LOCATION-AIDED ROUTING PROTOCOL IN HYBRID WIRED-WIRELESS NETWORKS

WU MINTAO
(B.Eng.(Hons.), NTU)

A THESIS SUBMITTED
FOR THE DEGREE OF MASTER OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE
2005

ACKNOWLEDGEMENTS

Many thanks are given to my supervisors, Prof Lawrence W.C. Wong and Dr. Winston K.G. Seah, for their guidance along the way, especially the discussions, critiques and advice during the course of paper and thesis writing.

Many thanks are due to lots of friends around ... Special thanks to Li Feng, Ricky C.H. Foo, Cho Chia Yuan , Tan Yick Fung, Dr Li Tong Hong for their friendship, for the hardship and fun we had working together, for making the past two years a memorable experience!

The dissertation is dedicated to my parents, my brother and my wife. It is their love, support and encouragement that made everything I have possible!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS OR ABBREVIATIONS

AODV	Ad-hoc On Demand Distance Vector Routing Protocol.
AP	Access Point, same meaning here as Base Station (BS) and Gateway (GW).
BS	Base Station, same meaning here as Access Point (AP) and Gateway (GW).
CBR	Constant Bit Rate.
CGSR	Clusterhead Gateway Switch Routing Protocol.
DSDV	Dynamic Destination-Sequenced Distance-Vector Routing Protocol.
DSR	Dynamic Source Routing Protocol.
FH	Fixed Host.
GPS	Global Positioning System.
GSM	Global System for Mobile Communications.
GW	Gateway, same meaning here as Access Point (AP) and Base Station (BS).
GWACK	Gateway Acknowledgement message.
GWAD	Gateway Advertisement message.
HSR	Hierarchical State Routing Protocol.
IEEE	Institute of Electrical and Electronics Engineers.
LANMAR	Landmark Ad Hoc Routing Protocol.
LAOD	Location-Aided On-Demand routing Protocol.
LAR	Location-Aided Routing Protocol.
LET	Link Expiration Time.
LLR	Link-Connectivity-Prediction-Based Location-Aided Routing Protocol.
MH	Mobile Host, same meaning here as Mobile Node (MN).
MN	Mobile Node, same meaning here as Mobile Host (MH).
OLSR	Optimized Link State Routing Protocol.

PDR	Packet Delivery Ratio.
RERR	Route Error message.
RET	Route Expiration Time.
RREP	Route Reply message.
RREQ	Route Request message.
SRREQ	Specific Route Request Message.
TORA	Temporally-Ordered Routing Algorithm.
TTL	Time-To-Live.
WR	Wireless Routing path.
WWR	Wireless-cum-Wired Routing path.
ZRP	Zone Routing Protocol.

SUMMARY

We propose a hybrid wired-wireless network that comprises a wireless ad hoc network combined with the fixed wired network with the latter forming a high-speed interconnected backbone. This hybrid network has a lot of potential economic applications. Routing is critical to achieve good performance in such a hybrid network environment. Previous research has not taken advantage of other research works on the routing protocols using location information. Here, we propose two different location-aided routing protocols, namely the Location-Aided On-Demand (LAOD) routing protocol, and the Link-Connectivity-Prediction-Based Location-Aided Routing (LLR) protocol, both of which make use of location information but in different ways. We also propose a gateway discovery algorithm to build the *K-hop* subnets around the gateways (GWs), which is fundamental to our proposed routing protocols. Simulation results using Network Simulator (NS2) show that our proposed routing protocols achieve better routing performance than the topology-based routing protocols, particularly Ad-hoc On Demand Distance Vector Routing (AODV).

Furthermore, a Hello message adjustment algorithm incorporated with LLR is also proposed. By dynamically adjusting the Hello message broadcasting interval with respect to the node mobility, the routing performance improves and power consumption is reduced. The simulation results demonstrate the routing performance improvement in terms of the packet delivery ratio (PDR), the end-to-end delay and as well as the overhead in the network.

CHAPTER I

INTRODUCTION

1.1 Hybrid Network

A hybrid wired-wireless network is defined to be a heterogenous hierarchical network that contains both mobile hosts (MHs) and access points (APs). MHs, or mobile nodes (MNs) can communicate with other MNs, which can be multiple hops away. APs, or gateways (GWs), or base stations (BSs), are nodes with both wireless and wired interface, e.g. Internet connectivity. GWs give MNs access to other MNs or fixed hosts (FHs) of the wired network. An example of such a network system is shown in Figure 1.1. In Figure 1.1, MN1 can reach MN4 in ad hoc mode, while MN1 can also reach MN7 despite the fact that they cannot communicate in ad hoc mode.

The hybrid network, as described above, can be considered as a wireless mobile ad hoc network (MANET) [1, 2] incorporated with wired backbone network connectivity. Thus it has dynamic network topology due to the fact that MNs change their physical locations by moving around, although the GWs are at fixed locations. By incorporating MANET with a wired network, typically Internet, the "range" of an GW can be extended to multiple hops away to allow for greater connectivity and provide connectivity outside the ad hoc network. For example, in Figure 1.1, the service of GW1 can be extended to MN3 and MN4 as opposed to just MN1 and MN2. Furthermore, when a MANET is incorporated, not all communication between MNs has to go through the GW, since the incorporated ad hoc network allows MNs to communicate directly without going through the GWs. This may ease the burden placed on the GWs, as economical consideration to have only a few GWs with a large number of MNs in such a hybrid network can be achieved.

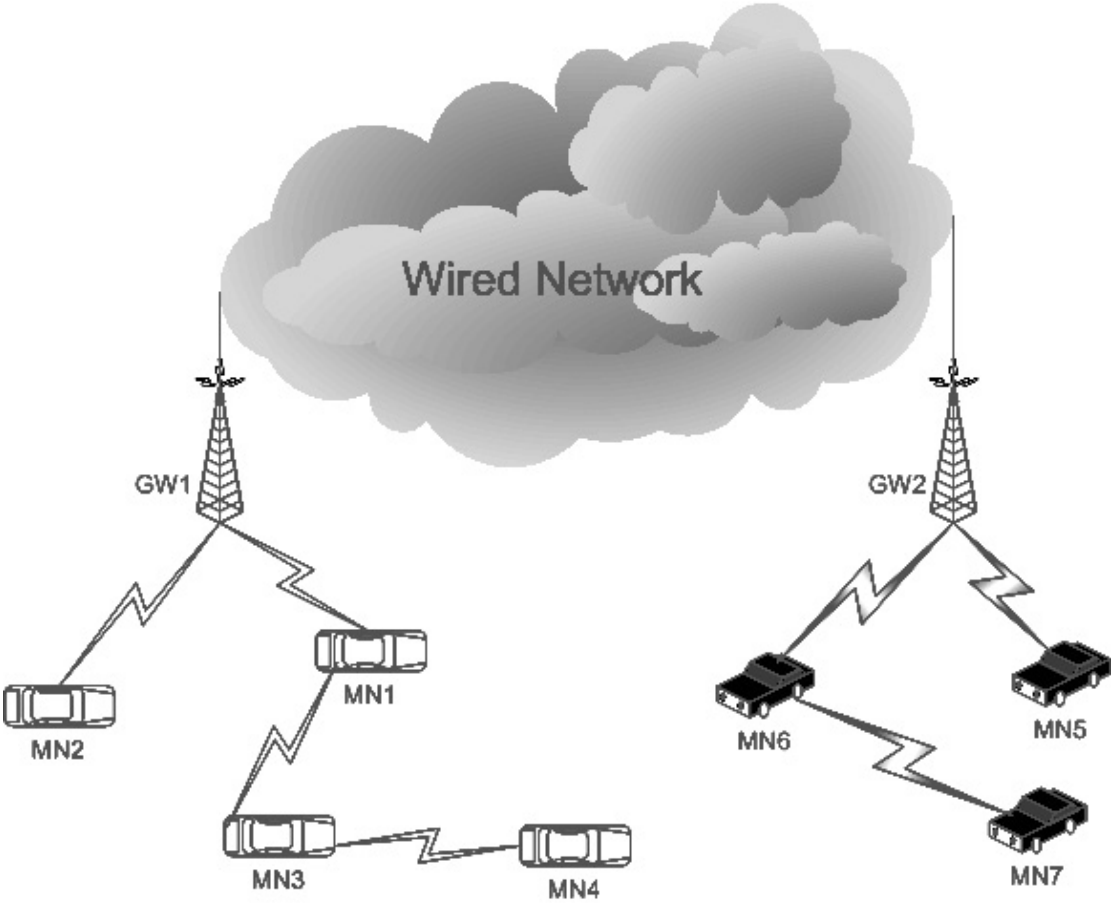


Figure 1.1: An example of hybrid wired-wireless network

Such a hybrid network has a lot of potential commercial applications. One possible useful application is an inter-vehicle hybrid network [3, 4, 5]. Vehicles in the network form an ad hoc network in order to share information between them. At the same time, passengers in vehicles can access the Internet through the connections between nearby vehicles and GWs, which are pre-placed and deployed along the roads. For example, you can communicate with other people in vehicles near to yours by chatting or playing interactive games, while at the same time you can check your email through the Internet.

1.2 Motivation

Routing in such a hybrid network is a challenging task since the network topology changes frequently due to the movements of the MNs. The communication in this hybrid network environment can be categorized into two scenarios: (1) routing between a fixed host (FH) within a wired network and an ad hoc MN and (2) routing between two peer-to-peer ad hoc MNs under the same GW or different GWs. The first scenario is also referred to as Internet connectivity. Several methods for achieving Internet connectivity have been proposed [6, 7]. In this thesis, a simple but efficient gateway discovery algorithm is presented to provide and maintain connectivity between the MNs and the GWs. However, since the focus here is on the peer-to-peer communication between MNs, which is the second scenario stated above, communication between FHs and MNs are not studied.

The existence of GWs makes the routing between two ad hoc MNs complicated. The routing path between two ad hoc MNs can be categorized into two types: Wireless Routing path (WR) and Wireless-cum-Wired Routing path (WWR). As shown in Figure 1.1, a WR path is a wireless multi-hop path directly from source to destination within an ad hoc network (e.g. MN1-MN3-MN4), while a WWR path is a wireless multi-hop path from source to destination via GWs (e.g. MN1-GW1-GW2-MN6-MN7).

Research effort has been carried out on such hybrid networks [8, 9] and most use traditional reactive routing protocols like Ad-hoc On Demand Distance Vector Routing (AODV) [10] for multi-hop peer-to-peer communication between MNs. However, those research works do not take advantage of the research that has been done in routing protocols [11, 12, 13] which make use of location information. Motivated by these research works on pure ad hoc network environments, it is worth studying routing performance for multi-hop peer-to-peer communications between MNs complemented with the additional location information in this hybrid network environment. One simple way to do the routing in this hybrid network is to use the GWs as the default route. This means that all communications between MNs has to go through the GWs. But routing this way may increase the burden placed on the GWs. Therefore, one of our concerns in the routing protocol design is to minimize the use of resources such as the GWs and wired network.

1.3 Assumptions

In this thesis, we assume that all MNs know their own location through Global Positioning System (GPS) devices [14, 15], or other means. The location here is represented in 2D Cartesian coordinate plane for simplicity. We further assume there is an appropriate working MAC layer under the designed routing protocol. The widely used IEEE 802.11 wireless network MAC [16] is adopted. The wired backbone network, where the GWs are interconnected with one another, is assumed to have a flat architecture. We consider the wired backbone network as one big virtual node. Any data packet going into one GW should seamlessly traverse through the wired network and arrive at a destination GW. Finally, we assume each MN or GW has a unique address. The addressing issue in such a hybrid wired-wireless network is already addressed in many research works [6, 7]. We believe that these works can be incorporated into our works in the future.

1.4 Contributions

The main contributions of this thesis are:

- A simple but efficient gateway discovery algorithm is presented. This gateway discovery algorithm works in conjunction with the routing protocol to provide local connectivity between GWs and their serving MNs.

- Location-Aided On-Demand (LAOD) routing protocol [17] is presented and simulation results show that this approach improve routing performance, in terms of packet delivery. However, LAOD has longer end-to-end delay and larger overhead. These pitfalls of LAOD make us move on to design another routing protocol which has better routing performance. The result is the proposed Link-Connectivity-Prediction-Based Location-Aided Routing (LLR) protocol.

- Link-Connectivity-Prediction-Based Location-Aided Routing (LLR) protocol [18] is presented and simulation results demonstrate that this approach improves routing performance in terms of packet delivery, end-to-end delay and overhead. Furthermore, a Hello message adjustment algorithm is presented, which is incorporated with the LLR protocol to further improve routing performance and reduce power consumption.

1.5 Organization

The remainder of the thesis is organized as follows. Chapter 2 reviews relevant background information and related works. Chapter 3 presents the gateway discovery algorithm, which works as a fundamental element for the proposed routing protocols. Chapter 4 presents a simple location-aware routing protocol, LAOD, which is an on-demand routing protocol incorporated with greedy packet forwarding scheme. Chapter 5 presents another location-aware routing protocol, LLR, which is specially designed for the hybrid network environment. Chapter 6 presents simulation results and performance evaluations. Finally Chapter 7 delivers some concluding remarks and directions for future work.

CHAPTER II

BACKGROUND AND RELATED WORKS

Research efforts have been carried out on such hybrid networks [3, 4, 5, 8, 9, 19, 20] and most use the reactive routing protocols like Ad-hoc On Demand Distance Vector Routing (AODV) [10] for multi-hop peer-to-peer communication between MNs. However, those research works do not take advantage of findings in routing protocols [12, 13, 21, 22, 23, 24] which make use of location information. Motivated by these research works on pure ad-hoc network environments, it is worth studying routing performance for multi-hop communications between MNs complemented with the additional location information in this hybrid network environment.

In this chapter, first, the hybrid network environment is described in detail. Next, a general overview of the routing protocols in wireless ad-hoc network is presented. After that, the link connectivity prediction algorithm is introduced, which is used to calculate the Link Expiration Time (LET) between two neighbors by using the location information.

2.1 Hybrid Wired-wireless Network Environment

With the advances in the wireless communication and the mobile computing technology, the wireless multi-hop network is expected to play an important role in modern personal ubiquitous communication system. The wireless multi-hop network, also known as ad hoc network or wireless mobile ad hoc network (MANET) [1, 2], enables the spontaneous establishment of communications between personal mobile communication systems (e.g. mobile phones, personal digital assistants, personal laptops), independent of pre-existing network infrastructure. Compared to the "conventional" wireless cellular systems, such as Global System for Mobile Communications (GSM) [25], the ad hoc network offers simple management and deployment, especially in applications where information must be distributed quickly and is only relevant in the area around the sender.

However, for many applications, it is desired that a self-organizing ad hoc network is somehow connected to a wired backbone network. For example, in a vehicular environment, there are some *info stations* [26], which are pre-placed along the roads and at the city entrances, to inform vehicle drivers and passengers, in a drive-by fashion, about nearby restaurants, the current traffic situation, cultural events, etc. A wired backbone network is formed among these *info stations* to share, maintain and update information on them. With ad hoc networking capabilities, vehicles in the transmission range of these *info stations* could then forward the information in a multi-hop fashion to other vehicles that have no direct wireless link to the *info stations*. Another example, vehicle drivers and passengers may want to access the Internet through the access points deployed along the roads. However, their vehicles may not be in the direct wireless transmission range of those access points. Thus, their communications with the access point need to go through multiple hops with other vehicles serving as intermediate nodes. With ad hoc networking capabilities, the vehicle drivers and passengers are thus able to get connected to the Internet. Therefore, the hybrid wired-wireless network is required for applications where it is necessary to provide connectivity both inside and outside the ad hoc network.

The hybrid wired-wireless network [8, 7, 27, 28, 29, 30] is a heterogenous hierarchical network for general purpose wide-area communication. There are two types of nodes in the hybrid wired-wireless network: gateways (GWs) and mobile nodes (MNs). GWs are nodes pre-placed throughout the network area. The GWs form the interface between the high-speed wired backbone and the wireless ad hoc network of the hybrid wired-wireless network. They improve ad hoc network routing scalability and provide the wired-network connectivity. MNs are nodes which can be moving freely. Each GW serves the MNs within a topological subnet around the GW. The coverage of the GW is determined not by the wireless transmission range of the GW but by a distance in wireless hops from it. Therefore, MNs are able to access the FHs in the wired network through the GWs even if they are multiple hops away from the GWs. At the same time, MNs can communicate with other MNs, which can be multiple hops away, through the ad hoc mode.

The benefits of such a hybrid wired-wireless network are numerous. The use of ad hoc network routing contributes to the robustness and adaptiveness of the system relative to the traditional wireless network, like GSM, because the ad hoc routing protocol is able to adapt to changes in the network topology and MN failures, as well as route around congested areas of the network. In addition, compared to the traditional wireless network, a hybrid network will have smaller number of GWs and due to the multi-hop routing capability of the ad hoc network, placement of the fixed GWs is significantly simplified over traditional architectures such as the cellular system. The exact placement, which requires topographical surveys, is not necessary. Furthermore, a lot of potential commercial applications can make use of such a hybrid wired-wireless network architecture. For example, the inter-vehicle hybrid network [3, 4, 5, 31] looks very promising to be the next "big thing" in communication networks. One typical usage of such an inter-vehicle hybrid network is in driver assistance: in case of accidents on the road, the vehicles that are involved in the accidents can send a notification message to the neighboring vehicles. Therefore, information of such accidents can be conveyed to other vehicles that might run into the accident.

Different kinds of wired backbone networks are proposed to be inter-connected with ad hoc wireless networks, in particular, the mobile cellular network and the Internet.

There have been several proposals [32, 33, 34] for a hybrid cellular and ad hoc networking infrastructure in which MNs within a cell use ad hoc network routing to reach the GWs, which are responsible for the cell. These proposals focus on the design and performance of the hybrid network within a single GW. However, they do not discuss the routing mechanisms for roaming between different cells. For example, Hsieh et al. [32] proposed a system for enhancing a cellular network with the wireless ad hoc network, in which MNs use ad hoc routing to reach the GWs along multiple hops and switch to cellular operation when the bandwidth available in the ad hoc mode is lower than that achievable in the cellular mode. In the proposal, the traditional cellular protocols are used instead of ad hoc routing protocols.

A number of approaches [8, 9, 30] have been proposed for connecting a wireless ad hoc network to the Internet. For example, Jetcheva et al. [9] described a hybrid network architecture connecting an ad hoc network running an extension of Dynamic Source Routing protocol (DSR). Their approach allows for roaming of MNs between different ad hoc network clouds and the Internet, and uses sub-netting to distinguish between MNs in different ad hoc network clouds. Their approach also emphasizes on on-demand routing within the ad hoc network. However, their approach does not make use of location information, which is obtainable through the GPS system. In this thesis, on the other hand, we assume that the MNs are equipped with GPS systems and location information is thus available.

2.2 Routing in Wireless Ad-hoc Network Environment

For multi-hop peer-to-peer wireless ad-hoc communication, there are already plenty of works on routing protocol design [10, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45]. They can be categorized into two approaches [35]: *topology-based routing* protocols and *position-based routing* protocols.

2.2.1 Topology-based Routing Protocols

Topology-based routing protocols use only the information about the network topology to perform packet forwarding. They can be further divided into *proactive routing*, *reactive routing*, and *hybrid routing* protocols.

- *Proactive routing* protocols, such as OLSR [36], DSDV [37], CGSR [38] and so on, normally employ classical routing strategies such as distance-vector algorithm or link-state algorithm. Nodes in the network maintain routing information about all the available paths in the network even if these paths are not currently used. Therefore, these protocols require each node in the network to maintain one or more routing-related tables and consistent, up-to-date routing information need to be available in the network. It is obvious that they are not suitable in networks with a large number of nodes, because the overhead will occupy more and more bandwidth as the number of participating nodes increases. It may reach such a point that the network is flooded with only the control packets with no real communication taking place.

- *Reactive routing* protocols, such as AODV [10], DSR [39], TORA [40], and so on, are source-initiated on-demand routing protocols. They do not require nodes to maintain routing information, at least not for long intervals. They create routes only when requested by the source node. The routes are first discovered, and then maintained if necessary. Therefore, the routing process is normally divided into two phases, *route discovery* and *route maintenance*. Although reactive protocols perform better in some aspects than proactive routing protocols, they still have some limitations. First, due to the on-demand characteristic, the route to the destination is searched before data communication starts. This leads to a delay for the first packet to be transmitted by the source. Second, in the *route discovery* process, it normally uses flooding to find the route to the destination. This may cause huge network traffic if the destination is far away from the source. Finally, although only the currently used route is tracked by the *route maintenance* process, it still generates significant amount of communication overhead if the network topology changes frequently.

As mentioned above, many reactive routing protocols have been proposed in the literature. Here, we are going to describe one of the protocols that have been standardized by the IETF, the Ad-hoc On Demand Distance Vector (AODV), which is used as the reference in our comparative simulation study.

AODV The AODV [10] routing protocol establishes routes only when the routing path is required. A source node wishing to communicate with a destination node initializes a route discovery process by broadcasting a Route Request (RREQ) message. The RREQ sets up a temporary reverse path to the source node. This temporary reverse path is used later. Only the destination node or an intermediate node with an up-to-date route to the destination can generate a Route Reply (RREP) message, which is sent back to the source node along the temporary reverse path. As the RREP travels along the reverse path, it sets up the forwarding path to the destination node. Upon receiving the RREP, the source node can begin sending data using the forwarding path set up by the RREP message. To avoid processing old control messages, each broadcasting message is uniquely identified by a <source, broadcast_id> tuple. Furthermore, destination sequence numbers are also used to determine the freshness of routes.

AODV provides good connectivity within the wireless network while reducing the overhead cost when the network is idle. It requires the MNs to store only the routes that are needed, and is scalable to large populations of MNs. Furthermore, the loop-free routes are achieved by use of the destination sequence numbers.

- *Hybrid routing* protocols, such as ZRP [41], LANMAR [42], HSR [43] and so on, try to achieve better performance by combining both the proactive and reactive routing protocols. These hybrid protocols may use locally proactive routing and globally reactive routing. Although research results shows that hybrid protocols perform better than any single proactive or reactive routing protocol mentioned above, the complexity of hybrid protocols is the main limitation. The cost of increasing complexity in this kind of protocol makes it doubtful to employ when the complexity outweigh the slight performance gain. Furthermore, position-based routing protocols may outperform hybrid routing protocols.

2.2.2 Position-based Routing Protocols

Position-based routing protocols [13, 21, 22, 23, 46, 47, 48, 49] make use of location information to forward data packets. They require information about the location of the participating nodes to be available. Location information is obtained via a location service. Location service provides a source node with the current location information of the destination node. More details on location service can be found in [11, 50, 51, 52, 53]. In these position-based routing protocols, each node maintains a location table that records the location of all other nodes and the time at which that location information is received. The source node then uses this information to improve efficiency in the transmission of packets. A review of some of these protocols is available in [11]. Most research results on the position-based routing protocols show that usage of location information significantly improves routing performance.

As mentioned above, many position-based routing protocols have been proposed in the literature. Here, we only describe two of the most well-known schemes, namely, the *greedy packet forwarding mechanism* and the *location-aided routing protocol* (LAR). These two schemes are very closely related to our routing protocol design in the later sections.

2.2.2.1 Greedy Packet Forwarding Mechanism

In the Greedy packet forwarding [11] mechanism, the source node will firstly choose a local optimal next-hop node based on the knowledge of the location information of the destination node and neighbor nodes. The selected next-hop node is normally the node which lies closer to the destination node than the source node. Then, the data packet will be forwarded to the desired intermediate node with the destination location information included. The receiving node repeats the next-hop selection, till the destination node is reached. One example of greedy packet forwarding is shown in Figure 2.1, where the source node is MN_S, while destination node is MN_D. MN_S has three neighbors, MN1, MN2, and MN3. As can be seen in Figure 2.1, MN3 is the node, which is closest

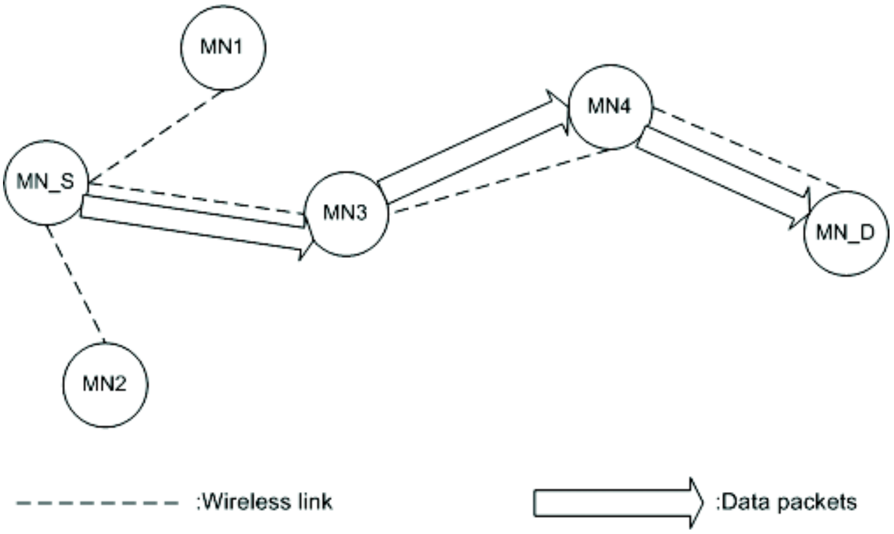


Figure 2.1: An example of greedy packet forwarding in wireless ad hoc network

to MN_D in terms of geographical distance. Therefore, MN_S will select MN3 as the next-hop to MN_D and forward data packets to MN3. Then, MN3 repeats the selection procedure and forwards data packets to MN4. This process continues till MN_D is reached.

But this routing protocol suffers one big problem, the so-called *Local Maximum* problem [11], especially in a sparse network. In Figure 2.2, MN_S has a transmission range r, which has center at MN_S, as shown by the dashed circle. Node MN_D is distance R away from Node MN_S. As can be observed from Figure 2.2, there is a valid routing path (MN_S-MN1-MN2-MN3-MN4-MN_D). The problem occurs because MN_S is closer to MN_D than any of its neighbor nodes. Therefore, by only using the forwarding technique stated above, greedy packet forwarding fails, because no other neighbor, except itself, is closer to the destination. In this case, it has reached the *Local Maximum*.

Although the greedy packet forwarding mechanism has the *Local Maximum* problem, it is still a simple but efficient forwarding technique, especially in a dense network. It only requires the location information of the destination node and the location information of the forwarding node's neighbors to deliver data packets to the destination.

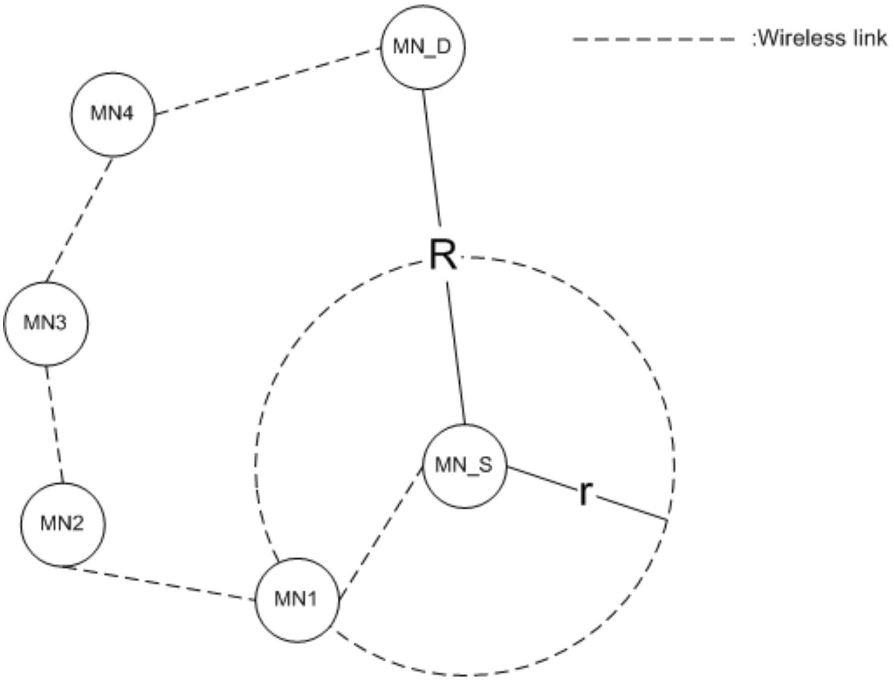


Figure 2.2: An example of *Local Maximum* problem

Therefore, with the location service present to provide frequently updated location information, the MNs neither have to store routing tables nor need to transmit control messages to keep the routing table up-to-date.

2.2.2.2 Location-Aided Routing Protocol (LAR)

LAR [13] is an on-demand routing protocol. It tries to search for a path from the source to the destination by flooding RREQ packets, similar to AODV [10]. But it uses the location information to restrict the flooding area of the RREQs. In LAR, before the route discovery phase, the source node defines a circular area, called *expected zone*, in which the destination may be located. The position and size of the circle is decided with the following information:

- The destination location known to source
- The time instant when the destination is located at that position
- The average moving speed of the destination

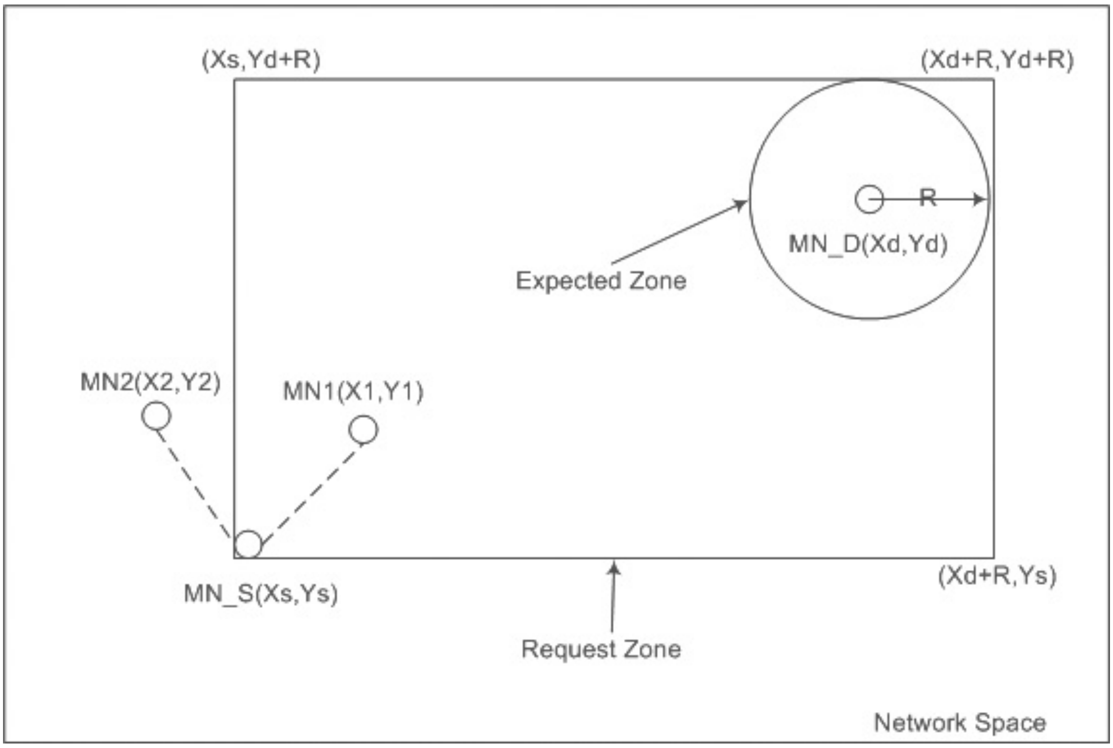


Figure 2.3: An example of LAR Scheme 1

Then the source node needs to define a *request zone*. Only the MNs inside such an area propagate the RREQ. Two ways of defining the *request zone* are proposed in [13]. In Scheme 1, the smallest rectangular area that includes the *expected zone* and the source is the *request zone*. This information is attached to the RREQ by the source and the RREQ is sent out. When an MN receives this packet, it checks whether it is inside the *request zone* and continues to relay the packet only if it is. Figure 2.3 shows an example. In this example, MN_S is the source node, and MN_D is the destination node. MN_S has two neighbors: MN1 and MN2. From Figure 2.3, it is obvious that MN1 is inside the *request zone*, while MN2 is outside the *request zone*. Therefore, MN1 will re-broadcast the RREQ from MN_S while MN2 will drop it instead.

In Scheme 2, the source node calculates the distance between the destination and itself. This distance, along with the destination location known to the source, is included in the RREQ and sent to the neighbors. When the MN receives this packet, it computes its distance to the destination, and continues to relay the packet only if its

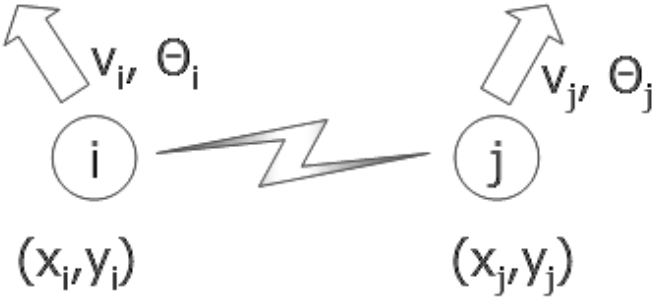


Figure 2.4: An example of Link Expiration Time (LET)

distance to the destination is less than or equal to the distance indicated by the packet. When forwarding the packet, the MN updates the distance field with its distance to the destination.

As can be observed, LAR is very simple to implement. It helps to reduce the overhead with the available location information. However, LAR uses the location information only to set up the routing path in an efficient way. The data packets are routed with a location-independent protocol. That means just like normal on-demand routing protocols, the MNs still have to store routing tables and need to transmit control messages to keep the routing tables up-to-date.

2.3 Link Connectivity Prediction Scheme

Su et al. [12] proposed to calculate the Link Expiration Time (LET) between two neighbors using location information. As shown in Figure 2.4, assume two nodes i and j are within the transmission range r of each other. Let (x_i, y_i) be the coordinate of node i and (x_j, y_j) be that of node j. Also let v_i and v_j be the speeds and θ_i and θ_j be the moving directions of nodes i and j, respectively. Then, the amount of time the two nodes will stay connected is predicted by:

$$LET = \frac{-(ab+cd) + \sqrt{(a^2+c^2)r^2 - (ad-bc)^2}}{(a^2+c^2)} \tag{2.1}$$

where:

$$a = v_i \cos\theta_i - v_j \cos\theta_j \tag{2.2}$$

$$b = x_i - x_j \tag{2.3}$$

$$c = v_i \sin\theta_i - v_j \sin\theta_j \tag{2.4}$$

$$d = y_i - y_j \tag{2.5}$$

This prediction scheme gives a quantitative estimated measurement of how long the two nodes will stay connected. The LET can then be applied to routing protocols as a metric for each link, and this metric can be utilized to anticipate when the routing path is going to break and action need be taken before it happens.

CHAPTER III

GATEWAY DISCOVERY ALGORITHM

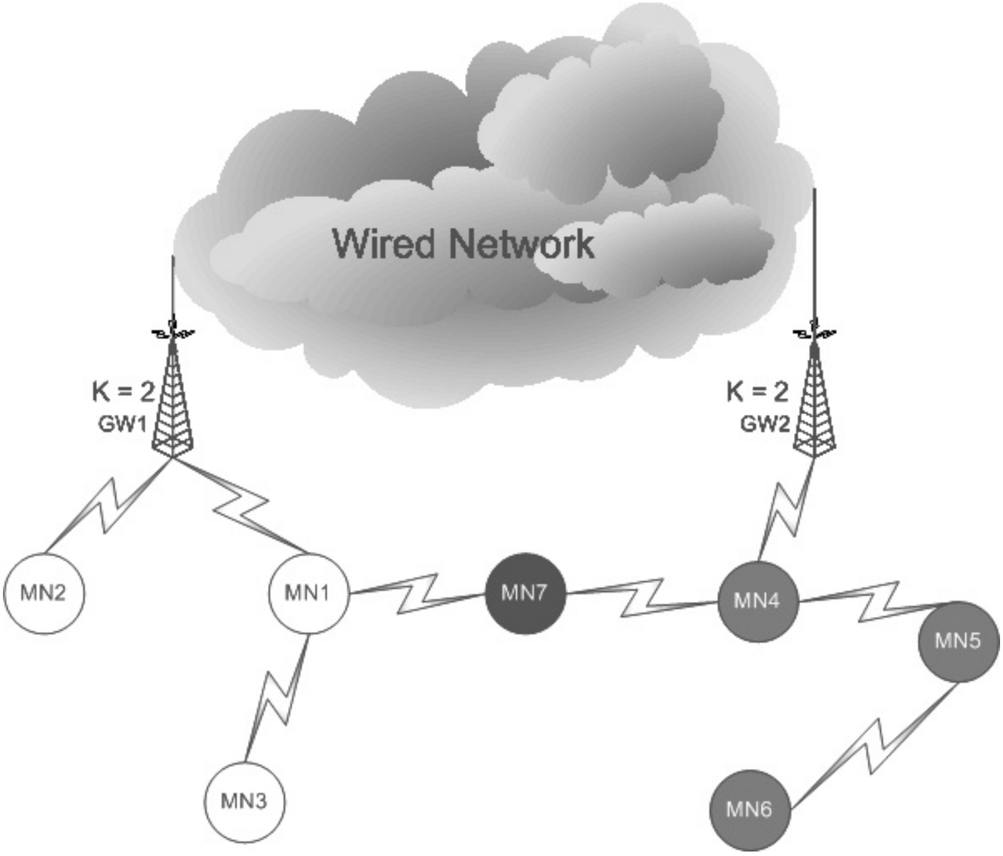


Figure 3.1: An example of *K-hop* subnet (*K*=2)

3.1 Introduction

In the hybrid network, as shown in Figure 3.1, the communication between the nodes is established through wireless multi-hop paths within an ad hoc subnet or across a wireless-wired-wireless hybrid network if the source MN and destination MN are located in different ad hoc subnets. The GWs provide the interface between the wireless ad hoc subnets to the wired backbone network. Therefore upon initialization, a MN should discover the existence of at least one GW within its reach. A few possible GWs could be selected by the MN to get connected to the Internet. However, we restrict our MNs in the proposed algorithm to select only one GW for the wired network connection purpose.

Our focus here is not about how the GWs in the wired backbone communicate with one another. Any data packet arriving at one GW is assumed to seamlessly traverse the wired backbone to an appropriate GW in order to reach the destination MN. Therefore, the GWs together with the wired backbone network are considered as one big virtual node. This assumption, as stated in Section 1.3 of Chapter 1, makes our research model less complicated.

In this chapter, we describe the *gateway discovery algorithm*, which is used to serve the purposes described above. First, we present the *K-hop* subnet concept.

3.2 K-hop Subnet

A *K-hop* subnet is a wireless subnet centred about a GW where MNs inside the subnet are at most *K* hops away from this particular GW. An example of *K-hop* subnets is shown in Figure 3.1. In Figure 3.1, with *K* equal to two, MN1, MN2, and MN3 form the *2-hop* subnet of GW1, while nodes MN4, MN5, and MN6 form the *2-hop* subnet of GW2. Note that in Figure 3.1, MN7 can be under *2-hop* subnet of GW1 , or GW2 , or both GW1 and GW2, since it is two hops away from both GW1 and GW2. The choice of selection depends on certain metrics (e.g. hop count, physical distance, load of GW, or combinations of these criteria). By using the *gateway discovery algorithm* described later, MN7 here can only choose one GW, either GW1 or GW2, to register with.

The formation of the *K-hop* subnet is essential in our routing protocol design. Inside the *K-hop* subnet, the GW proactively maintains the connectivity between itself and the MNs. In order to form such a *K-hop* subnet, a simple but efficient *gateway discovery algorithm* is described in the following subsection.

3.3 Gateway Discovery Algorithm

The *gateway discovery algorithm* is used to form the *K-hop* subnets around the GWs. After forming the *K-hop* subnets, connectivity between the GWs and the MNs is maintained. In other words, the local connectivity between each GW and its serving MNs is achieved. As shown in Table 3.1, each GW keeps track of an MN's address, MN's current location information, and the next-hop to this particular MN. At the same time, each MN in the hybrid network keeps track of the GW's address, geographical location information, the next-hop to this GW, and the number of hops away from this particular GW, as shown in Table 3.2. By using the *gateway discovery algorithm*, each MN should know how to reach its current registered GW, and the GW should know how to reach its serving MNs. Furthermore, the location information of MNs is collected and maintained at the GWs, which is then used later by the routing protocols, which will be described later in Chapter 4 and Chapter 5. The proposed *gateway discovery algorithm* consists of a gateway selection mechanism and a location update mechanism.

Table 3.1: Information kept by GW about its serving MNs

Information Field	Description
MN's address	The unique address of the MN
MN's location information	The location coordinate, speed and direction
Routing information to the MN	The next-hop to this particular MN

Table 3.2: Information kept by MN about its registered GW

Information Field	Description
GW's address	The unique address of the GW
GW's location information	The location coordinate
Routing information to the GW	The next-hop to this particular GW
Hop counts to the GW	The number of hops away from this particular GW

3.3.1 Gateway Selection Mechanism

Each GW periodically broadcasts Gateway Advertisement (GWAD) messages. The GWAD message contains the address of the originating GW, the location information of the originating GW, and the maximum number of hops it can propagate (equals to K). Thus the GWAD message is only propagated up to K hops away from the originating GW. The advertising interval of the GW must be chosen with care so that the network is not flooded unnecessarily many times. When a MN receives a GWAD message, it updates its routing table for the GW and responds with a Gateway Acknowledgement (GWACK) message only under three conditions: (a) First, if a MN is not registered with any other GW yet, it attempts to join the *K-hop* subnet of the originating GW by issuing a GWACK message; (b) The MN will also attempt to join the *K-hop* subnet from the GW with which it currently registers; (c) If a MN receives a GWAD message which originated from a GW different from its currently registered GW, a MN compares the hop count, and/or geographical distance from the GWs and selects which GW should be its current registered GW based on the rules stated in the following. Firstly, the number of hops away from GWs is compared, and the one with the smallest hop count is chosen. In case when the number of hops away from a GW is equal, the geographical distance away from the GW is calculated and the one with shortest distance is chosen. By this means, only one GW, which is closest to the MN, will be chosen to be registered with by the MN. Upon receiving the GWACK message, the new registered GW is responsible to inform the previous GW about the change and the previous GW will not maintain the information of this particular MN any more.

One thing to mention is that each GWAD message has a unique *broadcasting ID*, which is to prevent duplicate broadcast messages. When a MN receives a GWAD message, it first checks to determine whether the GWAD message with the same *originator address* and *broadcasting ID* already has been received previously. This means each MN needs to maintain a history table about the GWAD messages, which contain the pairs of *GW address* and *broadcasting ID*. If after checking, the MN finds that such a GWAD message has not been received, the GWAD message is rebroadcast if the GWAD message has not already propagated up to K hops yet. Otherwise, if such a

GWAD message has been received, the newly received GWAD message will be discarded. Furthermore, the serving area of the GW must overlap to ensure that each MN can receive advertisements from at least one particular GW. This means that either the K value should be large enough or there must be more than enough GWs in the network.

This proactive advertisement approach has one noticeable disadvantage, which is that the broadcast message is flooded through the local subnet periodically. This is a very costly operation, since limited resources in the wireless medium, such as bandwidth, will be used often. However, since only the local subnet is flooded, the periodic broadcasting of advertisement messages is acceptable with a carefully chosen interval. Furthermore, the proactive advertisement provides periodic link connectivity updates to the GW. This helps the MNs to be updated with relatively up-to-date routing information about its current registered GW.

3.3.2 Location Update Mechanism

In order to keep the routing and location information up-to-date at the GWs, a MN employs a periodic updating mechanism. A MN periodically sends out location update messages to its current registered GW. This location update message contains the current location information about the MN, which is unicasted towards the MN's current registered GW. Upon receiving the location update message, the GW will update its routing table and location information table about this MN.

In order to avoid potential problems, each MN needs to maintain not only the routing information to the current registered GW but also all the routing information to other GWs, which it has received through the GWAD message. Therefore, a MN can forward the location update message for other downstream MNs even if the destination of the location update message is not the current registered GW of the forwarding MN.

3.4 Conclusion

In this chapter, a *gateway discovery algorithm* has been presented. This algorithm is designed to partition a large wireless network domain into a number of smaller subnets (probably up-to a few hops away from the GW) with localized connectivity between the GW and MNs inside the subnet. Simulation results shown later in Chapter 6 prove this algorithm works fine with the associated routing protocols in the hybrid wired-wireless network.

CHAPTER IV

LAOD ROUTING PROTOCOL

4.1 Introduction

Location-Aided On-Demand (LAOD) routing protocol is designed to provide wireless multiple-hop paths, which can be wireless routing paths or wireless-cum-wired routing paths. The LAOD routing protocol aims to achieve better routing performance with the help of geographical location information in a hybrid wired-wireless network environment. A key concern of our routing protocol design is how to utilize the GWs without congesting them with excessive communication.

LAOD consists of two separate phases: (a) WR route discovery phase; and (b) Route maintenance phase. LAOD tries to combine on-demand routing with the greedy packet forwarding mechanism to achieve a more scalable routing protocol. LAOD uses the greedy packet forwarding mechanism when the destination location information is available. Here, the choice of the greedy packet forwarding mechanism is because of its simplicity and efficiency in a dense network. With the *gateway discovery algorithm*, information about the MNs is collected and stored at the GWs, as described in Chapter 3. This is then utilized by LAOD in the following manner, which is described in detail below.

4.2 WR Route Discovery

A source MN always tries to find the local routing path by initializing the local route discovery, which is called the WR route discovery process. This aims to find a WR path (which is explained in Chapter 1) when a source MN and a destination MN are in the same subnet, or are close to each other.

Whenever a source MN has data packets to send, it first checks its routing table to determine whether it has a current route to that destination MN. If none exists, it initiates the route discovery process similar to that of AODV. But unlike AODV, the RREQ message is broadcasted only to MNs in the region within a few hops away from the source MN instead of the whole network. This region should include the current registered GW of the source MN. This can be done by specifying the Time-To-Live (TTL) of the RREQ message to be the number of hops from the current registered GW of the source MN. This means the RREQ message can propagate at most K hops away, since the MN must be inside the *K-hop* subnet of one particular GW. There are a few possible cases that the RREP message can be generated in response to the RREQ message. Then, the source MN makes the routing decision according to where the RREP message is from and what kind of information it contains, as shown in Figure 4.1:

- If the RREP message is from the destination or an intermediate MN with an up-to-date route to the destination, the source MN sends data packets using the returned routing information. In this case, the RREP message sets up the forwarding route from the source to the destination in a similar style as AODV, then data packets are forwarded along the forwarding route. The RREP message also contains the location information of the destination, which will not be used, since we are not using the greedy packet forwarding mechanism in this case.
- If the RREP message is from the GW with the location information of the destination node, which means the destination is within the same subnet as the source, packets are sent towards the destination by the greedy packet forwarding mechanism. In this case, with the location information of the destination, the next-hop selection is based on the greedy packet forwarding mechanism described in Chapter 2. The data packets using the greedy packet forwarding mechanism will be marked in the packet header to indicate it is forwarded by the using greedy packet

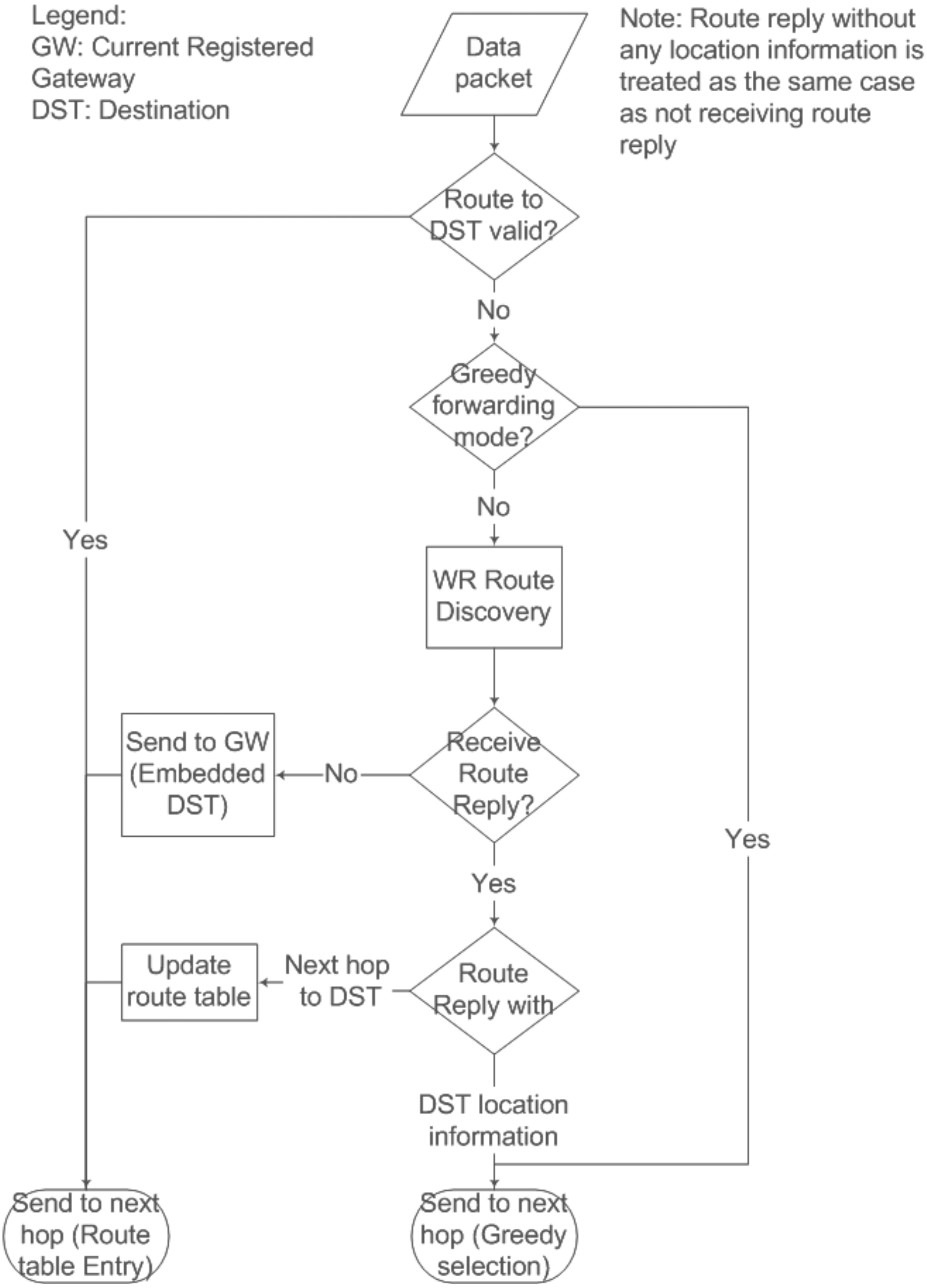


Figure 4.1: LAOD route selection flow chart for MN

forwarding mechanism.

- If the RREP message is from the GW but without any location information, or no reply is received for the RREQ message, the source MN sends the data packets towards its current registered GW with the destination address embedded. In this case, the WWR path (which is explained in Chapter 1) is used. The source MN forwards the data packets towards the current registered GW by using the route obtained during the *gateway discovery algorithm*. Here we assume GWs exchange information of MNs under them. Therefore, after the data packets reach the desired GW, the GW checks for the destination , then continues to forward the data packets towards the destination accordingly.

If there are both RREP messages from the destination or an intermediate node with an up-to-date route to the destination, and the RREP messages from GWs, the source MN prefers the first case of RREP message, which is the RREP messages from the destination or an intermediate node with an up-to-date route to the destination.

An intermediate MN follows the routing decision made by the source MN. Whenever an intermediate MN fails to find a next-hop, due to a broken route or the *Local Maximum* problem, it will send the data packets towards its current registered GW as a last resort, since the GW might be able to find an alternative route to the destination. If this still fails, it broadcast a route error message (RERR) to its neighbors.

4.3 Route Maintenance

After the routing path has been set up, it needs to be maintained during data communication. The data packets are delivered to the destination by one of the following mechanisms in LAOD, namely, *Normal Route Forwarding*, *Greedy Packet Forwarding* and *Gateway Packet Forwarding*. The route maintenance of these three different forwarding mechanisms are different processes, as shown below:

- *Normal Route Forwarding*, which uses a route obtained by the RREP message from the destination or an intermediate node with an up-to-date route to the destination. The maintenance of such a route is similar to AODV. When a MN detects that a route to a neighbor is no longer valid, it will remove the routing entry related with the neighbor and send a link failure message to other neighbors that are actively using the route, informing them that this route is not valid any more. The MNs that receive this message will repeat this procedure. This message will eventually be received by the affected source MN. The source MN can choose to either stop sending data packets or request a new route by sending out a new RREQ message.

- *Greedy Packet Forwarding*, which uses the destination location information. In this case, the routing path from the source to the destination is based on the hop-by-hop local optimal selection. Each intermediate MN forwards data packets only based on the location knowledge of the destination node and its neighbors. The location of a neighbor is obtained through the Hello message, which is periodically broadcasted. The location of the destination is forwarded together with the data packets from the source MN. The source MN obtains the location information of the destination during the WR route discovery process described above. Therefore only the destination location information needs to be updated at the source MN to keep the route up-to-date. The updating of the destination location information at the source MN is by the destination, which sends out its current location periodically through the reverse path from the destination to the source.

- *Gateway Packet Forwarding*, which uses a route via the GW. In other words, a WWR path is used. As described earlier in Chapter 3, WWR paths are maintained by the *gateway discovery algorithm*. The *gateway discovery algorithm* provides frequent route updates between the MNs and the GWs by exchanging *gateway advertisement messages*, *gateway acknowledgement messages* and *location update messages*. The freshness of a WWR path depends on how frequently these messages are exchanged. Once the data packets reach the GW from the MN through the WWR path, they will be passed to the appropriate GW which is responsible for the particular destination MN, as we assume the interconnected GWs are a big virtual node as explained in Section 1.3 of Chapter 1.

4.4 Conclusion

LAOD aims to achieve better routing performance than AODV for multiple-hop communications between MNs with the help of location information. The performance evaluation and comparison has been done through simulations in Chapter 6, which shows that LAOD achieves higher packet delivery at the expense of longer average end-to-end delay and higher overhead. These pitfalls of LAOD make us move on to design another routing protocol which can achieve better routing performance than LAOD. The result is the proposed Link-Connectivity-Prediction-Based Location-Aided Routing (LLR) protocol, which is presented in Chapter 5.

CHAPTER V

LLR ROUTING PROTOCOL

5.1 Introduction

As mentioned in Chapter 4, LAOD achieves higher packet delivery at the expense of longer average end-to-end delay and higher overhead compared to AODV. These are not satisfactory results. Therefore, we revamp our design into a new routing protocol, which is called the Link-Connectivity-Prediction-Based Location-Aided Routing (LLR) protocol. LLR aims to achieve better routing performance, like shorter average end-to-end delay, than LAOD.

LLR essentially consists of three separate phases: (a) WR route discovery phase; (b) Route maintenance phase; and (c) Route soft-handoff phase. A source MN always tries to find the local routing path by initializing local route discovery, which is called the WR route discovery process. This aims to find a WR path when the source MN and the destination MN are in the same subnet, or are close to each other. If no WR path is found, the source MN uses the WWR path by forwarding the data packet towards its currently registered GW, since the MN maintains connectivity with its currently registered GW through the *gateway discovery algorithm* described in Chapter 3. After the routing path has been set up, it needs to be maintained. The detailed algorithm is explained below.

5.2 WR Route Discovery

Whenever a source MN has data packets to send, it first checks its routing table to determine whether it has a current route to that destination MN. If none exists, it initiates the route discovery process similar to that of AODV. But unlike AODV, the RREQ message is broadcasted only to MNs in the region within a few hops away from the source MN instead of the whole network, which is the region within the current registered GW of the source MN. This can be done by specifying the Time-To-Live (TTL) of the RREQ message to be the number of hops away from the current registered GW of the source MN. This means the RREQ message can propagate at most K hops away, since the MN is inside the *K-hop* subnet of the GW. The reverse route is set up by the RREQ message, same as in AODV. A RREP message can be generated by the destination MN, or intermediate neighbors with an up-to-date route to the destination. The WR path from the source to the destination is then set up as the RREP message travels back to the source. Upon receiving the RREP message, the source MN starts sending data packets along the WR path. If the source MN receives no RREP message, the WWR path is used. The WWR path is always available since each MN establishes and maintains a route towards its current registered GW during the *gateway discovery process*. When the WWR path is used, the MN sends data packets towards its current registered GW and sets a flag for that destination MN in the routing table to indicate it is using the WWR path. Each data packet is then embedded with the address of the destination. After the data packets reach the destination GW, the destination GW checks its routing table for the next-hop node towards the destination MN and sends out the data packets accordingly. As we assume the interconnected GWs are a big virtual node as explained in Section 1.3 of Chapter 1, the GW node always knows which other GW node to send the data packet to. If the WR path is found, the source MN always prefers the WR path over the WWR path. During the connection, the source MN appends the following information to each data packet: (a) its current location information; and (b) a flag indicating whether it is the WWR path or the WR path.

5.3 Route Maintenance

As explained earlier, there are two possible routing paths: (a) a WR path, which is the shorter routing path without going through a GW and (b) a WWR path, which is the longer routing path via GWs. The maintenance of these two different paths is performed by different processes. At times, the movements of the source MN and the destination MN may request switching from WR path to WWR path or vice versa, in order to achieve better routing performance. This will be discussed later.

5.3.1 WWR Maintenance

WWR paths are maintained by the *gateway discovery algorithm*. The *gateway discovery algorithm* provides frequent route updates between the MNs and the GWs by exchanging *gateway advertisement messages*, *gateway acknowledgement messages* and *location update messages*. The freshness of a WWR path depends on how frequently these messages are exchanged.

5.3.2 WR Maintenance

Before the WR maintenance process is presented, we would like to introduce some terminology which will be used in the subsequent discussion: the Route Expiration Time (RET), which is the minimum LET along the path from the source to the destination. As shown in Figure 5.1, the RET of path from MN_S to MN_D is the minimum LET among the LETs along the path (MN_S-MN1-MN2-MN_D).

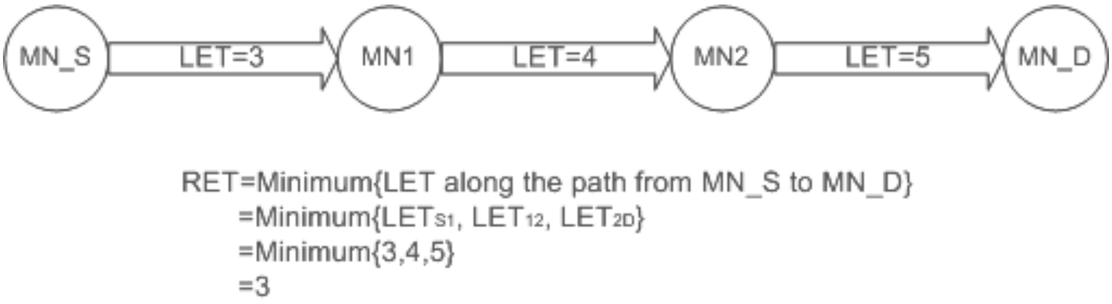


Figure 5.1: An example of Route Expiration Time (RET)

Another term used in the subsequent discussion is the *critical time*, T_c, given by:

$$T_c = RET - T_d \tag{5.1}$$

where T_d is the delay experienced by the latest packet which has arrived along the route.

During the duration of a WR connection, intermediate MNs keep updating the RET to each data packet based on the LET, enabling the destination MN to receive the RET prediction together with the latest source MN related information from each data packet. When the destination MN determines that the route is about to expire, at this *critical time* instant (Eqn 5.1), it computes both the *expected zone* and *request zone* using the latest source MN related information from the last data packet received, in a similar manner as LAR. It then attaches the information to a specific RREQ message, which is called SRREQ message, and then broadcasts the SRREQ message. The purpose of broadcasting the SRREQ message is to make sure that the source MN can receive such a message at a minimum network cost. If we depend only on the reverse path, which is obtained during the route discovery process from the RREP message, the SRREQ message may not reach the source MN, because the reverse path may be out-dated and invalid when the forwarding path is about to expire. Only the source MN can reply to this SRREQ message, which also contains the current RET. The receiving intermediate MN first checks whether it is inside the *request zone* and only MNs within the *request zone* can forward the SRREQ message. The MN then checks the LET of the last link that SRREQ message is received from and if the LET is less than or equal to RET embedded in the SRREQ message, the SSREQ message is dropped instead of being forwarded. Eventually, the source MN should receive one or more SRREQ messages. If there are alternative routes with better RET, the source MN chooses the best route on which to re-route the data packets based on the information contained in the SRREQ message (e.g. number of hops, destination sequence number, etc). After that, the source MN starts sending data packets along the new route.

5.3.3 Route Soft-Handoff

Here, we refer to route soft-handoff as either switching from WWR to WR or vice versa. It is sometimes necessary to do such a route handoff in order to achieve better routing performance. For example, when both the source MN and the destination MN move into the same subnet while the communication between them is still going through the GWs, it can be better to switch from WWR to WR.

A new metric is used to decide whether to do the route soft-handoff. The metric, called *percentage metric* is calculated by summing the percentage improvement in both the number of hop counts and RET. Let us assume two possible routing paths between the source and the destination are present. Let nh_1 be the hop count of route1 and nh_2 be that of route2. Also, let RET_1 and RET_2 be the route expiration time of route1 and route2 respectively. Then, the percentage improvement of route1 over route2, Δ_{12} is obtained as follows:

$$\Delta_{nh} = -\frac{nh_1 - nh2}{nh_1} * 100\% \tag{5.2}$$

$$\Delta_{RET} = \frac{RET_1 - RET2}{RET_1} * 100\% \tag{5.3}$$

$$\Delta_{12} = \Delta_{nh} + \Delta_{RET} \tag{5.4}$$

One example is shown in Figure 5.2. As can be seen, for route1, $nh_1 = 4hops$, for

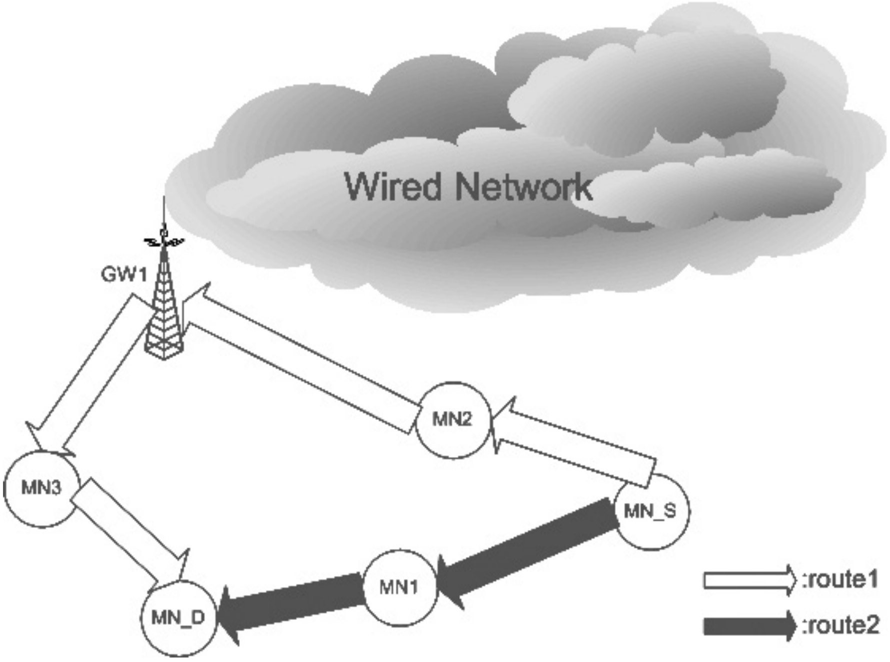


Figure 5.2: An example of route soft-handoff with *performance metric* calculation

route2, $nh_2 = 2hops$. Let us assume $RET1 = 5seconds$, $RET2 = 4seconds$. From the equations presented above, Δ_{nh}=-(4-2)/4=-50%, $\Delta_{RET} = (5-4)/5$=20%. Then, the *percentage metric*, Δ_{12} = -30%, which means route2 is better than route1 in terms of the combination of number of hop count and RET. This is reasonable since route2 is two hops shorter than route1 with only one second RET shorter than route1. Therefore, by using the *percentage metric*, route2 should be used.

The *gateway discovery algorithm* keeps GWs aware of where the source and destination MNs are, and which GWs they are currently registered with. When the source or destination MN registers with a new GW, the GW helps the destination MN initialize the handoff process by providing the destination MN with the routing information of the WWR path. The selection of the routing path then depends on the *percentage metric*. There are two possible scenarios, either switch from WWR to WR or from WR to WWR:

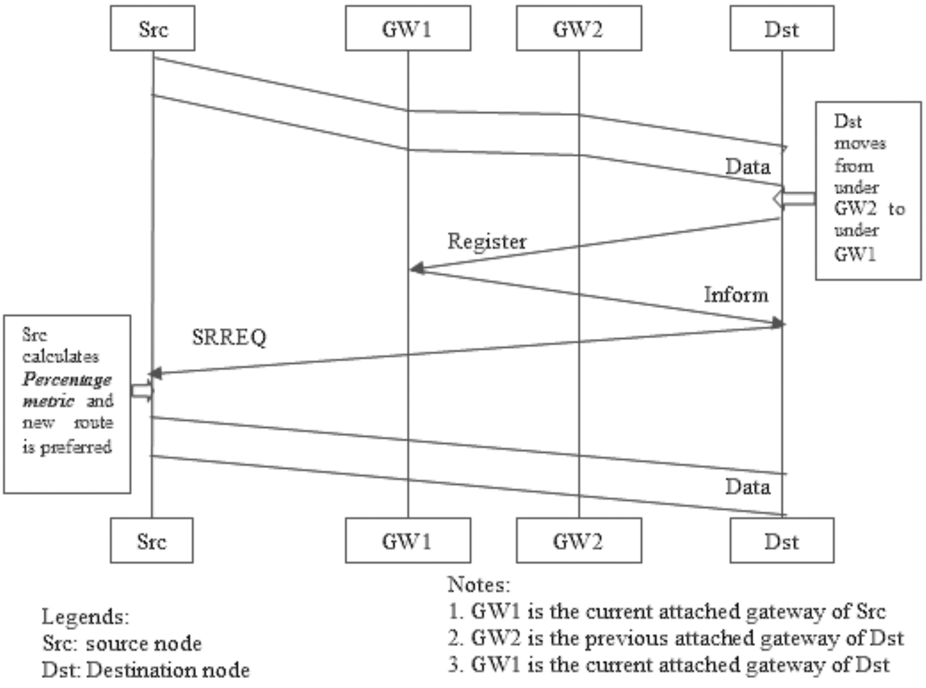


Figure 5.3: Messages exchange sequence during the WWR to WR handoff process when destination node moves under the same gateway as source node

WWR to WR When the source node and the destination node are under the same GW but still using the WWR path, the GW informs the destination node about that.

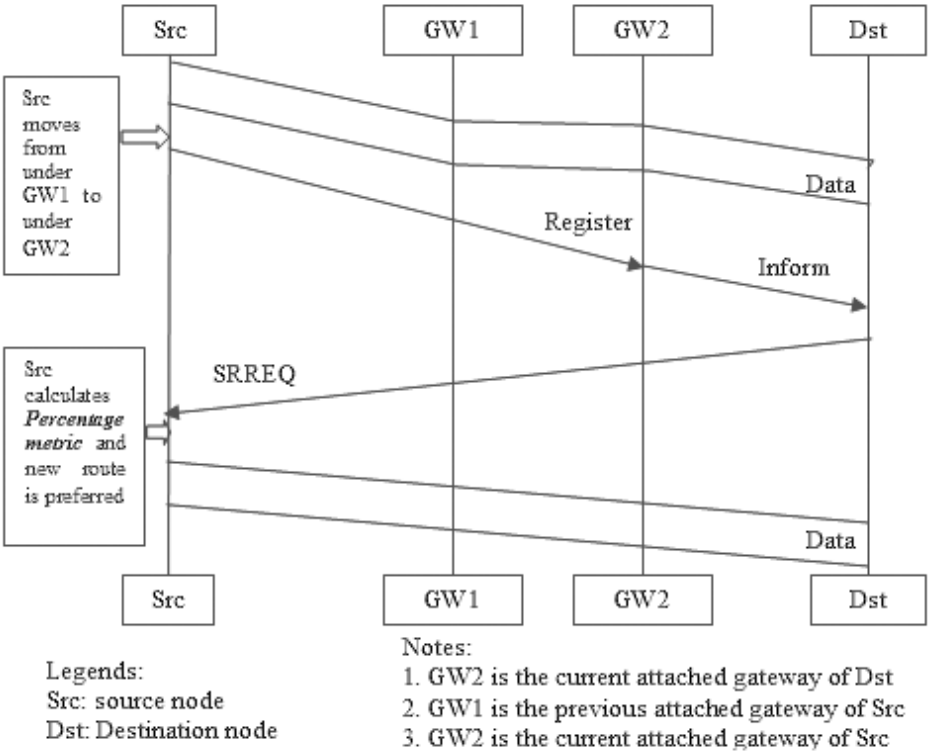


Figure 5.4: Messages exchange sequence during the WWR to WR handoff process when source node moves under the same gateway as source node

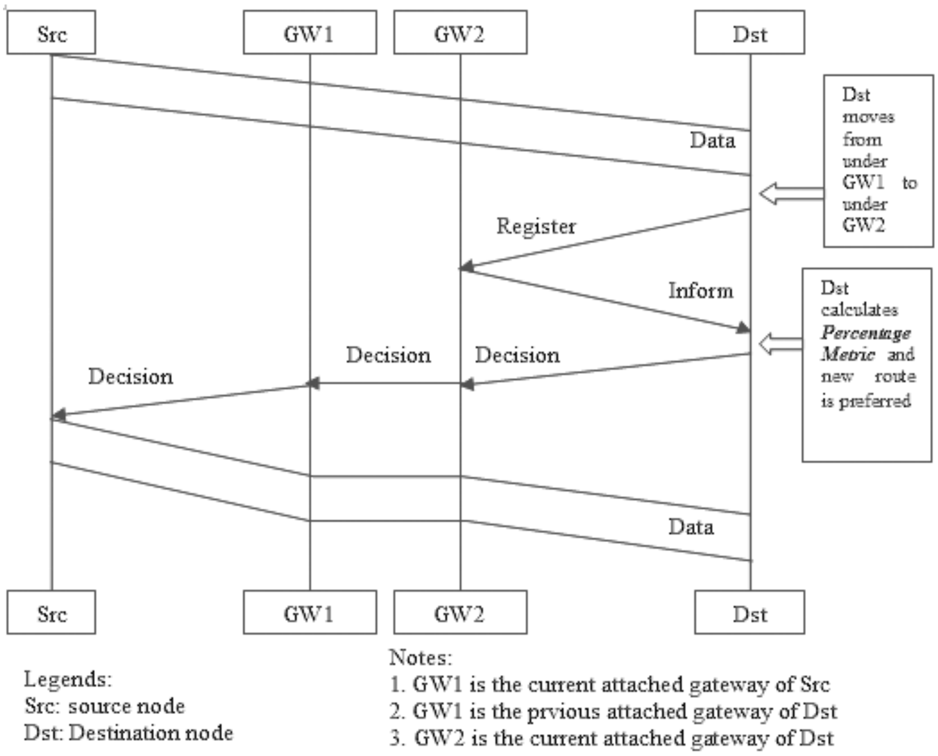


Figure 5.5: Messages exchange sequence during the WR to WWR handoff process when destination node moves under different gateway as source node

Then, the destination node calculates both the *expected zone* and *request zone* for the source node. After that, it attaches such information to a SRREQ message and broadcasts out. After the source node receives the SRREQ message, it determines the best routing path on which to route the data packets based on the *percentage metric* described above. Then the source starts sending data packets along the new WR path, only if the WR path is better in terms of the *percentage metric*. Figure 5.3 shows the message exchange sequence during the handoff process when the destination node moves under the same GW as the source node, while Figure 5.4 shows the case when the source node moves under the same GW as the destination node. The *gateway handoff messages* between GWs are not shown.

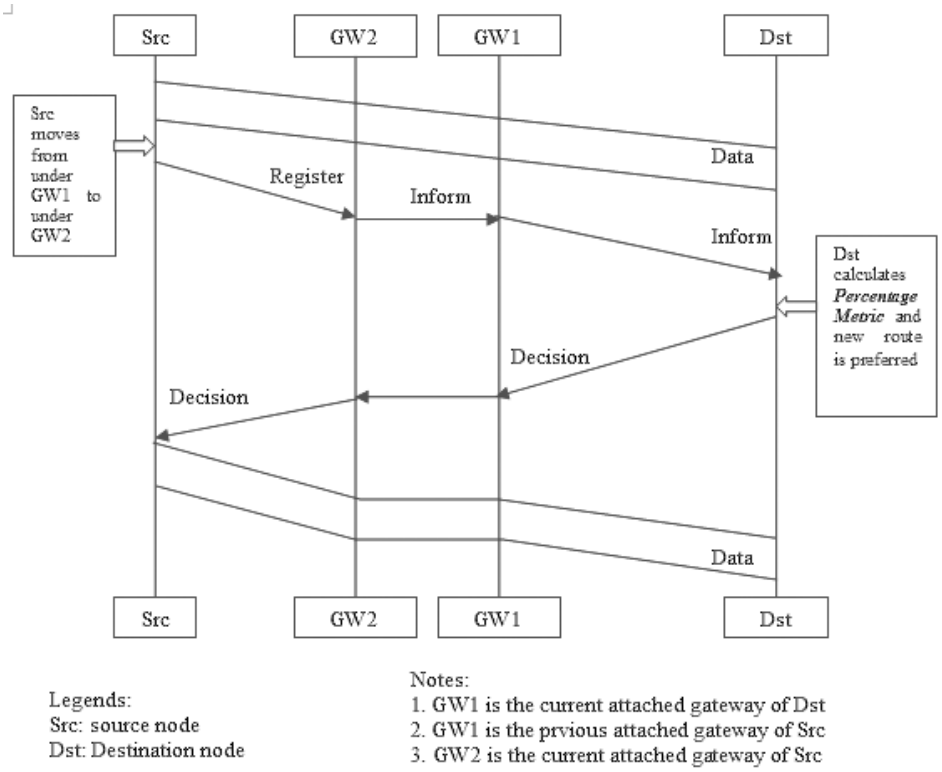


Figure 5.6: Messages exchange sequence during the WR to WWR handoff process when source node moves under different gateway as source node

WR to WWR When the source node and the destination node are under different GWs but still using the WR path, the GW calculates the hop count and RET of the

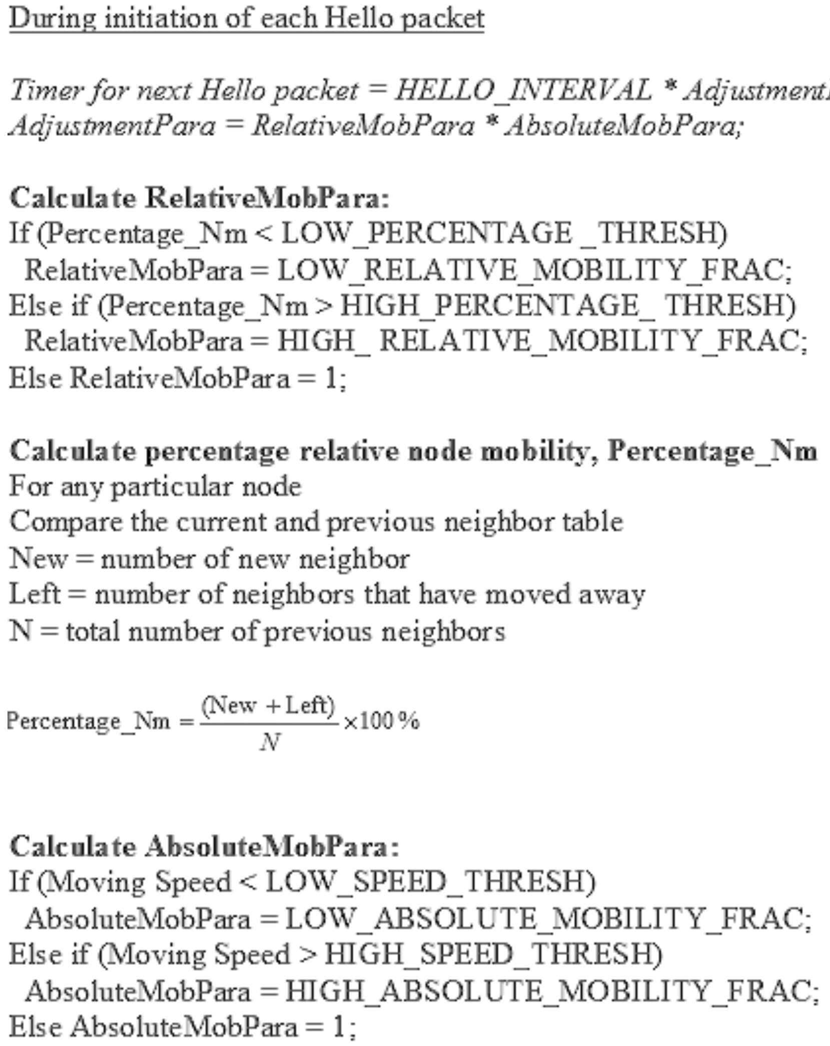
WWR path between the source node and the destination node. Then it informs the destination about these two parameters of the WWR path. After receiving the message with these two parameters, the destination node calculates the *percentage metric*, and then determines the best route to use. If the WWR path is better in terms of the *percentage metric*, the destination node informs the GW and the GW will then inform the source node. After that, the source node starts sending data packets along the WWR path. Figure 5.5 shows the message exchange sequence during the handoff process when the destination node moves under a different GW from the source node, while Figure 5.6 shows the case when the source node moves under a different GW from the destination node. The gateway handoff messages between GWs are not shown.

5.3.4 Hello Message Adjustment Algorithm

In LLR, each MN needs to periodically broadcast Hello messages to maintain the neighbor connectivity, just like AODV [10]. However, usage of Hello messages contributes to the overhead and affects the routing performance. Motivated by the research work in [54], we incorporated a Hello message adjustment algorithm into LLR, in order to further improve the routing performance. The purpose of this algorithm is to try to carry more data traffic on the network, (i.e. increase the network throughput) while still achieving similar routing performance compared with the one without the Hello message adjustment algorithm. At the same time, by reducing the broadcasting of Hello messages, we can save power consumption for the entire network. Here, the Hello packets are considered as overhead. There are two reasons to do that. First, we want to control the Hello packets to include some extra information, like location information, from the routing layer. Next, we want to have transparency on the MAC layer, i.e. we do not want our routing protocols, namely LAOD and LLR, to be limited by certain services from a particular MAC layer, like *link layer notification*. Furthermore, *RFC 3561* for AODV [55] states that Hello messages can be used to determine the local connectivity. This can be thought as a cross-layer optimization. Therefore, the Hello packets are considered as overhead.

The usage of Hello messages, which are being periodically broadcasted one-hop away by MNs, increases the network overhead. They contend with data packets and other important routing messages (like RREQ and RREP) for bandwidth. This may lead to high incidence of collision of packets, which in turn causes MAC backoff or even worse, packet drops. Therefore, it results in routing performance degradation, e.g. packet delivery ratio (PDR) reduction and end-to-end delay increase. However, as mentioned above, the Hello message is necessary to provide the local connectivity, and it cannot be eliminated completely. As such, it is worth studying how to adjust the broadcasting of Hello messages while it is still sufficient to provide local connectivity without degrading the overall routing performance, or even improving routing performance.

The Hello message adjustment algorithm varies the frequency of broadcasts of Hello messages from each MN, according to the relative mobility of MN's neighbors and MN moving speed. Figure 5.7 shows the pseudo-code for the Hello message adjustment algorithm, which also shows the definitions of the *RelativeMobPara* and the *AbsoluteMobPara*. During the initiation of each Hello message, each MN checks the *RelativeMobPara* and the *AbsoluteMobPara*. The *RelativeMobPara* is related to the percentage of neighbors' link change, while the *AbsoluteMobPara* is related to the moving speed of the MN. In summary, we vary the time interval of sending the Hello message with respect to the mobility of the MN: (a) When there is low relative mobility, we increase the Hello message sending interval, and vice versa; (b) When there is low absolute mobility, we further increase the Hello message sending interval, and vice versa. The reasons are: (a) When there are fewer relative node movements, the network is more stable, since there are fewer link breakages. Thus, the frequency of sending Hello messages should be reduced. On the other hand, when there are more neighbors changing, or more link breakages, the network is less stable. As such, the frequency of sending Hello messages at this time should be increased; (b) The slower the MN moves, the more stable the network is likely to become. Hence, the frequency of sending Hello messages should be



```
During initiation of each Hello packet

Timer for next Hello packet = HELLO_INTERVAL * AdjustmentPara
AdjustmentPara = RelativeMobPara * AbsoluteMobPara;

Calculate RelativeMobPara:
If (Percentage_Nm < LOW_PERCENTAGE _THRESH)
  RelativeMobPara = LOW_RELATIVE_MOBILITY_FRAC;
Else if (Percentage_Nm > HIGH_PERCENTAGE_ THRESH)
  RelativeMobPara = HIGH_ RELATIVE_MOBILITY_FRAC;
Else RelativeMobPara = 1;

Calculate percentage relative node mobility, Percentage_Nm
For any particular node
Compare the current and previous neighbor table
New = number of new neighbor
Left = number of neighbors that have moved away
N = total number of previous neighbors

$Percentage\_Nm = \frac{(New + Left)}{N} \times 100\,\%$

Calculate AbsoluteMobPara:
If (Moving Speed < LOW_SPEED_THRESH)
  AbsoluteMobPara = LOW_ABSOLUTE_MOBILITY_FRAC;
Else if (Moving Speed > HIGH_SPEED_THRESH)
  AbsoluteMobPara = HIGH_ABSOLUTE_MOBILITY_FRAC;
Else AbsoluteMobPara = 1;
```

Figure 5.7: Pseudo code of the Hello message adjustment algorithm

slightly reduced. On the other hand, the faster the MN moves, the more likely link status will change, and this means the network is likely to become less stable. Therefore, the frequency of sending Hello messages should be slightly increased. This is especially true when each MN can be considered as an individual entity.

5.4 Conclusion

LLR makes use of location information to predict link connectivity and restrict broadcasting of control messages so that more packets can be delivered to their destination successfully. In our comparative simulation study with LAOD and AODV, as shown later in Chapter 6, LLR achieves higher packet delivery ratio, less overhead and less end-to-end delay compared to LAOD and AODV.

CHAPTER VI

SIMULATION RESULTS

In this chapter we present our simulation studies. All the simulations were done using the Network Simulator (NS2) [56]. Two different mobility models are used to evaluate the performance, namely *Manhattan Grid* mobility model and *Graph-based* mobility model. The choice of these two mobility models are because we are interested to simulate a large vehicular bus network, which means the MNs in our simulations are buses that move by following some paths instead of totally random movements. These two models, which will be described later, provide what we need as the bus network. We compared the routing performance by varying two controlled input parameters as shown below:

1. The mobility of the MNs, i.e. the mean speed of the MNs in the network. This is to analyze the effect of mobility speed on the routing protocols.
2. The load of the network, i.e. the number of source and destination pairs in the network, or the number of Constant Bit Rate (CBR) connections. This is to analyze the effect of data traffic on the routing protocols.

One thing that we like to explain a bit more is about the range of the mean speed of the MNs, which is from $1m/s$ to $20m/s$. This is realistic as movement of buses in city area, especially in a very crowded area or downtown area, like the Orchard Road area or central business district of Singapore. The bus probably moves very slowly at an average speed around $10km/hour$ to $20km/hour$ ($2.78m/s$ to $5.56m/s$). It represents the repeated stop-and-go traffic pattern in modern urban environment [57].

In order to have a clearer view of performance, three different sets of simulations are run and demonstrated, which are listed below:

Simulations Set I The performance comparison between AODV, LAOD [17] and LLR [18] without the Hello message adjustment.

Simulations Set II The performance comparison between LLR and its variants of Hello message adjustment schemes.

Simulations Set III The performance under very high network data loading comparison between AODV, LAOD, LLR and LLR with Hello message adjustment scheme.

All simulations use Constant Bit Rate (CBR) traffic flows with sources and destinations chosen randomly. Each CBR flow sends data with packet size of 512 bytes. The IEEE 802.11 Medium Access Control (MAC) protocol is used and both GWs and MNs have the same transmission range of 250m. In all simulations, the number of nodes is fixed at 150. Each simulation lasts for 900 seconds of simulation time. Table 6.1 shows the key parameters in all the simulations.

Table 6.1: Key Parameters used during simulations

Parameter	**Value**
Data traffic	Constant Bit Rate(CBR)
Packet Rate	4 packets/s (Simulation Set I & II) 10 packets/s (Simulation Set III)
Packet Size	512 bytes
Transmitter Range	250 m
Number of MNs	150
Simulation Time	900 s

We choose to use only CBR data traffic in our simulations. There are three reasons for that decision. Firstly, CBR data traffic does not require us to model the variance of the data rate. It simplifies the communication model. Next, CBR data traffic has persistent flow information within each particular data stream. It is easy to manipulate the routing information in both the GWs and the MNs. Lastly, the applications we consider during our research are those with tight QoS requirements and multimedia traffic, which normally have continuous and constant data flows from the source to the destination.

The following metrics are used to evaluate the performance of the protocols:

- *Packet Delivery Ratio (PDR)*: The fraction of data packets sent that are successfully delivered to their destination.
- *End-to-End Delay*: The average time interval between a data packet sent by a source and its arrival at its destination. End-to-End delay is only measured for packets that are successfully delivered to their destinations.
- *Overhead*: The total number of control/routing packets transmitted, including all types of control messages, like Hello packets, as well as other control packets like RREQ, RREP, RERR, GWAD and GWACK, except the data message. It can be considered as the aggregate of control/routing packets.
- *Normalized Overhead*: The total number of control/routing packets transmitted per data packet delivered at the destination. Each hop-wise transmission of a control/routing packet is counted as one transmission.
- *Hello Overhead*: The number of the Hello packets transmitted during the simulation.

Each scenario is also run with different seed numbers and the measurements are averaged out to minimize any arbitrary randomness. A convergence factor of 5% is used for all the simulations. In other words, the following conditions apply for all results we obtained:

$$\Delta_{PDR} = \frac{|PDR(t+10) - PDR(t)|}{PDR(t)} * 100\% < 5\% \tag{6.1}$$

$$\Delta_{EtE} = \frac{|EtE(t+10) - EtE(t)|}{EtE(t)} * 100\% < 5\% \tag{6.2}$$

$$\Delta_{OH} = \frac{|OH(t+10) - OH(t)|}{OH(t)} * 100\% < 5\% \tag{6.3}$$

In Eqn 6.1, $PDR(t)$ denotes the packet delivery ratio (PDR) at time t and $PDR(t+10)$ denotes the PDR at time $t+10$, i.e. a time interval of 10s. From this equation, we obtain the convergence factor of PDR for the simulations we ran. All our simulation results of PDR are converged at less than 5%, after starting up. Similarly, from Eqn 6.2 and 6.3, the same convergence factor applies to both the end-to-end delay and the overhead.

6.1 Mobility Models

Two simulation mobility models are used here, namely, *Manhattan Grid* mobility model and *Graph-based* mobility model. The *Manhattan Grid* mobility model is more structured with less variability compared to the *Graph-based* mobility model. The purpose of using these two mobility models is to see the impact of randomness of GW placement. Next we discuss these two mobility models in detail below.

6.1.1 Manhattan Grid Mobility Model

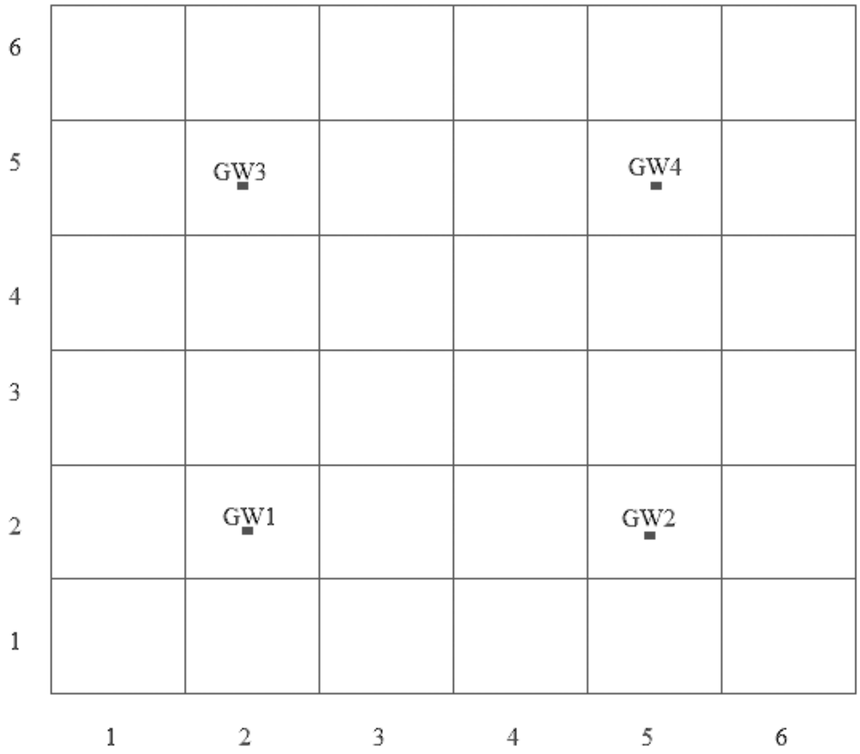


Figure 6.1: The *Manhattan Grid* mobility model graph used in the simulations

The *Manhattan Grid* mobility model is also known as the *City Section* mobility model. In this mobility model, the simulation area is a grid-like street network that represents a section of a city where the network exists [58]. The streets and speed limits on the streets are based on the type of city being simulated. For example, the streets may form a grid in the downtown area of the city with a high-speed highway near the border of the simulation area to represent a loop around the city. Each MN begins the simulation at a defined point on some street. An MN then randomly chooses a destination, also represented by a point on some street. The movement algorithm from the current position to the new destination locates a path corresponding to the shortest travel time between the two points. Upon reaching the destination, the MN pauses for a specified time and then randomly chooses another destination (i.e., a point on some street) and repeats the process.

Figure 6.1 shows an example of such a city graph, which is used later in the simulations. The graph contains 6X6 grid in 1600m by 1600m square area. Although this is unrealistic in the real world with such a small area, it is reasonable in simulation time. It is very difficult for us to collect, maintain and analyze the simulation data from a network with a very large size under some limited computing resource.

6.1.2 Graph-based Mobility Model

The *Graph-Based* mobility model [59], tries to provide a more realistic movement model by reflecting the spatial constraints in the real world. In this model, the graph is used to model the movement constraints imposed by the infrastructure of the real world. The *vertices* of the graph represents locations that the MNs might visit and the *edges* model the connections between these locations, e.g. streets or train connections. The graph is assumed to be connected, i.e. there is a path from any vertex to any other vertices in the graph. Each MN is initialized at a random vertex in the graph and moves towards another vertex, which is selected randomly as its destination. The MN always moves to the destination on the shortest possible path. After the MN reaches its destination, it pauses for a randomly selected period and then randomly picks out another

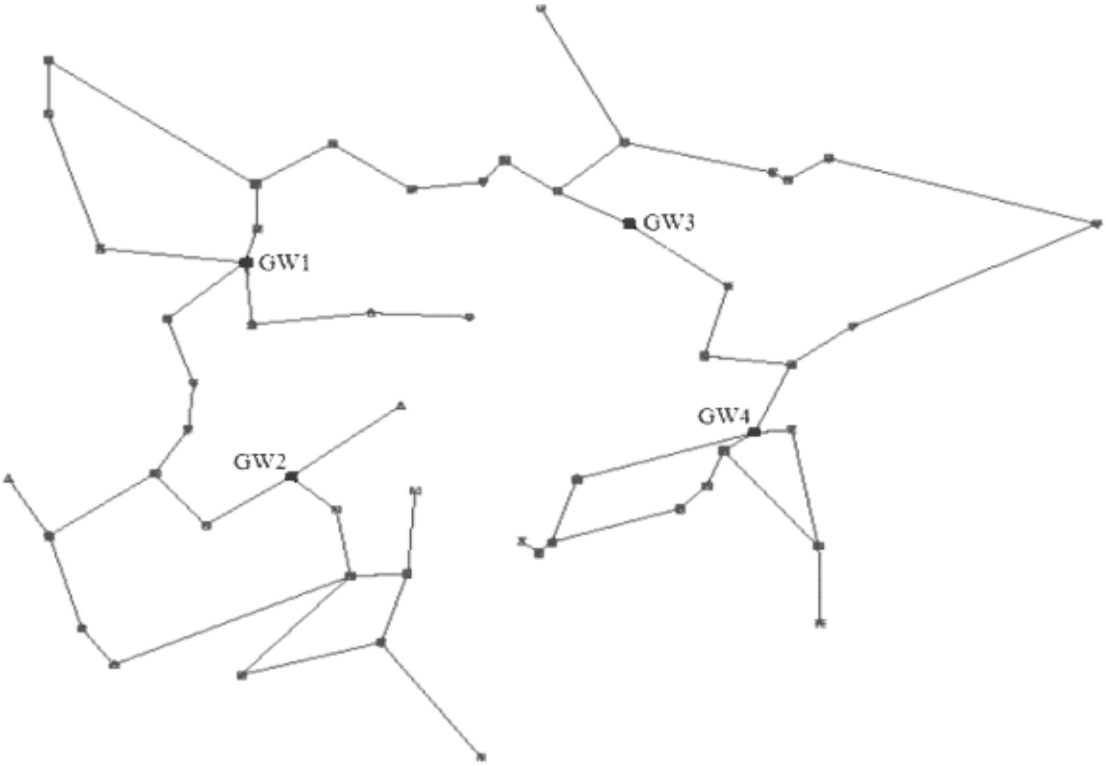


Figure 6.2: The city area graph used in the simulations

destination from other vertices for the next movement.

Compared to the *Manhattan Grid* mobility model, this *Graph-based* mobility model is more realistic since cities are not likely to be in a grid form in reality.

An example of graph mobility model is shown in Figure 6.2, which is used later in simulations. The graph contains 54 vertices representing significant locations and 59 edges representing road segments interconnecting them, covering an area of approximately 2500m by 1800m. The network size is a bit small. However, it is big enough for simulation purpose to yield some reasonable results.

6.2 Simulations Set I

Both the *Manhattan Grid* and *Graph-based* mobility models are used here. Besides the key simulation parameters shown in Table 6.1, other parameters used in the simulation are listed in Table 6.2:

Table 6.2: Parameters used during simulation set I

Parameter	**Value**	**Explanation**
Number of GWs	4	The location of GWs are shown in the Figure 6.1 and Figure 6.2 accordingly
Advertisement Zone	6 hops	The size of *K-hop* subnet, i.e. *K=6*
Advertisement Period	5 s	The period of sending gateway advertisement message by GW is 5s, refer to Chapter 3
Periodic Update Interval	5 s	The period of sending location update message by MN is 5s, refer to Chapter 3
Hello Beacon Interval	1 s	The period of sending neighbor Hello message by MN is 1s
Max Pause Time	20 s	The maximum time interval the MN will stay after reaching the destination but before heading towards the new destination

6.2.1 Simulation Results and Discussion (*Manhattan Grid* mobility model)

6.2.1.1 Varying Speed of MNs

Using the *Manhattan Grid* mobility model, the results are obtained by varying the mean speed of MNs from 1m/s to 20 m/s (i.e., the mobility of MNs), with the number of CBR connections fixed at 20. The network traffic load of 20 CBR connections is considered as medium loading for the network. We try to vary only one network parameter at one time. Therefore, once we vary the mean speed of MNs, we will fix all other network parameters, like network load (i.e. number of CBR connections).

Figure 6.3 shows the normalized overhead, while Figure 6.4 shows the actual number of control packets transmitted. Overhead here includes all the control messages used during the simulations. These are the *gateway discovery algorithm*'s control messages and routing control messages. *Gateway discovery algorithm*'s control messages consist of *gateway advertisement message* (GWAD), *gateway acknowledgement message* (GWACK), and *location update message*. Routing control messages consist of *route request message* (RREQ), *route reply message* (RREP), *route error message* (RERR) and *Hello beacon message*. All three protocols show more overhead as node speed increases because more route breaks occur, invoking route recovery procedures. However, LLR has the lowest overhead, in general, because the number of control messages during route recovery is reduced by limiting the broadcasting to a smaller region. Compared to AODV, LLR has achieved around 10% improvement at lower speed and 15% improvement at higher speed. On the other hand, LAOD has the highest actual overhead. The reason is that LAOD does not restrict the broadcasting of those route control messages when route recovery process is performed, unlike LLR. In the route maintenance process (Section 4.3 of Chapter 4), those source MNs which use normal route forwarding broadcast the RREQ messages in order to re-discover the path to the destination MNs. This broadcasting of the RREQ message is omni-directional, unlike the SSREQ message (Section 5.3.2 of Chapter 5), which is only broadcasted inside a particular area. Therefore, LAOD has much more overhead than LLR. On the other hand, LAOD has some extra *gateway discovery algorithm* control messages, which AODV does not have. Therefore, LAOD is the worst among the three protocols in terms of overall overhead, although AODV has created more routing control messages (e.g. RREQ and RREP) in general.

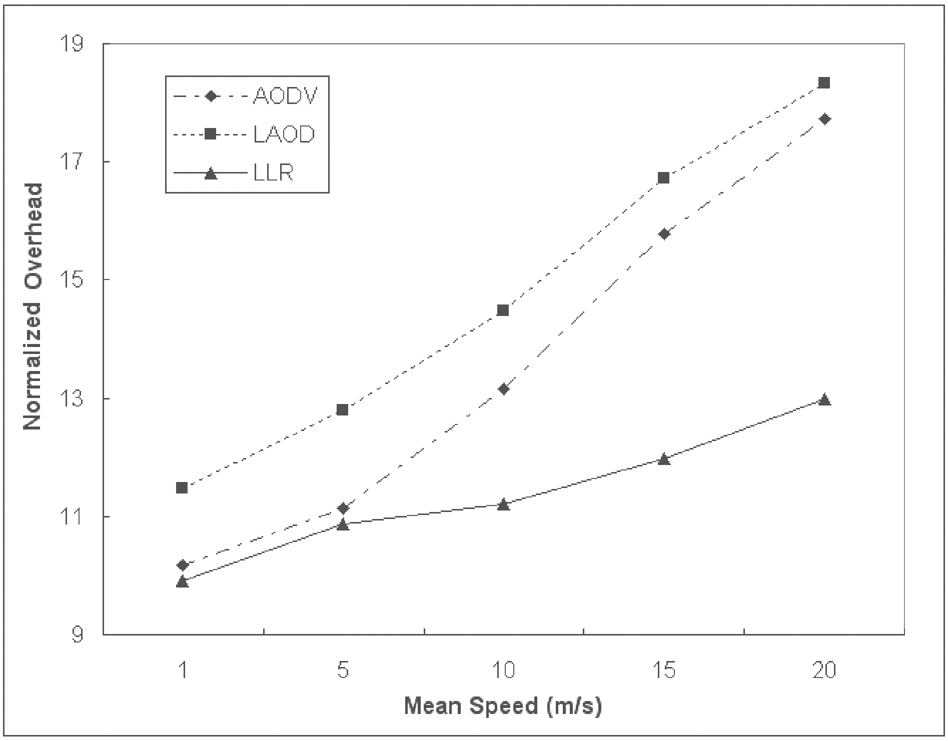


Figure 6.3: Normalized Overhead between LLR, LAOD and AODV using *Manhattan Grid* mobility model with different mobility speed

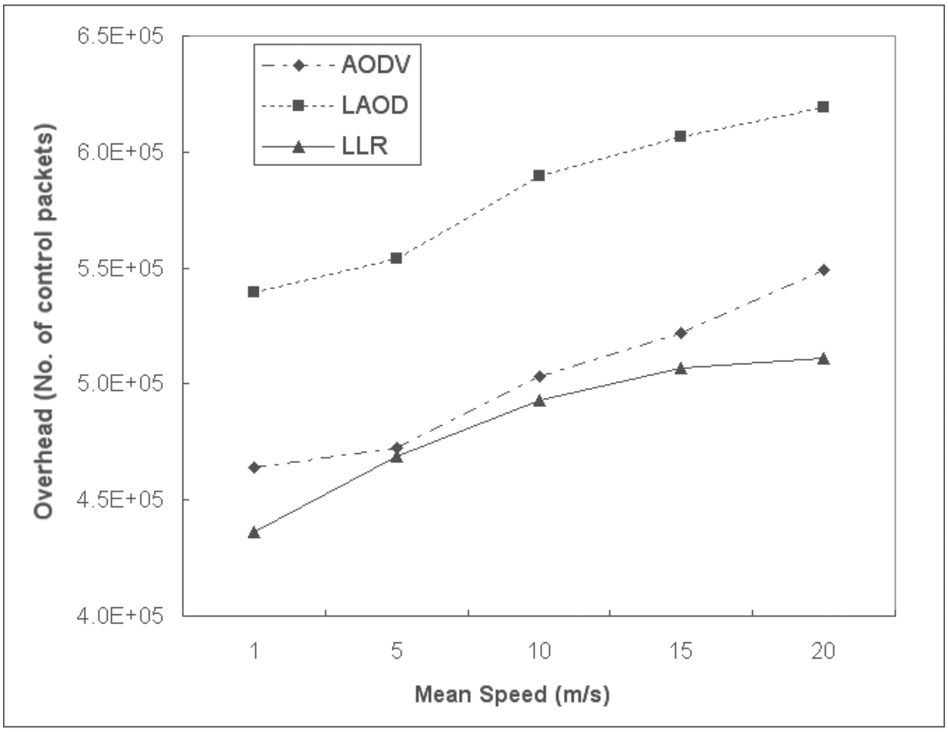


Figure 6.4: Overhead between LLR, LAOD and AODV using *Manhattan Grid* mobility model with different mobility speed

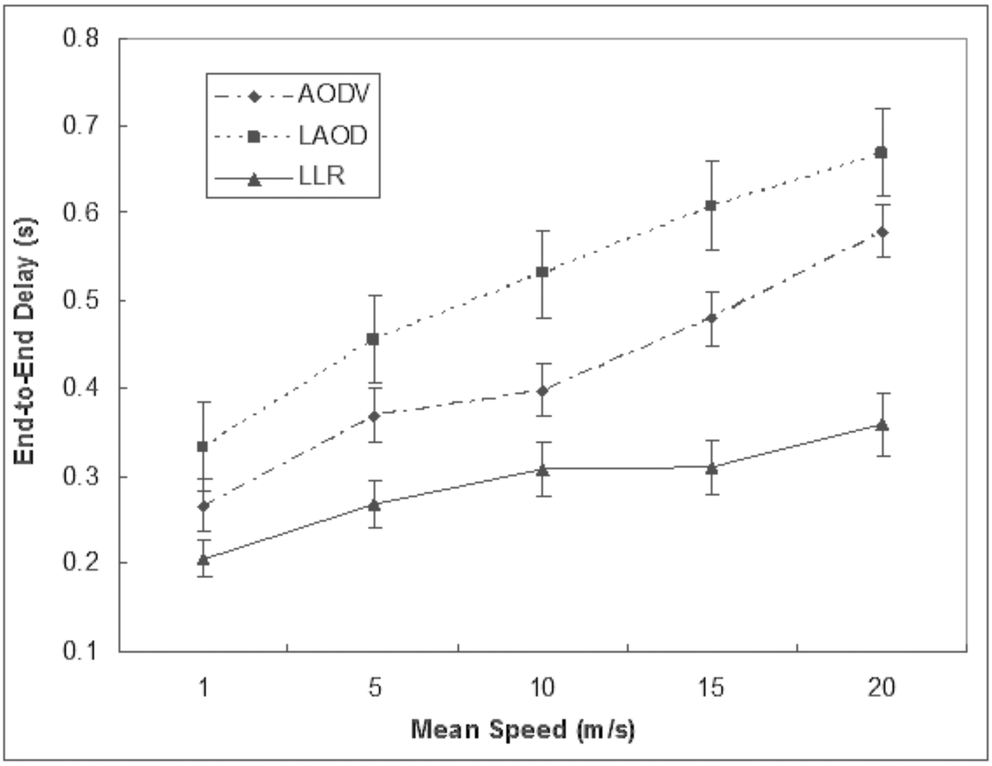


Figure 6.5: End-to-End Delay between LLR, LAOD and AODV using *Manhattan Grid* mobility model with different mobility speed

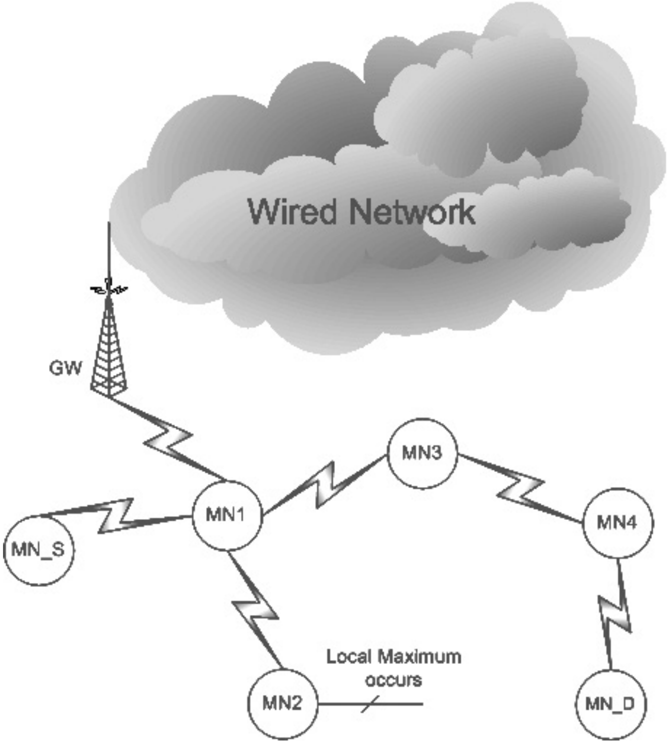


Figure 6.6: Scenario when greedy packet forwarding fails but gateway packet forwarding succeeds in LAOD

As shown in Figure 6.5, LLR outperforms the other two routing protocols in terms of end-to-end delay. Both AODV and LAOD have longer route (re)discovery latency after route breaks during which data packets are buffered while waiting for the new route to be constructed. LLR uses link connectivity prediction to perform rerouting prior to route disconnection, thus reducing the route (re)discovery latency. Compared to AODV, LLR has achieved around 15% improvement at lower speed and 40% improvement at higher speed. Surprisingly, LAOD has the worst end-to-end delay. This is because the scenario described below can easily happen in the network simulation, as shown in Figure 6.6. When the data packets, which are forwarded by the greedy packet forwarding mechanism from MN_S to MN_D fail at MN2 due to the *Local Maximum* problem, LAOD then tries to resolve this problem by forwarding the data packets via the GW from MN2. Some of these data packets do reach the destination node eventually. However, these data packets suffer longer delay. As shown in Figure 6.6, LAOD forwards data packets along the path MN_S-MN1-MN2-MN1-GW-MN1-MN3-MN4-MN_D, while AODV forwards data packets following the path MN_S-MN1-MN3-MN4-MN_D. Therefore, data packets delivered by LAOD travel almost twice the number of hops than those by AODV and thus incur extra delay.

Figure 6.5 also shows the delay variance/jitter of the end-to-end delay. Here the delay jitter is the standard deviation of the end-to-end delays experienced by the data packets between a source MN and a destination MN. As it can be seen, LAOD is again the worst one with around 7% variance, while both AODV and LLR are around 5%. The delay jitter does not show any significant difference between these three protocols.

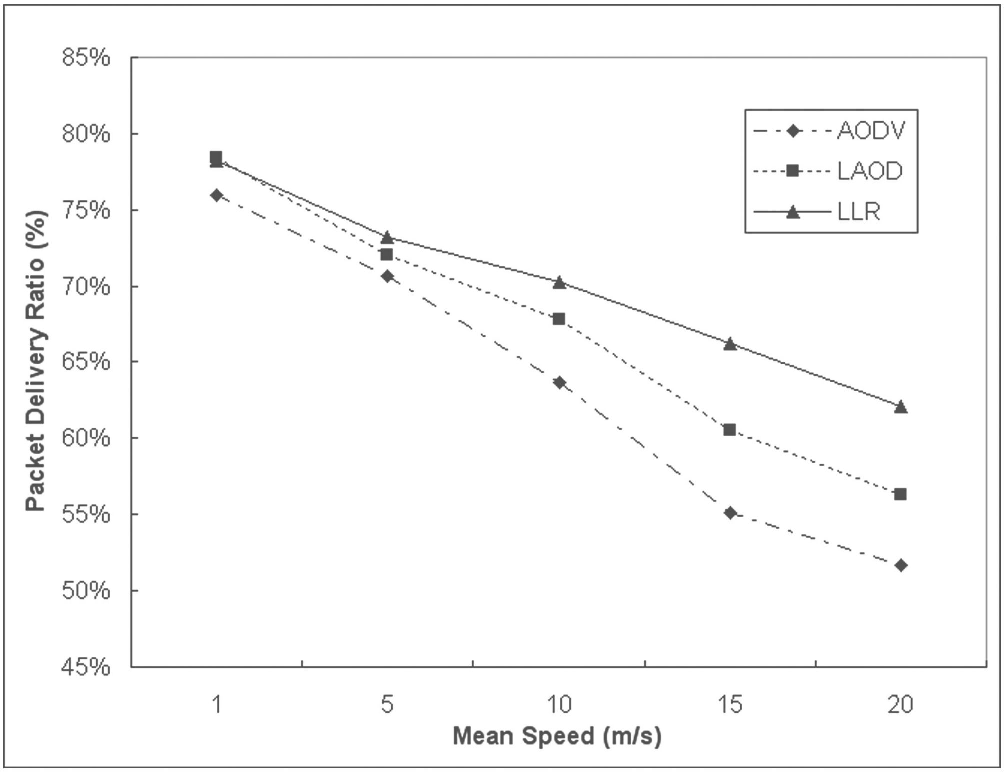


Figure 6.7: Packet Delivery Ratio (PDR) between LLR, LAOD and AODV using *Manhattan Grid* mobility model with different mobility speed

The packet delivery ratio (PDR) performance is shown in Figure 6.7. It is observed that fewer data packets are delivered as speed increases, which is expected. As MNs move faster, link connectivity changes more often and more control messages are broadcasted to make adjustments to the network topology change, contributing further to collisions, congestion, contention, and packet drops. LLR is least affected by mobility, since it limits the broadcasting of control messages for route discovery during the route recovery process. Besides reducing collisions, congestion, contention and packet drops with less broadcasting of control messages, LLR avoids route disconnection by using link connectivity prediction to perform rerouting prior to route disconnection. This helps to reduce packet drops too, since packets are more likely to be dropped during route disconnection due to buffer overflows, timeouts and other reasons. Compared to AODV, LLR has achieved around 2% improvement at lower speed and 10% improvement at higher speed. LAOD also performs slightly better than AODV. This is because LAOD uses the greedy packet forwarding mechanism as an alternative way to forward the data packets, which helps to deliver more data packets to the destination.

6.2.1.2 Varying Number of CBR Connections

The results are obtained by varying the number of CBR connections in the network from 10 to 50, with the mean speed of the MNs fixed at 20 m/s.

Figures 6.8-6.10 show the performance comparison between AODV, LAOD and LLR by varying the network traffic loads in the *Manhattan Grid* mobility model. We see, like before, LLR is the best protocol among these three routing protocols with highest PDR, lowest overhead, and lowest end-to-end delay. LAOD also achieves an improvement on PDR compared with AODV, but as before, at the expense of higher overhead and longer end-to-end delay. We can observer that LLR achieves around 2% improvement in PDR at lower speed and 10% improvement at higher speed compared to AODV. In terms of end-to-end delay, LLR is around 18% better at lower speed and 22% at higher speed than AODV. LLR also achieves around 3% improvement at lower speed and 6% improvement at high speed on overhead compared to AODV. However, the performance of these three routing protocols degrade when the data traffic in the network increases. Under heavy traffic loads (more than 30 data connections in the network), the overall performance deteriorates rapidly (all routing protocols deliver below 50% data packets, and end-to-end delay is around 10 times than light traffic loads). This is due to the excessive contention of the limited bandwidth in the network, leading to more collisions, more packets being lost and higher probability of congestion. Figure 6.10 also shows the delay variance/jitter of the end-to-end delay. As it can be seen, LAOD is again the worst one with around 5% variance, while both AODV and LLR are around 4%. This is consistent with the results we got by using the *Manhattan Grid* mobility model with varying speed.

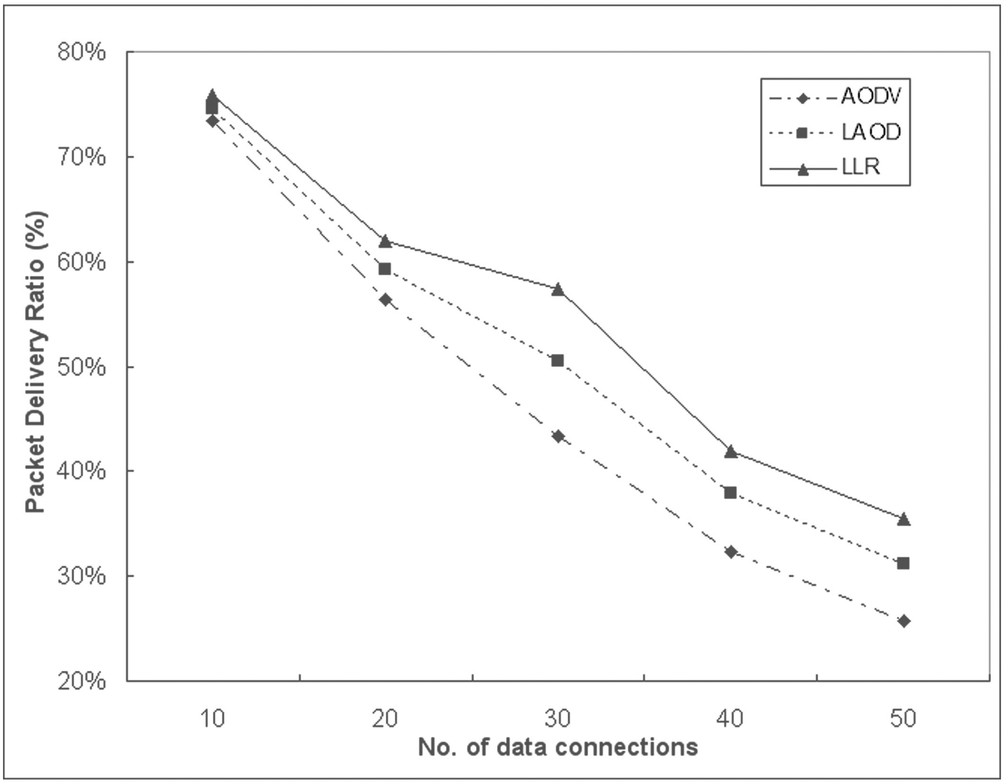


Figure 6.8: Packet Delivery Ratio (PDR) between LLR, LAOD and AODV using *Manhattan Grid* mobility model with different number of CBR connections

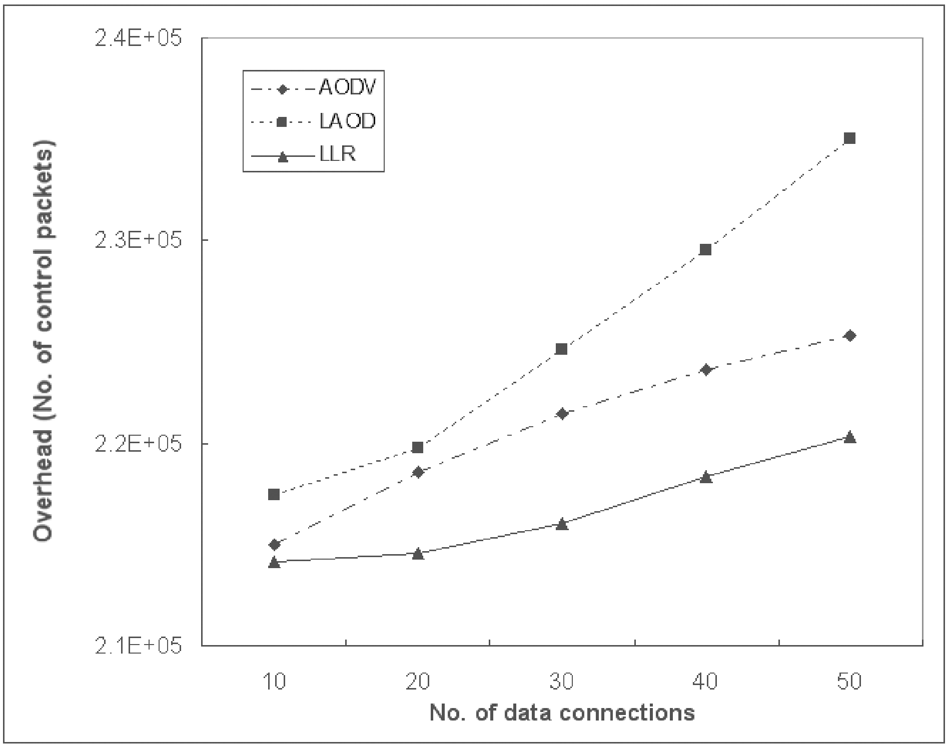


Figure 6.9: Overhead between LLR, LAOD and AODV using *Manhattan Grid* mobility model with different number of CBR connections

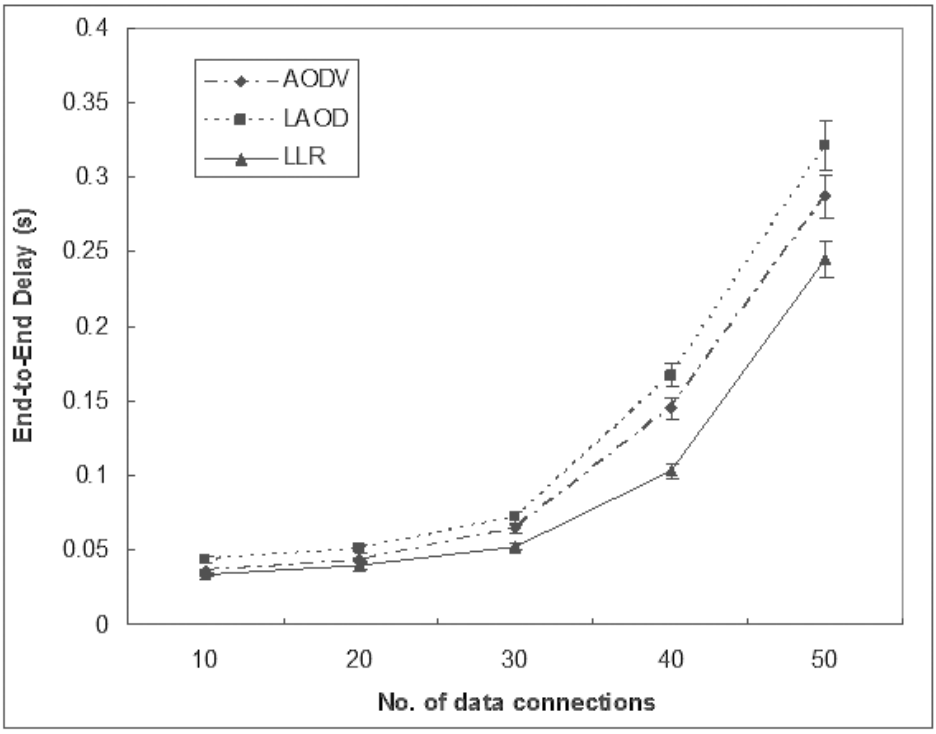


Figure 6.10: End-to-End Delay between LLR, LAOD and AODV using *Manhattan Grid* mobility model with different number of CBR connections

6.2.2 Simulation Results and Discussion (*Graph-based* mobility model)

6.2.2.1 Varying Speed of MNs

The *Graph-based* mobility model is used and the results are obtained by varying the mean speed of MNs from 1m/s to 20 m/s (i.e., the mobility of MNs), with the number of CBR connections fixed at 20. This means the network load for this set of simulations is fixed at 20 CBR connections, which is considered as medium loading for the network.

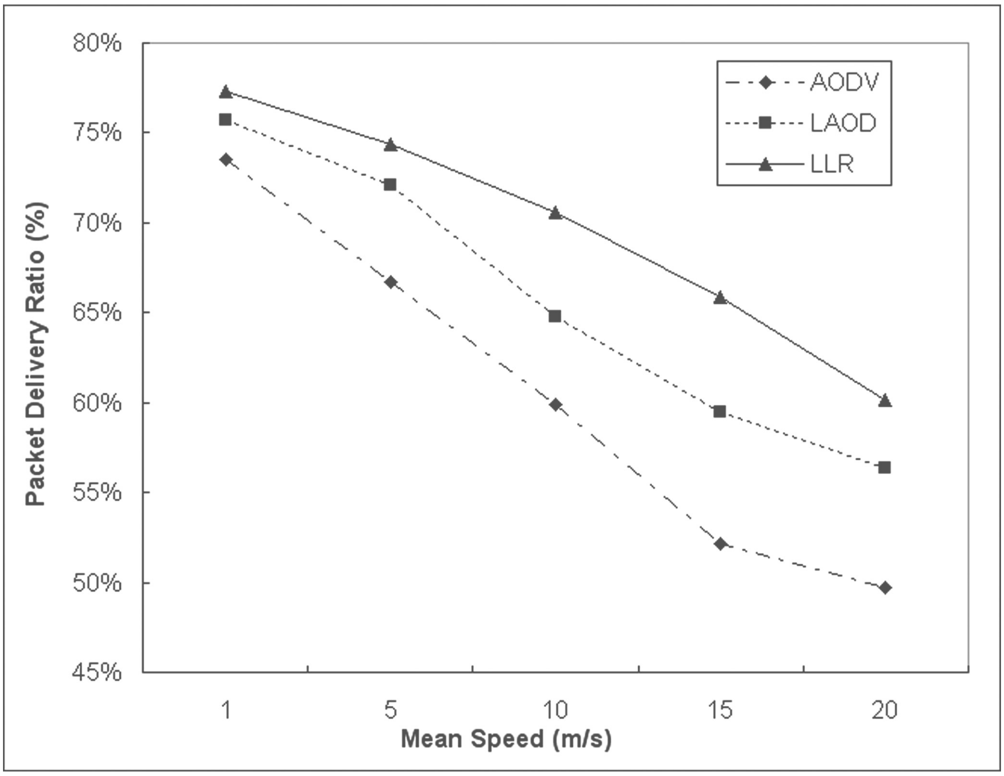


Figure 6.11: Packet Delivery Ratio (PDR) between LLR, LAOD and AODV using *Graph-based* mobility model with different mobility speed

Figures 6.11-6.14 show the performance in the *Graph-based* mobility model with varying speed. Like the case of *Manhattan Grid* mobility model, there are marked improvements by LLR in PDR, end-to-end delay, and the routing control packets being transmitted in the network. LAOD also achieves improvement on PDR compared with AODV, but at the expense of higher overhead and end-to-end delay. The reason is the same as that in *Manhattan Grid* mobility model.

In Figure 6.13, it can be observed that AODV and LAOD perform similarly at 10 m/s and above. At faster MN speeds, there are more route changes, which results in more overhead for both AODV and LAOD. However, the rate at which overhead in LAOD increases is a bit faster than that of AODV. This is due to the frequent occurrences of the *Local Maximum* problem, which is partly overcome when we revamped our design and developed LLR.

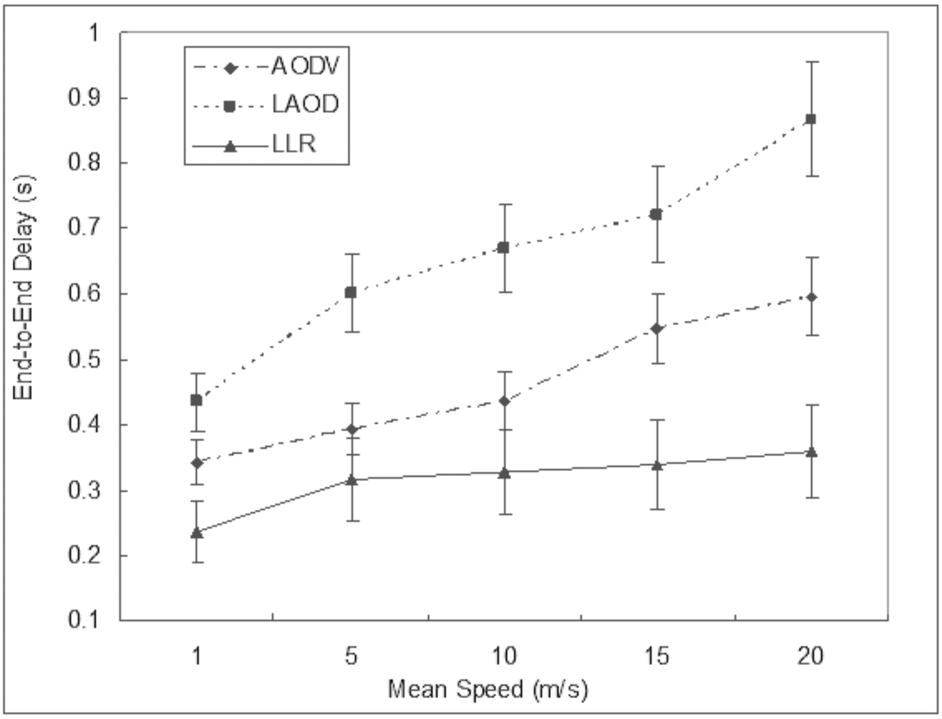


Figure 6.12: End-to-End Delay between LLR, LAOD and AODV using *Graph-based* mobility model with different mobility speed

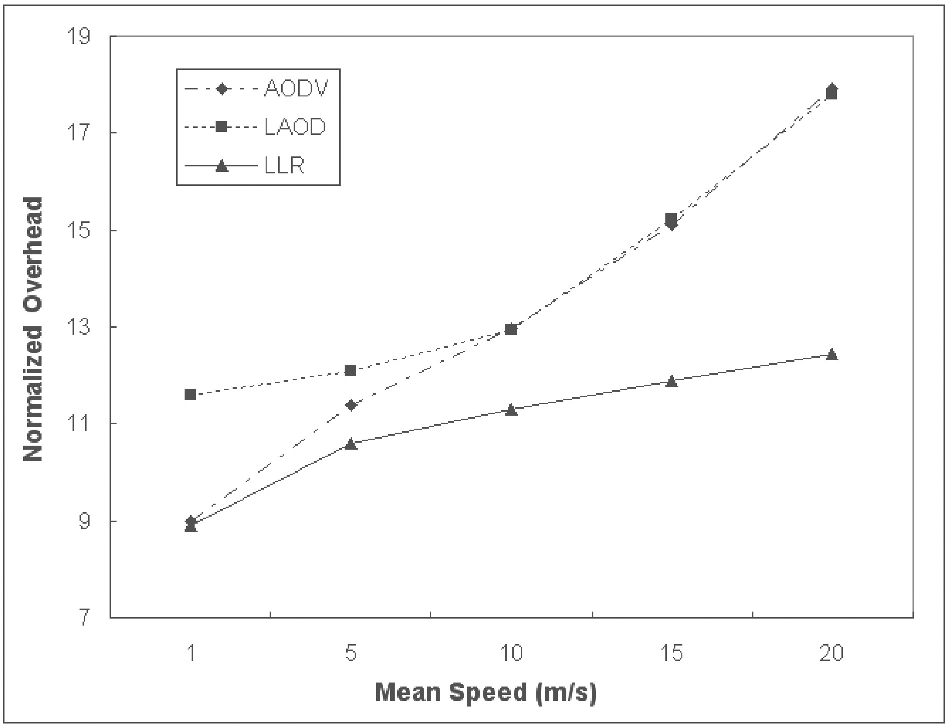


Figure 6.13: Normalized Overhead between LLR, LAOD and AODV using *Graph-based* mobility model with different mobility speed

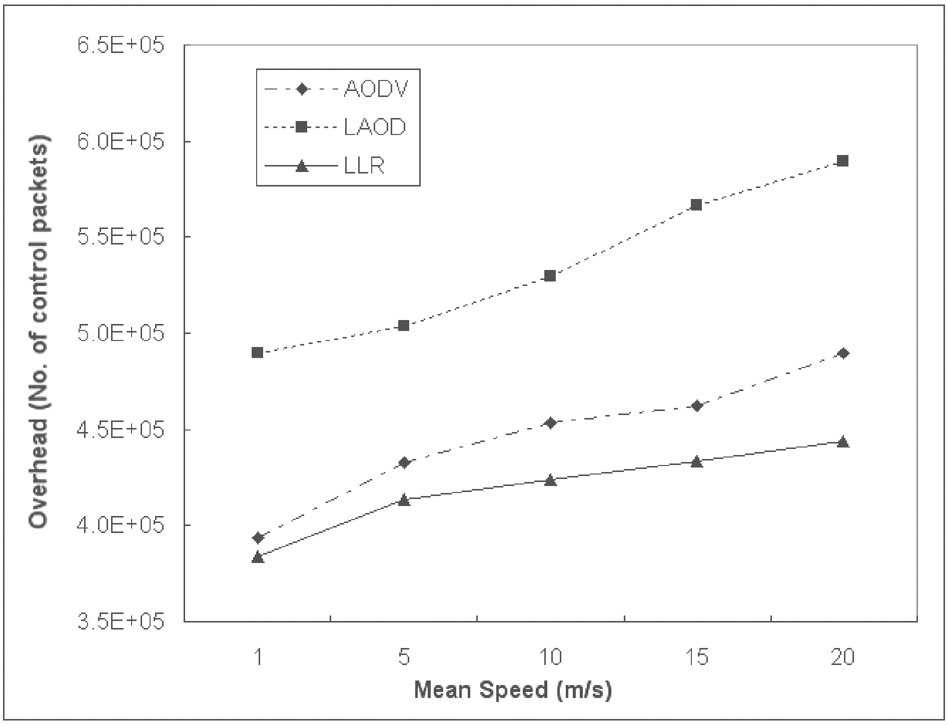


Figure 6.14: Overhead between LLR, LAOD and AODV using *Graph-based* mobility model with different mobility speed

6.2.2.2 *Varying Number of CBR Connections*

Here, the results are obtained by varying number of CBR connections in the network from 10 to 50, with the mean speed of the MNs fixed at 20 m/s.

Figures 6.15-6.17 show the performance in the *Graph-based* mobility model with varying traffic loads. We see that, like before, the same trend as those which were obtained earlier using *Manhattan Grid* mobility model with varying traffic loads. LLR is the best among the three routing protocols with highest PDR, lowest overhead, and lowest end-to-end delay. LAOD also achieves improvement on PDR compared to AODV, but at the expense of higher overhead and end-to-end delay.

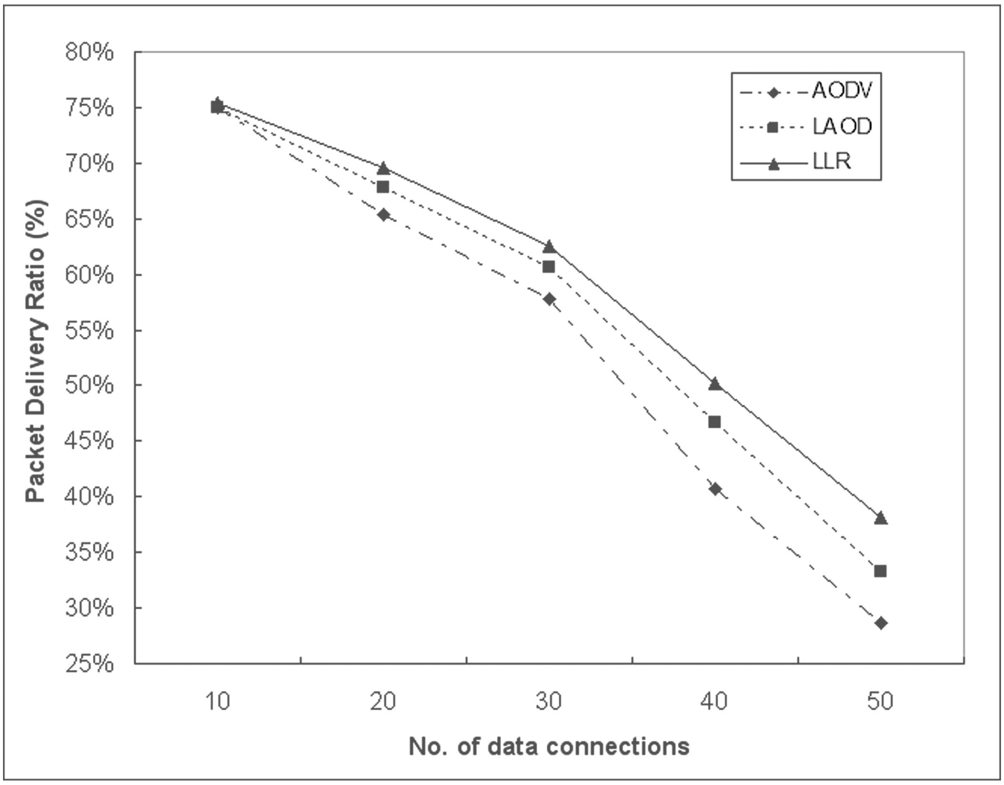


Figure 6.15: Packet Delivery Ratio (PDR) between LLR, LAOD and AODV using *Graph-based* mobility model with different number of CBR connections

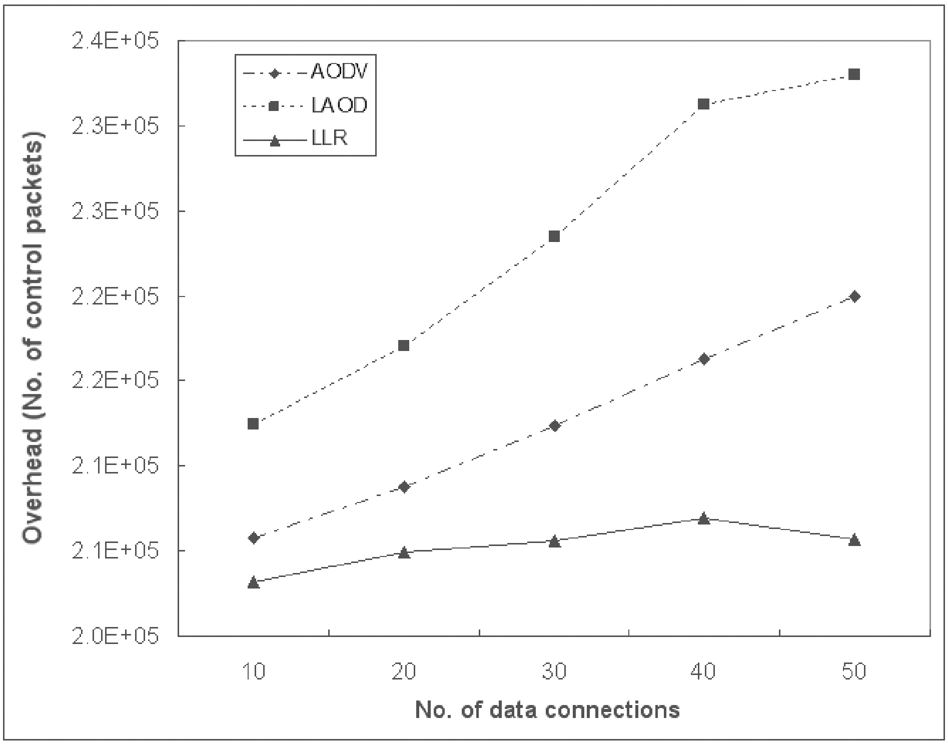


Figure 6.16: Overhead between LLR, LAOD and AODV using *Graph-based* mobility model with different number of CBR connections

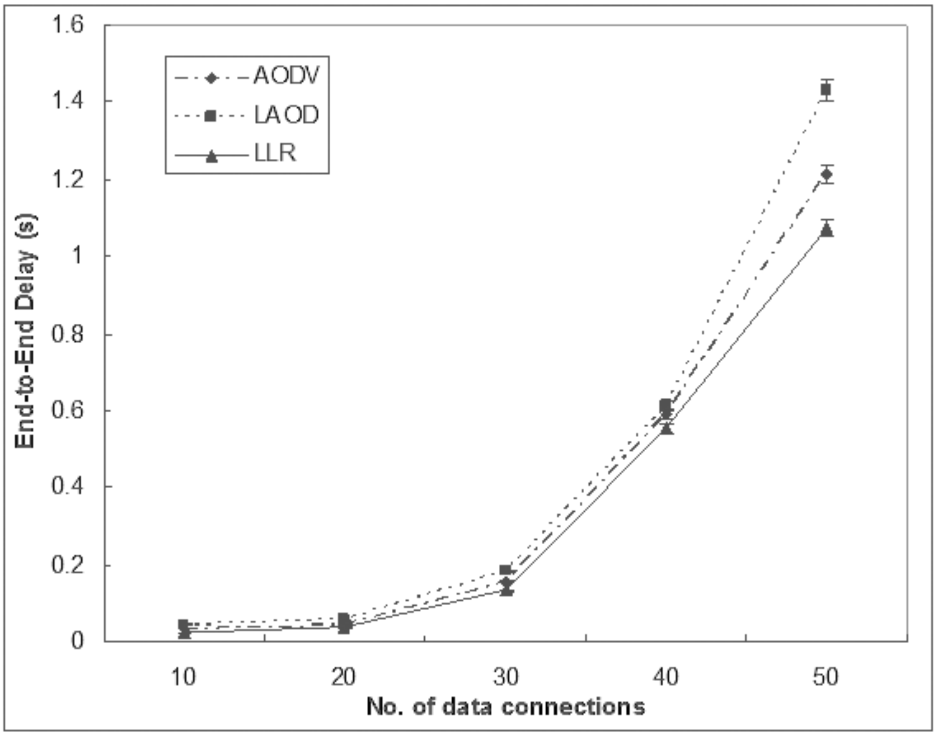


Figure 6.17: End-to-End Delay between LLR, LAOD and AODV using *Graph-based* mobility model with different number of CBR connections

6.2.2.3 Summary

From the results of Simulation Set I, we can observe the similarity of the results using the two mobility models, namely *Graph-based* and *Manhattan Grid* mobility model. The trends of the results (e.g. PDR, overhead, end-to-end delay, and delay jitter) are almost exactly the same. This is probably because of our simulation parameter settings. Therefore, we only present the simulation results using *Graph-based* mobility model in the following sections, since the simulation results we have using *Manhattan Grid* mobility model are consistent with the ones using *Graph-based* mobility model.

6.3 Simulations Set II

In this section, the performances of LLR and its various Hello message adjustment schemes are examined. Listed below are some parameter settings that we use in the simulations:

- *LOW_PERCENTAGE_THRESH=20%*
- *HIGH_PERCENTAGE_THRESH=50%*
- *LOW_SPEED_THRESH=5m/s*
- *HIGH_SPEED_THRESH=15m/s*
- *LOW_ABSOLUTE_MOBILITY_FRAC=1.05*
- *HIGH_ABSOLUTE_MOBILITY_FRAC=0.95*

Three different Hello message adjustment schemes with different relative mobility adjustment parameter are studied here, which are shown in Table 6.3.

Table 6.3: Different Hello message adjustment schemes' settings

Parameter	**Hello I**	**Hello II**	**Hello III**
LOW_RELATIVE_MOBILITY_FRAC	1.25	1.50	1.75
HIGH_RELATIVE_MOBILITY_FRAC	0.75	0.50	0.25

Only *Graph-based* mobility models is used here. Besides the key simulation parameters, as shown in Table 6.1, other simulations parameters are listed in Table 6.4:

Table 6.4: Parameters used during simulation set II

Parameter	**Value**	**Explanation**
Max Pause Time	20 s	The maximum time interval the MN will stay after reaching the destination but before heading towards the new destination
Number of GWs	4	The location of GWs are shown in the Figure 6.2
Advertisement Zone	6 hops	The size of *K-hop* subnet, i.e. *K*=6
Advertisement Period	5 s	The period of sending gateway advertisement message by GW is 5s, refer to Chapter 3
Periodical Update Interval	5 s	The period of sending location update message by MN is 5s, refer to Chapter 3

The purpose of the Hello message adjustment algorithm is to investigate how to effectively increase data traffic into the network, (i.e. increase the network throughput), and achieve similar routing performance compared to the one without the Hello message adjustment algorithm. However, the network density affects the Hello message adjustment. Furthermore, the nature of the MN mobility model, which has close correlation between MNs, also affects the Hello message adjustment. In our simulations, the MNs in the network always have sufficient neighbors, in a slowly-changing neighborhood. The simulation results in this section are consistent with these in Section 6.2. However, we are interested to investigate the performance of different Hello message adjustment schemes, particularly in their ability to save power consumption for the whole network as the result of reduced packet transmissions, which is demonstrated later.

6.3.1 Simulation Results and Discussion (*Graph-based* mobility model)

6.3.1.1 Varying Speed of MNs

The *Graph-based* mobility model is used here. The results are obtained by varying the mean speed of MNs from 1m/s to 20 m/s (i.e., the mobility of MNs), with the number of CBR connections fixed at 20, which is considered as medium load for the network.

Figure 6.18 shows a significantly lower number of Hello packet transmissions in

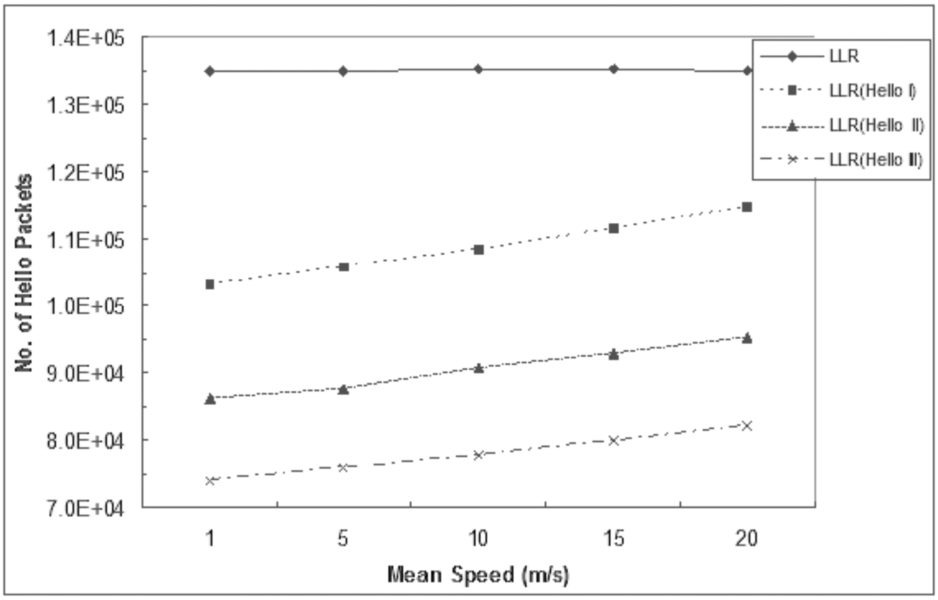


Figure 6.18: Hello Overhead comparison between LLR and its variants of Hello message adjustment schemes using *Graph-based* mobility model with different mobility speed

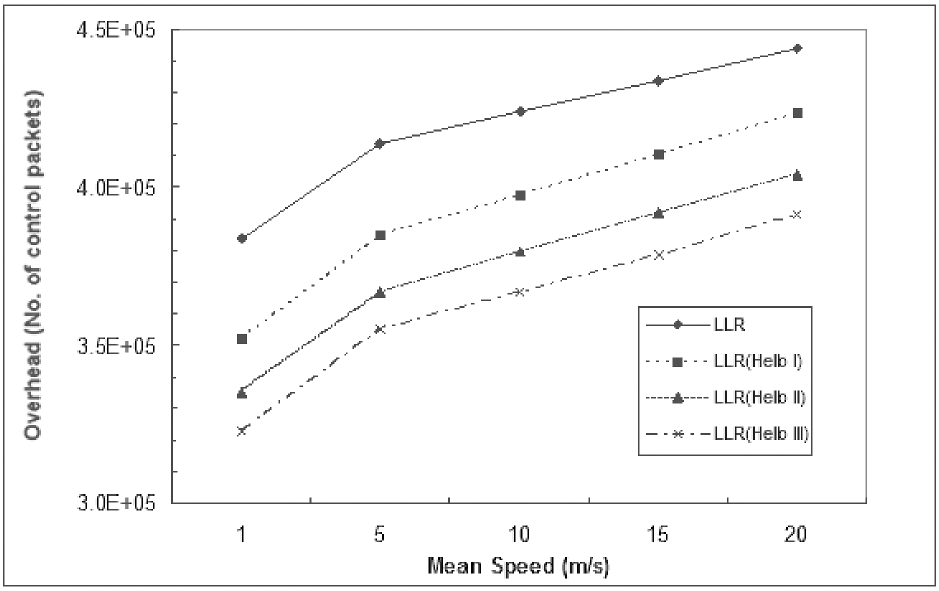


Figure 6.19: Overhead comparison between LLR and its variants of Hello message adjustment schemes using *Graph-based* mobility model with different mobility speed

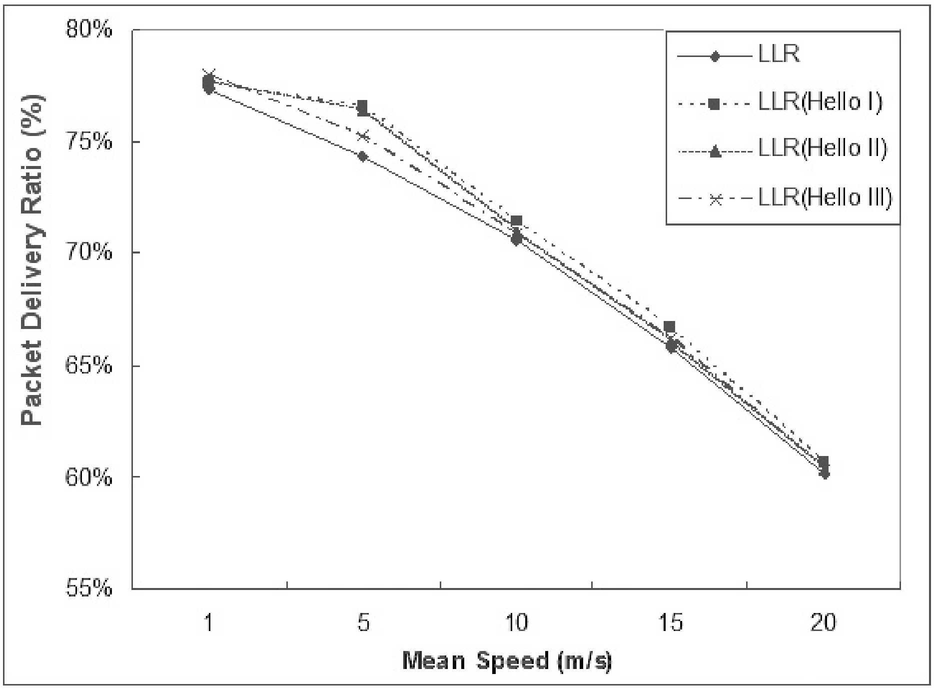


Figure 6.20: Packet Delivery Ratio (PDR) comparison between LLR and its variants of Hello message adjustment schemes using *Graph-based* mobility model with different mobility speed

the network by the different Hello message adjustment schemes. Compared to LLR, LLR (HelloI) broadcasts around 80% of Hello messages, LLR (HelloII) broadcasts around 70% of Hello messages, and LLR (HelloIII) only broadcasts around 60% of Hello messages at all speeds. The effect of reduced Hello packet broadcasting is shown in Figure 6.19, which shows the reduction on the overhead in the network for the different Hello adjustment message schemes. This causes less contention for bandwidth with data packets and important routing control packets like RREQ and RREP, which in turn, helps to reduce the probability of packet collision and leads to less MAC backoff time. Therefore, the PDR increases as shown in Figure 6.20, while end-to-end delay reduces also drops as shown in Figure 6.21.

The reduction in Hello message transmissions helps to achieve quite significant power saving for the entire network. Let us assume each control packet is sent by consuming exactly the same power. This is reasonable since RREQ, RREP, Hello message and so on are very small packet with similar size. Figure 6.22 shows the percentage

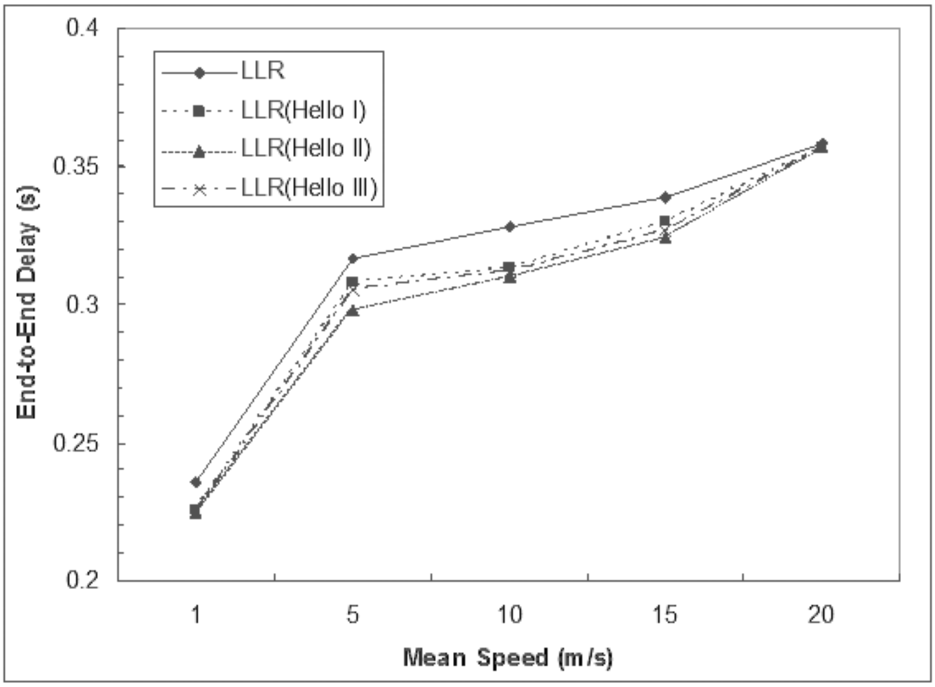


Figure 6.21: End-to-End Delay comparison between LLR and its variants of Hello message adjustment schemes using *Graph-based* mobility model with different mobility speed

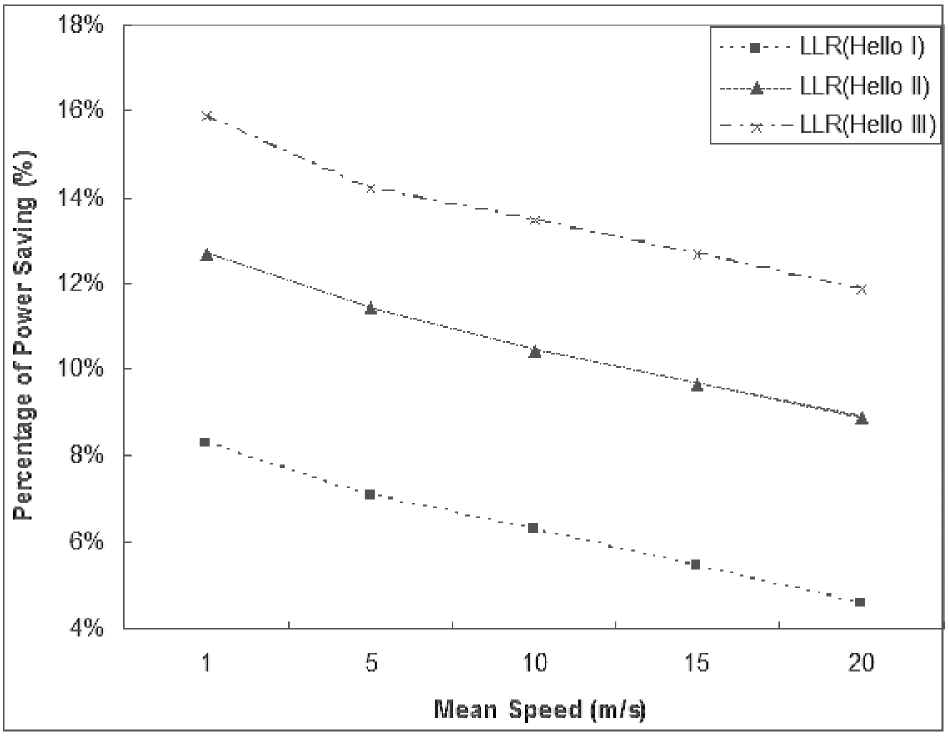


Figure 6.22: Percentage of Power Saving comparison between variants of Hello message adjustment schemes using *Graph-based* mobility model with different mobility speed

power consumption saving of different Hello message adjustment schemes with respect to the original LLR. As we can observe, all the Hello message adjustment schemes achieve more than 4% power consumption saving for the whole network. LLR (HelloIII) is the best, which achieves around 16% power saving for lower speed and 12% for higher speed. This is quite a significant improvement by simply adjust the broadcastings of the Hello message with respect to the network conditions, especially when the network is very large.

6.3.1.2 Varying Number of CBR Connections

Here, the results are obtained by varying the number of CBR connections in the network from 10 to 50, with the mean speed of the MNs fixed at 20 m/s.

Similar to pervious results, Figures 6.23-6.26 show that the routing performances of different Hello message adjustment schemes are better than the original one without it. Similar to previous results, Figure 6.26 shows a significantly lower number of Hello packet transmission in the network resulting from the different Hello message adjustment schemes. The effect of reduced Hello packet broadcasting results in an improvement of routing performance in terms of reduced routing control packets as shown in Figure 6.25, increased PDR as shown in Figure 6.23, and reduced end-to-end delay, as shown in Figure 6.24. Although, the improvements on PDR and end-to-end delay are not very significant, the reduction in the overhead means that more data traffic can be supported by the network. Furthermore, under heavy traffic loads, the overall routing performance are degraded for all schemes. As we can observe, only around 40% data packets are delivered to the destination at 50 data connections compared to 75% at 10 data connections, which is around 50% decrement. The end-to-end delay is more than 10 times at 50 data connections than that at 10 data connections. This is due to the excessive contention of the bandwidth in the network. Hence, the need to reduce unnecessary broadcasting of Hello packets is even more important at high data loads in the network.

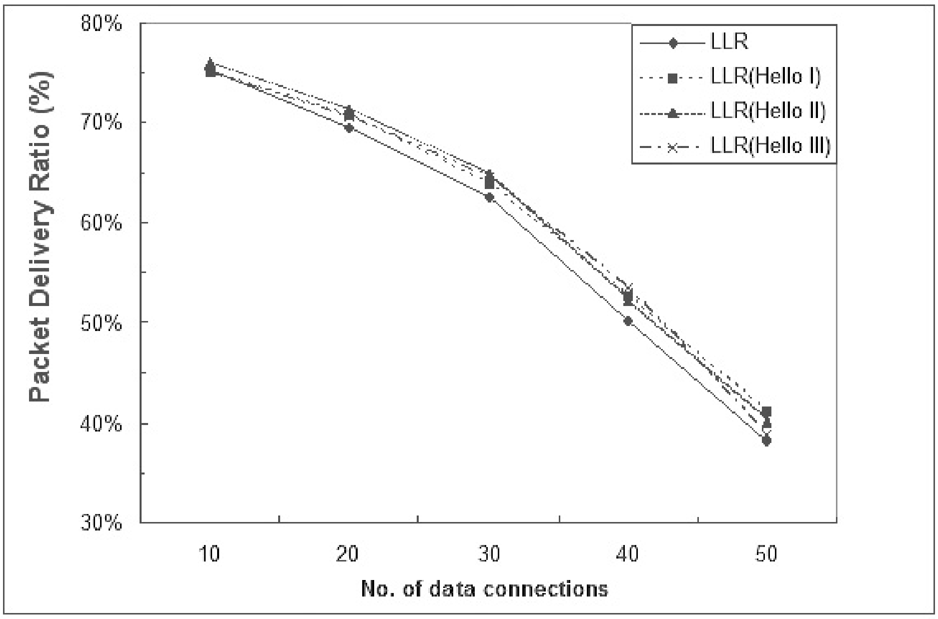


Figure 6.23: Packet Delivery Ratio (PDR) comparison between LLR and its variants of Hello message adjustment schemes using *Graph-based* mobility model with different number of CBR connections

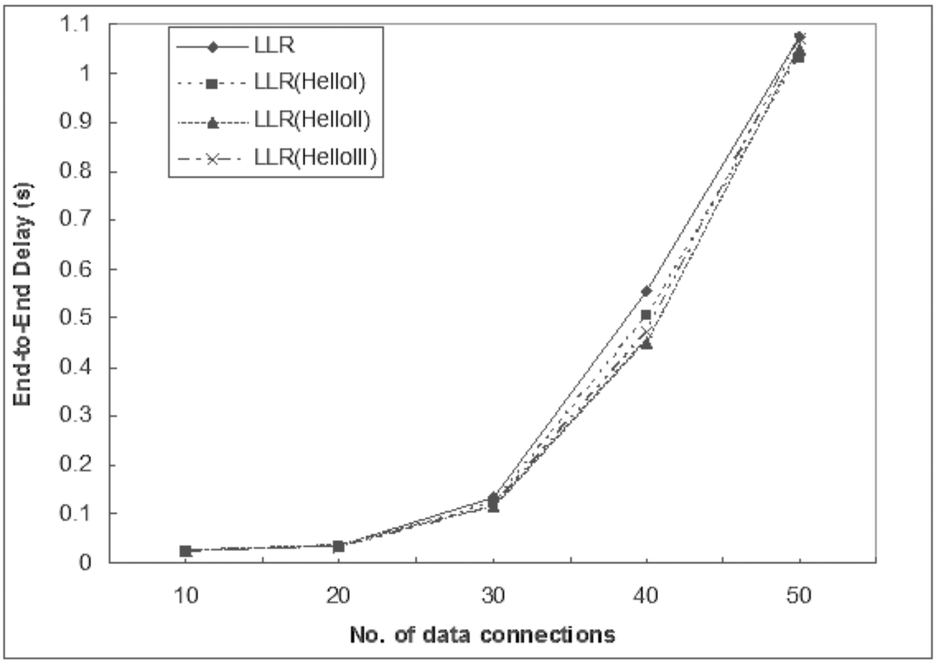


Figure 6.24: End-to-End Delay comparison between LLR and its variants of Hello message adjustment schemes using *Graph-based* mobility model with different number of CBR connections

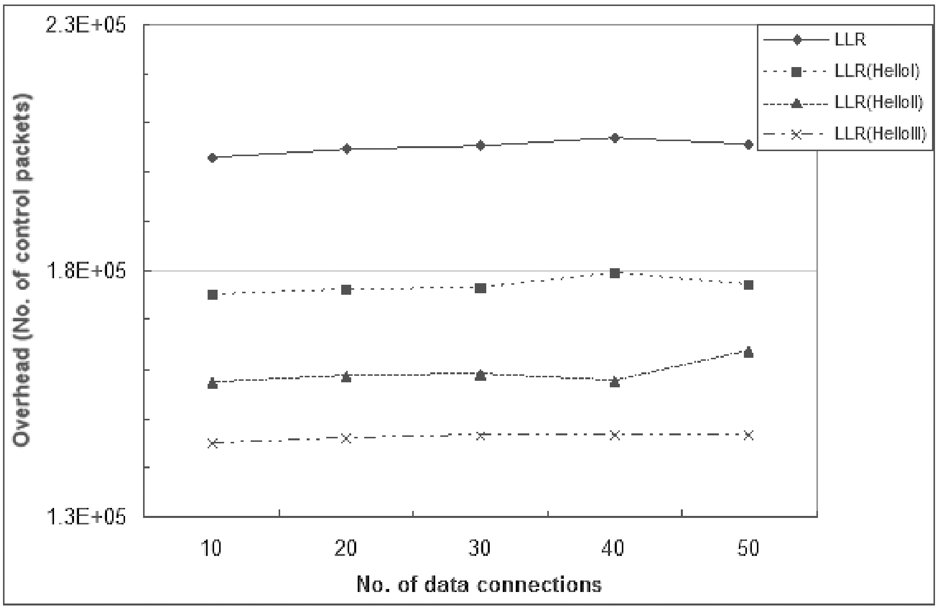


Figure 6.25: Overhead comparison between LLR and its variants of Hello message adjustment schemes using *Graph-based* mobility model with different number of CBR connections

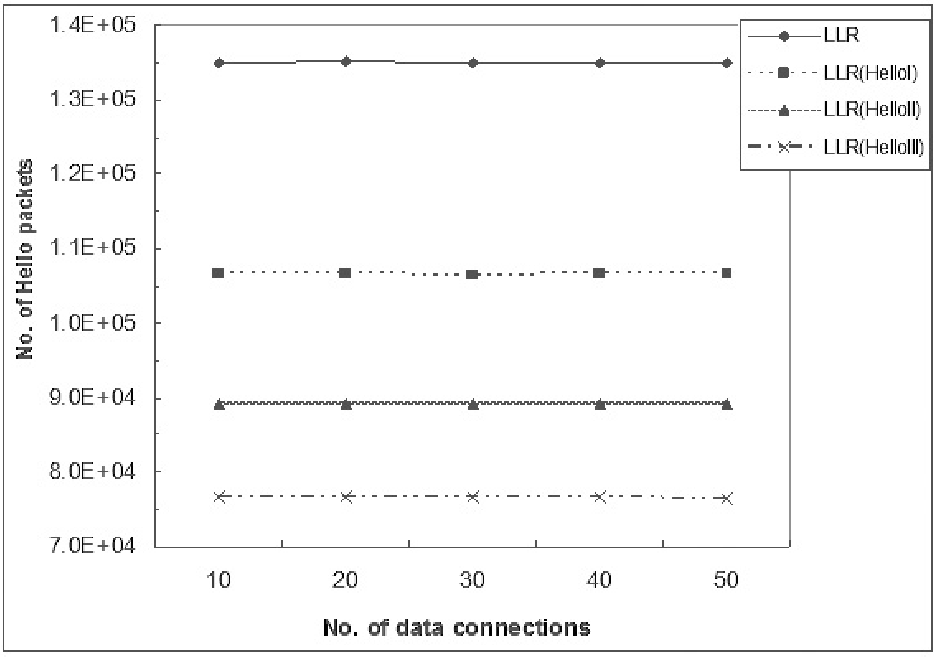


Figure 6.26: Hello Overhead comparison between LLR and its variants of Hello message adjustment schemes using *Graph-based* mobility model with different number of CBR connections

6.3.2 Simulation Results and Discussion for Different Relative Mobility Threshold Settings

In our simulations and discussions for the Hello message adjustment schemes of Section 6.3.1, we fix the relative mobility threshold, which are *HIGH_PERCENTAGE_THRESH=50%* and *LOW_PERCENTAGE_THRESH=20%*. The choice of an appropriate pair of threshold setting is dependent on network conditions like mobility pattern, network traffic load, and so on. We therefore carry out some simulation studies on them to get the optimal values of these two threshold settings for our specific network model. We choose to use the Hello message adjustment scheme with:

- *HIGH_RELATIVE_MOBILITY_FRAC=0.50*
- *LOW_RELATIVE_MOBILITY_FRAC=1.50*

The simulations are run using the *Graph-based* mobility model with different number of CBR connections. We vary the high and low relative mobility threshold settings to obtain different combinations, as shown in Table 6.5.

Table 6.5: Different combinations of high and low relative mobility threshold settings using in the simulations

Combinations	**HIGH_ PERCENTAGE_ THRESH**	**LOW_ PERCENTAGE_ THRESH**
Threshold (H>50%, L<20%)	50%	20%
Threshold (H>50%, L<0%)	50%	0%
Threshold (H>50%, L<10%)	50%	10%
Threshold (H>50%, L<30%)	50%	30%
Threshold (H>50%, L<50%)	50%	50%
Threshold (H>20%, L<20%)	20%	20%
Threshold (H>40%, L<20%)	40%	20%
Threshold (H>70%, L<20%)	70%	20%
Threshold (H>100%, L<20%)	100%	20%

From the simulation results shown in Figures 6.27-6.29, it can be observed that results from *Threshold (H>50%, L<30%)*, *Threshold (H>50%, L<50%)*, *Threshold (H>20%, L<20%)* ,*Threshold (H>40%, L<20%)*, and *Threshold (H>70%, L<20%)*

are identical with result from *Threshold (H>50%, L<20%)*. In other words, only settings with *Threshold (H>50%, L<0%)*, *Threshold (H>50%, L<10%)*, and *Threshold (H>100%, L<20%)* are different from *Threshold (H>50%, L<20%)*. This is because during the simulations, most of the MNs have less than 20% of neighbors changing across consecutive time instance. The typical value observed during the simulations is 0%. In other words, the network scenarios used are quite stable with minimal abrupt changes of MNs' speed or moving direction. Thus, the neighboring MNs of a particular MN remain the same for quite a long period.

The results from *Threshold (H>100%, L<20%)* is only slightly different from *Threshold (H>50%, L<20%)*. We observe in the simulations that the MNs do not have more than 20% of neighbors changing except at the start of the simulation, when they are just starting to know their neighbors after the Hello message broadcasting begins. We also observe some MNs have 100% changing in the middle of the simulations. This is because these are previously isolated MNs, which move out and get connected with others sometime during the simulation. However, these are very rare cases. Therefore, there is only a slight difference from *Threshold (H>100%, L<20%)* and *Threshold (H>50%, L<20%)*.

It can be seen that the results for different settings are very close, with *Threshold (H>50%, L<20%)* having highest PDR, lowest end-to-end delay and lowest number of Hello packet transmission. Therefore, it can be said that *Threshold (H>50%, L<20%)* is the optimal setting. However, this optimal setting is only applied to these network scenarios we used. For other network scenarios, this may not be true.

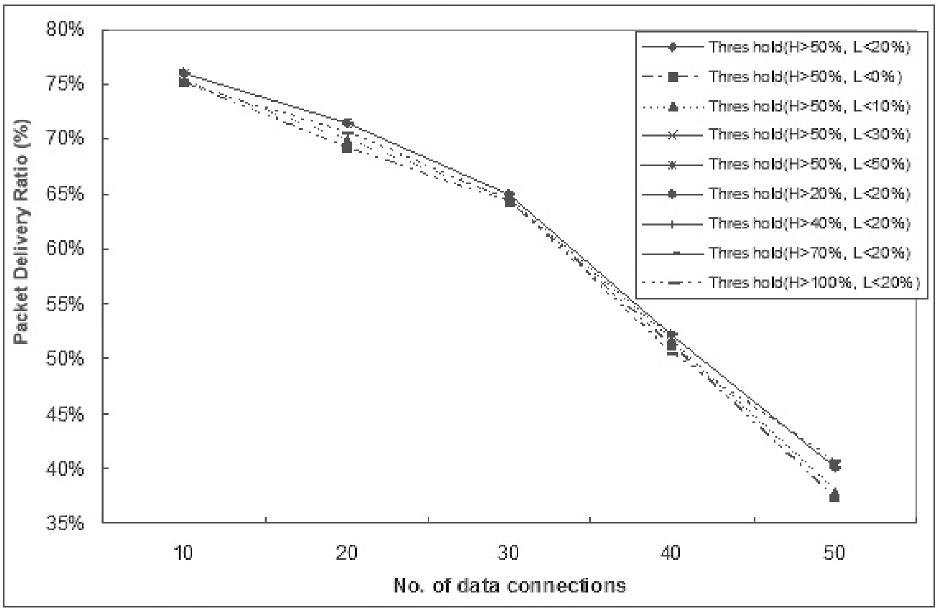


Figure 6.27: Packet Delivery Ratio (PDR) comparison between different relative mobility threshold settings of Hello message adjustment scheme using *Graph-based* mobility model with different number of CBR connections

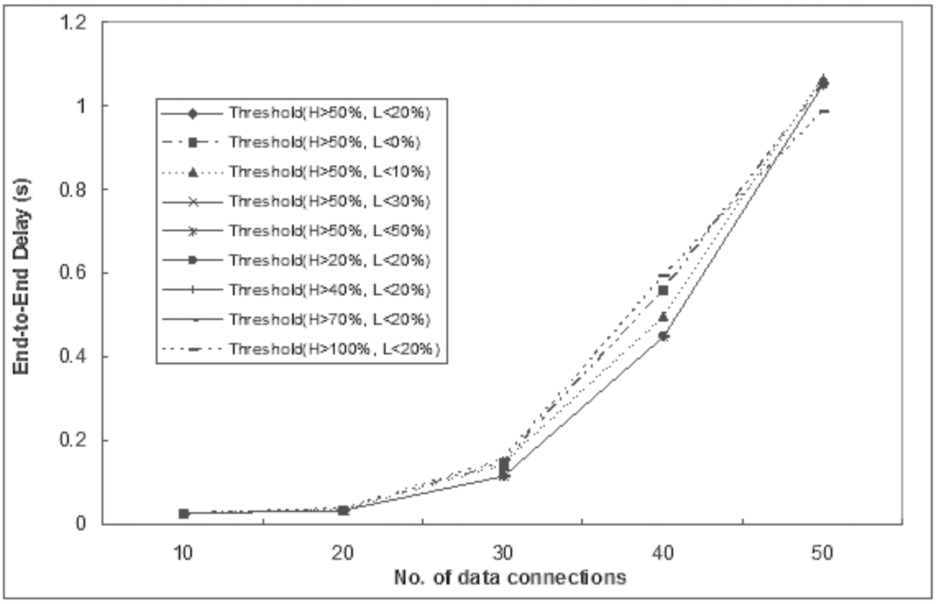


Figure 6.28: End-to-End Delay comparison between different relative mobility threshold settings of Hello message adjustment scheme using *Graph-based* mobility model with different number of CBR connections

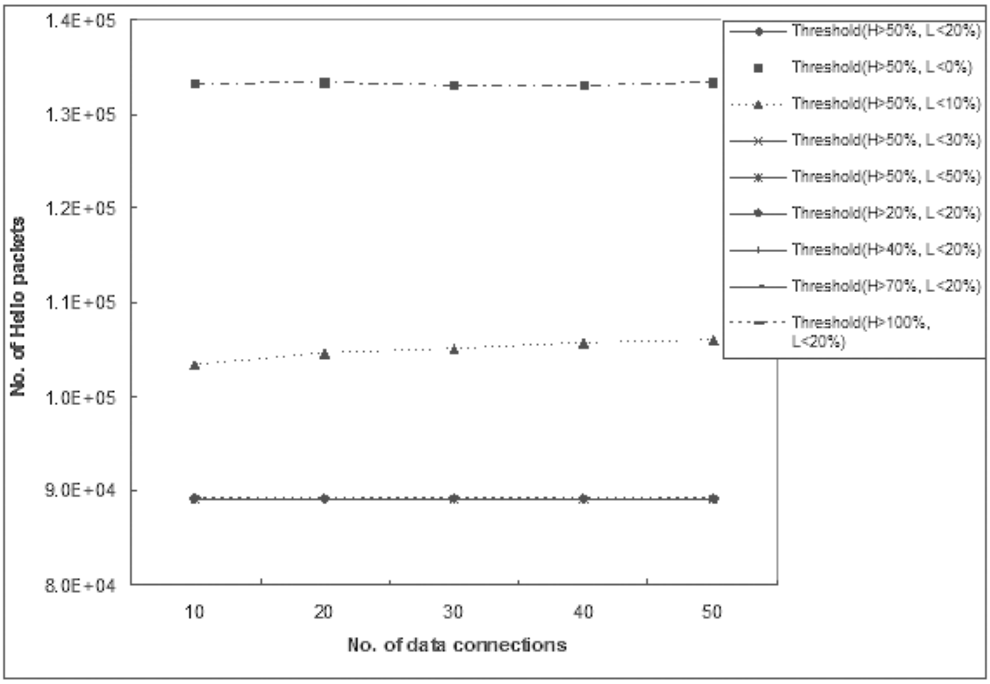


Figure 6.29: Hello Overhead comparison between different relative mobility threshold settings of Hello message adjustment scheme using *Graph-based* mobility model with different number of CBR connections

6.4 Simulations Set III

In this section, we demonstrate the routing performance of AODV, LAOD, LLR and LLR(Hello III) under very high network data traffic loads.

In our previous simulations, the results show that our routing protocols work fine under medium data traffic loads in the network. We would like to see how the protocols perform under very high data traffic loads in the network. AODV is used as the reference for comparison. Both LAOD and LLR are used to evaluate the routing performance. Furthermore, we also use one of the Hello message adjustment schemes, namely LLR(Hello III), which has the lowest overhead. The parameter setting of LLR(Hello III) is the same as in Simulation Set II of Section 6.3.

We still use CBR as the data traffic loads in the network. Each CBR flow sends data at 10 packets per second with packet size of 512 bytes. In the simulations, the number of CBR flows in the network varies from 100 to 140, with the number of MNs fixed at 150. In other words, there are always at least two-thirds of MNs generating 40.96 kbps data traffic in the network. The *Graph-based* mobility model and the same graph as shown in Figure 6.2 are used in the simulations. In summary, the simulation settings are shown in Table 6.6 below.

Table 6.6: Parameters used in the study of very high data traffic loads

Parameter	**Value**
Data traffic	Constant Bit Rate(CBR)
Packet Rate	10 packets/s
Packet Size	512 bytes
Number of CBRs	from 100 to 140
Mobility Model	*Graph-based* mobility model, as shown in Figure 6.2
Number of MNs	150
Simulation Time	900 s

Figures 6.30-6.32 shows the results obtained from simulations. As can be seen, under heavy data traffic loads, the routing performances of all the routing protocols are very bad. None of them achieve more than 2% of PDR, as shown in Figure 6.30. This is due to the excessive bandwidth contention in the network. This leads to extremely high number of collisions, low throughput and low PDR. This is also the cause of the extremely long end-to-end delay, as shown in Figure 6.31. Compared to the rest, LLR(Hello III) has the best routing performance among them, which is because of the reduction in the unnecessary overhead, the Hello messages, as shown in Figure 6.32. By reducing the Hello messages in the network, the control traffic is reduced, as shown in Figure 6.33. This helps to reduce a bit of contention for the bandwidth with the data packets and helps to achieve a bit better overall utilization of the network resources.

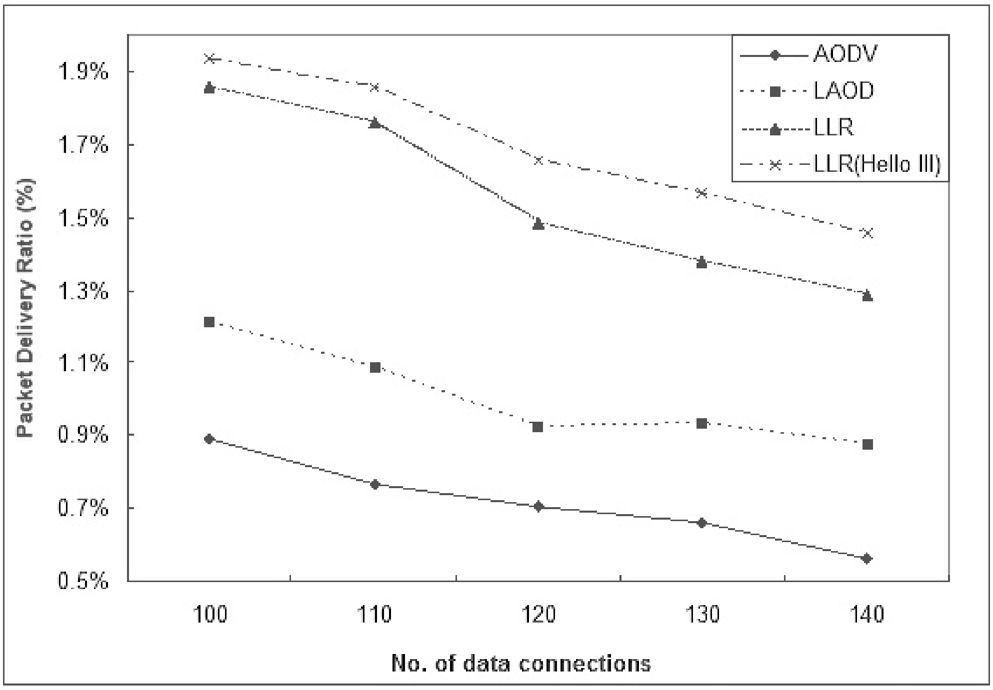


Figure 6.30: Packet Delivery Ratio (PDR) comparison between AODV, LAOD, LLR and LLR(Hello III) using *Graph-based* mobility model with very high network data loadings

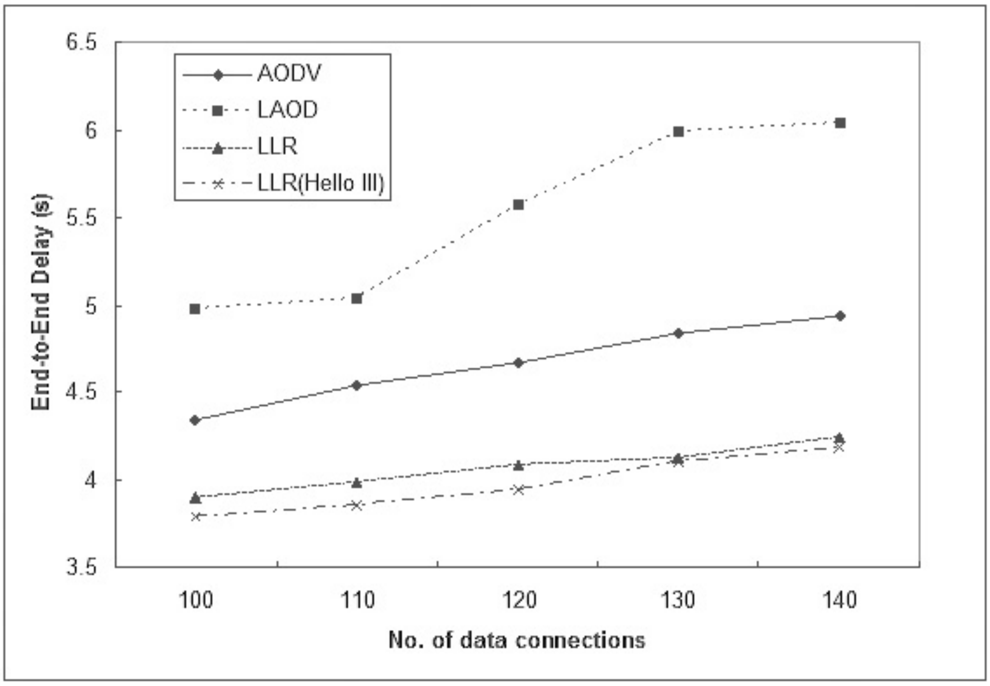


Figure 6.31: End-to-End Delay comparison between AODV, LAOD, LLR and LLR(Hello III) using *Graph-based* mobility model with very high network data loadings

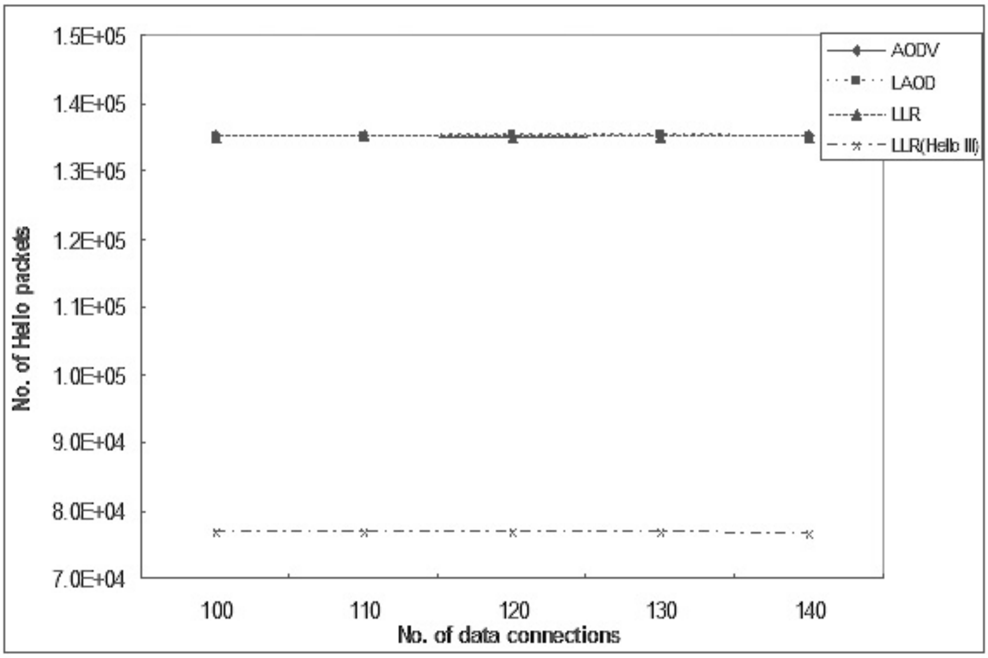


Figure 6.32: Hello Overhead comparison between AODV, LAOD, LLR and LLR(Hello III) using *Graph-based* mobility model with very high network data loadings

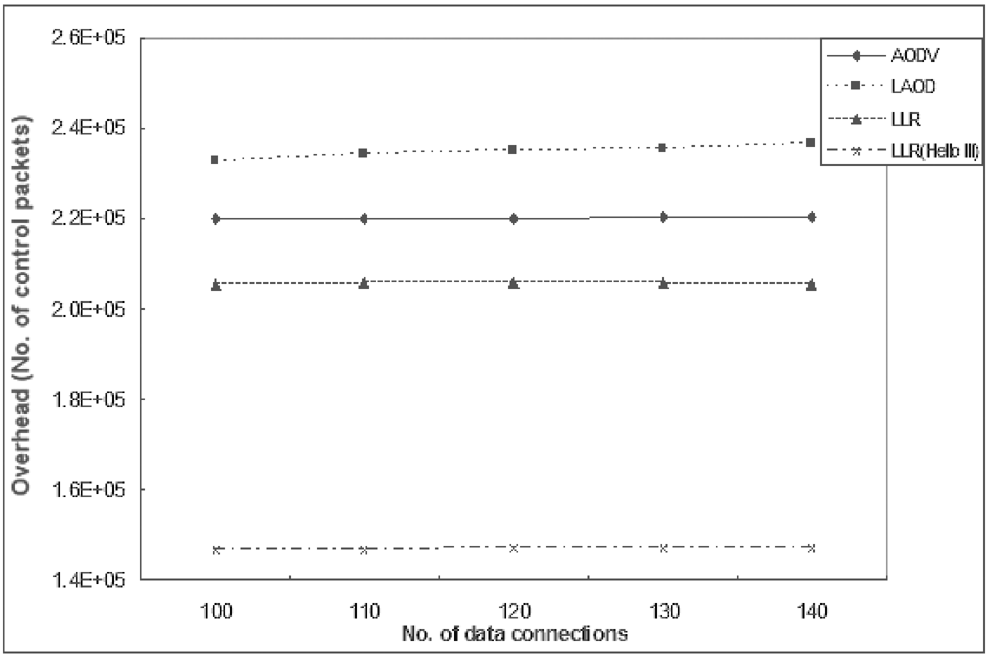


Figure 6.33: Overhead comparison between AODV, LAOD, LLR and LLR(Hello III) using *Graph-based* mobility model with very high network data loadings

6.5 Summary

In this chapter, we present our simulation results in three different sets. In Simulations Set I, we compared the routing performance between AODV, LAOD and LLR. LLR has overall best routing performance among the three routing protocols, with the highest PDR, the shortest end-to-end delay and the lowest overhead, especially at high mobility speed. In Simulations Set II, we showed that we are able to further improve routing performance and reduce power consumption, by dynamically adjusting the Hello broadcasting interval with respect to the network topology. In Simulations Set III, we demonstrated the performance under very high network data loading comparison between AODV, LAOD, LLR and LLR (Hello III). It can be observed from the results that LLR (Hello III) is best among the four routing protocols under such a heavy loaded network.

Compared to AODV, the performance improvement of LLR is quite significant, especially at high mobility speed. We usually can observe that LLR achieves around 10% more PDR, 15% less end-to-end delay, and 5% less overhead at the mean speed around $20m/s$.

We also found that the performance of proposed routing protocols are not very sensitive to mobility models under our simulation parameter settings. As we can see from the simulation results in Simulations Set I, the trends of results by using *Manhattan Grid* and *Graph-based* mobility model are very similar.

CHAPTER VII

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

Two location-aided routing protocols, namely Location-Aided On-Demand (LAOD) routing protocol and Link-Connectivity-Prediction-Based Location-Aided Routing (LLR) protocol, together with the supporting gateway discovery algorithm are presented. Both of these two routing protocols make use of location information to achieve better routing performance. We use simulation to verify the correctness of LAOD and LLR and compared with AODV, which does not use location information.

In our simulation study, compared to AODV, we find that LAOD has achieved higher packet delivery ratio (PDR) (i.e. more data packets are delivered) at the expense of longer end-to-end delay and more overhead. By incorporating the greedy packet forwarding mechanism into the on-demand routing protocol, LAOD is able to deliver slightly more data packets to the destination. However, LAOD does not provide any recovery technique when the greedy packet forwarding mechanism encounters the *Local Maximum* problem. This causes LAOD to have longer end-to-end delay. Furthermore, LAOD needs to use more control messages in order to work properly, which causes it to have more overhead. Therefore, the design of LAOD needs to be reconsidered. For example, a recovery technique should be provided when the greedy packet forwarding mechanism fails.

We also find that LLR has overall best routing performance among AODV, LAOD and LLR. From the simulation results, LLR has the highest PDR, the shortest end-to-end delay and the lowest overhead among these three routing protocols. By using the location information to predict link connectivity, LLR reduces the route (re)discovery latency by performing rerouting prior to route disconnections. This helps to reduce end-to-end delay. LLR also tries to restrict the broadcasting of the control messages during route recovery, which helps to reduce overhead. Furthermore, LLR tries to use shorter and more stable routing path, switching between the WR and the WWR accordingly. Consequently, these lead to lower network congestion, lower bandwidth contention between control messages and data packets, lower packet loss ratio and higher PDR. All these are crucial performance metrics in the hybrid network environment.

Furthermore, a Hello message adjustment algorithm incorporated with LLR has been proposed. By dynamically adjusting the Hello broadcasting interval with respect to the network topology, we are able to further improve routing performance and reduce power consumption. In general, the mobility of a MN is characterized by the rate of change of neighbor MNs and its moving speed. By varying the Hello message interval with respect to the mobility of MN, unnecessary broadcasting of Hello messages can be reduced. This has a number of desirable side effects, which includes less contention for bandwidth with the data packets that leads to higher PDR, decreased end-to-end delay and reduction of overhead. Furthermore, the simulation results have also shown that the power consumption decreased significantly as less transmissions of Hello messages.

We also demonstrated that our design is flexible enough to work under different mobility models such as the *Manhattan Grid* mobility model and the *Graph-based* mobility model. The simulation results from the two mobility models show similar trends.

In summary, we have shown that location-aided routing is likely to be an appropriate method for routing in hybrid wired-wireless networks. The routing performance can be further improved by tuning certain system parameters, e.g. the Hello broadcasting interval.

7.2 Future Work

While the work in this thesis only focuses on some issues in peer-to-peer communications between MNs in the hybrid network environment, there are a few more issues that can be addressed:

1. The gateway discovery algorithm presented here, uses a proactive mechanism to provide subnet connectivity. As explained in Chapter 3, this is not a very good method since a lot of control messages are generated. A better way to provide subnet connectivity is required in order to reduce the overall overhead in the network.

2. The design of LAOD should be reconsidered. From the simulation results obtained, it is obvious that using only the greedy packet forwarding mechanism is not good enough. One possible way to amend LAOD is to add some alternative recovery techniques when the greedy packet forwarding mechanism fails.

3. In LLR, the *percentage metric* is used to measure the routing path quality. The current design is to give equal priority/percentage to both *the number of hop counts* and *RET*. It will be interesting to study how the priority/percentage can be changed under different network data traffic types. For example, if the network data loads are real-time streams, the end-to-end delay is the most important metric to be considered. In this case, *the number of hop counts* may be assigned higher priority/percentage because smaller hop count means less distance that data packets have to travel, which means less end-to-end delay is incurred.

4. Only peer-to-peer communications between MNs is studied here. The routing performance between peer-to-peer communications together with communications between fixed host (FH) of wired network and an ad hoc MN will be challenging yet meaningful work. This will lead to an overall performance overview on the hybrid network environment.

5. The addressing issue is another challenging yet meaningful research to work on. One way is to use IPv6 in the hybrid network. If this is the case, how mobility management integrates with IPv6 in the hybrid network architecture is a key issue

to be studied. Some research studies [7, 27] have already been carried out in the area.

6. The two-dimension (2D) location information is assumed here for simplicity. But in reality, the three-dimension (3D) location information is more suitable as the physical location of the object. Furthermore, the accuracy of the positioning system may affect the routing performance dramatically. These should be further studied.
7. The Hello message adjustment algorithm demonstrates that it is possible to achieve network performance improvement by dynamically tuning certain network parameters. Currently we use the mobility of MNs as the tuning parameter. There are some other network parameters, which can be used as the tuning parameter, like network size, traffic characteristic, traffic patterns, etc. It will be interesting to study the effect by dynamically adjusting these different network parameters.

REFERENCES

[1] "IETF, Mobile ad hoc networks working group." [Online]. Available: http://www.ietf.org/html.charters/manetcharter.html

[2] C. E. Perkins, Ed., *Ad Hoc Networking*. Addison-Wesley, 2001.

[3] H. Hartenstein, B. Bochow, A. Ebner, M. Lott, M. Radimirsch and D. Vollmer, "Position-aware ad hoc wireless networks for inter-vehicle communications: the fleetnet project," in *Proceedings of the 2001 ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc2001)*, Long Beach, CA, USA, 2001.

[4] C. Lochert, H. Hartenstein, J. Tian, H. Fübler, 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, OH, USA, 2003.

[5] J. Tian, L. Han, K. Rothermel and C. Cseh, "Spatially Aware Packet Routing for Mobile Ad Hoc Inter-Vehicle Radio Networks," in *Proceedings of the IEEE 6th International Conference on Intelligent Transportation Systems (ITSC)*, Shanghai, China, 2003.

[6] Y. Sun, E. M. Belding-Royer and C. E. Perkins, "Internet Connectivity for Ad Hoc Mobile Networks," *International Journal of Wireless Information Networks*, vol. 9, 2002.

[7] U. Jonsson, F. Alriksson, T. Larsson, P. Johansson and G. Q. Maguire Jr., "MIP-MANET - Mobile IP for Mobile Ad Hoc Networks," in *Proceedings of the 1st ACM Interational Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2000*, Boston, Massachusetts,USA, August 2000.

[8] P. Ratanchandani and R. Kravets, "A Hybrid Approach to Internet Connectivity for Mobile Ad Hoc Networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2003)*, New Orleans, Louisiana, 2003.

[9] J. G. Jetcheva, Y.C. Hu, S. PalChaudhuri, A. K. Saha and D. B. Johnson, "Design and Evaluation of a Metropolitan Area Multitier Wireless Ad Hoc Network Architecture," in *Proceedings of the 5th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2003)*, Monterey, CA, USA, 2003.

[10] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vec-tor Routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Sys-tems and Applications*, New Orleans, LA, USA, 1999.

[11] M. Mauve, J. Widmer and H. Hartenstein, “A Survey on Position-Based Routing in Mobile Ad-Hoc Networks,” *IEEE Network Magazine*, vol. 15, no. 6, pp. 30–39, 2001.

[12] W. Su, S. Lee and M. Gerla, “Mobility Prediction and Routing in Ad Hoc Wireless Networks,” *International Journal of Network Management*, vol. 11, no. 1, pp. 3–30, 2001.

[13] Y. B. Ko and N. H. Vaidya, “Location-Aided Routing in Mobile Ad Hoc Networks,” in *Procceedings of ACM/IEEE Mobicom*, Dallas Texas, 1998.

[14] J. Hightower and G. Borriello, “Location Systems for Ubiquitous Computing,” *Computer Magazine*, vol. 34, no. 8, pp. 57–66, August 2001.

[15] B. Parkinson and S. Gilbert, “NAVSTAR: Global Positioning System - Ten Years Later,” in *Proceedings of IEEE*, 1983, pp. 1177–1186.

[16] IEEE Computer Society LAN WAN Standards Committee, “Wireless LAN Medium Access Control(MAC) and Physical Layer (PHY) Specifications,” 1997, IEEE std. 802.11-1997.

[17] M. T. Wu, Winston K. G. Seah and W. C. Wong, “Location-Aided On-Demand (LAOD) Routing Protocol for Hybrid Wired-Wireless Networks,” in *Proceeding of 2nd International Conference On Smart homes and health Telematic (ICOST2004)*, Singapore, 2004.

[18] M. T. Wu, Winston K. G. Seah and Lawrence W. C. Wong, “A Link-Connectivity-Prediction-Based Location-Aided Routing Protocol for Hybrid Wired-Wireless Networks,” in *Proceeding of Second International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2005)*, Osaka, Japan, January 2005.

[19] T. Kosch, C. Schwingenschögl and L. Ai, “Information Dissemination in Multihop Inter-Vehicle Networks - Adapting the Ad-hoc On-demand Distance Vector Routing Protocol (AODV),” in *Proceedings of IEEE Intern. Conf. on Intelligent Transportation Systems*, Singapore, September 2002.

[20] J. Xi and C. Bettstetter, “Wireless Multi-Hop Internet Access: Gateway Discovery, Routing, and Addressing,” in *Proceedings of the International Conference on Third Generation Wireless and Beyond (3Gwireless’02)*, San Francisco, CA, USA, May 2002.

[21] J. C. Navas and T. Imielinski, “Geographic Addressing and Routing,” in *Proceedings of the Third ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom’97)*, Budapest, Hungary, September 1997.

[22] S. Basagni, I. Chlamtac, S. R. yrotiuk and B. A. Woodward, “A Distance Routing Effect Algorithm for Mobility (DREAM),” in *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom’98)*, Dallas, TX, October 1998, pp. 76–84.

[23] B. Karp and H. T. Kung, “GPSR: Greedy Perimeter Stateless Routing for Wireless Networks,” in *Proceedings of 6th Annual Int’l. Conf. Mobile Computing and Networking (MobiCom 2000)*, Boston, MA, USA, 2000, pp. 243–254.

[24] S. H. Shah and K. Nahrstedt, “Predictive location-based QoS routing in mobile ad hoc networks,” in *Proceedings of IEEE International Conference on Communications*, Champaign, IL, USA, 2002.

[25] Y. B. Lin and I. Chlamtac, Ed., *Wireless and Mobile Network Architectures*. John Wiley & Sons, Inc, 2001.

[26] R. H. Frenkiel, B. R. Badrinath, J. Borras and R. D. Yates, “The Infostation Challenge: Balancing Cost and Ubiquity in Delivering Wireless Data,” *IEEE Personal Communications Magazine*, vol. 7, no. 2, pp. 66–71, April 2000.

[27] H. Lei and C. E. Perkins, “Ad Hoc Networking with Mobile IP,” in *Proceedings of Europ. Pers. Mobile Com. Conf. (EPMCC)*, Bonn, Germany, September 1997.

[28] M. S. Corson, J. P. Macker and G. H. Cirincione, “Internet-based Mobile Ad Hoc Networking,” *IEEE Internet Computing*, August 1999.

[29] J. Broch, D. A. Maltz and D. B. Johnson, “Supporting Hierarchy and Heterogeous Interfaces in MultiHop Wireless Ad Hoc Networks,” in *Proceedings of Workshop. on Mobile Computing, in Conj. Intern. Symp. on Parallel Architectures, Algorithms, and Networks*, Perth, Australia, June 1999.

[30] R. Wakikawa, J. T. Malinen, C. E. Perkins, A. Nilsson and A. J.Tuominen, “Global Connectivity for IPv6 Mobile Ad Hoc Networks,” November 2001, IETF Internet draft,draft-wakikawa-manet-globalv6.txt (work in progress).

[31] W. Kellerer, C. Bettstetter, C. Schwingenschlöl, P. Sties, K. E. Steinberg and H.-J. Vöel, “(Auto)Mobile Communication in A Heterogeneous & Converged World,” *IEEE Personal Communications Magazine. Special Issue on European R&D on Fourth-Generation Mobile and Wireless IP Networks*, vol. 8, no. 6, pp. 41–47, December 2001.

[32] H. Y. Hsieh and R. Sivakumar, “A Hybrid Network Model for Cellular Wireless Packet Data Networks,” in *Proceedings of the IEEE Global Communication Conference (GLOBECOM)*, Taipei, Taiwan, November 2002.

[33] H.Y. Hsieh and R. Sivakumar, “Internetworking WWANs and WLANs in Next Generation Wireless Data Networks,” in *Proceedings of the International Conference on 3G Wireless and Beyond*, San Francisco, CA, USA, May 2002.

[34] H. Y. Hsieh and R. Sivakumar, “On Using the Ad-hoc Network Model in Wireless Packet Data Networks,” in *Proceedings of the Third ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2002)*, Lausanne, Switzerland, June 2002.

[35] E. Royer and C. K. Toh, “A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks,” *IEEE Personal Communications*, vol. 6, no. 2, pp. 46–55, 1999.

[36] T. Clausen and P. Jacquet, “Optimized Link State Routing Protocol (OLSR),” 2002, IETF Internet draft, draft-ietf-manet-olsr-07.txt (work in progress).

[37] C. E. Perkins and P. Bhagwat, “Higly Dynamic Destination-Sequenced Distance Vector (DSDV) for Mobile Computers,” in *Proceedings of the SIGCOMM 1994 Conference on Communications Architectures, Protocols and Applications*, London, UK, August 1994, pp. 234–244.

[38] C. C. Chiang, “Routing in Clustered Multihop Mobile Wireless Networks with Fading Channel,” in *Proceedings of the IEEE SICON’97*, Singapore, April 1997, pp. 197–211.

[39] D. B. Johnson and D. A. Maltz, “Dynamic Source Routing in Ad Hoc Wireless Networks,” 2003, IETF Internet draft,draft-ietf-manet-dsr-09.txt (work in progress).

[40] V. Park and S. Corson, “Temporally-Ordered Routing Algorithm (TORA),” 2001, IETF Internet draft,draft-ietf-manet-tora-spec-03.txt (work in progress).

[41] Z. J. Haas and M. R. Pearlman, “The Performance of Query Control Schemes for the Zone Routing Protocol,” *IEEE/ACM Trans. Net.*, vol. 9, no. 4, pp. 427–438, August 2001.

[42] G. Pei, M. Gerla and X. Hong, “LANMAR: Landmark Routing for Large Scale Wireless Ad Hoc Networks with Group Mobility,” in *Proceedings of IEEE/ACM MobiHOC 2000*, Boston, MA, August 2000, pp. 11–18.

[43] G. Pei et al., “A Wireless Hierarchical Routing Protocol with Group Mobility,” in *Proceedings of IEEE WCNC’99*, New Orleans, LA, September 1999.

[44] I. Stojmenovic, *Handbook of wireless networks and mobile computing*. New York: Wiley, 2001.

[45] X. Hong et al., “Scalable routing protocols for mobile ad hoc networks,” *IEEE Network*, vol. 16, no. 4, pp. 11 –21, September 2002.

[46] I. Stojmenovic, “Position Based Routing in Ad Hoc Networks,” *IEEE Commmunications Magazine*, vol. 40, no. 7, pp. 128–134, July 2002.

[47] S. Giordano, I. Stojmenovic, *Ad Hoc Wireless Networking*. Kluwer, 2004, ch. Position Based Routing Algorithms for Ad Hoc Networks: A Taxonomy, pp. 103–136.

[48] P. Bose, Pat Morin, I. Stojmenovic and J. Urrutia, “Routing with Guaranteed Delivery in Ad Hoc Wireless Networks,” *ACM Wireless Networks*, vol. 7, no. 6, pp. 609–616, November 2001.

[49] I. Stojmenovic, M. Russell and B. Vukojevic, “Depth First Search and Location Based Localized Routing and QoS Routing in Wireless Networks,” *Computers and Informatics*, vol. 21, no. 2, pp. 149–165, 2002.

[50] N. Guba and T. Camp, “Recent Work on GLS: a Location Service for an Ad Hoc Network,” in *Proceedings of the Grace Hopper Celebration (GHC’02)*, Vancouver Canada, October 2002.

[51] J. Li, J. Jannotti, D. De Couto, D. Karger and R. Morris, “A Scalable Location Service for Geographic Ad-hoc Routing,” in *Proceedings of MobiCom’00*, Boston, Massachusetts, USA, August 2000, pp. 120–130.

[52] Z. J. Haas and B. Liang, “Ad Hoc Mobility Management with Uniform Quorum Systems,” *IEEE/ACM Trans. On Networking*, vol. 7, no. 2, pp. 228–240, April 1999.

[53] L. Stojmenovic, “A Routing Strategy and Quorum Based Location Update Scheme for Ad Hoc Wireless Networks,” Computer Science. SITE. University of Ottawa,” Technical Report TR-99-09, 1999. [Online]. Available: http://www.mwif.org

[54] H. X. Tan and Winston K.G. Seah, “Dynamically Adapting Mobile Ad Hoc Routing Protocols to Improve Scalability - The AODV Example,” in *Proceedings of IASTED Interna-tional Conference on Communication Systems and Networks (CSN2004)*, Marbella, Spain, September 2004.

[55] C. E. Perkins, E. M. Belding-Royer and S. Das, “Ad Hoc On Demand Distance Vector (AODV) Routing,” IETF RFC 3561.

[56] “The Network Simulator ns-2 notes, documentation and source codes.” [Online]. Available: http://www.isi.edu/nsnam/ns

[57] N. Gartner, C. J. Messer and A. K. Rathi, “Traffic flow theory -A state-of-the-art report: revised monograph on traffic flow theory,” 1998. [Online]. Available: http://www.tfhrc.gov/its/tft/tft.htm

[58] T. Camp, J. Boleng and V. Davies, “A Survey of Mobility Models for Ad Hoc Network Research,” *Wireless Communication and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no. 5, pp. 483–502, 2002.

[59] T. Jing, H. Jörg, B. Christian, S. Illya and R. Kurt, “Graph-Based Mobility Model for Mobile Ad Hoc Network Simulation,” in *Proceedings of the 35th Annual Simulation Symposium*, San Diego, California, 2002.