard them when they are received. If a route to the destination is not found within a timeout, the source falls back to flooding to perform a new route discovery to the same destination.

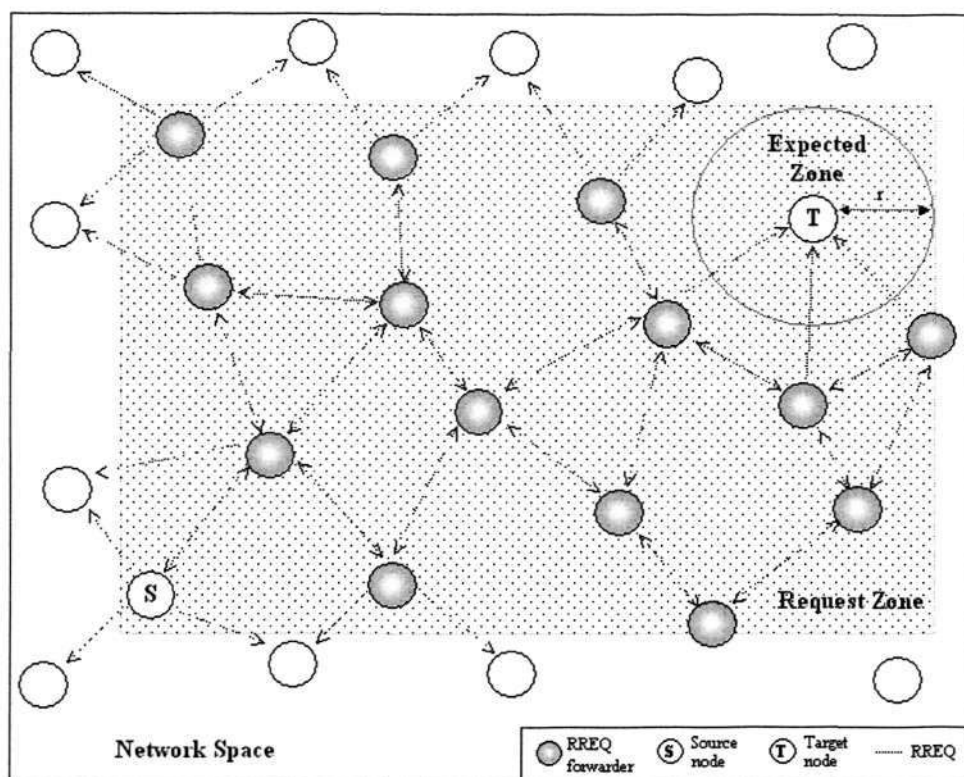


Figure 2.3: LAR-1: concepts of request and expected zones

## 2.6.2 LAR Scheme 2

In LAR2, the source uses location information to compute its physical distance to the destination, and then attaches this distance value, and the destination location into the

RREQ. An intermediate node that receives this RREQ computes its own distance to the destination, and compares with the distance value in the RREQ. If the node finds that it resides closer to the destination than the node from which it receives the RREQ, then it overwrites the previous distance value with its own value and forwards the RREQ. Otherwise, the RREQ is discarded. Similarly, if a route is not found upon timeout, a new route discovery is initiated using flooding. Figure 2.4 illustrates the LAR-2 scheme: The RREQ broadcast by source  $S$  is received by nodes 1 to 4, but only nodes 1 and 2 forward the RREQ because they are closer to target  $T$ , than is from  $S$  to  $T$ . Hence, RREQ only gets progressively closer to the target upon each forwarding and not further away from it.

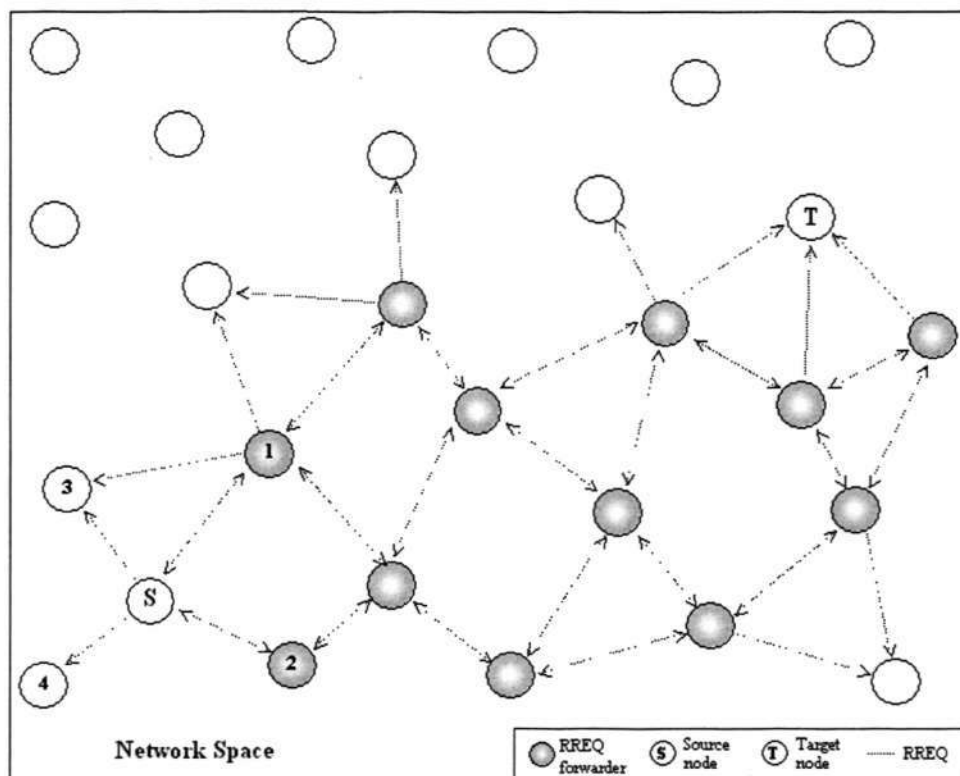


Figure 2.4: LAR-2: progressive forwarding of RREQ

The notion of progressive forwarding in LAR-2 is similar to position-based routing, as mentioned in Section 2.2. However, position-based routing uses location information for data delivery, while LAR uses it for route discovery. Another difference is that LAR does not require nodes to know the location of their one-hop neighbors, thus does not require nodes to regularly exchange their location information, e.g. using HELLO messages, unlike position-based routing.

### 2.6.3 Further Optimizations

Authors of the LAR [5] further suggested (but did not experimentally evaluate) a number of ways to enhance the overhead performance of the two basic LAR schemes. They include:

i) Alternative definition of request zone; ii) Adaptation of request zone; iii) Propagation of location and speed information; and iv) Local search.

- *Alternative definition of request zone:* It is proposed that shape of the request zone for LAR-1 may take an alternative form. For example, instead of defining it as a rectangle with sides parallel to the XY axes as shown in Figure 2.3, the alternative request zone is defined as a rectangle whose long and short sides are parallel and perpendicular respectively, to an imaginary line connecting source and destination (see figure 14 in [5]), resulting in a smaller request zone. It further proposed an “expanding zone” strategy (similar to expanding ring in Section 2.5.1) to increase the request zone size *gradually*, following each unsuccessful LAR attempt.
- *Adaptation of request zone:* Both LAR schemes can adapt their operation to more updated destination location information at intermediate nodes. For example in

LAR-1, an intermediate node receiving the RREQ can adapt the request zone to include the newer expected location of the destination. In LAR-2, the node can calculate distances from the more recent location of the destination. The existing request zone/distance information in the RREQ is then replaced with the adapted information by the node before it rebroadcasts the RREQ.

- *Propagation of location and speed information:* Each node can propagate its location and speed information to other nodes by piggybacking such information on any packet (data or routing packet), which it originates or forwards for others. The acquisition of these information by the nodes shall allow them to carry out more route discoveries using the LAR technique.
- *Local search:* An intermediate node that detects a broken link in a currently used route can conduct a local route pair (similar to local repair in Section 2.5.2) using LAR. This may result in a smaller request zone, compared to one that is conducted from the source. As a consequence, route pair time and overhead may be reduced.

## 2.7 Other Related Work

The problem of uncontrolled flooding and its effects on network performance have been well-studied in literature. Prominent among the works is a classical study of the so-called *broadcast storm* problem in MANET [4, 37], where the authors described a phenomenon of excessive contention and collisions experienced during the rebroadcasting of messages. Several methods for reducing broadcast redundancy and overhead of broadcast protocols

are presented, including: *probabilistic*, *counter-based*, *distance-based*, *location-based*, and *cluster-based* schemes.

In the probabilistic scheme, each node rebroadcasts the first copy of a received message with a probability  $p$ . The counter-based scheme inhibits the rebroadcast if the message has been received for  $> C$  times. In the distance-based scheme, the message is rebroadcast only if the distance between the current node and each of its neighbors from which it receives the message is  $>$  threshold  $D$ . The location-based scheme rebroadcasts the message if the extra coverage area by this new emission is  $>$  threshold  $A$ . Finally, in the cluster-based scheme, only nodes designated as cluster-heads or gateways rebroadcast the message using any of the above four methods.

The problem of flooding is also studied in [60], where the author proposed a scheme called Position-based Selective Flooding (PSF). Similar to LAR, geographic position is used for selecting a subset of nodes to forward a broadcast message. However unlike LAR, PSF does not propagate the message *towards* any specific direction (e.g. towards a destination), but only propagates *away* from source.

MPR flooding [61] is another scheme proposed to reduce the overhead of message broadcast. In MPR flooding, a broadcasting node selects from its 1-hop neighbors a set of multi-point relays (MPRs) that can ensure a rebroadcast by these nodes will allow its message to reach all nodes in its 2-hop neighborhood. Upon receiving, the MPRs will also select their own MPR set for the next rebroadcast, and so on, until the message is heard by the entire network. This

is an innovative scheme, but it requires each node to proactively maintain 2-hop neighbor information, which can be quite costly, especially in node high density.

It is worth recalling that the objective of a broadcast protocol is to disseminate any given broadcast message to the entire network. The broadcast schemes proposed in [4] seek to accomplish this objective with minimal retransmission frequency, i.e. seeking to allow a broadcast message to be received by all nodes, while using a minimal number of nodes to rebroadcast the message. However, in route discovery, it may not be necessary to always broadcast the RREQ to the *entire* network in order to seek a route to the destination. Thus, unnecessary message overhead may still be incurred, in addition to the energy needed for each node to receive and process these messages.

Another approach to reduce flooding well cited in literature is *query localization* [38]. Here, routing histories are used to localize the RREQ flood to a limited region in the network. Two heuristics: *path-* and *node-locality*, are proposed that work on the principle that many parts of a recently broken route may still be used to rebuild a new route to the destination. The path locality expects the new route to be no more than  $k$  nodes different from the older route. The node locality expects the destination to be within  $k$  hops away from any node on the older route. Each RREQ carries a counter that increments to keep track of its locality (path or node) differences. The RREQ is dropped when the counter exceeds the threshold value  $k$ . Essentially, this technique is used for *route pair* soon after a route failure. Thus, the route prolonging problem in local repair (Section 2.5.2) can similarly happen here. For a route discovery with no route to the destination is previously known,

RREQ will still be flooded to the entire network. It is also envisaged that this approach is more useful to routing protocols with only a single route per destination, e.g. AODV, but less effective to those with multiple routes, e.g. DSR, since the alternate routes available would obviate the need to carry out route repair.

Finally, some recent works on efficient route discovery could be found in [62] and [63]. FRESH [62] is a scheme that works on the notion of 'encounter ages', i.e. it assumes a node that encounters a destination more recently, should also be spatially closer to that destination. Thus, instead of flooding, a node may direct its message to another node that has seen the destination more recently than itself. This assumption may be true if all nodes travel at about the same speed with time. However, if nodes can travel at different speeds, or even remain stationary, then the time-space assumption may be violated.

Different but similar in spirit to FRESH is a scheme called CBRD [63]. The scheme replaces a single network-wide search by several local searches that involves locally flooding a query to find a node that has closer distance to the destination than itself. This node in turn does another local flood to find the next closer node towards destination, and so on. It is noted that the proposed scheme requires a periodic exchange of distance information among nodes, and this may constitute a significant overhead in the case of dense networks.

## Chapter 3

# UNIQUE Optimization for RREQ Reduction

### 3.1 Introduction

This chapter presents the first optimization technique proposed in this thesis, called the UNIQUE technique for RREQ reduction. Our design technique is primarily motivated by the observation that the existing LAR [5] technique can still incur a high overhead if the source and target of the route discovery are physically far from each other, leading to a large request zone that renders the LAR less effective. This observation along with the intuition that the efforts and resources expended to discover existing routes should be more effectively utilized, provided us the impetus to propose the UNIQUE technique. As in LAR, the technique proposed herein is location-based and can be used in conjunction with existing LAR and other native optimizations of (reactive) base routing protocols to further decrease the overhead of route discovery.

#### 3.1.1 Route Discovery with UNIQUE

We begin with the rationale behind our proposal: For a network with moderate to high node density, each node can be expected to have a sizeable number of neighbors, i.e. a high node degree. Intuitively, even though a source  $S$  may not yet have a route to some target  $T$ , it is however likely that  $S$  will have a route to at least one of many nodes near  $T$ , especially when some traffic involving such nodes is already flowing. Then  $S$  can utilize this route to first ‘transport’ (unicast) the query closer to  $T$  before launching a localized

broadcast-based search for  $T$ , e.g. by LAR. Thus, compared to the original LAR scheme, the forwarding of RREQ first as an unicast message to the target neighbor in UNIQUE can result in fewer nodes being disturbed as only those on the route and in the vicinity of  $T$  will participate in the route discovery, as shown in Figure 3.1.

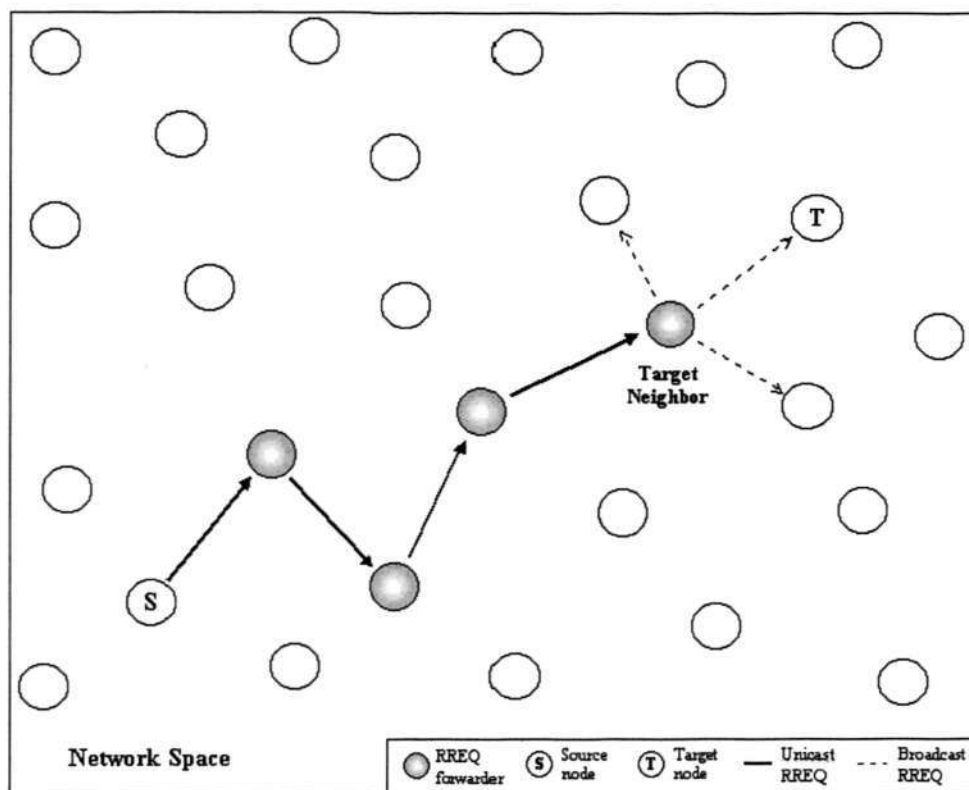


Figure 3.1: UNIQUE: RREQ forwarded to target neighbor by unicast

To illustrate this concept by example, we first assume  $S$  knows its own location, the location of  $T$  and those of nodes to which it has a route. As in position-based routing, a node can learn its *own* location through GPS [32] or non-GPS methods [39-41]. Location of other nodes can be known by, e.g. the location propagation mechanism proposed for LAR [5]. Alternatively, distributed location servers [42] can be deployed.

With knowledge about the location of nodes of interest,  $S$  can make an informed selection of an existing route (from its route cache or routing table) to *unicast* a query (RREQ) to some node  $V_i$  whose location is closest to  $T$ , and which must be closer than  $S$  itself. Expressed in more formal terms, if  $\text{dist}(u, v)$  is the distance between  $u$  and  $v$ , then  $\text{dist}(V_i, T) = \min[\text{dist}(V\{i, j, k, \dots\}, T)]$ , and  $\text{dist}(V_i, T) < \text{dist}(S, T)$ , where  $V$  is a set of nodes  $\{i, j, k, \dots\}$  to which  $S$  has a route, and  $V_i \in V$ . We call this node *target neighbor* (TN) and the route selected *unicast query path* (UQPath).

Each node visited by the RREQ will be queried for a route to  $T$ , and when found, the route is returned to  $S$  in a RREP and route discovery ends. If not, the RREQ is *unicast* to the next hop on the *UQPath*. This repeats until the RREQ reaches *TN*. If still no route is found, the RREQ is *broadcast* but only in the direction of  $T$  in a manner similar to LAR. The pseudo-code for UNIQUE is presented in Figure 3.2. Note however, that since this directed broadcast is only performed within close vicinity of the target (from *TN*), the overall fraction of nodes that will receive the RREQ is almost certainly lower compared to LAR where RREQ is broadcast from  $S$ . This translates into fewer RREQ and RREP messages that will be generated in a route discovery.

In the event that the *UQPath* is invalidated as a result of node movements, the node upstream of the link in error may still forward the current RREQ if it has an alternate *UQPath* to the vicinity of  $T$ . Otherwise, it informs  $S$ , which then proceeds to issue a new RREQ using the next preferred choice of technique, which can be UNIQUE if a different *UQPath* exists, or the original LAR scheme as  $S$  already knows about  $T$ 's location, or

simply flooding. Note that our proposal does not define the exact ‘fall back’ solution to use or number of stages that route discovery is allowed to perform, as these are more a matter for a full protocol than for an optimization technique. However, we shall define them when we discuss how various techniques considered in this study are applied to a base routing protocol in Section 3.2.

```

UNIQUE OPTIMIZATION TECHNIQUE

Let  $S$  and  $T$  be the RREQ's source and target node respectively
Let  $TN$  be the closest neighboring node of  $T$  (known by  $S$ )
Let  $UQPath$  be the shortest known route from  $S$  to  $TN$ 
Let  $V$  be the set of nodes to which  $S$  has a route

// At source node
When a route discovery to  $T$  is required:
    Check route cache to determine if an  $UQPath$  to node  $V_i$  exists
        where  $\text{dist}(V_i, T) = \min[\text{dist}(V\{i, j, k, \dots\}, T)]$ 
        and  $\text{dist}(V_i, T) < \text{dist}(S, T)$ 
    If ( $UQPath$  exists)
        Unicast RREQ to  $V_i$  // i.e.  $TN = V_i$ 
    Else
        Perform route discovery by an alternative technique
        e.g. Ring-Zero, LAR, etc.
    Return

// At intermediate nodes
When an unicast RREQ  $q$  is received:
    Check route cache to determine if a route  $r$  to  $T$  exists
    If ( $r$  exists)
        Send RREP with  $r$  to  $S$ 
    Else If (current node is not  $TN$ )
        Unicast  $q$  to next-hop on  $UQPath$ 
    Else // if current node is  $TN$  (i.e. end of  $UQPath$  reached)
        Broadcast  $q$  in the direction of  $T$ 
    Return

```

Figure 3.2: Pseudo-code for UNIQUE

### 3.1.2 UNIQUE with Query Redirection

Generally, it can be expected that as the RREQ edges closer to the target, more updated location of the target will be available, which should be utilized to redirect the RREQ if necessary in order to minimize the impact of any less-than-current information about the target's location at the source. For example, suppose the RREQ carries the location tuple  $(x, y, t)$  of the target, where  $x$  and  $y$  is the longitude and latitude of target respectively, while  $t$  is the time at which this information is created. When a certain node  $n$  receives this RREQ, it checks if a more recent location of the target is available by comparing their timestamps (or alternatively use sequence numbers as in [23] if no timing synchronization among nodes is assumed, to indicate the freshness of node location information received). If so,  $n$  can redirect the RREQ to a different target neighbor (to which it has a route) whose location is closest to the newer target location. This feature will be useful in some real implementations where only approximate locations are known, and where locations could change over the course of route discovery.

## 3.2 Protocol Description

The following sections describe how the UNIQUE and LAR techniques are applied to optimize a reactive base routing protocol. We considered both IETF MANET protocols DSR and AODV, and subsequently selected DSR as our base routing protocol because it consistently demonstrated a lower routing load than AODV in previous studies [43, 44], primarily due to several native optimization techniques it employs to improve efficiency (Section 2.4). A DSR protocol used with these optimizations will thus provide a more

challenging base than a simple flooding protocol (as used for comparison in [5]) to evaluate the efficacy of UNIQUE and LAR. Three protocols are being studied in this chapter. They are the (natively optimized) DSR protocol, DSR protocol optimized with LAR (DSR+L), and DSR protocol optimized with both UNIQUE and LAR (DSR+L+U). The route discovery features of these protocols are highlighted in the following.

As mentioned, the DSR protocol in this study is used with all its native optimizations, including the route discovery optimizations listed in Section 2.4. There are two stages to the route discovery. In the initial stage, the source performs a 1-hop route discovery using *Ring Zero* search (Section 2.4.2). If it does not receive a RREP within a timeout period, the source resorts to flooding the RREQ through the entire network.

### 3.2.1 DSR+L

DSR+L is the combination of the DSR protocol with a LAR scheme, or specifically the LAR-2 scheme. We chose to apply LAR-2 because it performed better than LAR-1 in our preliminary studies, concurring with the results reported in [5]. In DSR+L, the LAR is used when the initial route discovery attempt by Ring-Zero search is unsuccessful. If this follow-up attempt by LAR is still unable to discover a route to the desired destination, the protocol resorts to network-wide flooding.

In addition to implementing the basic LAR scheme, we evaluated all other optimizations (except the alternative definition of request zone) described in Section 2.6.3. Of the three evaluated optimizations, namely i) adaptation of request zone; ii) propagation of location

information (speed information is not needed by LAR-2); and iii) local search (LS); we chose to exclude LS in our implementation as it does not seem to improve the protocol performance. The reason may be understood by studying the packet dropped summary for LAR as shown in Table 3.1, with and without LS.

It can be seen that most data packets are dropped either because of *No Route* or *IFQ full*. IFQ refers to the node interface queue, and *IFQ full* is often an indication of local congestion. Expectedly, fewer packets are dropped due to *No Route* when LS is used since packets in error (those sent to a broken link) can be rerouted via another path found using local search, and not simply dropped when LS is not used. However, LS results in more packets dropped due to *IFQ full* or congestion, bringing the total packets dropped to exceed that without LS.

Table 3.1: Data packet dropped summary for LAR with and without LS (50 nodes, 20 sources, 4 packets/s, 64 bytes, 1500m x 300m)

Data Packet Dropped Summary	LAR	
	without LS	with LS
No Route	872	264
TTL Expired	0	0
RTR Queue Full	0	0
Timeout	69	77
Routing Loop	0	0
IFQ Full	1224	3034
ARP Full	50	92
MAC Callback	0	0
Simulation End	43	35
Total Packets Dropped	2258	3502

Table 3.2 further shows that LS generates more routing packets than without LS, which explains why it has higher congestion. The reason for the lower routing load generated in the case without LS, is because the source does not perform route discovery each time a

route in use fails, since it can switch to using another route to the same destination, given that DSR (which supports multiple routes per destination) is the base routing protocol used in our LAR implementation.

Table 3.2: LAR performance results (with and without LS)

Metrics	LAR	
	without LS	with LS
No. of route discoveries	856	1299
Route request packets	25,241	33,177
Route reply packets	31,662	43,219
Total route discovery packets	56,903	76,396

To propagate location information, each node piggybacks its own location onto packets that it sends or forwards. Since DSR+L uses *source routing*, the packet header carries the address with the location of nodes listed on the source route. Any node receiving the packet can learn the location of *every* other node through which the packet has visited. Such location information can be learned not only by nodes on the route, but also other nodes in the surrounding area through promiscuous listening. Initially, the performance of DSR+L is similar to DSR, as the location of other nodes may not yet be known. However, as time progresses, more information about others' location would be gathered, resulting in more route discoveries to be performed using the LAR scheme.

### 3.2.2 DSR+L+U

DSR+L+U is an improvement over DSR+L. It takes a step further to reduce the resources required for a route discovery by incorporating the UNIQUE technique. Figure 3.3 shows the algorithm used by the protocol to obtain a route. Notice that UNIQUE is used as the

first choice of technique for route discovery instead of Ring Zero. The reason for this is that in UNIQUE, only the next addressed node on the *UQPath* will react (reply) to the RREQ, whereas in Ring Zero, potentially all nodes in the range of the source will react, leading to a local congestion and higher routing overhead. With this next node on the *UQPath* likely to lie closer to the target than the source itself, the node would possibly return a route if one exists in the source's neighborhood. The source is thus able to acquire a route with both minimal overhead and latency. Especially in dense networks, UNIQUE should be the preferred choice compared to Ring Zero.

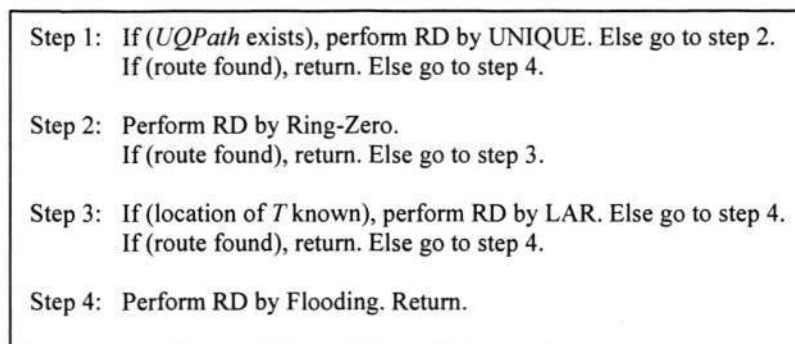


Figure 3.3: Route Discovery (RD) algorithm for DSR+L+U

Notice also that if a route cannot be found upon timeout using UNIQUE, flooding will be performed instead of LAR. This design choice is made based on our observation that LAR usually fails following an unsuccessful attempt by UNIQUE, which is attributed to some misinformation about the target location. If flooding is performed immediately after UNIQUE, we could reduce not only the delay, but also the overhead of route discovery that would otherwise be higher if flooding follows an unsuccessful LAR.

### 3.3 Simulation Model and Methodology

We conduct our experiments using VINT *ns-2* network simulator [6] with MONARCH wireless extensions [45]. The simulator provides a detailed model of the physical and data link layer behaviors of a wireless network. The physical layer includes a channel and radio propagation model supporting propagation delay, capture effects, and carrier sense. The link layer properly models the wireless medium contention, channel reservation and interference. A network of 50 nodes placed randomly within a 1500m x 300m area is simulated for 900s of simulation time

#### 3.3.1 Channel and Radio Model

The simulator uses a combination of free space propagation and two-ray ground reflection models [46] to properly model the attenuation of radio waves between the antennas close above the ground. When a transmitter is within a certain *reference distance* of the receiver, the free space model in which the signal attenuates as  $1/r^2$  (where  $r$  is the spatial separation between the antennas) is used. Beyond this reference distance, the simulator uses the ground reflection model where the signal falls off as  $1/r^4$ . The power level of a received signal is compared to two different values: the *carrier sense threshold* and the *receive threshold*. If the received power level is below the carrier sense threshold, the packet is discarded as noise. If the power level is above the carrier sense threshold but is below the receive threshold, it is marked as a packet in error. Otherwise, the packet is simply handed up to the MAC layer. The radio interface uses characteristics similar to the commercial WaveLAN [47] radios. The transmission range for each node is about 250m.

### 3.3.2 Medium Access Control (MAC) Protocol

In order to accurately model the contention of nodes for the wireless medium, the link layer of the simulator includes a simulation model of the complete IEEE 802.11 wireless LAN standard [48] Distributed Coordination Function (DCF) MAC protocol. DCF is an access scheme used by mobile nodes to share the wireless channel under independent peer-to-peer configuration. The access scheme is based on Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) [49], which uses both physical and virtual carrier sense to reduce the probability of collision by random backoff when senses the channel is busy. The virtual carrier sense is implemented by an exchange of short control packets called Request-to-Send (RTS) and Clear-to-Send (CTS) between the sender and receiver nodes before the actual packet transmission [50, 51]. Other nodes hearing this RTS/CTS exchange defer their transmission for a time period necessary for the sender node to send its packet. Each correctly received packet is followed by an Acknowledgement (ACK) by the receiver to the sender. If the sender does not receive an ACK, or if it fails to receive a CTS after transmitting a RTS, the sender will retry sending several times before giving up and reports a link breakage to its upper layer.

The 802.11 MAC does not apply the RTS/CTS exchange and ACK described above to broadcast transmissions since this will lead to CTS and ACK “implosion”. Instead, they are mainly used in the transmission of a unicast packet with a specified neighbor as the MAC destination. Examples of unicast packets include data packets and routing control packets such as RREP and RERR in the DSR protocol. RREQ packets, on the other hand, are typically treated as broadcast packets in the MAC.

### 3.3.3 Traffic and Mobility Pattern

The traffic pattern is consisting of 20 constant bit rate connections with sources and destinations randomly selected from all nodes. Each source sends data packets at a rate of 4 packets per second, each packet with a data payload size of 512 bytes. All connections start at times uniformly distributed during the first 180 seconds of simulated time and remains active until the simulation ends.

Each node moves according to a mobility pattern based on the random waypoint model [52], which is characterized by a *pause time*. The value of the pause time determines how frequent the nodes move, which in turn affects the amount of topology change. Each node begins the simulation by remaining stationary for *pause time* seconds. It then randomly selects a location inside the simulation site and moves to that location at a predefined speed. Upon reaching the location, it pauses again for *pause time* seconds, selects another location, proceeds there, then stays for *pause time* seconds, and so forth, repeating this behavior throughout the simulation. The speed at which the node moves in our simulation varies uniformly between 0 and 20 m/s (equivalent to a vehicle speed of 72 km/h).

The pause time used in the simulation is varied between 0 and 900 seconds. A 0 second pause time corresponds to continuous mobility, while a 900 seconds pause time, which equals the length of the simulation time, corresponds to no mobility. We simulate each pause time with 10 mobility scenarios, each generated with a different seed number, and present the mean of each performance metric over these 10 runs. The following describes our performance metrics of interest.

### 3.3.4 Performance Metrics

- *Route discovery overhead.* The sum of routing packets generated by each node due to route discovery. The routing packets include both RREQ and RREP packets. Each hop-wise transmission is counted as one transmission.
- *Total routing overhead.* The sum of routing packets generated by each node due to both route discovery and route maintenance. The routing packets generated by basic route maintenance are the RERR packets.
- *Total MAC signaling overhead.* The sum of MAC signaling packets generated during the duration of the simulation. The MAC signaling packets include RTS, CTS and ACK packets generated for all unicast traffic.
- *Route discovery latency:* The average time from originating a RREQ at the source to receiving the first RREP that answers the request. This metric measures the amount of time needed to acquire a route to the destination.
- *End-to-end delay.* The average time from originating a data packet at the source to delivering the packet to its final destination. This includes all possible delays such as route discovery latency, queuing delay at network interface queue, propagation and retransmission delays in MAC and physical layers.
- *Packet delivery ratio.* The percentage of data packets delivered to the destinations with respect to number of data packets sent by the sources. This metric measures the extent of packet loss due to such reasons as routing failure and network congestion.

### 3.4 Evaluation Results

The following sections present the evaluation results achieved by the three protocols under study using *ns-2* simulator. Table 3.3 summarizes the simulation parameters used, which are similar to those used in literature to evaluate many ad hoc routing protocols [43, 44]. In all the plots in this chapter, results are shown with 95% confidence intervals.

Table 3.3: Summary of simulation parameters

Parameter	Setting
Mobility model	Random waypoint
Traffic model	20 CBR sources
Network space	1500m x 300m
Number of nodes	50 nodes
Maximum node speed	20 m/s
Packet sending rate	4 packets/s
Data payload	512 bytes

#### 3.4.1 Route Discovery Overhead

Figure 3.4 shows the route discovery packets (RREQ and RREP) generated under different mobility conditions. In all protocols, the route discovery packet decreases as pause time increases. This is expected since lower mobility increases the stability of the routes, and consequently requires fewer route reconstructions once they are established. At pause time of 0 seconds when all nodes are in constant motion (highest mobility), DSR+L and DSR+L+U generate less overhead packets by 6.3% and 36.1% respectively over DSR. As pause time increases, the margin of overhead reduction diminishes. This is because the impacts from the optimization techniques (LAR and UNIQUE) on these protocols are less felt with fewer route discoveries performed at lower node mobility. At pause time of 900 seconds when all nodes are stationary, no remarkable difference in overhead between the protocols are observed.

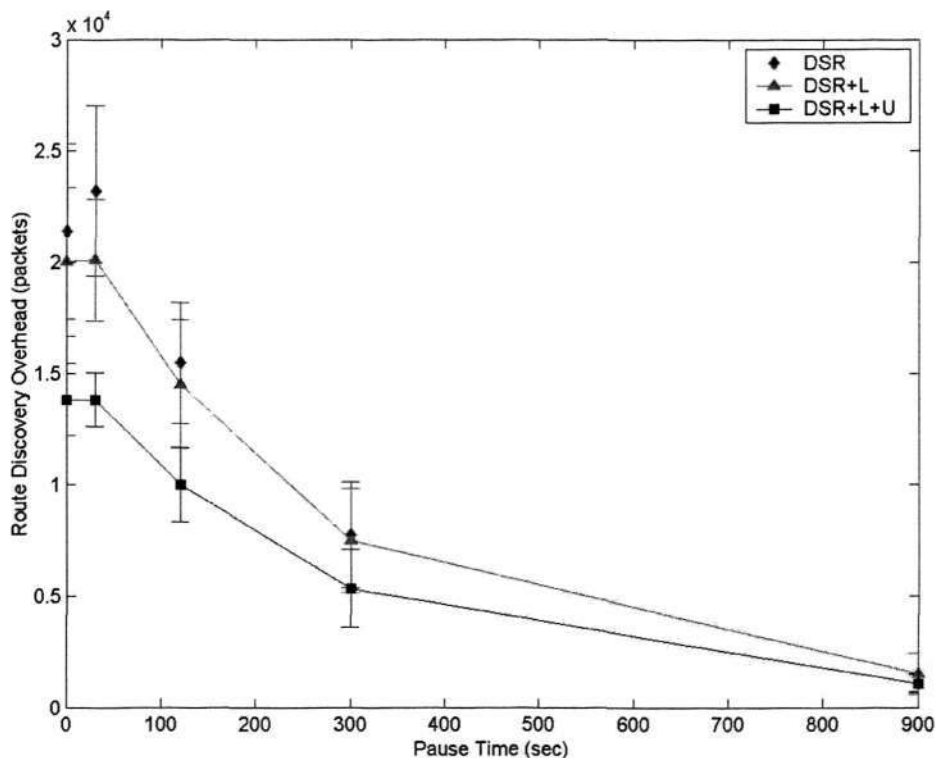


Figure 3.4: Number of routing packets generated by route discoveries

Figure 3.5 shows the number of route discoveries performed. Interestingly DSR, although it generates the highest routing load, performs the least route discoveries. On the other hand, both DSR+L and DSR+L+U perform more route discoveries than DSR, though they generate a lower routing load.

This phenomenon is the result of two factors from using route discovery techniques with limited scope. First, if a route is not found in an initial restricted search, another attempt will be made to search over a wider area. An example is the fall back to flooding if the initial attempt by UNIQUE or LAR is unsuccessful. Second is an outcome specific to

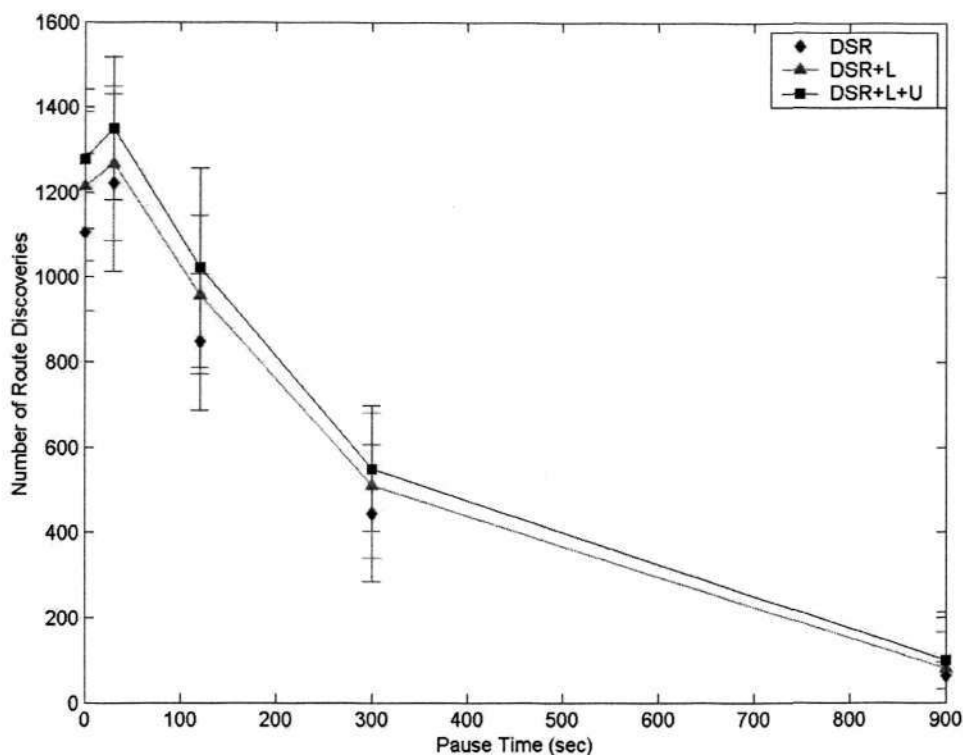


Figure 3.5: Number of route discoveries performed by each protocol

DSR, where nodes may cache multiple routes to a target (AODV allows nodes to store only one route per target). Generally, fewer alternate routes exist to be found within a reduced search area. Therefore, when a route to some target fails, a replacement route may not be available in the node cache, and route discovery may need to be performed. Indeed, this is reflected in the cache hit results in Table 3.4. A cache hit refers to a cache access that successfully finds the requested route. Results show that protocols performing more ‘focused’ route searches have, on average, a lower cache hit rate.

Table 3.4: Cache hit rates of protocols operating under constant node mobility

Protocol	Cache hit rate
DSR	0.668
DSR+L	0.625
DSR+L+U	0.388

A higher cache hit and thus fewer route discoveries do not necessarily mean that DSR will have a lower packet delay, as there are other factors besides route discovery latency that can influence the time required to deliver a packet. We shall further discuss this when we present the results for end-to-end delay in Section 3.4.3.

Since the route discovery can be performed using a combination of techniques, i.e. Ring Zero, LAR, UNIQUE or FLOOD, some statistics to show the extent to which a particular technique is utilized will be helpful. Table 3.5 shows the percentage breakdown of: a) number of route discoveries; b) number of route discovery packets; due to each technique employed. For each protocol, the contribution by each technique is read left-to-right of the result table. Values shown are for the worst case at pause time of 0 seconds.

Table 3.5: Percentage breakdown of the number of route discoveries and route discovery packets (RREQ+RREP) by each technique employed in each protocol

Protocol	Route discovery technique			
	Ring Zero	LAR	UNIQUE	Flooding
DSR	74%	--	--	26%
DSR+L	70%	26%	--	4%
DSR+L+U	0.4%	0.2%	91.2%	8.2%

(a) Route discoveries

Protocol	Route discovery technique			
	Ring Zero	LAR	UNIQUE	Flooding
DSR	24%	--	--	76%
DSR+L	29%	62%	--	9%
DSR+L+U	0.1%	0.4%	49.3%	50.2%

(b) Route discovery packets

From Table 3.5a, it can be seen that a majority of route discoveries in DSR and DSR+L are performed by means of Ring Zero. In DSR, 74% of route discoveries are by Ring Zero, while the rest resort to Flooding when Ring Zero fails. From this, we can work out the success rate of Ring Zero in DSR (or the success of finding a route in a neighbor's cache), which turns out to be 0.649, not far from the cache hit rate of 0.668 in Table 3.4. Similarly, Ring Zero is the most utilized technique in DSR+L, while LAR is used in only 26% of the route discoveries performed. This perhaps explains why DSR+L is only marginally better than DSR in terms of overhead performance (Figure 3.4).

DSR's existing optimization techniques such as Ring Zero has been found to be quite effective in decreasing the frequency and spread of Flooding, and therefore limits the potential contribution of LAR. For example, some Flooding can be averted when using Ring-Zero search. DSR's caching of overheard routing information also enables nodes to perform fewer route discoveries, and increases their responsiveness to route requests from other nodes, thereby curbing the spread of these broadcast messages. Thus, while there is little doubt that LAR would perform well over a simple pure flooding protocol (no optimizations or whatsoever), it may not have much performance advantage to offer when applied to a natively optimized protocol such as DSR.

Now coming back to our analysis of the results in Table 3.5a. Recall that for DSR+L+U, UNIQUE is the first choice of technique used (Section 3.2.2), so more route discoveries are performed using this technique. We have earlier experimented with a version that uses Ring Zero as the first choice, followed by UNIQUE, LAR, and Flooding. However, the

overhead results were not very impressive due to a larger number of RREP (especially cached replies), though it was still better compared to DSR+L. Under the present version (see Figure 3.3), Flooding is mostly performed after the initial attempts by UNIQUE to discover a route are unsuccessful. Thus, the small proportion of route discoveries (8.2%) resorting to Flooding reflects that only a small fraction ( $< 10\%$ ) of search attempts by UNIQUE is unsuccessful.

From comparing Tables 3.5a and 3.5b, we also made the following observations. Though Flooding in DSR is used in only 26% of route discoveries, the packet implosion caused by Flooding is responsible for 76% of route discovery packets generated, highlighting its expensiveness. In DSR+L+U, UNIQUE generates less than half (49.3%) the total route discovery packets, though it was used in over 90% of route discoveries. The savings come from the limited extent to which RREQ can be propagated, and RREP can be returned, by virtue of using unicast transmission.

### **3.4.2 Total Routing and MAC Overhead**

In the previous section, we consider only the overhead of route discovery. However, the total overhead of a routing protocol will also comprise of route maintenance packets, which are packets sent to monitor or maintain routes in use. They include RERR and gratuitous RREP sent to shorten the routes in use over time (an optimization to route maintenance in DSR [22]). Figure 3.6a shows the total routing packet overhead of each protocol.

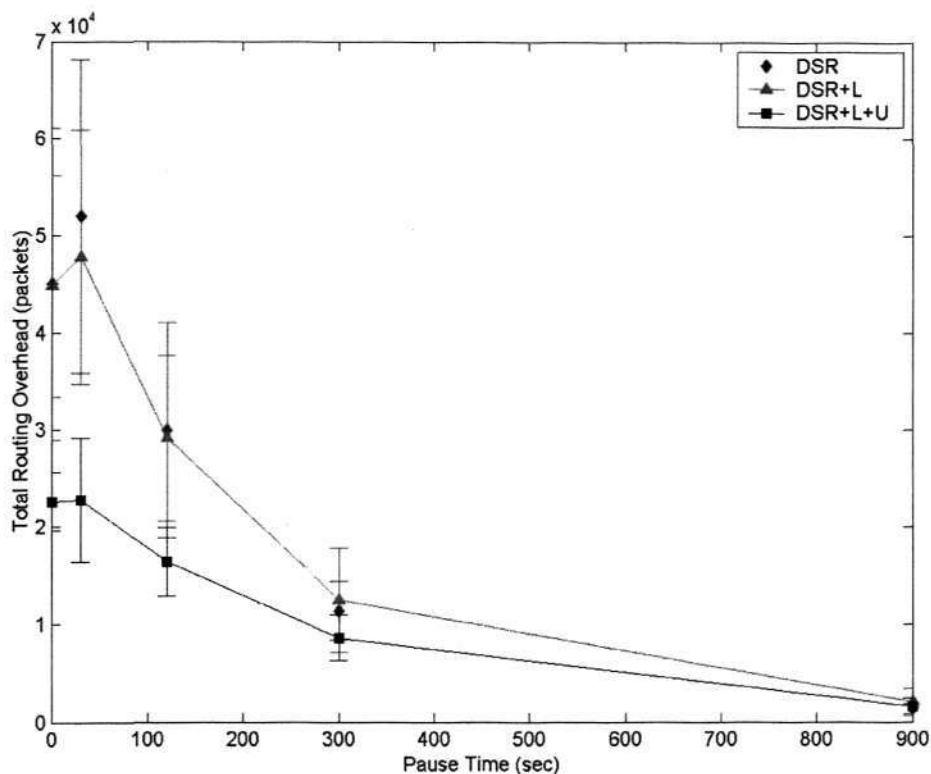


Figure 3.6a: Total routing packets generated

Comparing Figure 3.4 and 3.6a, the first thing we note is that the general pattern of results remains unchanged. However, the overhead difference between DSR+L and DSR is more subtle after we include the overhead of route maintenance. This is because both of these protocols generate a fairly similar amount of route maintenance packets, which consist of mostly RERR packets. The LAR scheme in DSR+L mainly affects the RREQ and RREP packets due to route discovery.

Another observation is the higher margin of overhead improvement of DSR+L+U over the other two protocols due to a much smaller number of RERR packets, which can be

attributed to the following reasons: First, *data packets* are generally routed through more *recently* discovered routes as a result of higher frequency of route discovery due to a lower cache hit probability. Also recall that in DSR, except for routes that are in use, other routes acquired and cached are *not maintained*. If these routes are used to deliver packets at a later point in time, and if node mobility is sufficiently high, the possibility of encountering a link breakage on these routes is non-negligible. With UNIQUE, however, the RREQ that is unicast through a cached route also acts like a *probe packet*, verifying the connectivity of each link that it crosses. Thus, some outdated links can be uncovered and removed early by nodes from their caches before they can be used by data packets to reach their destinations, causing subsequent routing errors.

Since location information is included in the routing packets of both UNIQUE and LAR, their routing packet sizes would be larger than that of DSR. Thus, we further present the overhead results in terms of bytes, as shown in Figure 3.6b. Indeed, presenting in bytes gives a slightly different picture, with DSR+L+U becoming slightly more expensive than DSR when location bytes are included, i.e. with more bytes per routing packet, DSR+L+U generated more routing bytes than DSR, but the increase has been limited by its greatly reduced number of routing packets. Given the conventional wisdom that the expense of acquiring the air medium to transmit a packet dominates with the 802.11 MAC relative to per byte transmission cost, we believe the incorporating of extra location bytes in a packet could be justified, if by doing so it saves a significant number of routing packets.

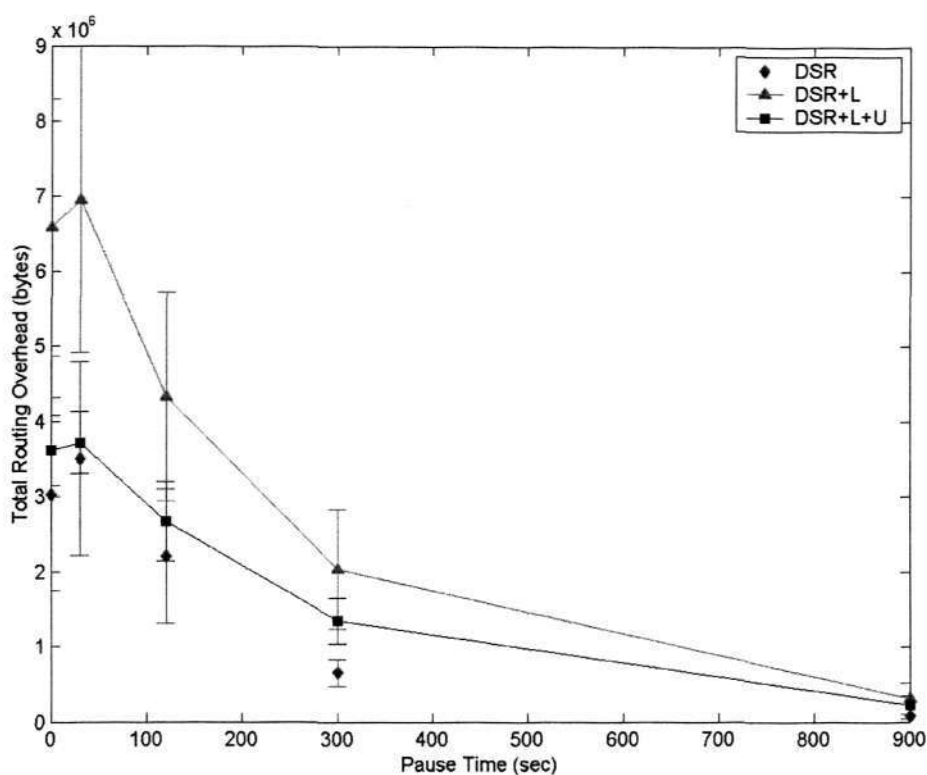


Figure 3.6b: Total routing bytes generated

As a point of comparison, we also experimented with a simple flooding protocol built using the same code base of DSR in *ns-2*. This protocol, which we call DSR (Flood), performs *all* route discoveries by Flooding. However, the Flooding here differs from the previous Flooding in DSR in that: (i) All intermediate nodes receiving the RREQ must rebroadcast it, and must not reply to it with their cached routes; (ii) All nodes do not operate in promiscuous mode, thus they cannot overhear and cache routing information in packets not meant for themselves. In brief, none of the optimizations for route discovery or route maintenance in DSR are present in DSR (Flood).

To give ourselves a clearer picture of what extent UNIQUE and LAR can decrease the overhead of route discovery in the *absence* of DSR's optimizations, we rerun DSR+L and DSR+L+U with those optimizations off. Thus, Ring Zero is removed from the algorithm of these two protocols. In DSR+L, the number of route discovery stages reduces from three to two, as LAR replaces Ring Zero as the first choice of technique, followed by Flooding. This makes it similar to the original LAR scheme studied in [5]. DSR (Flood) is also similar to the basic flooding used for comparison in the same paper. For both of these protocols, *every* non-target node receiving the RREQ will forward it, and will not reply to it with their cached routes. For DSR+L+U, it means the RREQ will be unicast all the way to the target neighbor, as nodes on the UQPath will not reply. For DSR+L and DSR+L+U protocols, which operate with no optimizations from DSR, we refer them to as DSR+L (noOPT) and DSR+L+U (noOPT) respectively. Figure 3.7 shows the modified algorithm, while Figure 3.8 shows the overhead results.

Without the previous optimizations, the total overhead of DSR (Flood) almost triples compared to DSR (Figure 3.6a). The overhead of DSR+L (noOPT) also increases, but to a lesser extent. At pause time of 0 seconds, its overhead improvement over DSR (Flood) is 45.5%, not far from the 40% reported in [5]. This margin of improvement is also higher than when we compared DSR+L to DSR. This reaffirms our earlier assessment that LAR may not give much further benefit to a DSR protocol already operating with all its native optimizations. DSR+L+U (noOPT) remains the most efficient, with less than half the routing load of DSR (Flood) generated at higher mobility.

Step 1: If ( $UQPath$  exists), perform RD by UNIQUE. Else go to step 2.  
 If (route found), return. Else go to step 3.

Step 2: If (location of  $T$  known), perform RD by LAR. Else go to step 3.  
 If (route found), return. Else go to step 3.

Step 3: Perform RD by Flooding. Return.

Figure 3.7: Modified route discovery algorithm for DSR+L+U (noOPT)

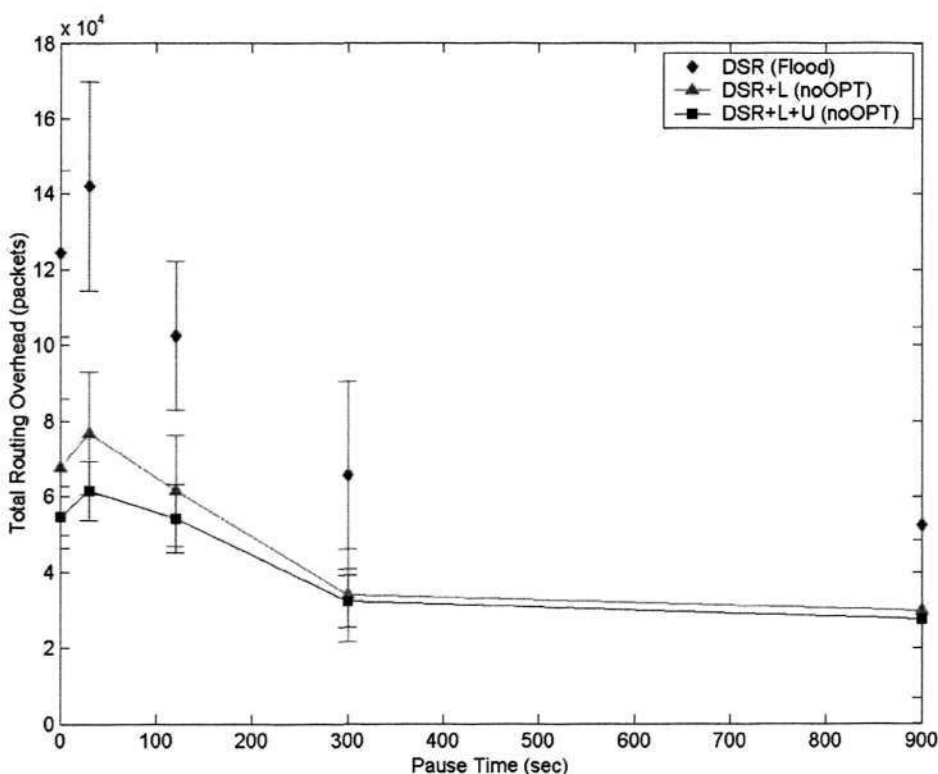


Figure 3.8. Total routing overhead with no optimizations from DSR

Figure 3.9 summarizes the overhead results and shows the composition of routing packets by each protocol at pause time of 0 seconds. Note the much smaller proportion of RERR in DSR (Flood), DSR+L (noOPT) and DSR+L+U (noOPT). This is because of end-to-end

RREQ forwarding, which gives routes that are more updated and therefore less susceptible to breakage than those obtained from caches of intermediate nodes.

In UNIQUE, the RREQ are sent by means of unicast transmission, and thus incurs extra overhead at the MAC layer due to RTS/CTS exchanges and ACK packets. However, under the worst case scenario at highest mobility (pause time = 0s), we found that the MAC layer packets generated by DSR+L+U is still less than DSR, though by a lesser margin of 14.8%, compared to about 50% margin of packet reduction at the routing layer (refer to Figure 3.6a). This is because the extra MAC overhead incurred by UNIQUE's RREQ packets has been more than offset by the decrease in MAC overhead as a result of fewer RREP and RERR packets, which are unicast packets. As for DSR+L, its overhead at MAC layer is found to be very comparable to DSR.

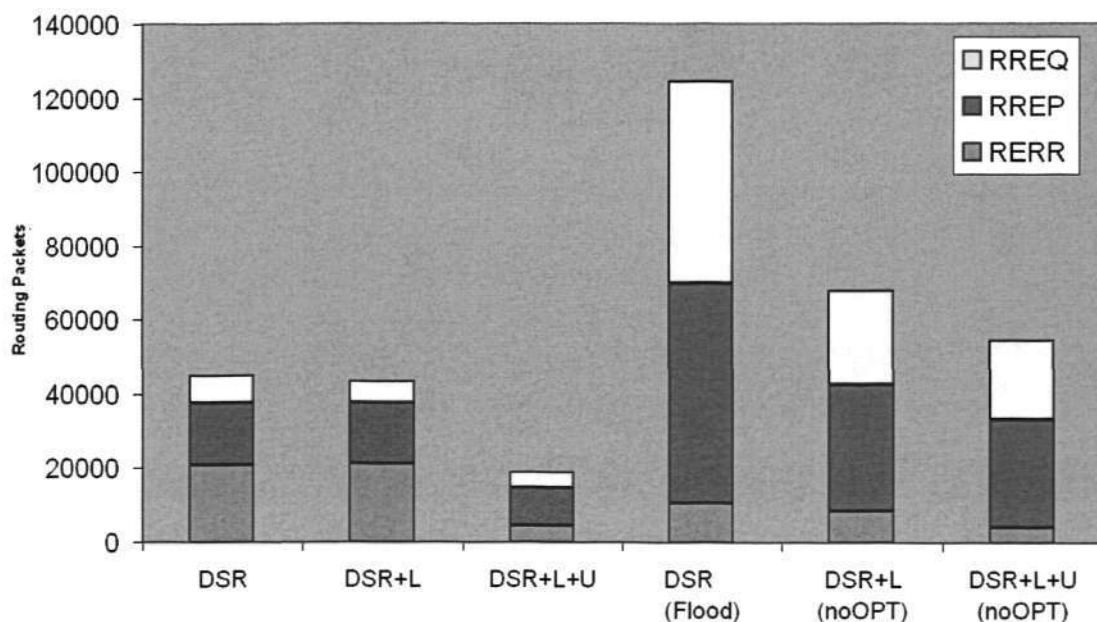


Figure 3.9: Composition of routing packets

### 3.4.3 Route Discovery Latency

In Section 3.4.1, we evaluated the route discovery overhead. In this section, we would evaluate the route discovery latency. This is the time it takes a source to obtain a reply to its request for a route to the destination. This metric contributes to the end-to-end delay of data packets (to be discussed next in Section 3.4.4), as without a route, the packets cannot be transmitted but buffered at the source while a route discovery is in progress.

There are basically two factors that might influence the route discovery latency. First, is the per-hop forwarding delay of RREQ and RREP packets, which largely depends on the congestion level of the network. Second, is the number of hops (or path length) traversed by the route discovery packets to acquire a route to the destination. This often depends on the distribution of routing information among nodes within the network.

Figure 3.10 shows the route discovery latency evaluated under different levels of mobility. Under a low mobility, where routing load of each protocol is similar and light, the route discovery latency is minimal, and there appears no remarkable difference in the latencies between the protocols. But as mobility increases, the growing difference in routing loads between the protocols causes each to produce a very different performance. In the highest mobility scenario, all protocols experience a higher latency level due to network congestion caused by greater routing traffic.

The lower routing load arising from using UNIQUE allows DSR+L+U to minimize the forwarding delays of RREQ and the subsequent return of RREP, as less routing traffic

reduces channel contention, while less backlog in interface queue accelerate the process of forwarding packets. The reduced forwarding delay in turn shortens the time it takes to complete the route discovery operation.

Between DSR and DSR+L, the latter exhibits higher latency, especially when mobility is high. This result should not be surprising if we recall that DSR+L has a similar routing load as DSR (Figure 3.6a), and so experiences a similar level of congestion. However, its routing packet size is larger than DSR since the packets also carries location information. Consequently, the RREQ and RREP packets queuing for transmission in a node must wait longer since the backlog in the interface queue will drain away more slowly.

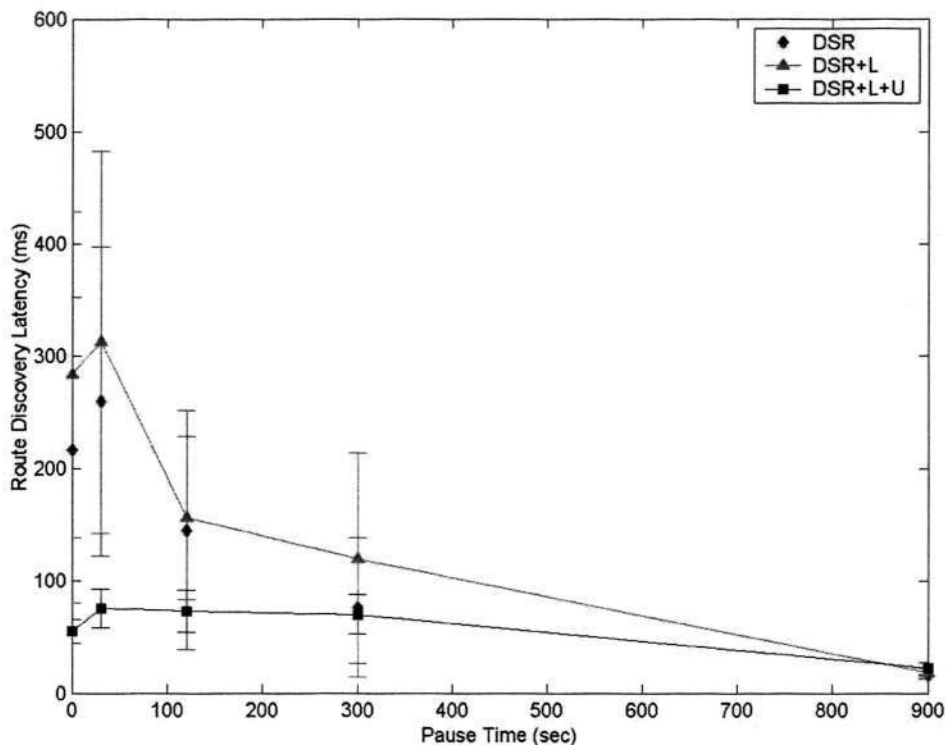


Figure 3.10: Latency in route discovery

As mentioned, the number of hops traversed by route discovery packets is another factor that might influence the route discovery latency. Note that the hop count traversed by a RREQ would be similar to its RREP, as the latter is using the reversed route of the RREQ to traverse back to the source. We evaluated the average number of hops traversed by the RREP (more specifically, the first RREP received by the source for a RREQ it sent), and we found that DSR+L+U has actually a higher query reply path length as compared to DSR and DSR+L. This is shown in Table 3.6 for the highest mobility case.

Table 3.6: Number of hops traversed by first received RREP

Protocol	Hops traversed by first RREP
DSR	2.41
DSR+L	2.40
DSR+L+U	3.01

We believe this higher number of hops taken is related to the lower cache hit rate of the DSR+L+U (Table 3.4), since a lower cache hit means the RREQ needs to travel further before it reaches a node with a route to the destination. The delaying effect of this extended path taken is however, small relative to that of congestion at high mobility, and therefore DSR+L+U is able to achieve the lowest latency. At low mobility where congestion is light, e.g. at pause time of 900s, this extended path effect may have led to a higher latency in DSR+L+U, but not to an appreciable extent as we observed from Figure 3.10.

#### 3.4.4 End-to-End Delay

The end-to-end delay of a data packet is contributed by the route discovery latency, as well as the cumulative forwarding delays at each hop of its route between the source and destination.

This includes delays due to queuing at interface queues, propagation and retransmissions in MAC and physical layers (as defined in Section 3.3.4).

Similar to route discovery, the forwarding delay of data packets also largely depends on the congestion level of the network. Furthermore, the end-to-end delay is proportional to the number of hops taken by the data packets to reach their respective destinations. This depends on the optimality of the routing path used, which in turn might be influenced by the technique of route discovery.

Figure 3.11 shows the average end-to-end delay of data packets. In general, it reveals a similar trend as that in Figure 3.10 for route discovery latency, since the congestion effect that impacts the route discovery packets also impacts the data packets in a similar way. To investigate the effect of path optimality, we obtained the distribution of the path length difference between the actual number of hops taken and optimal number of hops required by data packets to reach their destinations. The result is shown in Figure 3.12.

Here, a difference of zero means the data packets have taken an optimal (shortest) path, whereas a difference  $> 0$ , i.e. 1, 2, 3, indicates the extra number of hops the packets have incurred. We observe that all protocols have approximately 80% of their data packets routed through optimal paths, while the majority of remaining packets that took a non-optimal path, incurs only an extra hop from optimal. The performance impact of non-optimal routes is thus believed to be quite minimal.

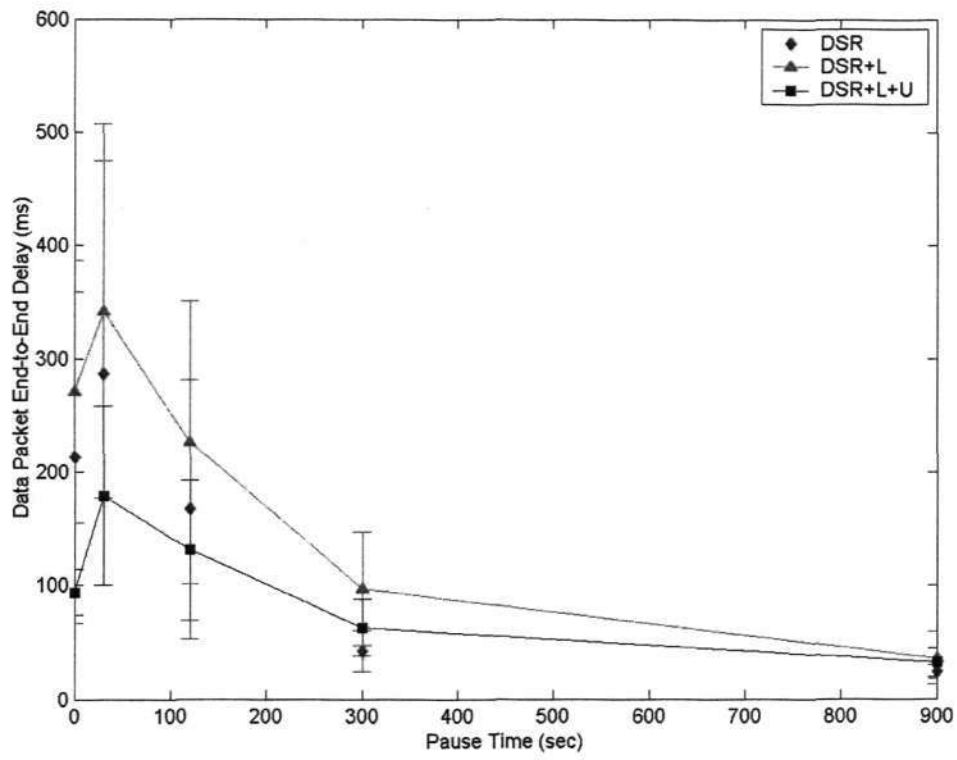


Figure 3.11: End-to-end delay of data packets

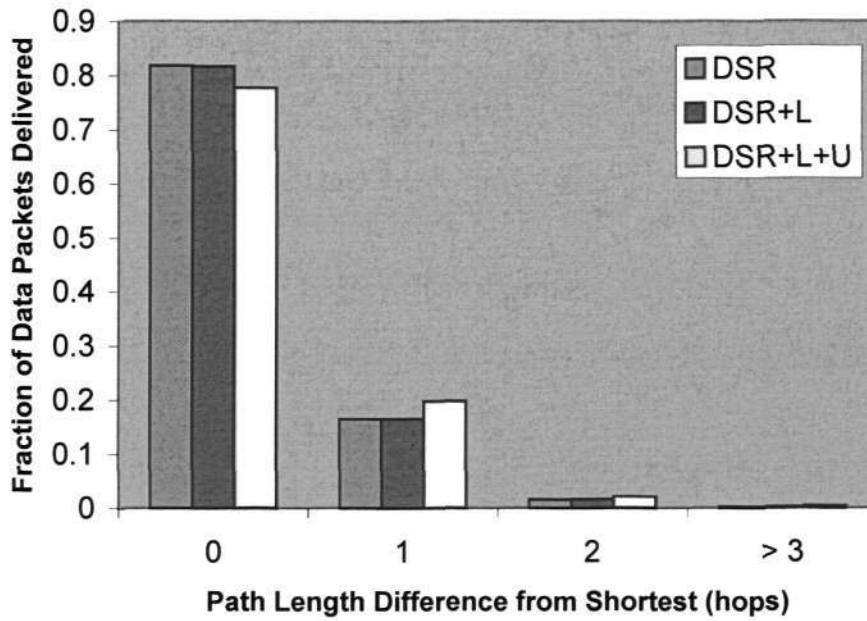


Figure 3.12: Path optimality of data packets

DSR+L+U does not perform better than DSR and DSR+L in terms of path optimality, which is expected since the route acquired by UNIQUE is seldom to be the most optimal that exists between the node-pair in communication. However, its close optimal performance is better than initially expected, and we attribute it to a DSR feature that allows the route to self-optimize its path length over time [22]. Note that this does not restrict the application of UNIQUE to DSR, as there are general schemes such as SHORT [59] for the specific task of optimizing the length of paths in use by reactive protocols.

### 3.4.5 Packet Delivery Ratio

Figure 3.13 shows the success of delivering data packets to their destinations. In general, the fraction of data packets successfully delivered during low mobility (high pause time) approaches 100%. At higher mobility (low pause time), packet delivery performance deteriorates mainly because of two reasons: i) more packets are dropped in transit due to route disconnections, i.e. a packet can be dropped by an intermediate node when the primary route fails and no other routes exist between itself and the destination; ii) packets are also dropped when the interface queue becomes full as a result of more routing traffic introduced by the increase in route discoveries.

Table 3.6 summarizes the reasons for packet loss for the three protocols at highest mobility. It is observed that DSR+L+U benefited greatly from its lower routing load, which led to a 95.3% reduction in packet loss (relative to DSR) due to IFQ (interface queue) full. A second major saving comes from the 83.1% reduction in packet loss due to no route, as a result of fewer route breaks or routing errors (RERR) discussed in Section 3.4.2

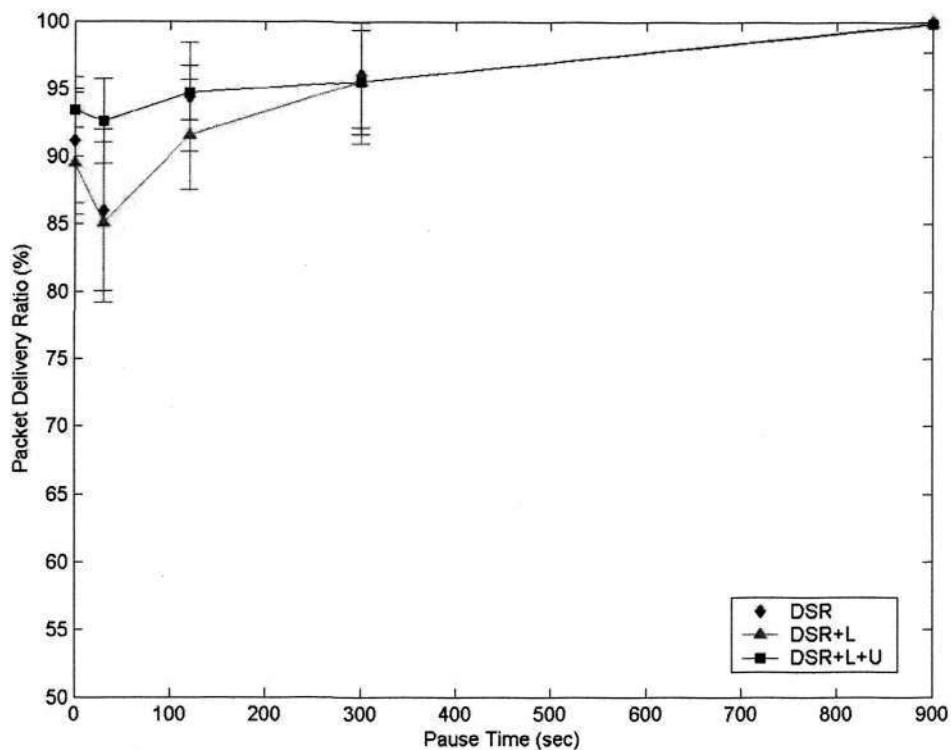


Figure 3.13: Percentage of data packets delivered

Table 3.7: Data packet dropped summary for the three protocols

Data Packet Dropped Summary	DSR	DSR+L	DSR+L+U
No Route	1341	1265	226
TTL Expired	0	0	0
RTR Queue Full	0	0	0
Timeout	55	38	91
Routing Loop	0	0	0
IFQ Full	2170	2617	101
ARP Full	101	100	68
MAC Callback	0	0	0
Simulation End	18	28	9
Total Packets Dropped	3686	4004	497

### 3.4.6 Effect of Network Size and Traffic Load

The previous sections investigated the effect of mobility on the protocols by varying the pause times for a given network size and traffic load. This section further investigates the impact of changing the network size and traffic load on the performance metrics. In order to isolate the effect of these changes, we chose to vary one parameter at a time, i.e. either network size or traffic load, but not both at the same time, while keeping other parameters unchanged. Network size is the number of nodes in the network, while traffic load is the number of data packets injected into the network, which can be varied by changing the number of data sources. Different numbers of nodes and data sources were experimented but only the most significant results for a network size of 100 nodes (up from 50 nodes) and traffic load of 30 sources (up from 20 sources) are presented. An attempt to simulate above 100 nodes was aborted due to slow speed and very large memory requirement of the simulator. A traffic load of 30 sources is also the highest that can be injected without severely congesting the network for an otherwise meaningful comparison.

Before we discuss the effect of network size and traffic load, we summarize the previous results of the protocols for each performance metric (listed in Section 3.3.4) in Table 3.8, shown for a 50 node model with 20 sources, under highest mobility (pause time = 0s). Note that DSR+L is omitted for brevity since its performance is mostly similar to DSR. Furthermore, all the overhead results shown are not in absolute terms but normalized to the total number of data packets delivered, i.e. number of overhead packets transmitted per data packet delivered to the destination. This is to facilitate a meaningful comparison with results of the 100 node model, which we simulated for 500 instead of 900 seconds,

in order to allow a reasonably timed simulation. Normalizing the overhead packets is to take into account of the difference in number of data packets and hence overhead packets generated between the 50 and 100 node model because of different simulation duration. The percentage improvement of DSR+L+U over DSR for each metric is also shown.

Table 3.8: Summary of results for a 50 node model with 20 sources

Performance metrics	DSR	DSR+L+U	% Improvement
Normalized route discovery overhead	0.36	0.23	36.1
Normalized total routing overhead	0.76	0.37	51.3
Normalized total MAC overhead	11.47	9.77	14.8
Route discovery latency (ms)	216	55	74.5
End-to-end delay (ms)	212	93.7	55.8
Packet delivery ratio (%)	91.2	93.4	2.41

Table 3.9 shows the resulting effect of changing the network size and traffic load on the protocols performance. In general, both protocols show a deterioration in performance (higher overhead, longer end-to-end delay and lower packet delivery ratio), as network size and traffic load increases. However, DSR's performance seems to deteriorate much more rapidly than DSR+L+U. For example, when network size is increased to 100 nodes (Table 3.9a), the packet delivery ratio of DSR decreased to only 72.6%, while that of DSR+L+U remained above 90%. This is due to reduced congestion by routing and MAC overhead packets in DSR+L+U, which is almost 80% and 52% less, respectively, than DSR. The percentage improvement of all metrics is generally higher over that achieved in the 50 node model (Table 3.8).

Table 3.9: Effect of changing network size and traffic load

Performance metrics	DSR	DSR+L+U	% Improvement
Normalized route discovery overhead	2.69	0.62	77.0
Normalized total routing overhead	3.87	0.81	79.1
Normalized total MAC overhead	24.3	11.7	52.0
Route discovery latency (ms)	1511	84.8	94.4
End-to-end delay (ms)	746	76.9	89.7
Packet delivery ratio (%)	72.6	90.4	24.5

(a) Network size: 100 nodes. Traffic load maintained at 20 sources

Performance metrics	DSR	DSR+L+U	% Improvement
Normalized route discovery overhead	0.91	0.29	68.1
Normalized total routing overhead	2.00	0.52	74.0
Normalized total MAC overhead	21.7	13.5	37.6
Route discovery latency (ms)	1281	293	77.1
End-to-end delay (ms)	1631	564	65.4
Packet delivery ratio (%)	54.5	70.9	30.1

(b) Traffic load: 30 sources. Network size maintained at 50 nodes

Comparing Tables 3.9a and b, it is also worth noting that a larger network size appears to have a greater effect on increasing the overhead packets than a higher traffic load. However, the end-to-end delay and packet delivery ratio appear worse with higher traffic load, in particular for DSR, because of more congestive effects from a greater number of data packets, which are also larger in size than most routing and MAC overhead packets. A clearer picture of the impact from changing network size and traffic load can be seen from Table 3.10. Clearly, increasing network size directly increases the routing packets generated in a route discovery, since more nodes will receive and react to the route

request, e.g. in a flood attempt. In DSR, for instance, the route discovery cost - defined as number of routing packets generated per route discovery, increases from about 19 to 44 packets as network size increases from 50 to 100 nodes. However, increasing traffic load does not effect a similar change in route discovery cost, which remains relatively stable as traffic load increases from 20 to 30 sources.

Table 3.10: Number of routing packets per route discovery

Number of Nodes ( $n$ ) and Sources ( $src$ )	Number of Routing Packets per Route Discovery	
	DSR	DSR+L+U
$n = 50, src = 20$	19.35	10.95
$n = 100, src = 20$	44.00	28.83
$n = 50, src = 30$	19.37	9.17

### 3.5 Chapter Summary

This chapter proposes a route discovery optimization technique for reactive MANET routing protocols. Named UNIQUE, the proposed optimization is designed to reduce RREQ (or *query*) packets by allowing the source to take advantage of any existing route to carry the RREQ (as a *unicast* message) closer to the destination, before a low-overhead localized flood is attempted to find the required route. By using both location and *route* information in its optimization process, UNIQUE outperforms the well-known Location Aided Routing (LAR) scheme that only uses location information, for reducing routing overhead of the Dynamic Source Routing (DSR) protocol – a well-studied reactive routing protocol. Even though we developed and evaluated UNIQUE for DSR, it should be applicable for other reactive protocols of similar nature, such as AODV (Section 2.3.3)

Experimental studies carried out across various scenarios with different node mobility, network size, and traffic loads also show that UNIQUE improves route discovery, data packet delivery and end-to-end delay performances as a result of reduced network congestion and routing errors. The improvement is more marked with increasing severity of the network environment, such as with larger number of nodes and data sources.

## Chapter 4

# SURE Optimization for RREP Reduction

### 4.1 Introduction

This chapter presents the second optimization technique proposed in this thesis, called the SURE technique for RREP reduction. The proposed technique is targeting RREP packets generated by and carrying cached routes from non-destination nodes (herein we call them Cached RREP). SURE is designed to complement the UNIQUE technique developed for RREQ reduction in the previous chapter. As mentioned before, RREQ and RREP together constitute the overhead of route discovery. SURE exploits a rarely noted but commonly occurring phenomenon to provide intermediate nodes with query state information that can prevent them from replying to a RREQ (even if received for the first time) if they do not offer a more effective route to the destination. We illustrate this previously unreported phenomenon through examples in the following.

#### 4.1.1 Observation

Referring to Figure 4.1a, suppose a source node  $A$  broadcast at time  $t_0$  a RREQ, which is received by two intermediate nodes  $B$  and  $C$ . Assume  $B$  has a cached route to the RREQ destination, while  $C$  does not. Also assume  $B$  successfully contended for the channel and transmits ahead of  $C$  at time  $t_1$  its Cached RREP, which is overheard by node  $D$ . After transmission from  $B$  is over,  $C$  rebroadcast the RREQ at time  $t_2$ , which is also received by  $D$ . Therefore,  $D$  notices the RREP for a RREQ it receives only later.

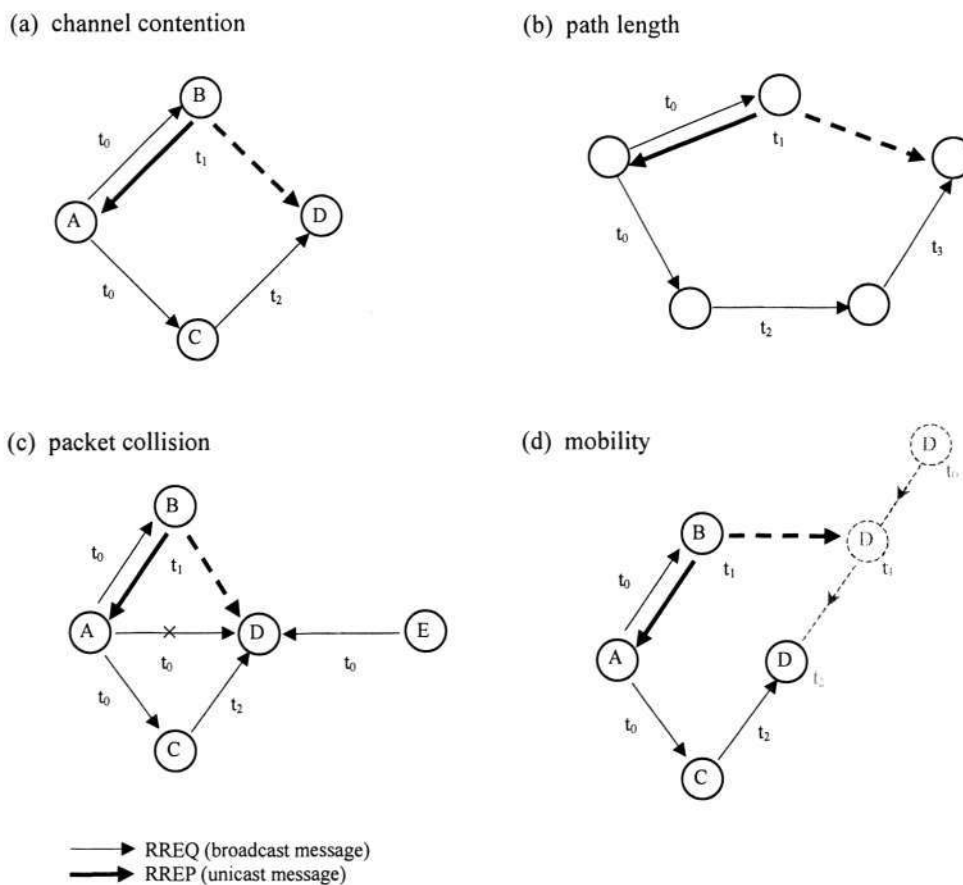


Figure 4.1: Example scenarios that illustrate the effects of channel contention, path length, packet collision and mobility, which are potential factors leading to the observed phenomenon

This phenomenon also occurs in other scenarios. For example, Figure 4.1b illustrates a case where the RREQ travels over a longer path ( $A-C-E$ ) than the RREP ( $A-B$ ) to reach  $D$ . In Figure 4.1c,  $D$  is in direct range of  $A$ , but did not receive its RREQ due to a packet collision caused by a simultaneous broadcast from  $E$  at time  $t_0$  for a separate route discovery. Then as in Figure 4.1a,  $D$  hears the RREP at time  $t_1$  and receives the RREQ at time  $t_2$ . Figure 4.1d further considers the effects of mobility, where  $D$  moves into range of

$B$ , and later into range of  $C$  at times  $t_1$  and  $t_2$  respectively, thereby observing the RREP *before* receiving the RREQ for the same route discovery.

#### 4.1.2 Proposed Optimization

We refer to this phenomenon as *reversed query/response arriving order* as it reverses a commonly assumed packet arriving order of query (RREQ) precedes response (RREP). This phenomenon can be put to good use to minimize the number of Cached RREP. For example, the overheard RREP contains information about i) the hop length of the returned source route, and ii) the ID of the RREQ, for which this RREP is generated. Together with the source address of the RREQ, which is found from either the returned source route, or destination address of the RREP, these three pieces of information could be used to decide if a node should reply upon receiving a RREQ, even for the first time.

Existing route discovery algorithms [22, 23] dictate that a node will reply if it receives a RREQ for the first time, for which it has a route to the destination. It is almost certain that node  $D$  in Figure 4.1 would send a Cached RREP, since it knows at least a route to the destination, which is that it overhears from the RREP. However, the returned route may not be useful if it is longer than the one previously returned, since route selection at the source is typically based on the shortest path.

We thus propose that if a node overhears a RREP for a RREQ it has not seen before (known by the RREQ ID and source address), the node shall record the three pieces of information from the RREP, namely i) the hop-length of returned source route, ii) RREQ

ID, and iii) RREQ source address, as mentioned before. Subsequently, if the node receives this RREQ, it will compare the hop-length of its route (to be returned) with that seen previously. It will reply if it has a shorter route, and discard otherwise. Figure 4.2 shows the pseudo-code for this algorithm.

```

SURE OPTIMIZATION TECHNIQUE

Let  $p$  be the RREP packet overheard from a neighboring transmission
Let  $q$  be the RREQ packet for which  $p$  is generated
Let  $H_s$  be the hop-length of the returned source route  $s$  in  $p$ 
Let  $H_r$  be the hop-length of the desired source route  $r$  in cache
Let  $ID$  be the identification value of a RREQ packet
Let  $S$  and  $T$  be the RREQ's source and target node respectively

// At intermediate nodes
When  $p$  is received:
    Check  $S$  and  $ID$  of  $q$  in  $p$  to determine if  $q$  has been seen
    If  $q$  has not been seen before
        Record  $H_s, S$  and  $ID$  of  $q$ 
    Return

When  $q$  is received:
    Check route cache to determine if a route  $r$  to  $T$  exists
    If ( $r$  exists) AND ( $H_r < H_s$ )
        Send RREP with  $r$  to  $S$ 
    Else // if no route exists, or if  $r$  exists but  $H_r \geq H_s$ 
        Discard  $q$  // do not forward or reply to  $q$  which
        Return // has been replied with  $p$  previously

```

Figure 4.2: Pseudo-code for SURE

Recall that DSR has an optimization scheme with some similarity for preventing “Route Reply Storms” (Section 2.4.3). However, the scheme does not propose the use of other information received from the RREP as we mentioned above. Furthermore, the scheme listens for shorter routes only *after* RREQ is received. This inherently introduces a delay, which adds to the route acquisition latency.

## 4.2 Protocol Description

For this study, we again use DSR [22] as our base routing protocol to implement SURE. As in Chapter 3, the DSR is used with all its native optimizations to provide a more challenging base for comparison. The route discovery is consisting of two stages: an initial 1-hop Ring Zero search (Section 2.4.2), which if not successful, is followed by a flooding attempt that searches the entire network.

### 4.2.1 DSR+S

DSR+S is the combination of the DSR protocol with SURE. It similarly performs a two-stage route discovery as DSR. The principal difference lies in the processing of received RREP and RREQ at intermediate nodes, which is modified to perform according to the pseudo-code given in Figure 4.2. Note that SURE is not used in every route discovery, but only during flooding in the second stage where the observed phenomenon can occur. Therefore its performance impact can be limited by the use of Ring Zero search. We shall revisit this issue when discussing the performance results in Section 4.4.2.

## 4.3 Simulation Environment and Parameters

The *ns-2* [6] simulator as described in Section 3.3 is used for evaluating the proposed technique in this study. However, we use different values for the simulation parameters (except for the mobility model and node speed) in order to effect an increase in Cached RREP generated by the DSR protocol, so that we can evaluate the extent to which the

SURE technique can improve the protocol performance. The following summarizes the simulation parameters used.

A total of 100 nodes are simulated for 500s over a network space of 1342m x 1342m. The network traffic is modeled as 40 CBR sources with data sent in 64-byte packets at 2 packets/s. Five mobility scenarios are generated based on random waypoint model [52] and the mean of each performance metric described in Section 3.3.4 is presented over these 5 runs. The pause time is varied between 0 and 500 seconds. A pause time of 0s corresponds to continuous motion (at speed of up to 20m/s), while a pause time of 500s (length of simulation) corresponds to no motion.

Note that a square site is used instead of a rectangular site, for the network space in this study. A square site is known to have a higher average node degree and better distribution of routing information [55]. A higher node degree facilitates the propagation of RREQ, while a better distribution of routing information increases the ability of queried nodes to reply using their route caches. More nodes would thus receive and response to the RREQ by sending Cached RREP. A wider site of 1342m x 1342m is used to increase the average distance (and thus the cost of discovering routes) between source and destination pairs. A network size of up to 100 nodes is used to similarly allow a reasonably timed simulation. To avert congestion from a higher traffic load of 40 CBR sources, the data sending rate is reduced to 2 packets/s, while a smaller data payload of 64 bytes is used. The metrics used for evaluating the protocol performance are as given in Section 3.3.4.

## 4.4 Performance Results and Analysis

### 4.4.1 Route Discovery Overhead

We first refer to Figure 4.3, which compares the number of RREP generated by both DSR and DSR+S. The RREP are further segregated into Cached RREP and Target RREP, the former being generated by non-destination (intermediate) nodes, the latter by destination nodes. Due to source routing and aggressive caching, DSR has an inherent high hit rate for its route caches [44], which could explain why there is much more Cached RREP than Target RREP as shown in the figure.

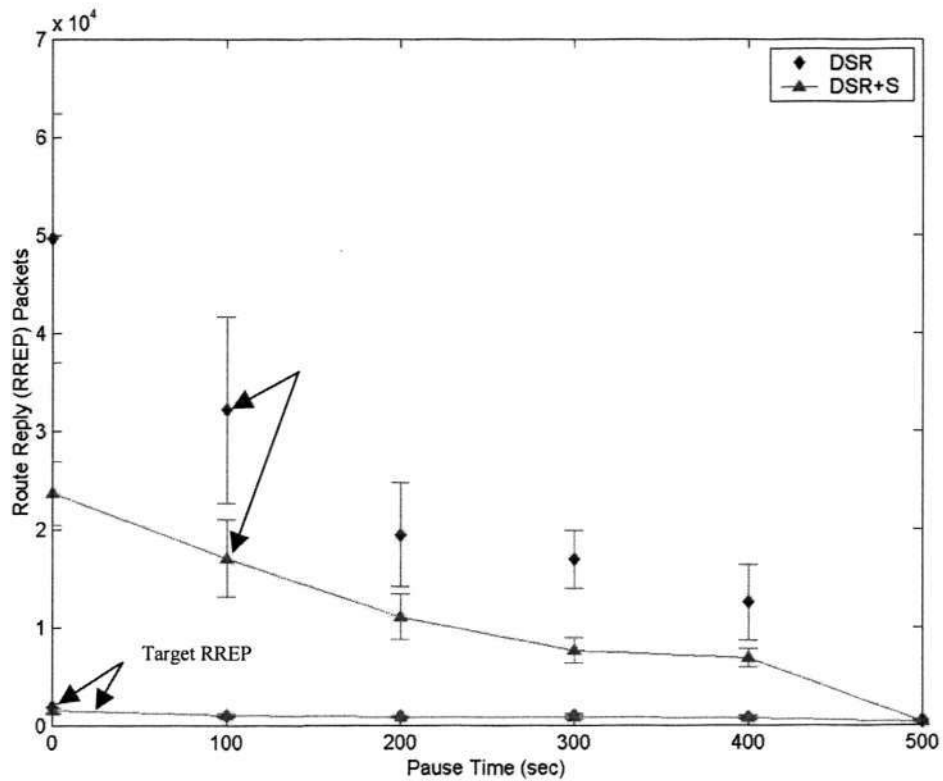


Figure 4.3: Comparison of RREP packets between DSR and DSR+S

The results above show that the SURE technique reduces the number of Cached RREP significantly, in particular at higher node mobility (lower pause time). At pause time of 0s (highest mobility) where all nodes are in continuous motion, the number of Cached RREP is reduced by more than 50%. And expectedly, this margin of improvement decreases with mobility, since lower speed leads to fewer route discoveries to be performed. At pause time of 500s (lowest mobility) where all nodes are stationary, no significant difference in Cached RREP is observed. Also, since our optimization is aimed at Cached RREP, the number of Target RREP remains relatively unchanged. Figure 4.4a and b shows the overall route discovery overhead, comprising both RREQ and RREP, of DSR and DSR+S,

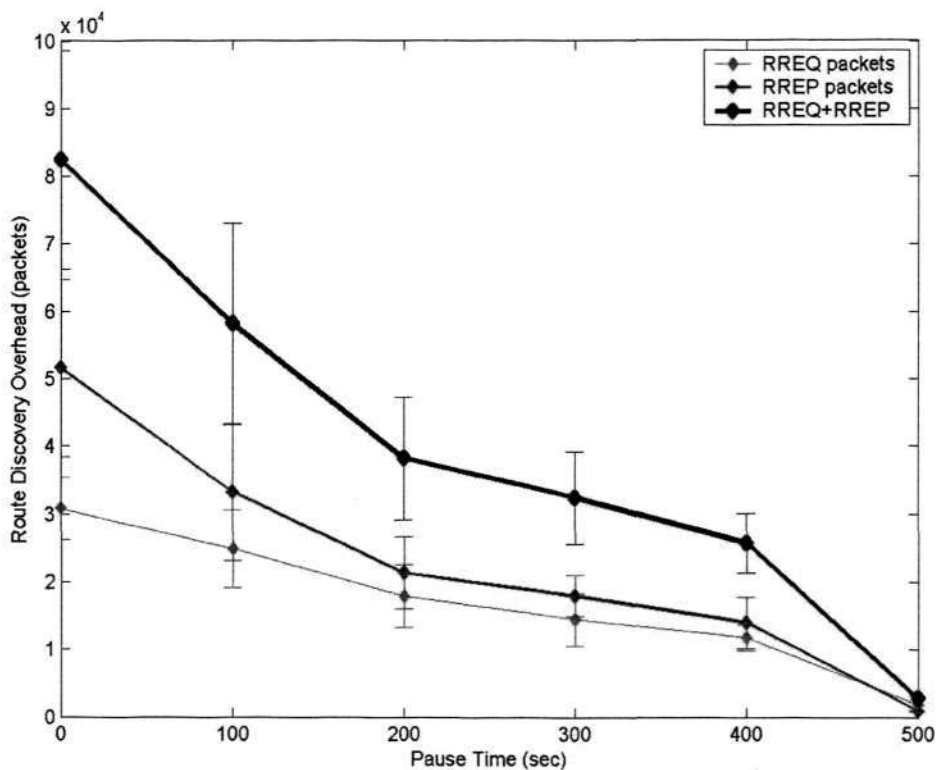


Figure 4.4a: Number of route discovery packets by DSR

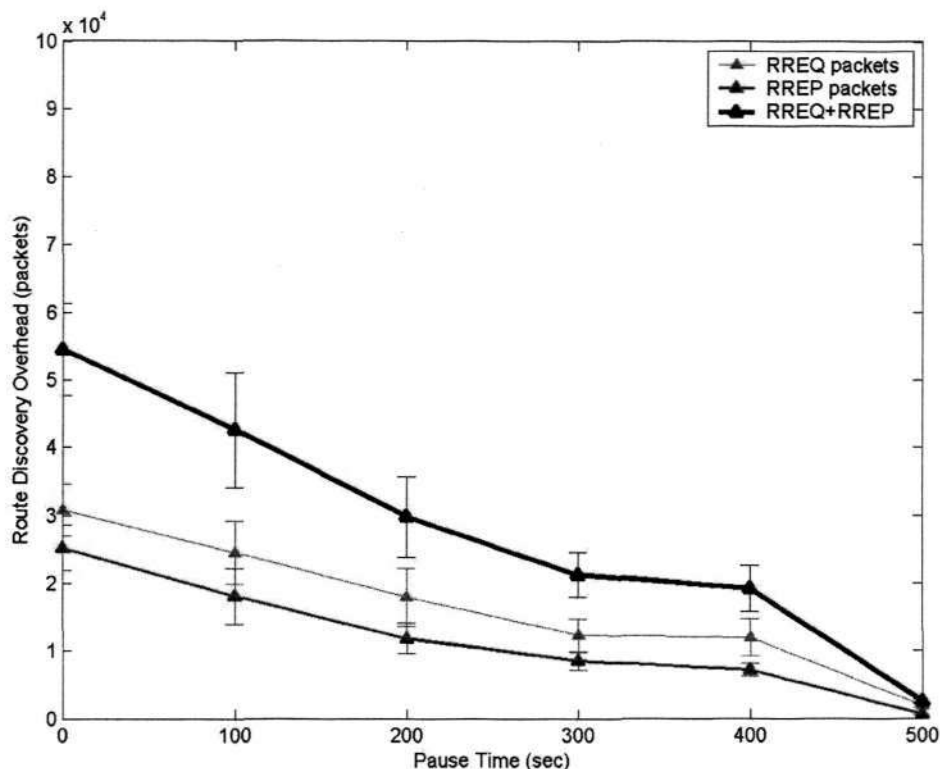


Figure 4.4b: Number of route discovery packets by DSR+S

respectively. By taking RREQ into account, DSR+S achieves an overall reduction of 37.3% under highest mobility (zero pause time).

#### 4.4.2 Total Routing Overhead and MAC Overhead

The previous section reported the route discovery overhead. Here, we consider the total routing overhead that will also comprise route maintenance packets, which are packets sent to monitor or maintain routes in use. These are RERR packets plus a small number of gratuitous RREP sent to optimize (shorten) the routes in use over time. Figure 4.5 shows the total routing overhead as a function of pause time.

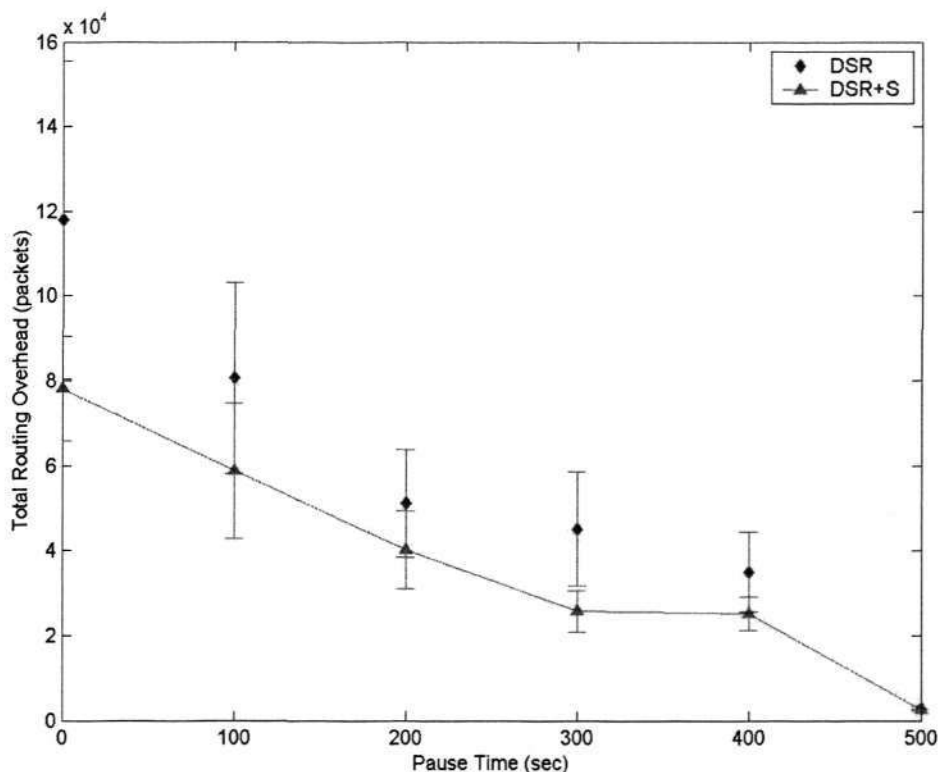


Figure 4.5: Total routing overhead vs. pause time

The results show that the total routing overhead of DSR is reduced by as much as 37.5% under highest mobility (pause time = 0s). This percentage of improvement is not far from that of the route discovery overhead (Section 4.4.1), due to a similar margin of reduction in route maintenance packets. Although SURE is targeting at reducing Cached RREP, the suppression of RREP that contain less effective (longer) routes, reduces the potential for route breaks, which in turn reduces the RERR. Also, by having less to route using longer routes, fewer gratuitous RREP are sent to optimize the route length.

Figure 4.6 further shows the overhead results of both protocols with no Ring Zero (noRZ) incorporated, i.e. each route discovery is performed via flooding. Using this protocol

configuration is to evaluate the maximum improvement achievable by SURE, which is otherwise limited by the successes of Ring Zero, as briefly mentioned in Section 4.2.1. The results show that without Ring Zero, the total routing overhead of DSR and DSR+S increases by as much as 2.5 and 1.5 times, respectively, while the maximum (relative) percentage improvement increases to 60%.

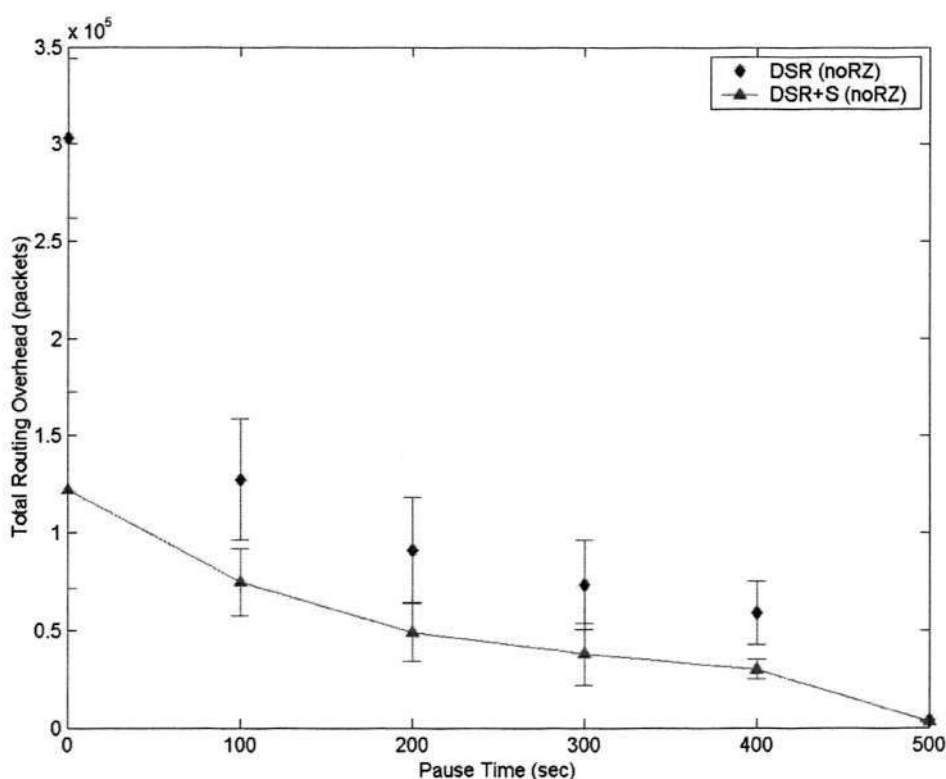


Figure 4.6: Total routing overhead with no Ring Zero search

Figure 4.7 summarizes the overhead results and shows the composition of routing packets for each protocol under highest mobility (pause time = 0s). Note the proportion of Cached RREP (CREP) in the figure for DSR and DSR (noRZ). CREP is a significant component that accounts for almost half the sum of routing packets generated, which highlights the

need to control transmission of this type of routing packet. In DSR, the transmission of unicast packets such as RREP, RERR and DATA packets, incur extra overhead at the MAC layer due to the use of RTS, CTS and ACK packets. Under the worst cast scenario with zero pause time, the MAC overhead packets generated by DSR+S is 30.5% less than DSR due to fewer RREP and RERR transmitted.

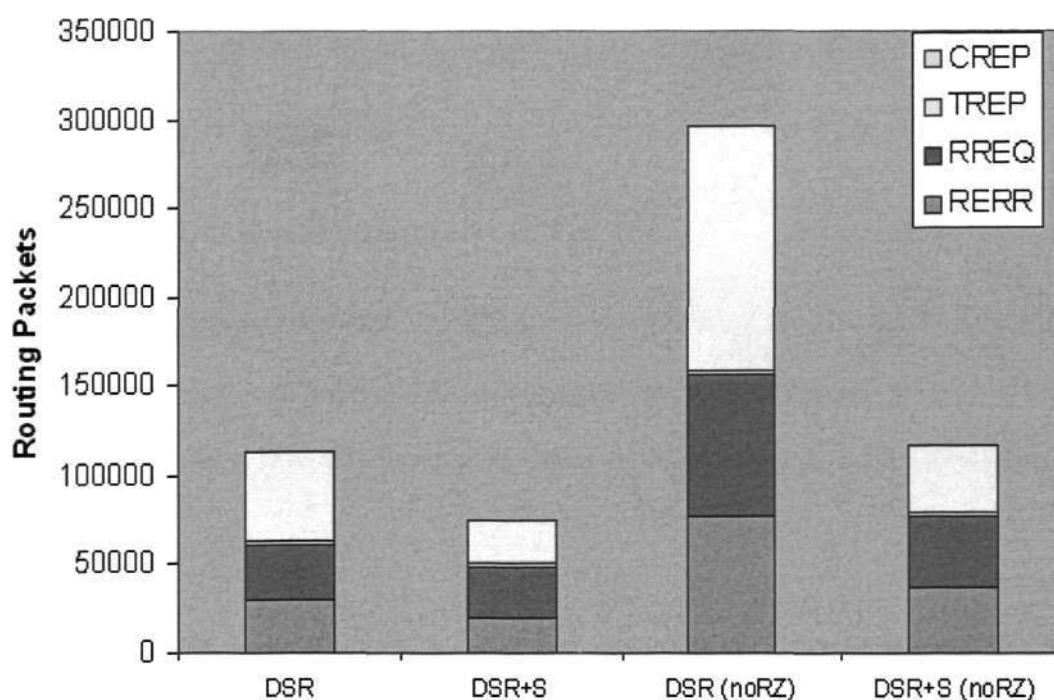


Figure 4.7: Composition of routing packets of each protocol

### 4.4.3 Route Discovery Latency

Figure 4.8 shows the route discovery latency as a function of pause time. In general, the latency increases with mobility. For DSR, the variability of its latency, as observed from its confidence intervals, also clearly increases with mobility, which is a sign of congestion.

This would impact the forwarding delay of RREQ and RREP packets. The effect of the number of hops traversed by these packets on route discovery latency is also investigated. However, no major difference in the hop-count traversed is found, as shown in Table 4.1. Thus, the observed latency differential between the protocols at higher mobility should be caused by mainly the higher forwarding delays due to routing loads.

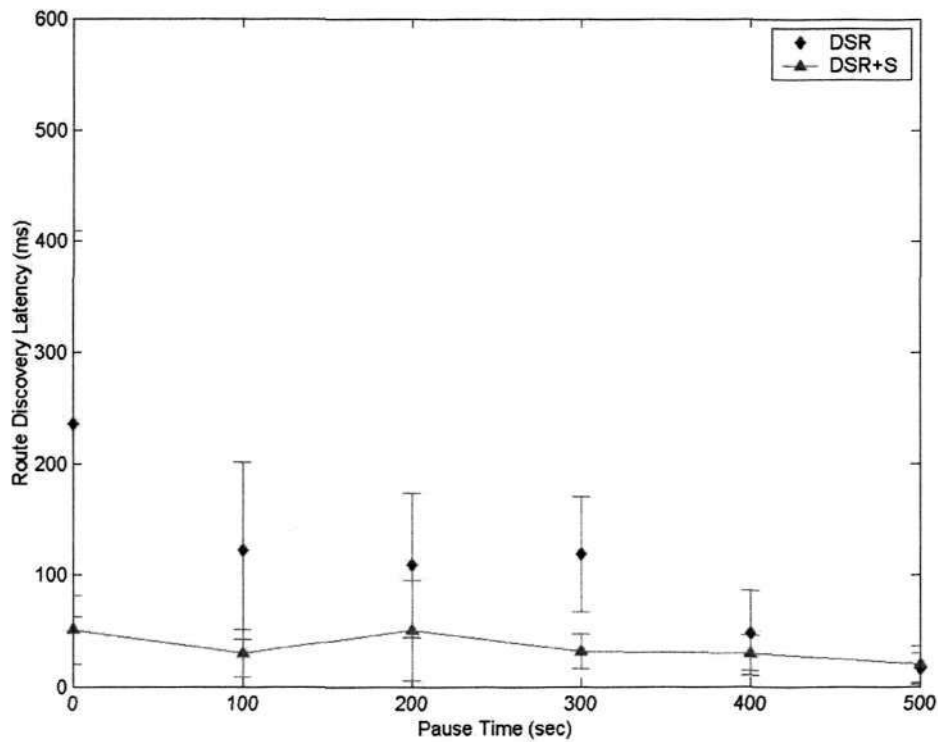


Figure 4.8: Route discovery latency vs. pause time

Table 4.1: Number of hops traversed by first received RREP

Protocol	Hops traversed by first RREP
DSR	2.80
DSR+S	2.90

#### 4.4.4 End-to-End Delay

The end-to-end delay of a data packet includes the initial route discovery latency, and the subsequent forwarding delays it experiences along the route to its destination. Similar to route discovery, the end-to-end delay shown in Figure 4.9, exhibits a similar pattern of change with mobility. For DSR+S, a lower routing load means fewer routing packets compete with data packets for channel access, resulting in fewer packet collisions and backoffs that may prolong end-to-end delay. Given that routing packets are often given a higher priority to transmit than data packets in interface queues, fewer routing packets also lead to shorter queuing time for data packets waiting to be transmitted.

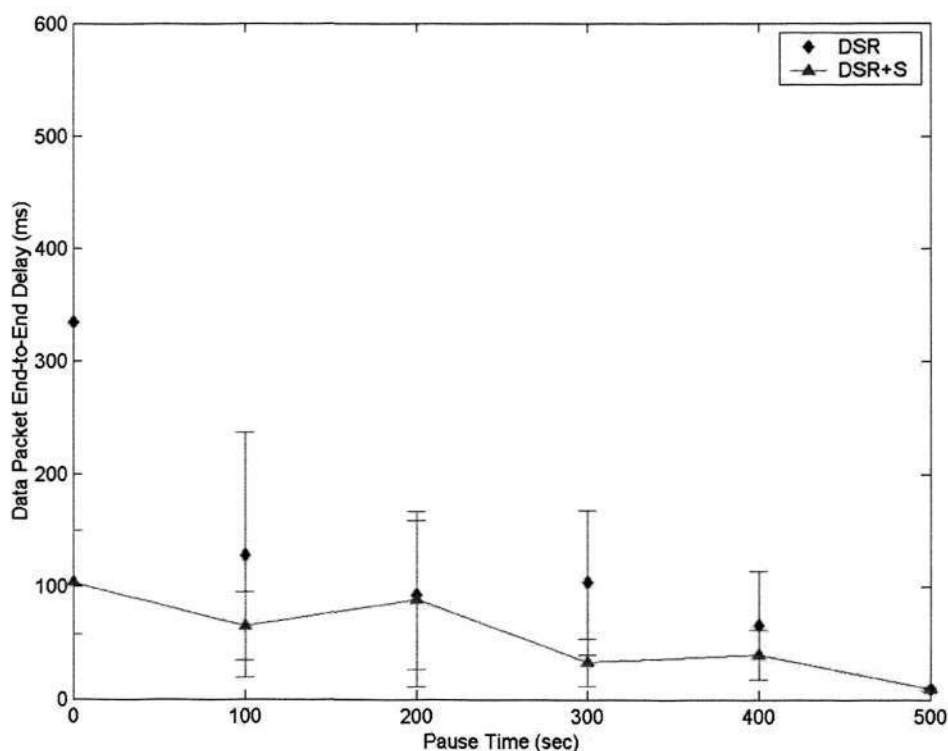


Figure 4.9: End-to-end delay vs. pause time

Besides the congestion effect given above, the number of hops traversed by the data packet to reach its destination may also impact the end-to-end delay. Figure 4.10 shows the hop-wise optimality of the paths used in each protocol for packet delivery. DSR+S is found to have slightly better path optimality than DSR, which could be due to its suppression of RREP packets that contain longer routes during route discovery (Section 4.1.2). But in general, the optimality of both protocols is fairly similar, with a large fraction of packets delivered using optimal (shortest), or near-optimal paths with one or few extra hops.

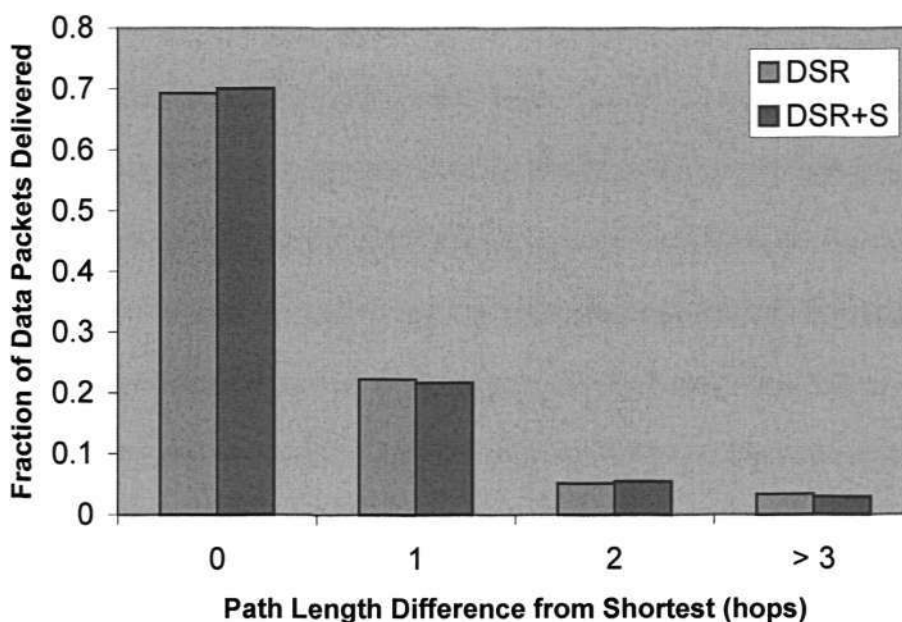


Figure 4.10: Path optimality of data packets

#### 4.4.5 Packet Delivery Ratio

The percentage of data packets successfully delivered to their destination decreases with increasing mobility (lower pause time), as shown in Figure 4.11. At the highest mobility (pause time = 0s), the packet delivery ratio of DSR decreases to below 90%, while that of DSR+S still remains above. This can be attributed in part to *reduced routing failure* and in part to *reduced congestion*, which can be observed from the packet dropped summary in Table 4.2 (shown for pause time = 0). Fewer packets encountered routing failure (no route) in DSR+S, as reflected in the table and from the reduced RERR in Figure 4.7, because less effective routes are prevented from being returned and used for routing packets when SURE is employed. The resulting lower routing load also contributes to reduced congestion, which in turn reduces packet drops as a result of buffer or queue overflow.

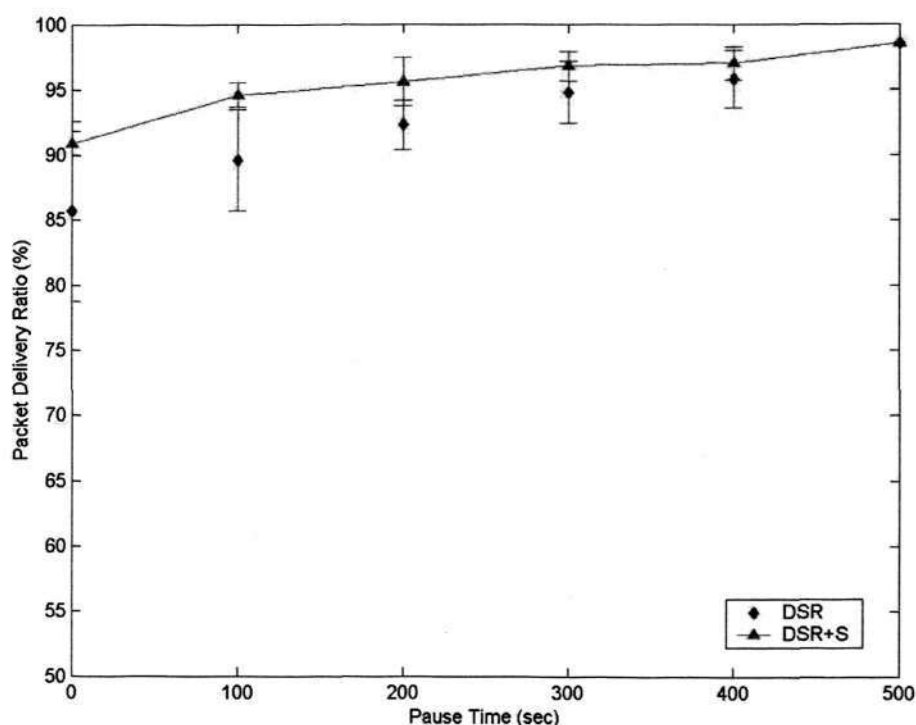


Figure 4.11: Packet delivery ratio vs. pause time

Table 4.2: Data packet dropped summary for the two protocols

Data Packet Dropped Summary	DSR	DSR+S
No Route	2401	2307
TTL Expired	0	0
RTR Queue Full	0	0
Timeout	137	141
Routing Loop	0	0
IFQ Full	757	36
ARP Full	138	153
MAC Callback	0	0
Simulation End	107	44
Total Packets Dropped	3540	2681

#### 4.4.6 Effect of Network Size and Traffic Load

This section further investigates the effect of different network size and traffic load on the performance metrics. As in the previous chapter, we chose to vary either the network size (number of nodes) or the traffic load (number of sources), but not both at the same time, while keeping other parameters unchanged. We present results for a higher traffic load of 50 sources (highest without causing severe congestion) and a reduced network size of 50 nodes. A network with more than 100 nodes is not used because of simulation constraints as described in Section 3.4.6. Table 4.3 summarizes the previous results for a 100 node model with 40 sources, under highest mobility (pause time = 0s). As before, the overhead

Table 4.3: Summary of results for a 100 node model with 40 sources

Performance metrics	DSR	DSR+S	% Improvement
Normalized route discovery overhead	2.92	1.83	37.3
Normalized total routing overhead	4.19	2.62	37.5
Normalized total MAC overhead	31.7	22.0	30.5
Route discovery latency (ms)	236	41.7	82.3
End-to-end delay (ms)	334	104	68.9
Packet delivery ratio (%)	85.7	90.9	6.07

results are normalized to the total data packets delivered so that they can be compared to results of the 50 node model, which we simulated for a longer duration of 900s.

The effect of increased traffic load is shown in Table 4.4a. Both protocols generated more routing and MAC overhead as more routes are discovered and maintained for the increase number of data sources. Route discovery, end-to-end delay and packet delivery performance deteriorated due to congestion and collisions caused by higher data and control traffic. However, DSR+S still performed significantly better than DSR across all metrics. The margin of improvement is higher with a larger number of sources.

Table 4.4: Effect of changing network size and traffic load

Performance metrics	DSR	DSR+S	% Improvement
Normalized route discovery overhead	7.42	2.45	67.0
Normalized total routing overhead	11.2	4.31	61.5
Normalized total MAC overhead	56.0	28.9	48.4
Route discovery latency (ms)	1387	589	57.5
End-to-end delay (ms)	1641	649	60.5
Packet delivery ratio (%)	52.6	76.1	44.7

(a) Traffic load: 50 sources. Network size maintained at 100 nodes

Performance metrics	DSR	DSR+S	% Improvement
Normalized route discovery overhead	1.12	1.0	9.82
Normalized total routing overhead	1.63	1.5	10.4
Normalized total MAC overhead	17.2	16.3	5.07
Route discovery latency (ms)	55.6	37.6	32.4
End-to-end delay (ms)	567	539	4.94
Packet delivery ratio (%)	87.7	90.1	2.74

(b) Network size: 50 nodes. Traffic load maintained at 40 sources

With a smaller network size of 50 nodes, the overhead generated is smaller and so is the margin of performance improvement (Table 4.4b). Packet delivery performance seems to be relatively unchanged, suggesting that the network is still sufficiently connected even though the number of nodes is reduced. An interesting observation, though, is the increase in end-to-end delay, which contradicted our earlier intuition that less overhead and hence lower congestion would reduce the end-to-end delay. We checked that both 50 and 100 node models have similar average route length of approximately 4 hops. Thus, it is not due to an increase in route length that increases the end-to-end delay.

Instead, buffering delay at the source nodes for route discovery is the primary factor. This is based on our finding that cache hit rate of the 50 node model is much lower than that of 100 node model by an order of magnitude (Table 4.5). This higher cache misses means that more data packets must wait at the source (in a buffer) for a route to their destinations to be found, and not send immediately, leading to higher delivery delay. The lower cache hit rate of the 50 node model is attributed to reduced route diversity or fewer number of routes that can exist with a reduced number of nodes.

Table 4.5: Cache hit rates with 40 sources and zero pause time

Number of nodes	Cache hit rates	
	DSR	DSR+S
50	0.0286	0.0294
100	0.3431	0.2472

## 4.5 Chapter Summary

In this chapter, we first report and investigate an observation of a previously unreported phenomenon of *reversed query/response arriving order* that is seen to occur at some nodes during a route discovery. Our investigation attributes this rarely noted but commonly occurring phenomenon to several effects, including channel contention, path length, packet collision, and mobility, which causes a reversal of the arriving order of RREQ and RREP packets at these nodes. The observation inspired the proposal of a second optimization known as the SURE (SUppress REply) technique, which is targeting RREP packets. The proposed optimization is simple in concept and implementation, and its application to the DSR reactive routing protocol, particularly in stressful environments such as under high mobility and traffic load, is found to yield significant improvements in both protocol efficiency and performance.

## Chapter 5

### Joint Application of UNIQUE and SURE

This chapter is a logical follow-up from the previous two chapters that presented the UNIQUE and SURE optimizations. In this chapter, we study the joint application of these two independent and complementary techniques in an effort to attain even better routing performance than when either one is applied alone.

#### 5.1 Protocol Description

Both UNIQUE and SURE are jointly applied to DSR [22] which is the underlying routing protocol used throughout this thesis. Four protocols are studied: i) DSR; ii) DSR+L+U; iii) DSR+S; and iv) DSR+AllOpt. The description of DSR and DSR+L+U are as given in Section 3.2. DSR+L is excluded since it did not perform well relative to the other two protocols in the previous study as we discussed in Chapter 3. DSR+S is as described in Section 4.2.1, while DSR+AllOpt is the combination of ii) and iii), with UNIQUE and SURE optimizations jointly applied.

In DSR+AllOpt, UNIQUE and SURE are implemented as given in the pseudo-code of Figure 3.2 and 4.2, respectively. However, SURE is slightly modified to include a prior check on the RREQ header and ignore any RREQ that is sent as a unicast message for a UNIQUE route discovery. This check is not originally required in DSR+S because every RREQ then is a broadcast message.

## 5.2 Simulation Environment and Parameters

As before, the *ns-2* [6] simulator is used. The simulation parameters are the same as that given in Section 4.3, except for the data traffic which now consists of 20 sources, each sending 512-byte packets at 4 packets per second. This is the same traffic pattern used in Chapter 3 for the evaluation of UNIQUE. We use this traffic pattern because it injects a reasonably heavy load into the network and, more importantly, the larger packet size of 512 bytes reduces the overhead of appending location information (needed by UNIQUE) to each packet as a percentage of data payload. With each location tuple  $(x, y, t)$  requiring 12 bytes and an average route length of 4, the location bytes are  $< 10\%$  of data payload, compared to  $> 50\%$ , if a packet size of only 64 bytes is used.

## 5.3 Performance Results and Analysis

The protocols are evaluated according to the performance metrics given in Section 3.3.4. Figure 5.1 shows the results of route discovery overhead. Remaining much the same is the pattern of change in overhead with pause time, i.e. higher overhead with lower pause time, and vice-versa, and also the increasing performance differential between protocols as mobility increases. We first observe the relative ranking of the protocols, and then their performance differentials, especially at highest mobility with zero pause time.

Much expectedly, DSR+AllOpt, with UNIQUE and SURE jointly applied, has the lowest overhead among the protocols. DSR+L+U is the next more efficient protocol, followed by DSR+S, and finally DSR. In terms of percentage improvement, DSR+AllOpt reduces the route discovery overhead of DSR by as much as 76%, while DSR+L+U and DSR+S each

reduces by 68% and 43%, respectively. This result suggests that between UNIQUE and SURE, the former is more effective in reducing overhead of route discovery. Although UNIQUE is primarily designed for RREQ reduction, it also inherently (and effectively) reduces the RREP by: i) limiting the propagation area of RREQ, thus limiting the nodes that will receive and subsequently reply to the RREQ; ii) sending the RREQ as a unicast message, thus only the addressed node and not all nodes that receive the RREQ can reply. However, for its better performance, UNIQUE requires nodes to be location-aware, while SURE does not and is comparatively simpler to implement. Therefore, as always, there is a tradeoff between complexity and performance.

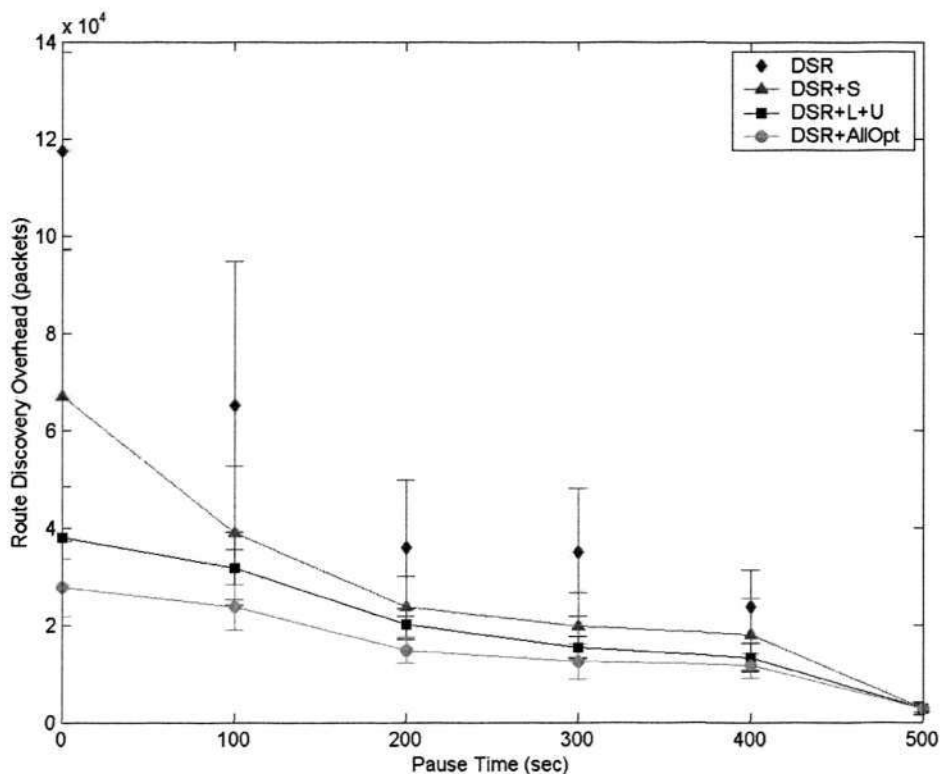


Figure 5.1: Route discovery overhead as a function of pause time

It is also worth noting that in a joint application, the performance contribution of SURE may be marginalized by the RREP reduction by UNIQUE, resulting in only a marginal further reduction in overhead, as shown by the performance differentials in Figure 5.1 between DSR+AllOpt and DSR+L+U.

Figure 5.2 shows the total routing overhead. This includes both route discovery and route maintenance packets. The latter are mostly RERR packets. The ranking of the protocols remains unchanged. As for their performance differentials, which are especially marked at pause times < 200s, DSR+AllOpt reduces DSR's total routing overhead by as much as 78%, while DSR+L+U and DSR+S each reduces by 72% and 30%, respectively.

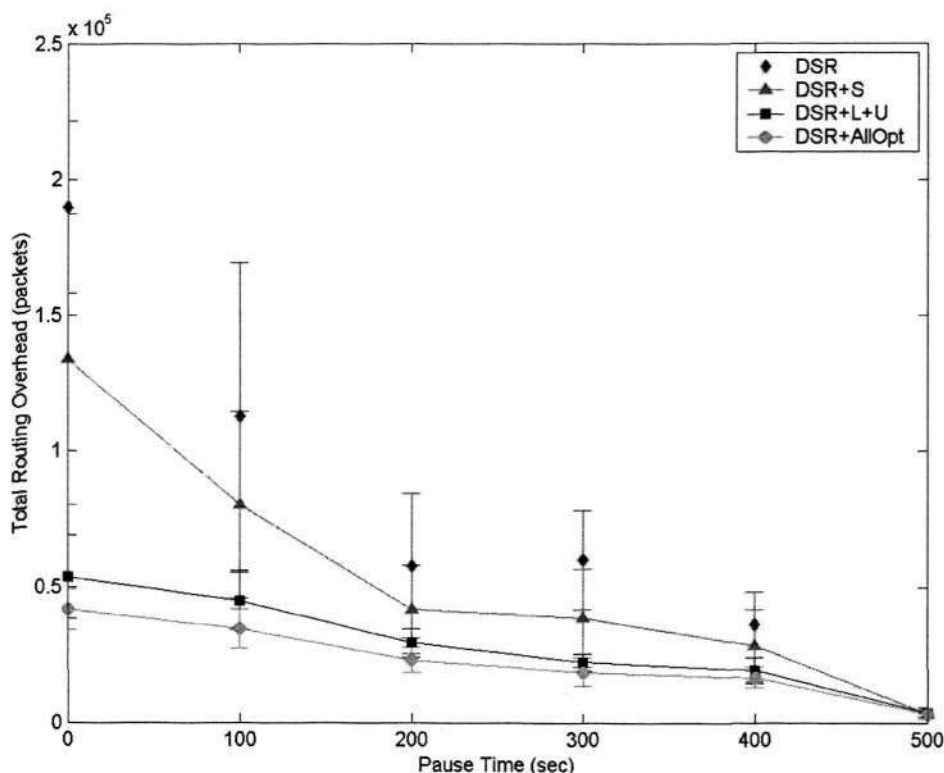


Figure 5.2: Total routing overhead as a function of pause time

It may be interesting to note that the inclusion of route maintenance packets increases the percentage improvement of DSR+L+U, but reduces that of DSR+S. For DSR+L+U, the increased improvement is the result of a much smaller number of RERR, as mentioned in Section 3.4.2. Comparatively, DSR+S has more RERR as seen from the constituent data shown for zero pause time in Figure 5.3.

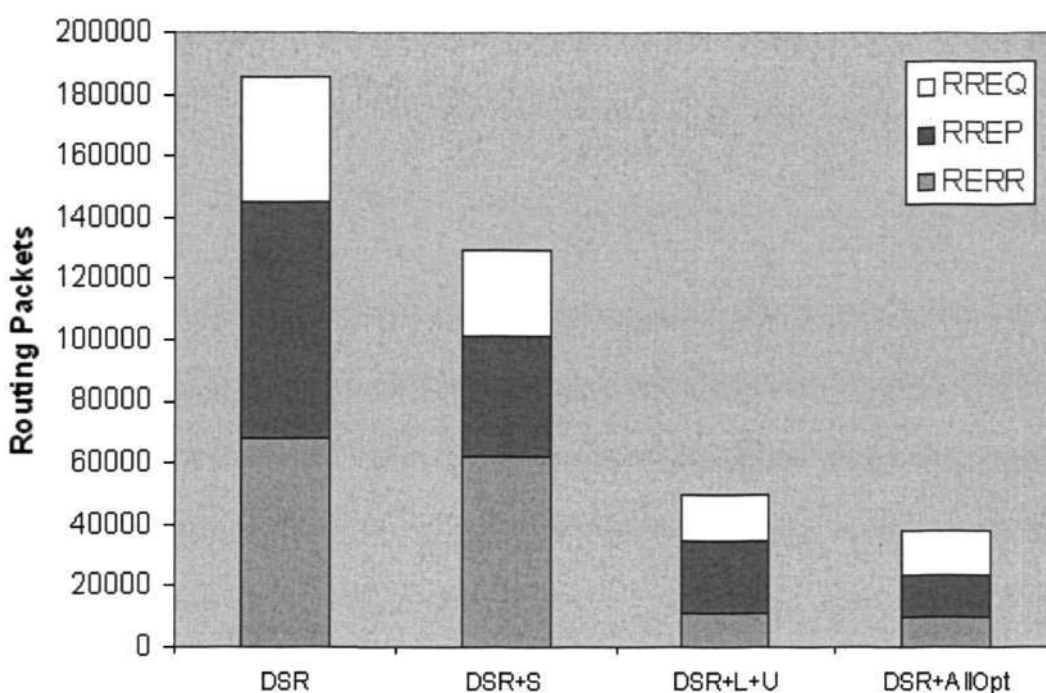


Figure 5.3: Histogram showing the constituents of routing packets

The reason for this could be that the larger packet size used in this study increases the packet’s delivery time due to increased transmission and queuing delays, which in turn increases the packet’s vulnerability to route errors. That is, the longer time it takes for a packet to reach its destination, the higher the probability that the packet will encounter a broken route due to topology changes with mobility over time.

In fact, the same could be said about DSR+L+U, since the same packet size is used. However, both protocols could react differently to route error, as each may use a different technique to discover a new route. For example, DSR+L+U uses UNIQUE, while DSR+S uses flooding (optimized with SURE). Since routing load is higher in DSR+S, there is higher potential for route error to occur, as packet delivery could be further delayed as a result of higher congestion.

In terms of MAC signaling overhead, the performance differentials with respect to DSR are shown in Table 5.1. However, in order for this result to be understood more clearly, we defer its discussion until we present the data packet delivery ratio given in Figure 5.6. Results for the route discovery latency, and data packet end-to-end delay are shown in Figure 5.4 and 5.5, respectively.

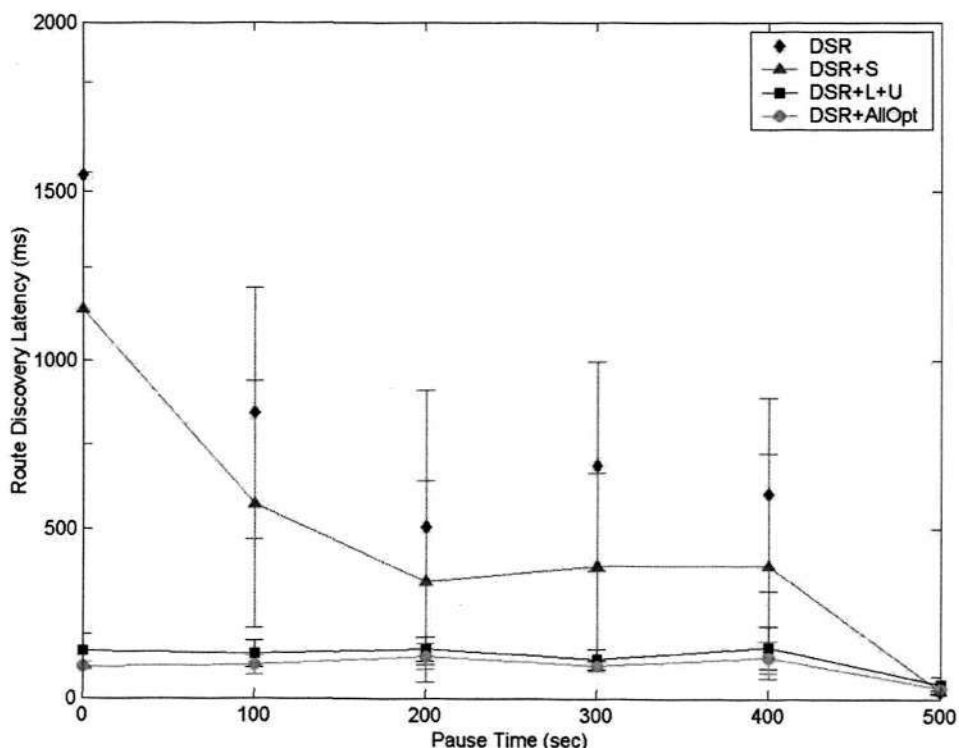


Figure 5.4: Route discovery latency

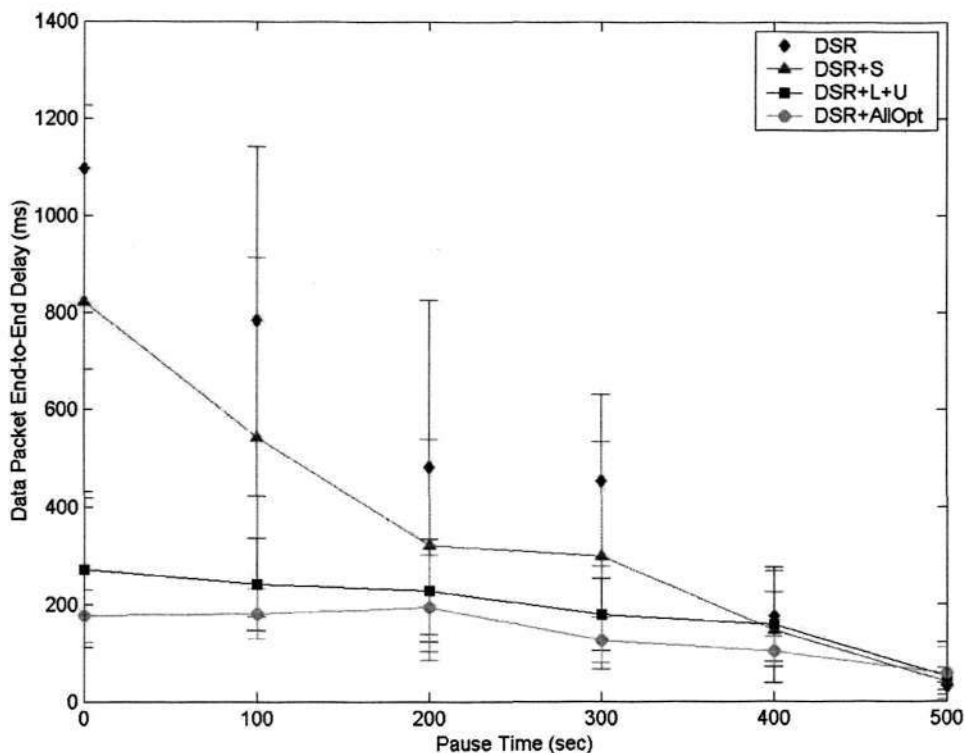


Figure 5.5: Data packet end-to-end delay

Apart from reporting the results follow a similar trend to that of the total overhead, it may be interesting to notice that route discovery latency can be higher than the end-to-end delay. This occurs for DSR and DSR+S at higher mobility, e.g. pause time zero. This is possible because the end-to-end delay is averaged over all data packets, and not every data packet incurs a route discovery latency, i.e. a route acquired for one packet can be used again for other packets as long as it remains valid.

Figure 5.6 shows DSR+AllOpt delivering the most data packets, with approximately 30% more packets than DSR, at highest mobility. Coming back to the MAC overhead results of Table 5.1. The results predictably show that protocols that incur a high routing overhead, similarly incur a high MAC overhead, and vice-versa. However, the reduction margin in

Table 5.1: MAC signaling overhead under constant mobility

Protocol	MAC packets	% Improvement over DSR
DSR	976,311	--
DSR+S	824,802	15.56%
DSR+L+U	564,495	42.18%
DSR+AllOpt	512,259	47.53%

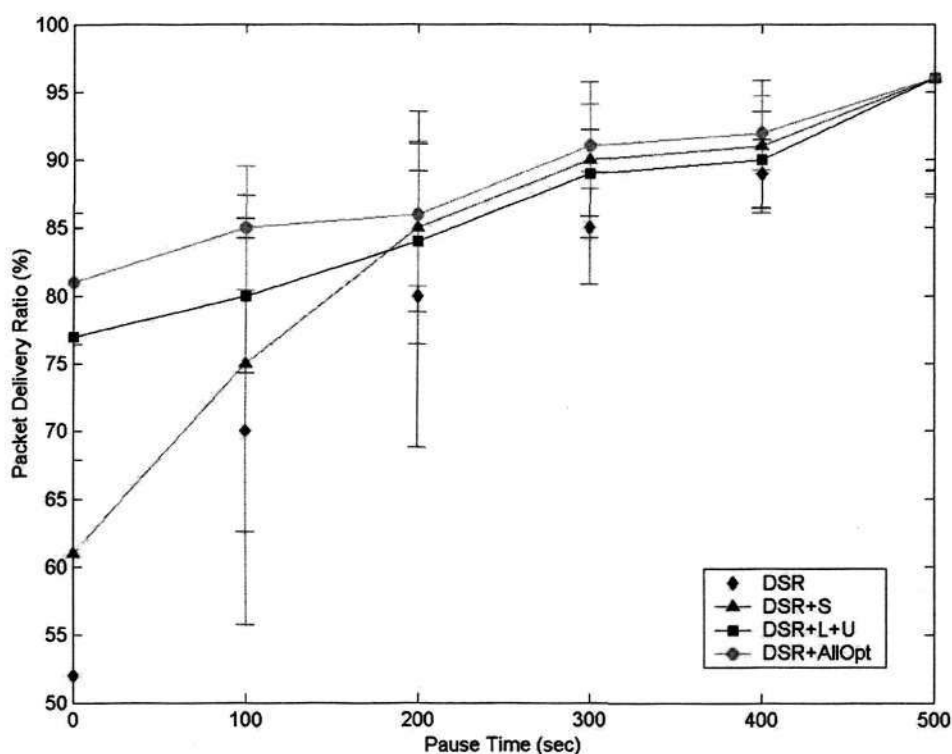


Figure 5.6: Data packet delivery ratio

MAC overhead is clearly much lower than that of routing overhead, as mentioned for Figure 5.2. For DSR+L+U and DSR+AllOpt, the reduced margin is in part due to the extra MAC overhead introduced by UNIQUE's RREQ packets, which are unicast packets that incur RTS/CTS/ACK overhead under the IEEE 802.11 MAC. A further reason, which also

led to the reduced margin in DSR+S, is the lower MAC overhead contributed by DATA packets in DSR, since a large number of them were not delivered (but dropped at high mobility), as shown in Figure 5.6.

## **5.4 Chapter Summary**

In this chapter, we investigate the possibility for further enhanced performance through joint application of the proposed techniques in this thesis. We have shown that with both UNIQUE and SURE jointly applied, the resulting protocol (DSR+AllOpt) excels in all metrics studied, the performance of which exceeds that when either one is applied alone. Between UNIQUE and SURE, the former is found to be more effective in reducing the overhead of route discovery, but at the cost of increased complexity.

## Chapter 6

### Conclusions and Future Work

This thesis focuses on optimizing the efficiency of the route discovery process, which is a major source of control overhead for reactive MANET routing protocols. The routing control packets will not only consume the limited wireless bandwidth and battery power, but also impact the delay and delivery of the data packets because of additional multiple access interference. This chapter summarizes the work of this thesis and outlines some possible areas for future work.

#### 6.1 Conclusions

##### 6.1.1 UNIQUE Optimization for RREQ Reduction

In Chapter 3, we presented a new route discovery optimization technique called UNIQUE which utilizes both location and route information to reduce RREQ of reactive MANET routing protocols. Key results of experimental studies to evaluate relative performance of UNIQUE, LAR and DSR are summarized as follows.

- The route discovery overhead of DSR is found to decrease only marginally when LAR is used. This is because DSR's native optimizations, which include caching and Ring-Zero search, are quite effective in limiting the use of flooding, and thus limits the contribution of LAR in DSR's overhead reduction. Using UNIQUE, however, decreases further significantly the DSR's route discovery overhead. UNIQUE's localizing of RREQ flooding to the vicinity of destination, and the

unicasting of RREQ which thus allow only the addressed node to reply, results in significant reduction of not only RREQ, but also RREP packets.

- Although UNIQUE decreases the route discovery overhead, it results in a higher number of route discoveries performed. This is the result of two outcomes from using route discovery techniques with limited search scope. First, if a route is not found in an initial restricted search, another attempt will be made to search over a wider area. Second, fewer alternate routes exist to be found within a reduced search area. Therefore, when a route to some target fails, a replacement route may not be available in the node cache, and route discovery may need to be performed. Indeed, we found that using route discovery techniques that carry out ‘focused’ route searches, e.g. UNIQUE and LAR as opposed to using flooding in a network-wide search, reduces the cache hit rate of the routing protocol.
- Under the worst case scenario at highest mobility (pause time = 0s), we found that UNIQUE still generates fewer MAC layer packets than DSR, despite incurring extra MAC overhead (RTS, CTS and ACK packets) for transmitting its RREQ packets as unicast messages. This is because the extra MAC overhead incurred by UNIQUE’s RREQ has been more than offset by the decrease in MAC overhead as a result of fewer RREP and RERR, which are unicast packets.
- The fewer RERR packets in UNIQUE can be attributed in part to *data packets* being routed through more *recently* discovered routes as a result of more frequent route discovery, and in part to RREQ that also acts like a *probe packet*, verifying the connectivity of each link along the cached route through which it is unicast.

Thus, some outdated links can be uncovered and removed early by nodes from their caches before they can be used by data packets to reach their destinations, causing subsequent routing errors.

- The lower routing load arising from using UNIQUE minimizes the forwarding delays of RREQ and the subsequent return of RREP, as less routing traffic reduces channel contention, while less backlog in interface queue accelerate the process of forwarding packets. The reduced forwarding delay in turn shortens the time it takes to complete the route discovery operation, i.e. route discovery latency.
- Similar to route discovery, the forwarding delay of data packets also largely depends on the routing load in the network. The reduced forwarding delay in turn shortens the time it takes to complete the data packet delivery from its source to destination, i.e. end-to-end delay.
- UNIQUE also allows more data packets to be delivered to their destinations, since fewer packets are dropped in transit due to routing congestion, and routing errors (fourth point of this section) at forwarding nodes .

### **6.1.2 SURE Optimization for RREP Reduction**

In Chapter 4, we further presented a second new optimization technique called SURE which exploits an observed route discovery phenomenon to reduce RREP of reactive MANET routing protocols. The following summarizes the key results of performance comparison between SURE and DSR.

- DSR is found to generate a large number of RREP packets for its route discovery, a significant majority of which are Cached RREP from intermediate nodes, due to its source routing and aggressive caching policy.
- When SURE is applied, the Cached RREP is reduced significantly, in particular at high node mobility (low pause time). Since the SURE technique targets primarily Cached RREP, the number of Target RREP (from destination nodes) is found to remain relatively unchanged.
- Although SURE is developed as an optimization to route discovery, it contributes also to better route maintenance by preventing the return and use of less effective (longer) routes, thereby reducing the potential for route breaks and RERR packets. With fewer RREP and RERR, both of which are unicast packets, SURE decreases DSR's MAC overhead (RTS, CTS, and ACK packets) significantly as well.
- With SURE, the resulting lower routing load not only decreases route discovery latency, but also improves the data end-to-end delay performance. A lower routing load means fewer routing packets compete with data packets for channel access, resulting in fewer packet collisions and backoffs that may prolong end-to-end delay. Given that routing packets are often given a higher priority to transmit than data packets in interface queues, fewer routing packets also lead to shorter queuing time for data packets waiting to be transmitted.
- SURE allows more data packets to be delivered to their destinations. This can be attributed in part to *reduced routing failure* and in part to *reduced congestion*.

Fewer packets encountered routing failure as reflected by the reduced RERR, because less effective routes are prevented from being returned and used for routing packets when SURE is employed. The resulting lower routing load also contributes to reduced congestion, which in turn reduces packet drops as a result of buffer or queue overflow.

### **6.1.3 Joint Application of UNIQUE and SURE**

In Chapter 5, we investigated the possibility for further enhanced performance through joint application of these two independent and complementary techniques: UNIQUE and SURE, proposed in this thesis for optimizing route discovery. The following reports two key findings from the investigation.

- With both UNIQUE and SURE jointly applied, the resulting protocol excels in all metrics studied (route discovery overhead, total routing and MAC overhead, route discovery latency, end-to-end delay, and packet delivery ratio), the performance of which exceeds that when either one is applied alone.
- We also found that between UNIQUE and SURE, the former is more effective in reducing the overhead of route discovery, but at the cost of increased complexity. As mentioned, although UNIQUE is primarily designed for RREQ reduction, it also inherently reduces the RREP by: i) limiting the propagation area of RREQ, thus limiting the nodes that will receive and subsequently reply to the RREQ; ii) sending the RREQ as a unicast message, thus only the addressed node and not all nodes that receive the RREQ can reply. However, for its better performance,

UNIQUE requires nodes to be location-aware, while SURE does not and is comparatively simpler to implement. Therefore, as always, there is a tradeoff between complexity and performance.

## 6.2 Future Work

We conclude this thesis by outlining some possible areas for future work.

### i) Extension for Load Balancing Support in UNIQUE

In this thesis, we have not considered load distribution issue in the selection of UQPath (Section 3.1.1). As a result, some routes could be more loaded with data traffic than others. For example, if an existing route already in use for a data session is selected as the UQPath and utilized subsequently for another data session, the resulting traffic concentration may cause traffic along this route to experience higher delay and packet loss. More intelligent selection of UQPath is therefore desired. Traffic load could be used as a further selection criterion. Extending UNIQUE to incorporate support for load balancing, by either developing new or adapting existing (e.g. [56]) methods for monitoring congestion status of routes and dissemination of traffic load information, should be explored.

### ii) An Alternative UNIQUE Strategy

This sub-section proposes a possible alternative UNIQUE (UNICAST QUERY) strategy that does not depend on node location information. It works on the hypothesis that if some source node  $A$  initiates a route discovery to a destination node  $B$ , then  $A$  and perhaps any of its direct neighbors could be potential sources of route to  $B$ . If another node  $C$  wants to

obtain a route to  $B$ , then instead of flooding the entire network,  $C$  may *unicast* its query (or RREQ) to  $A$ , or to any of  $A$ 's direct neighbors to which  $C$  has a route. These direct neighbors, which could be known from the received RREQ from  $A$ , may have overheard  $A$ 's transmission to  $B$ , thus knowing the route to  $B$  as well.

After obtaining the route to  $B$ , the routing path may be post-processed to optimize its hop length prior to being used for data routing. As an example, for source routing from  $C$  to  $B$ , if  $C-F-A$  is the sub-path from  $C$  to  $A$ , and  $A-D-E-B$  is the sub-path from  $A$  to  $B$ , then the full path from  $C$  to  $B$  is  $C-F-A-D-E-B$ , which has a hop length of 5. But  $C$  may have a shorter route to certain nodes on this path. For instance, if  $C$  has an existing route to  $E$  through a single node  $G$ , then the full path could be shortened to  $C-G-E-B$ , which has a hop length of 3. An illustration is shown in Figure 6.1. Further refinements to the basic idea presented herein could be worth exploring.

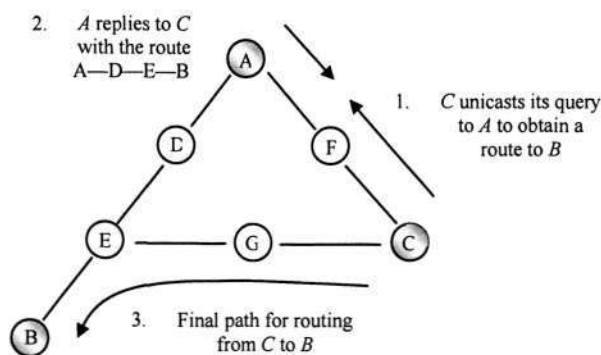


Figure 6.1: An alternative UNIQUE strategy

### iii) Another SURE Technique

Previously, SURE (SUPpress REply) aims to suppress only the *origination* of RREP, i.e. preventing a node from sending RREP if it does not have a shorter route to the destination than the one previously observed. We now further propose that SURE also suppresses the *forwarding* of RREP. That is, preventing a node from forwarding to the source, a RREP originated from another node, if the route carried by this RREP is not shorter than the one previously observed. An illustration is provided as shown in Figure 6.2.

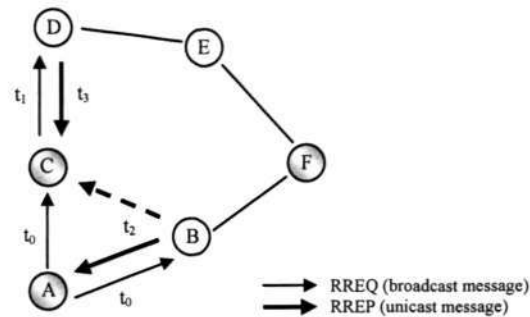


Figure 6.2: Suppress the forwarding of RREP

At time  $t_0$ , the source node  $A$  broadcasts a RREQ seeking a route to destination node  $F$ . This RREQ is received by both nodes  $B$  and  $C$ . At time  $t_1$ ,  $C$  rebroadcasts the RREQ to  $D$ . On the other hand,  $B$  has a route to  $F$  ( $B-F$ ), and replies to  $A$  at time  $t_2$ . This RREP is also observed by  $C$ , which is within range of  $B$ . Similarly,  $D$  has a route to  $F$  ( $D-E-F$ ) and replies at time  $t_3$ . However, when  $C$  receives this RREP from  $D$ , it finds that the route ( $D-E-F$ ) returned by  $D$  is longer than that ( $B-F$ ) returned by  $B$ . Thus,  $C$  discards the RREP from  $D$ , instead of forwarding it to  $A$ .

For a large network where it may take RREP many hops to be forwarded to the source, the proposed approach can potentially result in significant savings in RREP packets, and thus deserves to be further investigated.

As an ending note, it is conceivable that the SURE technique developed in this thesis for suppressing ineffective Cached RREP can be adapted for similar suppression of undesired *Cached Reply* messages in applications that utilize a similar reactive flooding approach for item search or discovery. Likely applications includes (destination) location discovery in Reactive Location Services (RLS) [57], and content searching in MANET Peer-to-Peer (P2P) file sharing applications [58].

## Bibliography

- [1] T. Rappaport, "Wireless Communications: Principles and Practice", 2nd Edition, Prentice Hall, December 2001.
- [2] Internet Engineering Task Force (IETF), <http://www.ietf.org>
- [3] E. Royer and C-K Toh "A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks", in *IEEE Personal Communications Magazine*, April 1999, pp. 46-55.
- [4] Y.-C. Tseng, S.-Y. Ni, Yuh-Shyan Chen, and J.-P. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," in *ACM Journal of Wireless Networks*, vol. 8, no. 2, March 2002, pp. 153-167.
- [5] Y. Ko and N. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks", in *ACM/Kluwer Journal of Wireless Networks (WINET)*, vol.6, no.4, July 2000, pp. 307-321.
- [6] VINT network simulator version 2 (ns-2), <http://www.isi.edu/nsnam/ns/>
- [7] IEEE P802.11: The Working Group for Wireless LAN Standards, <http://grouper.ieee.org/groups/802/11/>
- [8] Robert E. Kahn, Steven A. Gronemeyer, Jerry Burchfiel, and Ronald Kunzelman, "Advances in Packet Radio Technology", in *Proceedings of the IEEE*, vol. 66, no.11, November 1978, pp. 1468-1496.

- [9] J. Jubin and J. D. Tornow, "The DARPA Packet Radio Network Protocols", in *Proceedings of the IEEE (Special Issue on Packet Radio Networks)*, vol. 75, no. 1, January 1987, pp. 21–32.
- [10] N. Schacham and J. Westcott, "Future Directions in Packet Radio Architectures and Protocols", in *Proceedings of the IEEE (Special Issue on Packet Radio Networks)*, vol. 75, no. 1, January 1987, pp. 83–99.
- [11] J.L. da Silva Jr., J. Shamberger, M.J. Ammer, C. Guo, S. Li, R. Shah, T. Tuan, M. Sheets, J.M. Rabaey, B. Nikolic, A. Sangiovanni-Vincentelli, P. Wright, "Design Methodology for PicoRadio Networks," in *Proceedings of Design, Automation, and Test in Europe (DATE) Conference*, Munich, Germany, March 2001.
- [12] J. Gomez, A. T. Campbell, M. Naghshineh and C. Bisdikian, "Conserving Transmission Power in Wireless Ad Hoc Networks" in *Proceedings of IEEE 9th International Conference on Network Protocols (ICNP)*, Riverside, California, USA, November 2001.
- [13] G. Aggelou and R. Tafazolli, "On the Relaying Capacity of Next-Generation GSM Cellular Networks", *IEEE Personal Communications Magazine*, vol. 8, no. 1, February 2001, pp. 40–47.
- [14] R. Hermann, D. Husemann, M. Moser, M. Nidd, C. Rohner, and A. Schade, "DEAPSpace-Transient Ad-Hoc Networking of Pervasive Devices", in *Proceedings of the First Workshop on Mobile and Ad Hoc Networking and Computing (MobiHoc)*, Boston, USA, August 2000.

- [15] T. M. Siep, I. C. Gifford, R. C. Braley, and R. F. Heile, "Paving the Way for Personal Area Network Standards: An Overview of the IEEE P802.15 Working Group for Wireless Personal Area Networks", in *IEEE Personal Communications*, vol. 7, no. 1, February 2000, pp. 37 – 43.
- [16] C. Lochert, H. Hartenstein, J. Tian, H. Fussler, D. Hermann, and M. Mauve, "A Routing Strategy for Vehicular Ad Hoc Networks in City Environments", in *Proceedings of the IEEE Intelligent Vehicles Symposium*, Columbus, Ohio, USA, June 2003.
- [17] I. Gruber, G. Bandouch, and H. Li, "Ad Hoc Routing for Cellular Coverage Extension", in *Proceedings of the 57th IEEE Vehicular Technology Conference*, Jeju, Korea, April 2003.
- [18] J. Moy, "OSPF Version 2", *Internet RFC 2328*, April 1998.
- [19] B. Patil, Y. Saifullah, S. Faccin, S. Sreemanthula, L. Aravamudhan, and S. Sharma, "IP in Wireless Networks", Prentice Hall, February 2003.
- [20] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)", *Internet RFC 3626*, October 2003.
- [21] R. Ogier, F. Templin, and M. Lewis, "Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)", *Internet RFC 3684*, February 2003.

- [22] D. B. Johnson, D. A. Maltz, and Y.-C. Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)", *Internet Draft (work-in-progress)*, draft-ietf-manet-dsr-09.txt, April 2003.
- [23] C. E Perkins, E. M Royer, and S. R Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing", *Internet RFC 3561*, July 2003.
- [24] Z. Hass, M. R. Pearlman, and P. Samar, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks", *Internet-Draft (work in progress)*, draft-ietf-manet-zrp-04.txt, July 2002.
- [25] Z. Haas, M. Pearlman, and P. Samar, "The Interzone Routing Protocol (IERP) for Ad Hoc Networks", *Internet-Draft (work-in progress)*, draft-ietf-manet-zone-ierp-01.txt, July 2002.
- [26] Z. Haas, M. Pearlman, and P. Samar, "The Intrazone Routing Protocol (IARP) for Ad Hoc Networks", *Internet-Draft (work-in progress)*, draft-ietf-manet-zone-iarp-02.txt, July 2002.
- [27] Z. Haas, M. Pearlman, and P. Samar, "The Bordercast Resolution Protocol (BRP) for Ad Hoc Networks", *Internet-Draft (work-in progress)*, draft-ietf-manet-zone-brp-02.txt, July 2002.
- [28] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks", in *Proceedings of 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing Communications (DIALM)*, Dallas, USA, August 1999.

- [29] B. Karp and H. T. Kung, "Greedy Perimeter Stateless Routing for Wireless Networks, in *Proceedings of the Sixth ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom)*, Boston, MA, August 2000.
- [30] R. Jain, A. Puri, and R. Sengupta, "Geographical Routing Using Partial Information for Wireless Ad Hoc Networks", in *Proceedings of IEEE INFOCOM*, Anchorage, Alaska, USA, April 2001.
- [31] L. Blazevic, L. Buttyan, S. Capkun, S. Giordano, J. Hubaux, and J. Le Boudec, "Self-Organization in Mobile Ad-hoc Networks: The Approach of Terminodes", in *IEEE Communications Magazine*, June 2001.
- [32] G. Dommety and R. Jain, "Potential Networking Applications of Global Positioning Systems (GPS)", in *Technical Report TR-24*, Computer Science Department, The Ohio State University, April 1996.
- [33] J. Li, J. Jannotti, D. Decouto, D. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing", in *Proceedings of the Sixth ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom)*, Boston, USA, August 2000.
- [34] IETF MANET Working Group, <http://www.ietf.org/html.charters/manet-charter.html>
- [35] Andrew S. Tanenbaum, "Computer Networks", 4th Edition, Prentice Hall, August 2002.

- [36] C. Huitema, "Routing in the Internet", Prentice-Hall, 1999.
- [37] Y.-C. Tseng, S.-Y. Ni, E.-Y. Shih, "Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Network", in *IEEE Transactions on Computers*, vol. 52, no. 5, May 2003, pp. 545-557.
- [38] R. Castenada, S. R. Das, and M. K. Marina, "Query Localization Techniques for On-Demand Routing Protocols in Ad Hoc Networks", in *ACM/Kluwer Journal of Wireless Networks (WINET)*, vol.8, no.2, March 2002, pp 137-151.
- [39] S. Capkun, M. Hamdi and J. P. Hubaux, "GPS-Free Positioning in Mobile Ad-Hoc Networks", in *Journal of Cluster Computing*, vol. 5, no.2, April 2002, pp. 157-167.
- [40] A. Harter and A. Hopper, "A New Location Technique for the Active Office", in *IEEE Personal Communications*, October 1997, pp. 42-47.
- [41] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System", in *Proceedings of the Sixth ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom)*, Boston, USA, August 2000.
- [42] I. Stojemnovic, "Location Updates for Efficient Routing in Ad Hoc Wireless Networks", in *Handbook of Wireless Networks and Mobile Computing*, John Wiley and Sons, January 2002, pp. 451-471.

- [43] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols", in *Proceedings of the Fourth ACM International Conference on Mobile Computing and Networking*, Dallas, Texas, USA, October 1998.
- [44] S. R. Das, C. E. Perkins and E. M. Royer, "Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks", in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, Tel Aviv, Israel, March 2000.
- [45] Rice Monarch Project Extensions to ns-2, <http://www.monarch.cs.rice.edu/cmuns.html>
- [46] Theodore. S. Rappaport, "Wireless Communications: Principles and Practice", Prentice Hall, July 1999.
- [47] A. Kamerman and L. Monteban, "WaveLAN-II: A High-Performance Wireless LAN for the Unlicensed Band", in *Bell Labs Technical Journal*, vol. 2, no. 2, Summer 1997 pp. 118-133.
- [48] IEEE 802.11 WG, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", ANSI/IEEE Std 802.11, 1999.
- [49] W. Stallings, "Wireless Communications and Networks", Prentice Hall, August 2001.

- [50] P. Karn, "MACA - A New Channel Access Method for Packet Radio", in *Proceedings of the 9th ARRL/CRRL Amateur Radio Computer Networking Conference*, September 1990.
- [51] Bharghavan V. et al. "MACAW: A Media Access Protocol for Wireless LANs", in *Proceedings of ACM SIGCOMM*, London, UK, August 1994.
- [52] D. B. Johnson and D. A. Maltz. "Dynamic Source Routing in Ad Hoc Wireless Networks", in *Mobile Computing*, edited by Tomasz Imielinski and Hank Korth, Chapter 5, pages 153-181, Kluwer Academic Publishers, 1996.
- [53] M. Mauve, J. Widmer and H. Hartenstein, "A Survey on Position-based Routing in Mobile Ad Hoc Networks", in *IEEE Network Magazine*, November 2001, pp. 30-39.
- [54] C. Perkins and P. Bhagwat, "Highly Dynamic Destination Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", in *Computer Communications Review (CCR)*, October 1994, pp. 234-244.
- [55] D. A. Maltz, J. Broch, J. Jetcheva, and D. B. Johnson, "The Effects of On-Demand Behavior in Routing Protocols for Multi-Hop Wireless Ad Hoc Networks", in *IEEE Journal on Selected Areas in Communications*, August 1999, vol. 7, no.8, pp. 1439-1453.
- [56] S.-J. Lee and M. Gerla, "Dynamic Load-Aware Routing in Ad Hoc Networks", in *Proceedings of IEEE International Conference on Communications (ICC)*, Helsinki, Finland, June 2001.

- [57] Käsemann, H. Füßler, and M. Mauve, “A Reactive Location Service for Mobile Ad-Hoc Networks”, in *Technical Report TR-14-2002*, Department of Computer Science, University of Mannheim, Germany, November 2002.
- [58] A. Klemm, C. Lindemann, and O. Waldhorst, “A Special-Purpose Peer-to-Peer File Sharing System for Mobile Ad Hoc Networks”, in *Proceedings of IEEE Vehicular Technology Conference*, Orlando, USA, October 2003.
- [59] C. Gui and P. Mohapatra, “SHORT: Self-Healing and Optimizing Routing Techniques for Mobile Ad Hoc Networks”, in *Proceedings of ACM Symposium on Mobile Ad Hoc Networking and Computing*, Maryland, USA, June 2003.
- [60] M. Abolhasan and T. Wysocki,, “GPS-based Route Discovery Algorithms for On-Demand Routing Protocols in MANETs”, in *Proceedings of the first Annual Conference on Wireless On-Demand Network Systems*, Madonna di Campiglio, Italy, January 2004.
- [61] T. H. Clausen, P. Jacquet, L. Viennot, and G. Hansen, “Optimizing Route Discovery in Reactive Protocols for Ad Hoc Networks”, *Technical Report RR-4509, INRIA*, July 2002.
- [62] H. Dubois-Ferriere, M. Grossglauser and M. Vetterli, “Age Matters: Efficient Route Discovery in Mobile Ad Hoc Networks Using Encounter Ages”, in *Proceedings of the fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Maryland, USA, June 2003.

- [63] X. T. Hoang, S. Ahn, N. Kim and Y. Lee, “Efficient Route Discovery for Reactive Routing Protocols with Lazy Topology Exchange and Condition Bearing Route Discovery”, in *Proceedings of the International Conference on Information Networking (ICOIN)*, Busan, Korea, February 2004.

## Appendix A

### List of Publications

#### A.1 For This Thesis

##### A.1.1 Book Chapter

- B.-C. Seet, B.-S. Lee and C.-T. Lau, "Route Discovery Optimization Techniques in Ad Hoc Networks", Invited book chapter, in *Handbook of Ad Hoc Wireless Networks*, edited by M. Ilyas, CRC Press, Boca Raton, Florida, December 2002.

##### A.1.2 Journals

- B.-C. Seet, B.-S. Lee and C. T.-Lau, "Route Discovery Optimization for Dynamic Source Routing in Mobile Ad Hoc Networks", in *IEE Electronics Letters*, vol. 36, no. 23, November 2000, pp. 1963-1964.
- B.-C. Seet, B.-S. Lee and C.-T. Lau, "On Optimizing Route Discovery of Topology-based On-Demand Routing Protocols for Ad Hoc Networks", in *IEEE ComSoc/KICS Journal of Communications and Networks*, vol. 5, no. 3, September 2003, pp. 266-274.
- B.-C. Seet, B.-S. Lee and C.-T. Lau, "Optimisation of Route Discovery for Dynamic Source Routing in Mobile Ad Hoc Networks", in *IEE Electronics Letters*, vol. 39, no. 22, October 2003, pp. 1606-1607.

- B.-C. Seet, B.-S. Lee and C.-T. Lau, “Suppression of Non-Optimal Replies in Mobile Ad Hoc Networks”, submitted.

### **A.1.3 Conference**

- B.-C. Seet, B.-S. Lee and C.-T. Lau, “Study of a Unicast Query Mechanism for Dynamic Source Routing in Mobile Ad Hoc Networks”, in *Lecture Notes on Computer Science: Proceedings of the First International Conference on Networking*, France, Springer-Verlag, vol. 2094, July 2001, pp. 168-176.

## **A.2 For Other Related Research**

### **A.2.1 Journals**

- C.-H Foh, G. Liu, B.-S Lee, B.-C Seet, and K.-J. Wong, “Network Connectivity of One-Dimensional MANETs with Random Waypoint Movement”, to appear in *IEEE Communications Letters*.
- B.-S. Lee, G. Liu and B.-C. Seet, “Anchor-based Street and Aware Routing for Metropolis Vehicular Ad Hoc Networks”, submitted.

### **A.2.2 Conferences**

- K.-J. Wong, B.-S. Lee, B.-C. Seet, G. Liu and L. Zhu, “BUSNet: Model and Usage of Regular Traffic Patterns in Mobile Ad Hoc Networks for Inter-Vehicular

Communications”, in *Proceedings of the Second International Conference on Information and Communication Technologies*, Thailand, April 2003.

- B.-S. Lee, K.-J. Wong, B.-C. Seet, L. Zhu and G. Liu, “Performance of Mobile Ad Hoc Network in Constrained Mobility Pattern”, in *Proceedings of International Conference on Wireless Networks*, Las Vegas, Nevada, USA, June 2003.
- G. Liu, K.-J. Wong, B.-C. Seet, B.-S. Lee and L. Zhu, “PATCH: A Novel Local Recovery Mechanism for Ad Hoc Networks”, in *Proceedings of the 58th IEEE Vehicular Technology Conference*, Orlando, Florida, USA, October 2003.
- L. Zhu, B.-S. Lee, B.-C. Seet, K.-J. Wong, G. Liu, S.-Y. Huang and K.-K. Lee, “Performance of New Broadcast Forwarding Criteria in MANET”, in *Proceedings of the International Conference on Information Networking*, Busan, Korea, February 2004.
- B.-C. Seet, G. Liu, B.-S. Lee, C.-H. Foh, and K.-J. Wong, “A-STAR: A Mobile Ad Hoc Routing Strategy for Metropolis Vehicular Communications”, in *Lecture Notes on Computer Science: Proceedings of the Third IFIP-TC6 Networking Conference*, Athens, Greece, Springer-Verlag, vol. 3042, May 2004, pp. 989-999, (acceptance rate < 20%).

- B.-C. Seet, B.-S. Lee and C.-T. Lau, “An Epidemiology-Inspired Approach for Modeling Broadcast Route Discovery in Wireless Ad Hoc Networks”, in *Proceedings of the IEEE Conference on Networks*, Singapore, November 2004.

### **A.2.3 Technical Report**

- B.-C. Seet and T.-H. Cheng, “Mobile Ad Hoc Communication Infrastructure for Military Applications”, in *DSTA (Defense Science & Technology Agency) Report*, Contract P0D0001523, Network Technology Research Centre (NTRC), Nanyang Technological University, Singapore, March 2001.

## Appendix B

### Acronyms

ACK	: Acknowledgement
AODV	: Ad-hoc On-demand Distance Vector
CA	: Collision Avoidance
CBR	: Constant Bit Rate
CSMA	: Carrier Sense Multiple Access
CTS	: Clear To Send
CPU	: Central Processing Unit
CREP	: Cached Route Reply
DCF	: Distributed Coordinated Function
DSR	: Dynamic Source Routing
GPS	: Global Positioning Systems
IETF	: Internet Engineering Task Force
IFQ	: Interface Queue
LAN	: Local Area Network
LAR	: Location Aided Routing
LS	: Local Search
MAC	: Medium Access Control
MANET	: Mobile Ad Hoc Network
P2P	: Peer-to-Peer
RERR	: Route Error

RREP : Route Reply  
RREQ : Route Request  
RTS : Request To Send  
TREP : Target Route Reply  
TTL : Time-To-Live  
WPAN : Wireless Personal Area Networks