PREDICTIVE METHODS FOR LOCATION SERVICES IN MOBILE AD HOC NETWORKS

by

Xinwei Luo

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Mathematical and Computer Sciences).

Golden, Colorado
Date __________________

Signed: ______________________________
Xinwei Luo

Approved: ______________________________
Dr. Tracy Camp
Associate Professor

Approved: ______________________________
Dr. William Navidi
Professor

Golden, Colorado
Date __________________

____
Dr. Graeme Fairweather
Professor and Head
Department of Mathematical and
Computer Sciences

ABSTRACT

Mobile Ad hoc NETworks (MANETs) consist of a group of Mobile Nodes (MNs) which form a communication network without prior infrastructure. Each MN in the network is responsible for providing services to other MNs in order to realize the ad hoc communication capability. A key component and primary challenge in MANETs is routing data packets over multiple hops between MNs. MANET routing has recently received considerable research interest. Applying location information in ad hoc network routing protocols has demonstrated performance improvements and promised dramatic scalability.

Some routing protocols assume that a location service exists to provide location information about all the MNs in the network. In recent years, many location service protocols have been developed for ad hoc networks, including the Grid Location Service (GLS), the Reactive Location Service (RLS), the Simple Location Service (SLS) and the DREAM (Distance Routing Effect Algorithm for Mobility) Location Service (DLS).

In this paper, GLS, RLS, SLS and DLS are compared in simulation. In the location service protocols, information about individual MNs are shared among all the MNs in the network. In the aforementioned location services, location information about the other MNs is usually saved in a table maintained by each MN. And when an MN's location is needed, the previously saved information is used. In this thesis, a new location service, the Prediction Location Service (PLS), is proposed. In PLS, an MN uses the previous state of an MN to predict that MN's future state. Our evaluation shows that PLS has low overhead and lower location error than GLS, RLS, SLS, and DLS.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to extend my sincerest thanks to Dr. Tracy Camp and Dr. William Navidi, for not only providing me with the opportunity to attain this degree, but also for patiently providing guidance in my research. I would also like to thank Dr. Colagrosso and Dr. Mehta for contributing their time to serve on my committee. In addition, I gratefully acknowledge Michael Kaseman for providing the code for the Grid Location Service. Finally, this research has been supported by the NSF Grant #ANI-0073699. I would like to express my appreciation to NSF for its generous funding and support of this project.

Chapter 1

INTRODUCTION

1.1 Overview of Mobile Ad Hoc Networks

An ad hoc wireless network is “a collection of two or more devices equipped with wireless communications and networking capability” [1], that creates a network “on the fly” without any network infrastructure. In order to realize the communication capability, each Mobile Node (MN) in the ad hoc network, whether it be a laptop, PDA, autonomous agent, or sensor, is responsible for providing information/services to other MNs. The environment for creating an ad hoc network is a broadcast physical medium with limited range, such as the physical medium offered by infrared or radio frequency wireless communications.

There are numerous scenarios that do not have an available network infrastructure which could benefit from the creation of an ad hoc network:

- Rescue/emergency operations: rapid installation of a communication infrastructure during a natural/environmental disaster that has destroyed the previous communication infrastructure;

- Tactical/military missions: rapid installation of a communication infrastructure in a hostile/unknown territory;

- Commercial projects: simple installation of a communication infrastructure for commercial gatherings such as conferences;

- Educational classrooms: simple installation of a communication infrastructure to create an interactive classroom on demand.

There are many challenges in the creation of an ad hoc network:

- Heterogeneity: (e.g., how to communicate among the different devices, such as laptops, PDAs, autonomous agents, or sensors in the ad hoc network);
- Routing challenges: (e.g., how to route information to a fast moving MN);
- Wireless medium challenges: (e.g., lower bandwidths, higher error rates, more frequent disconnections, and less security than fiber lines);
- Portability challenges: (e.g., lower power and smaller storage capacity than desktop computers);
- Security: (e.g., how to prevent eavesdropping, or how to prevent malicious MNs from damaging the ad hoc network);
- Scalability: (e.g., how to efficiently share information among the MNs as the size of the network grows).

1.2 Existing Location Services in Mobile Ad Hoc Networks

Location information has recently been applied to MANET routing protocols. Using location services to provide location information to MANET routing protocols have demonstrated performance improvements and promised dramatic scalability [2, 3]. These routing protocols forward a packet (i.e., the unit of data that is transmitted between an origin and a destination on the network) based on the geographical location of that packet's destination. In these algorithms each MN obtains its own location using a mechanism such as the Global Positioning System (GPS) [4, 5, 6, 7], and shares its location with other MNs in the network. While some of the location-based routing protocols (e.g., DREAM [8] and LAR [9]) include the exchange of location information as a part of its protocol, most of them assume a location service is available to provide them with accurate MN locations (e.g., [3, 10, 11]).

There are two main categories of location services: proactive and reactive. An MN using a proactive protocol periodically shares its location information with other MNs. In a reactive protocol, an MN transmits a query when it needs the location information for another MN; location information is only transmitted when requested. Several proposed location services are hybrid protocols, combining proactive and reactive features.

In this thesis, four location services – the Grid Location Service (GLS) [12], the Simple Location Service (SLS) [13], the DREAM Location Service (DLS) [13], and the Reactive Location Service (RLS) [13] – are compared by simulation. In these four location services, location information for the other MNs is maintained by each MN in a table called a location table. When an MN's location is needed, the previously saved information in the location table is used. Details about these location services and the simulation results are presented in Chapter 3.

1.3 Motivation and Research Overview

Our research goals are aimed at using prediction methods to determine an MN's current location by combining the MN's previous location with its approximate velocity. More specifically, this research enables MANET routing protocols to improve the accuracy in determining an MN's location via prediction. In this thesis, a new location service, the Prediction Location Service (PLS), is proposed. In PLS, an

MN's previous state (location and velocity) is utilized to predict its future state. The simulation results show that PLS has lower overhead and lower location error when compared to GLS, RLS, SLS, and DLS. Details about PLS are presented in Chapter 4.

Chapter 2

LITERATURE REVIEW

2.1 Location Based Protocols in Ad Hoc Networks

In recent years, many location based routing protocols have been developed for ad hoc networks: GRID [14], Location Aided Routing protocol (LAR) [9], Terminodes Routing [15, 16, 17], Greedy Perimeter Stateless Routing (GPSR) [3, 18], and Distance Routing Effect Algorithm for Mobility (DREAM) [8]. Surveys of location based routing protocols are available in [19, 20, 21]. These protocols make forwarding decisions based on the geographical position of a packet's destination. The application of location information in ad hoc network routing protocols has demonstrated performance improvements and promised dramatic scalability [2, 3].

In location based routing protocols, the routing table stored at each Mobile Node (MN) contains location information for the other MNs in the network (e.g., geographic coordinates that can be obtained by the use of a system such as GPS [4, 5, 6, 7]). Some of these protocols assume a location service exists which provides location information on all the MNs in the network. The sender uses the location service to learn the destination's location.

2.2 Location Services in Ad Hoc Networks

This section contains an overview of the location services previously developed. A location service provides location information for an MN in a mobile ad hoc network by implementing a mechanism for an MN to track the location of other MNs in the network topology. Many location services have been developed recently. See [20, 22] for surveys of location services.

Location services are classified into three types on the basis of when location information is obtained:

- Proactive

  A proactive location service compute locations in a proactive fashion. It requires an MN to share its location information with other MNs periodically.

- Reactive

  A reactive location service compute locations on demand in a reactive fashion. Location information is only transmitted when requested.

- Proactive and reactive hybrid

  Several proposed location services are hybrid protocols, combining both proactive and reactive features.

Grid Location Service (GLS)

Grid Location Service (GLS) was proposed in 2000 [12], and was analyzed in many papers [23, 24, 25]. GLS is a proactive and reactive hybrid protocol. In GLS, each MN periodically updates a small set of other MNs (its location servers) with its current location proactively. An MN sends its position updates to its location servers without knowing their actual identities, assisted by a predefined ordering of node identifiers and a predefined geographic hierarchy. Queries for an MN's location information also use the predefined identifier ordering and spatial hierarchy to find a location server for that MN reactively. GLS is explained in detail in Section 3.2.1.

Dream Location Service (DLS)

DLS was proposed in 2002 [13]. DLS is a proactive and reactive hybrid protocol. The proactive piece has DLS periodically flood location information. The reactive piece searches for the location of an MN if the desired location information is unknown. In [13] the authors compared DLS with other location services and concluded that DLS is unable to provide accurate location information (especially at high speeds). DLS is explained in detail in Section 3.2.2.

Reactive Location Service (RLS)

Two almost identical location services were proposed in [13] and [26] in 2002. RLS is a reactive location service that queries location information on an as needed basis. Each MN in RLS maintains a location table; entries in the location table of an MN are purged periodically based on the age of the location information. RLS is explained in detail in Section 3.2.3.

Simple Location Service (SLS)

SLS was also proposed in 2002 [13]. SLS is a proactive and reactive hybrid protocol that periodically shares location table entries with neighbors and floods location requests when needed. SLS offers advantages in terms of simplicity, overhead, and performance [27]. SLS is explained in detail in Section 3.2.4.

Legend Exchange and Augmentation Protocol (LEAP)

LEAP was proposed in 2003 [28]. LEAP is a proactive protocol. LEAP consists of a legend (i.e., an explanatory list) that migrates (or leaps) from one MN to another one in a heterogeneous network [28]. In LEAP, each MN periodically broadcasts a packet to its neighbors to announce its existence. This broadcast packet includes the source MN's location information and the time the packet was sent. The information in these broadcast packets are used to update an MN's local location table entries on the MN's neighbors. The legend is a global location table and includes location information on all MNs as well as information to decide where to send the legend [28]. The legend traverses the network sharing location information.

Uniform Quorum System (UQS)

A distributed mobility management scheme using a class of Uniform Quorum Systems (UQS) was proposed in 1999 [29]. In this proactive scheme, location databases are stored in the MNs. The databases are dynamically organized into quorums, every two of which intersect at a constant number of databases. Upon receipt of a location update or call, an MN's location information is written to or read from all the databases of a quorum, chosen in a non-deterministic manner [30].

An important aspect of quorum-based location services is the following tradeoff. The larger the quorum sets, the higher the cost for position updates and queries. However, the larger the quorum sets the higher possibility of obtaining an intersection of two quorums [20].

Unfortunately, "original" quorum systems, also termed strict quorum systems, do not apply well to highly dynamic environments [30]. The very construction of these quorums is not a trivial task, as their outcome is strongly subject to membership changes [31].

Home Region Location Services

A home region based location service was independently proposed in [32] and [33] in 1999. This proactive location service uses the concept of a virtual homezone where position information for an MN is stored. The source MN issues several search 'tickets' (each ticket is a 'short' message containing the sender's id and location, the destination's best known location and the time that location is reported) that will look for the exact position of the destination MN. When the first ticket arrives at the destination MN D, D will report back to the source with a brief message containing its exact location. The source MN then sends the data message (a 'long' message) toward the exact location of the destination [33].

Dead Reckoning Method (DRM)

The Dead Reckoning Method (DRM) was proposed in 2003 [34]. DRM is a location-based routing protocol with a location service included, similar to LAR [9] and DREAM [8]. DRM is a predictive mobility tracking method where every MN predicts or tracks the movement of every other MN in the network. Once an MN can predict all other MNs' locations, DRM has the MN compute the topology of the network, and then uses Dijkstra's shortest path algorithm to route packets. In [34], the authors use geographic forwarding to route packets. DRM's performance was evaluated in [35].

In Chapter 3, we evaluate GLS, DLS, RLS and SLS in simulation. GLS is often referenced, and since DLS, RLS and SLS were developed by the Toiler's Research Group, we have simulation code for these three protocols. We wanted to evaluate DRM in simulation as well; however, we were not able to obtain the DRM code from the authors. We also did not evaluate LEAP, because the details on LEAP were not finalized during this research. The Home Region Location Services was not evaluated because it is not a fully distributed protocol while GLS, DLS, RLS and SLS are. UQS was not evaluated because previous research concluded that it does not apply well to highly dynamic environments [30].

Chapter 3

A COMPARISON OF LOCATION SERVICES

3.1 Introduction

Applying location information in Mobile Ad hoc NETworks (MANET) routing protocols has demonstrated improvements in performance and promised dramatic scalability [2, 3]. This success coupled with the recent development of protocols that require location information highlights the need to investigate protocols for delivering location information, a.k.a., location services. An effective location service can be used to improve the performance and scalability of routing protocols that need location information.

Four location services, the Grid Location Service (GLS), the DREAM Location Service (DLS), the Reactive Location Service (RLS), and the Simple Location Service (SLS), are explained in detail in Section 3.2 and are compared by simulation in Section 3.4.

3.2 Existing Location Services

This section detail the functionality of GLS, DLS, RLS, and SLS.

3.2.1 The Grid Location Service (GLS)

GLS is a hierarchical location service. In GLS, the area covered by the ad hoc network is divided into squares of the same size, called grids, and the grids are predefined into a hierarchy of grids. The smallest square is called an order-1 grid.

Four order-1 grids make up an order-2 grid, four order-2 grids make up an order-3 grid, and so on.

Each Mobile Node (MN) in the network chooses other MNs to be its location servers. Periodically, each MN sends location information to its location servers. Each MN maintains a location table that contains the location information for all the other MNs in the network.

GLS is a proactive and reactive hybrid protocol. When an MN wants to send a packet to another MN, it first accesses its location table for the destination MN's location. If the required location information is either unknown or expired, the MN initiates a location query. The query is sent to the nearest potential location server for the destination MN. If the destination's location information is not known, the query is forwarded to a potential location server in the next highest order grid. This process continues until a location server is found that knows the location of the destination. That location server then forwards the query directly to the destination MN, which responds to the location query request with its most recent location.

In summary, the three main activities in GLS are defining the location grids, selecting and updating the location servers, and sending the location queries. These activities are explained in detail below.

- Defining the Location Grids

  Figure 3.1 shows an example for the grids. This example was taken from [12]. We note that an order-n square's lower left corner's coordinates must be of the form $(a2^{n-1}, b2^{n-1})$ for integers a and b. Therefore the lightly shaded square is shown as an example of a 2×2 square which is not an order-2 square because of its location.

- Selecting and Updating the Location Servers

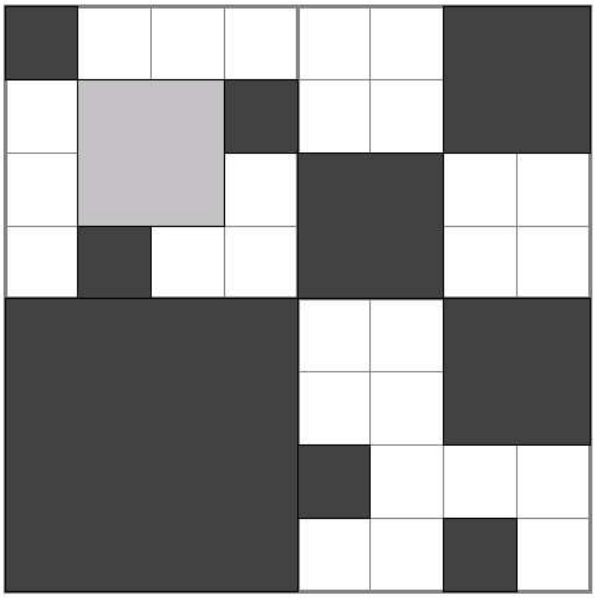


FIG. 3.1. Example of the Grids in GLS

Each MN is allocated a unique ID. The set of location servers for an MN is chosen by the predefined geographic grid and the predefined ordering of mobile node identifiers in the ad hoc network. Specifically, an MN chooses its location servers in a grid by selecting a set of MNs in that grid with IDs close to its own ID. Figure 3.2 shows an example of the location server selection. This example was taken from [12]. The inset squares are regions in which B will seek a location server. The MNs circled are selected to become B's location servers. B recruits three servers in order-1 squares, three servers in order-2 squares, and three servers in order-3 squares. B chooses the MN closest to itself in ID space as a server [12].

When an MN moves out of the threshold distance of one of its location servers, which is calculated according to that location server's square order, the MN

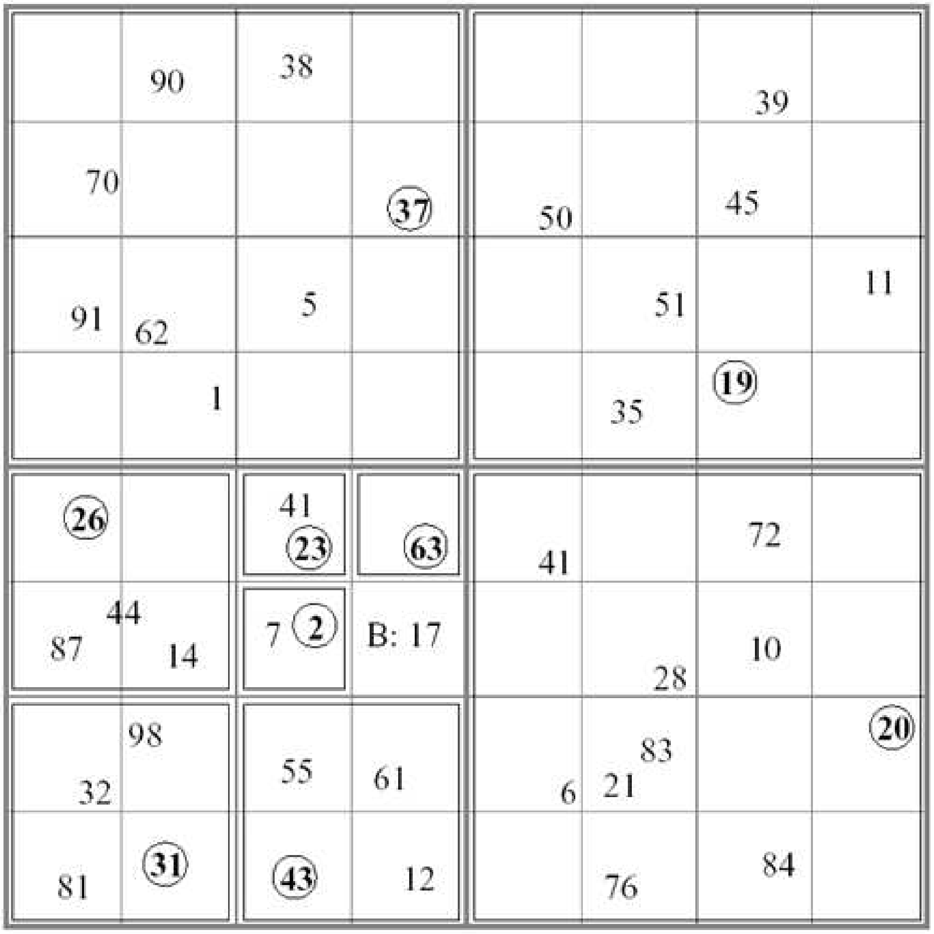


FIG. 3.2. Example of Server Selection in GLS

sends an update packet to that location server. Therefore, an MN updates its location servers at a rate proportional to its speed, and the distant location servers are updated less frequently than the nearby location servers.

Each location server maintains two tables: the location table which holds the location of MNs that have selected it as their location server, and a location cache which holds information from the update packets an MN has forwarded.

- Sending Location Queries

  When an MN needs a location for a destination, it initiates a location query request. Since each MN knows all MNs within its order-1 square, the request is first sent to a potential location server for the destination desired in the requesting MN's order-2 square, i.e., to an MN whose ID is the least one that is greater than or equal to the ID of the destination within the order-2 square. If the MN is not a location server for the destination, the MN forwards the query using the same algorithm until it reaches a location server for the destination. This location server then forwards the query directly to the destination, which responds to the location query request with its most recent location.

3.2.2 The DREAM Location Service (DLS)

DLS was proposed in 2002 [13]. It is named the DREAM Location Service because it is similar to a location service proposed by the authors of DREAM [8].

DLS is a proactive and reactive hybrid protocol. DLS periodically shares individual MN location information by sending a Location Packet (LP), which contains the coordinates of the source MN based on some reference system, the source MN's speed, and the time the LP was transmitted. Each MN sends an LP to nearby MNs at a higher rate, and to faraway MNs at a lower rate. By differentiating between nearby and faraway MNs, the overhead of location packets is limited. The DLS location update time is calculated by the following equations:

transmit nearby LP: $(\frac{T_{range}}{\alpha}) \times (\frac{1}{\nu}) = \frac{T_{range}}{\alpha\nu}$,

transmit faraway LP: one for every X nearby LPs or at least every Y seconds, where T_{range} is the transmission range of the MN, ν is the average speed of the MN, and α is a scaling factor.

DLS uses promiscuous mode operation, which means that each MN updates its location table whenever a new location packet is overheard/received. Therefore, LPs to faraway MNs update all MNs in the network, including nearby MNs.

When an MN requires the location of another MN, it accesses its location table first. The MN initiates a location request packet if the required location information is either unknown or expired. The location request packet is flooded in the network. When another MN receives this location request packet, it propagates the flooded location request if it is not the requested MN, or it sends back a location reply packet with its current location information if it is the requested MN.

If a location entry is older than $locationGoodTime$ seconds, it's considered outdated and deleted from the location table. The parameter $locationGoodTime$ is set at a constant 46 seconds in DLS.

3.2.3 The Reactive Location Service (RLS)

RLS was proposed in 2002 [13]. In RLS, when an MN requires a location for another MN, it will first check its location table. If the required location information is either unknown or expired, the requesting MN will send a query packet to the network by binary flooding. This means that the requesting MN first asks its 1-hop neighbors for the requested location information. If no response is received within a timeout period, then the MN will flood a location request packet in the entire network.

When an MN receives a location request packet and has information about the requested MN in its location table, it returns a location reply packet via the reverse source route obtained in the location request packet. Otherwise, it propagates the flooded location request. Same as DLS, a location information entry is deleted if it is 46 seconds old.

3.2.4 The Simple Location Service (SLS)

SLS was proposed in 2002 [13]. Similar to DLS, SLS is a proactive-reactive hybrid protocol, which means location information is shared in the network periodically. If the required location information is not found in the MN's location table then the MN transmits a request packet. Old entries are deleted from the location table periodically. The difference between DLS and SLS is that SLS periodically shares multiple location table entries with neighbors, instead of sending its own location information to all the MNs in the network. The update conditions of SLS are as follows:

1. Every $(\frac{T_{range}}{\alpha}) \times (\frac{1}{\nu}) = \frac{T_{range}}{\alpha\nu}$ seconds or at least every X seconds, a location packet is transmitted. T_{range} is the transmission range of the MN, v is the MN's velocity, α is a scaling factor, and X is a constant.

2. If the MN's speed is 0 (i.e., the MN is paused) when it sends an update packet, the next update is scheduled to be sent after $\frac{\overline{T_{pause}}}{2}$ seconds, where $\overline{T_{pause}}$ is the average pausing time.

The SLS update algorithm is implemented by scheduling the MN's next update every time it transmits an update packet, according to its velocity at the time. According to [13], the implementation details of SLS are as follows: the scaling factor α is set to 4 and X is set to 13. Similar to DLS, a location information entry is deleted if it is 46 seconds old.

3.3 SLS Overhead Analytical Analysis

In this section, we review the steps we took to calculate the overhead of SLS analytically. We first calculated the average distance an MN travels between two broadcast updates. Then we calculated the total number of update packets and the total number of flooding packets.

The four calculation steps are as follows:

1. We calculated the distance an MN travels between two updates.

   Update condition 1 of SLS is equivalent to the following. An MN updates its location information whenever it travels m meters away from the previous update location, or at least every X seconds. To simplify the problem, we assumed that the upper boundary X is never reached. We also assumed that the current destination of the MN is a random point in the network when the MN sends an update. Therefore, the probability that an MN sends an update before reaching the destination d_1 is as follows:

   $$p_1 = 1 - \frac{\pi \times m^2}{a \times b}, \tag{3.1}$$

   where a and b are the width and length of the network area, $m =$ (transmission range/$scalingFactor$) $= 25$ m, $a = 300$ m, and $b = 600$ m. Therefore, $p_1 \approx 98.91\%$. According to this result, the probability that an MN does not send an update before reaching the destination is about 1%. Among this 1% of cases, we assumed that every MN sends an update before reaching the second destination. The expected distance traveled between two consecutive location updates under this assumption can be calculated as follows:

$$E(distance) = m \times p_1 + [m + E(d_1)] \times (1 - p_1), \tag{3.2}$$

where d_1 is the distance between the first destination and the update location, and $E(d_1)$ is given by

$$E(d_1) = \int_0^m x \times \frac{2\pi x}{\pi \times m^2} dx = \frac{50}{3}. \tag{3.3}$$

Therefore,

$$E(distance) = 25 \times 0.9891 + \frac{125}{3} \times 0.0109 = 25.1817. \tag{3.4}$$

2. We calculated the expected time between two updates $E(T)$.

   Update condition 2 of SLS is if the MN is paused when it sends an update packet, the next update will be scheduled to be sent after $\frac{\overline{T_{pause}}}{2}$ seconds, where $\overline{T_{pause}}$ is the average pausing time. Therefore, $E(T)$ can be calculated by the following equation:

$$E(T) = P(moving) \times \frac{E(distance)}{E(v)} + [1 - P(moving)] \times \frac{\overline{T_{pause}}}{2}, \tag{3.5}$$

   where $E(v)$ is the expected value of the MN's speed, and $P(moving)$ is the probability that the MN is moving when it transmits an LP. $E(v)$ can be calculated by

$$E(v) = \frac{v_1 - v_0}{\log(v_1) - \log(v_0)}, \tag{3.6}$$

   where v_1 is the highest speed of the MN, and v_0 is its lowest speed.

The parameter $P(moving)$ can be calculated by

$$P(moving) = \frac{E(n_{moving})}{E(n_{moving}) + E(n_{paused})}, \tag{3.7}$$

where n_{moving} is the number of LPs sent between two pauses and n_{paused} is the number of LPs sent during a pause. $E(n_{moving})$ can be calculated by $\frac{E(D^*)}{E(distance)}$, where D^* is the distance between two path endpoints. In general, for a rectangular region with sides a and b, a result in [36] can be used to show that

$$\begin{gathered} E(D^*) = \tfrac{1}{6}[\tfrac{b^2}{a}\log(\sqrt{\tfrac{a^2}{b^2}+1}+\tfrac{a}{b}) + \tfrac{a^2}{b}\log(\sqrt{\tfrac{b^2}{a^2}+1}+\tfrac{b}{a})] \\ +\tfrac{1}{15}(\tfrac{a^3}{b^2}+\tfrac{b^3}{a^2}) - \tfrac{1}{15}\sqrt{a^2+b^2}(\tfrac{a^2}{b^2}+\tfrac{b^2}{a^2}-3). \end{gathered} \tag{3.8}$$

According to the mobility files we used, $a = 300$ and $b = 600$; therefore $E(D^*) = 241.4316$ and $\frac{E(D^*)}{E(distance)} = 9.5876$.

Because the next update is scheduled to be sent after $\frac{\overline{T_{pause}}}{2}$ seconds, we assumed that no more than two update packets are sent when an MN is paused. We also assumed that the MNs' velocity and pausing time are both constants. Therefore, $E(n_{paused})$ can be calculated by the following:

$$\begin{aligned} &if\ \tfrac{E(distance)}{v} > \tfrac{T_{pause}}{2}: \\ &E(n_{paused}) = 2 \times [\tfrac{T_{pause}}{2} \times \tfrac{v}{E(distance)}] + 1 \times [1 - \tfrac{T_{pause}}{2} \times \tfrac{v}{E(distance)}]; \\ &if\ \tfrac{E(distance)}{v} \leq \tfrac{T_{pause}}{2}: \\ &E(n_{paused}) = 2. \end{aligned} \tag{3.9}$$

3. We calculated the expected total number of update packets.

   The expected total number of update packets is equal to $totalTime/E(T)$, where $totalTime$ is the length of the simulation, or 1000 seconds in our mobility files.

4. Finally, we calculated the flooding packets.

   The number of flooding packets can be calculated by the simulation results of $lrra$ (percent of queries resolved locally) and lrl (percent of queries resolved later). $lrra$ and lrl are the results of the simulations.

   The upper boundary of the number of flooding packets is $1900 \times (1 - lrra) \times (NumberOfMNs + 2)$ because $(1 - lrra)$ queries are flooded since they are not found in the location table. The approximate lower boundary is $1900 \times lrl \times (NumberOfMNs + 2)$ because at least lrl queries are flooded in the network. Theoretically the lower boundary could be lower since the flooding packet might not reach every MN in the network. This equation is an approximation.

The analytical calculation results are listed in Table 3.1.

Table 3.1. SLS Overhead Table

Speed (m/sec)	1	5	10	15	20
Simulation Results					
lrra (%)	99.23	99.96	100	100	100
lrl (%)	0.15	0.02	0	0	0
Analytical Calculation Results					
$E(v)$	0.9967	4.9833	9.9666	14.9499	19.9332
P(moving)	0.8889	0.8282	0.8274	0.8274	0.8274
E(T)	23.0136	5.0441	2.9535	2.2567	1.9083
Update Packets	43.4526	198.2514	338.5813	443.1249	524.0266
Flooding Packets –					
Upper Boundary	760.76	39.52	0	0	0
Lower Boundary	148.2	19.76	0	0	0

Comparison of Analytical Results with Simulation

The analytical calculation of the number of SLS update packets are compared with the simulation results in Figure 3.3. As shown, the analytical calculation result is very close to the simulation result. At 1 m/sec, the analytical calculation is lower than the simulation result. This result is because we assumed that the maximum-next-update-time X is not used in our analysis. However, at low speeds, the maximum next update time X is sometimes applied in the simulation. At 20 m/sec, the analytical calculation is higher than the simulation result. We believe the approximation of $E(distance)$ has a fast moving MN update too often.

The comparison of the SLS flooding packets is shown in Figure 3.4. We plot both the upper and lower boundary for the analytical result in the figure. As shown, except at 1 m/sec, the analytical calculation result is very close to the simulation result. Furthermore, Figure 3.4 shows that the confidence interval of the simulation result at 1 m/sec is contained within the analytical calculation's lower and upper boundaries.

3.4 Evaluation of Location Services

In this section, we compare the performance and overhead of GLS, SLS, DLS, and RLS. We first present the simulation environment and then the simulation results.

3.4.1 Simulation Environment

Each location service was implemented in NS-2 [37] version 2.1b8a according to the previously mentioned protocol descriptions. Table 3.2 lists the simulation parameters that we used. The area of the simulated network is 300 m × 600 m. We set the transmission range to be 100 m, and we used IEEE 802.11 medium access

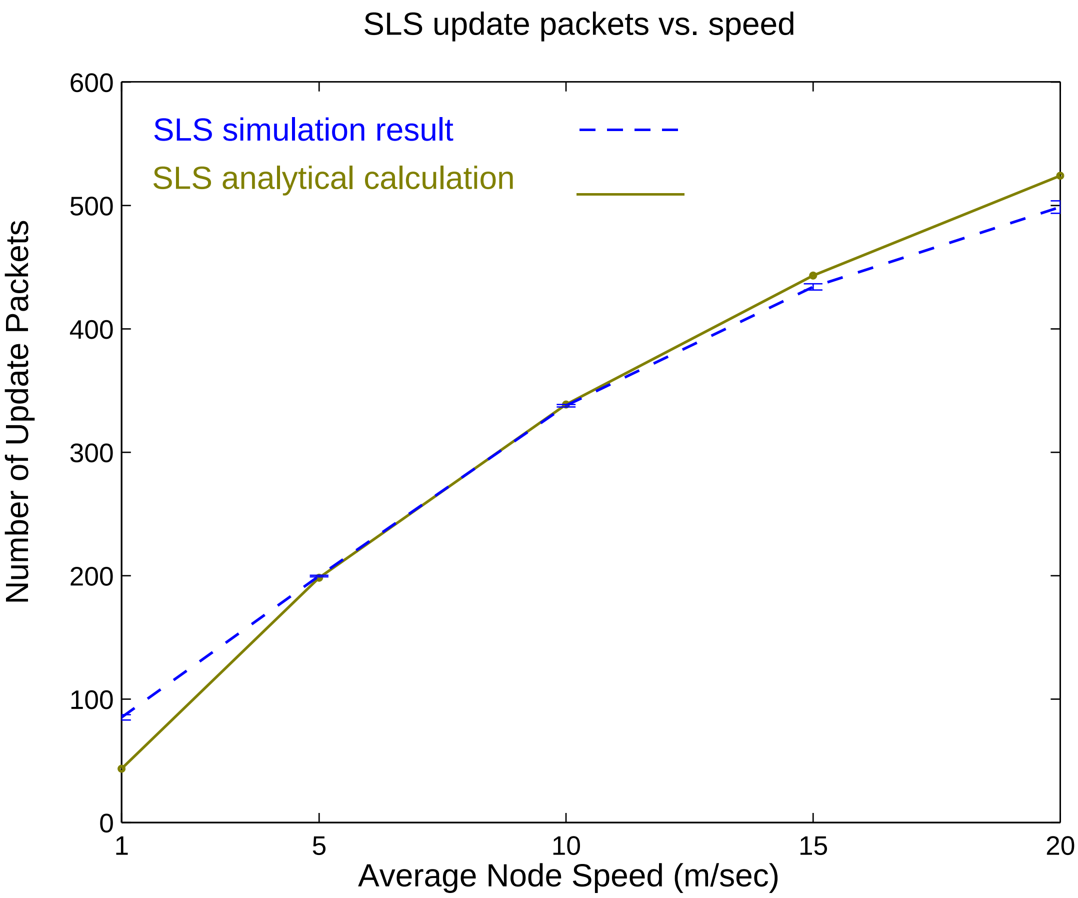


Fig. 3.3. Comparison of SLS Update Packets between the Analytical Calculation and Simulation Result

protocol [38] and set 2Mbps as a link's bandwidth. Many published simulation results of MANET use the random waypoint mobility model [39] (e.g., [12, 13, 24, 28]). In [39], each MN is assigned a random initial location. We used the steady-state random waypoint mobility model [40], in which the initial locations and speeds of the MNs are chosen from the stationary distribution of the random waypoint mobility model. When an MN reaches each destination, the MN pauses for 10 seconds ± 10% before choosing a new destination. With the steady-state random waypoint

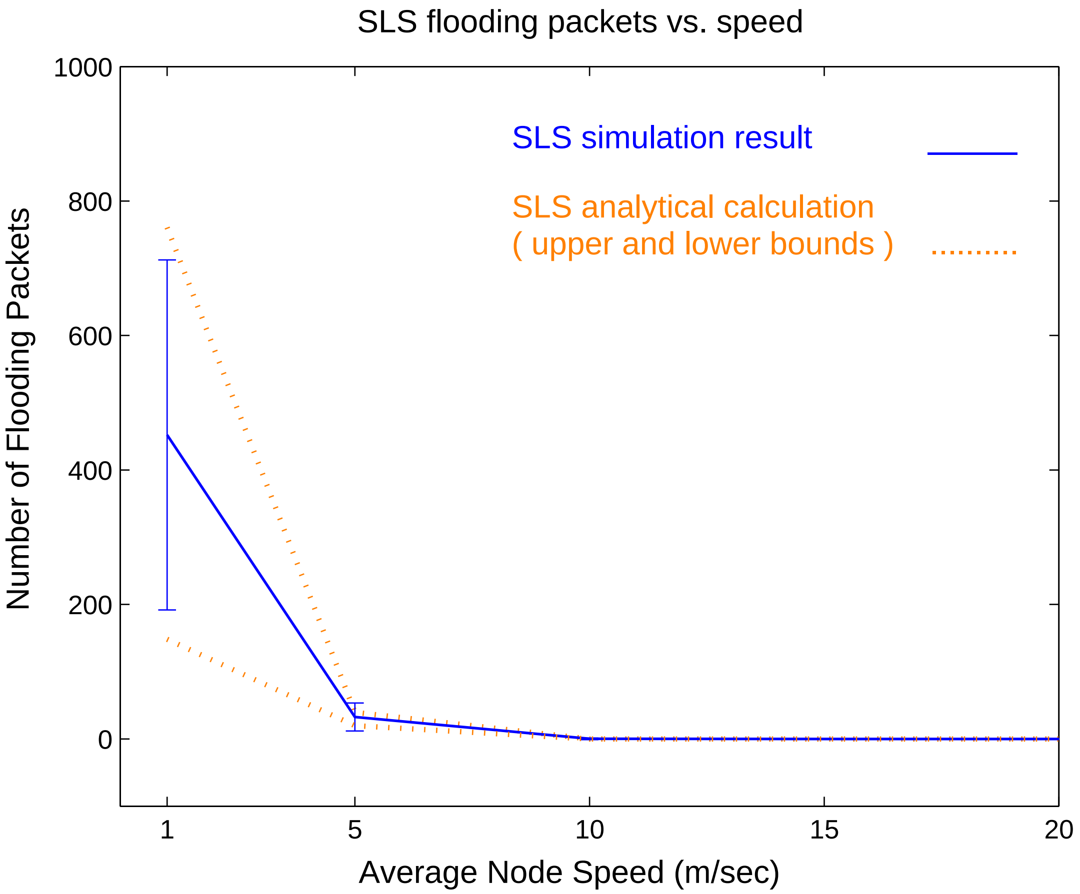


Fig. 3.4. Comparison of SLS Flooding Packets between the Analytical Calculation and Simulation Result

model, one is able to construct more reliable simulations [40]. Each simulation is 1000 seconds long. In the simulation of protocols that have a proactive part (i.e., SLS, DLS, and GLS), location queries begin after 50 seconds to allow location information to propagate in the network before data packets begin transmitting. Two location requests are generated per second per MN. We performed 10 simulation trials for each pair of network settings (network density and MN speed) and plotted the results with 95% confidence intervals.

Table 3.2. Simulation Details

Input Parameters	
Simulation Area Size	300 m × 600 m
Transmission Range	100 m
Simulation Duration	1000 seconds, location requests generated after 50 seconds
Mobility Model	Steady-State Random Waypoint
Pause Time	10 seconds ±10%
Location Request Frequency	2 requests per second per node
Simulator	
Simulator	NS-2 version 2.1b8a
Medium Access Protocol	IEEE 802.11
Link Bandwidth	2 Mbps
Number of Trials	10
Confidence Interval	95%

3.4.2 Simulation Results

Figure 3.5 shows the performance (percentage of location queries answered and the location error) of the four protocols in a network of 50 MNs. Figure 3.6 compares the overhead (packets and bytes) of the four protocols in a network of 50 MNs.

From Figure 3.5 (a), we notice that the percentages of location queries being answered for DLS, RLS, and SLS ranged between 99% and 100%. The percentage of location queries being answered for GLS is only 88-90%. This result is because GLS does not perform well in networks of this density. According to [12], GLS performs best in a network of 400 nodes within a 2000 m × 2000 m area with a transmission range of 250 m. (This scenario is equivalent to 112.5 nodes in our simulation environment). The reason why GLS does not perform well at low network density is because of the "location server" mechanism; if an MN needs location information for another

MN and it is not found in the querying MN's location table, the querying MN must obtain the information from the queried MN's location servers. In networks of low density, there might be no route to the nearest location server. Figure 3.5 (b) shows that GLS also has the highest location error. Therefore, we conclude that GLS is not preferred in a network of 50 MNs.

We also note in Figure 3.5 (b) that DLS was unable to provide accurate location information (especially at high speeds). This result is due to contention issues from the large number of MNs flooding LPs at high speeds. SLS had the lowest location error although RLS's query success ratio was slightly higher.

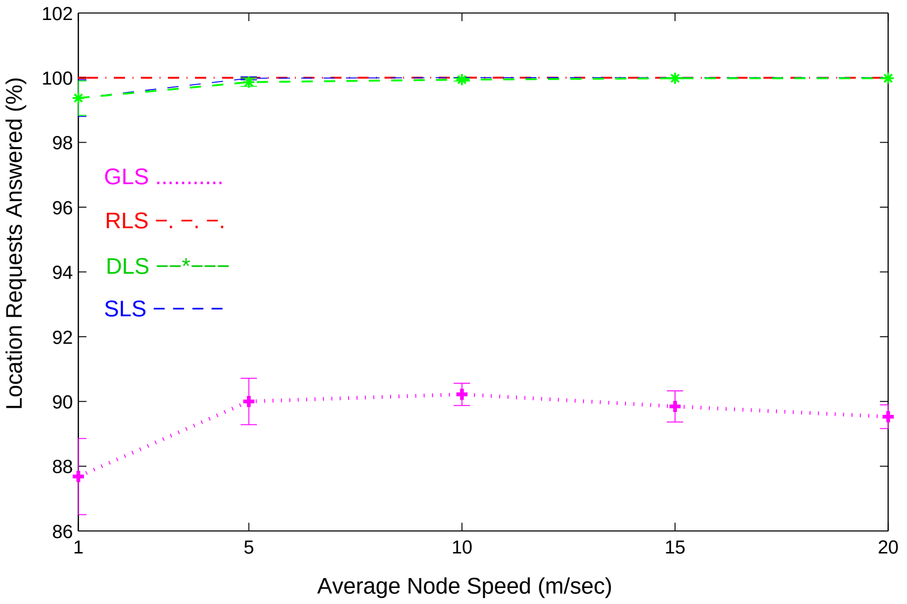
According to Figure 3.6 (a), SLS has the lowest overhead in terms of the number of packets sent. SLS sends more bytes than DLS (see Figure 3.6 (b)), but minimizing the number of packets is more important in wireless communications. Hence, SLS is preferred over GLS, RLS, and DLS.

Figure 3.7 shows the performances of the four protocols, with regard to the percentage of location requests answered and the average location error, in a network with speed 10 m/sec as the network density increases. Although GLS is supposed to perform better with a higher network density, its performance was still not as good as SLS, DLS, and RLS. In fact, once again the performance of SLS is the best in terms of location error (see Figure 3.7 (b)).

Figure 3.8 shows the overhead (packets and bytes) for the four protocols in a network with speed 10 m/sec as the network density increases. The overhead of RLS is much higher than the other three protocols (GLS, DLS, and SLS); most of the results for RLS were out of the range of our plot. This is due to the costly flooding requirements of the reactive protocol. We conclude that RLS should not be used in a network with more than 60 MNs.

Comparing SLS and DLS, SLS has less overhead packets while DLS has less byte overhead; that is, DLS uses smaller LPs that only contain the source MN's location information. Overall, SLS is preferred over GLS, RLS, and DLS.

(a) Location requests answered vs. speed.



(b) Error of location responses vs. speed.

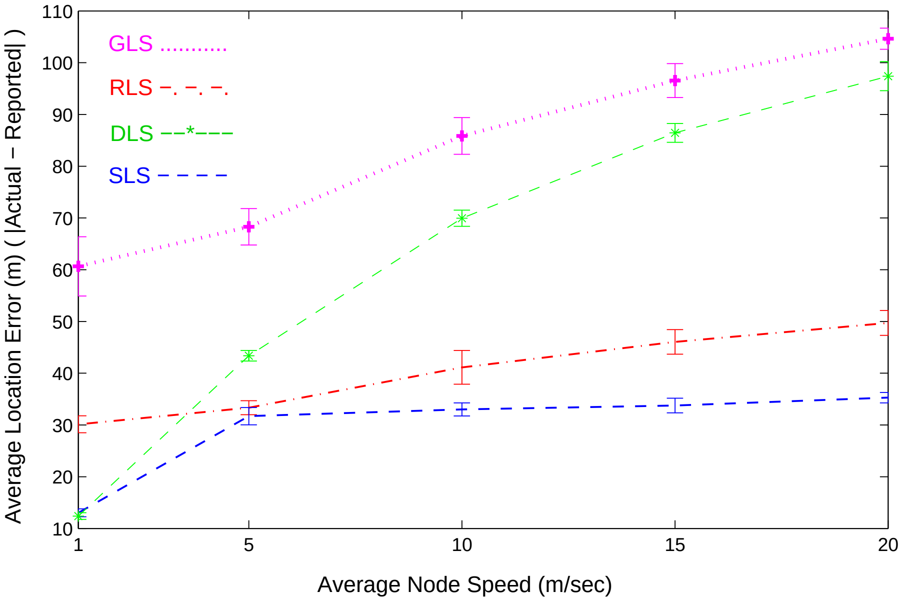
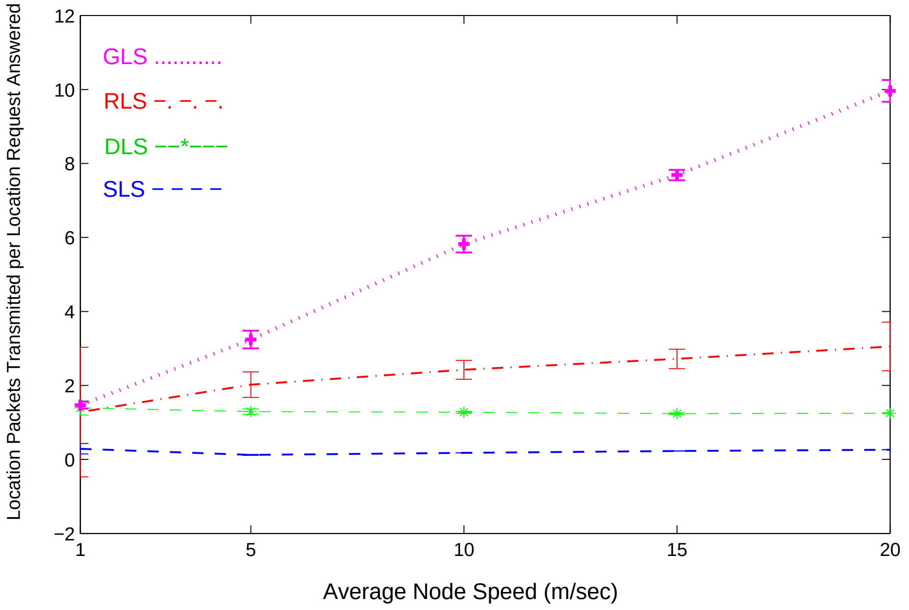


Fig. 3.5. Performance Comparison of GLS, SLS, DLS, and RLS in a Network with 50 MNs

(a) Location packet overhead vs. speed.



(b) Location byte overhead vs. speed.

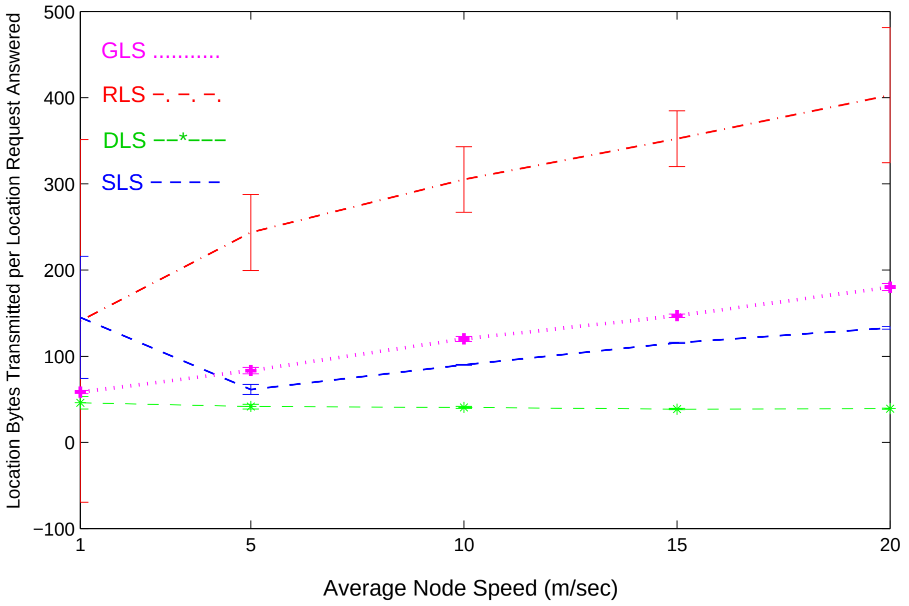
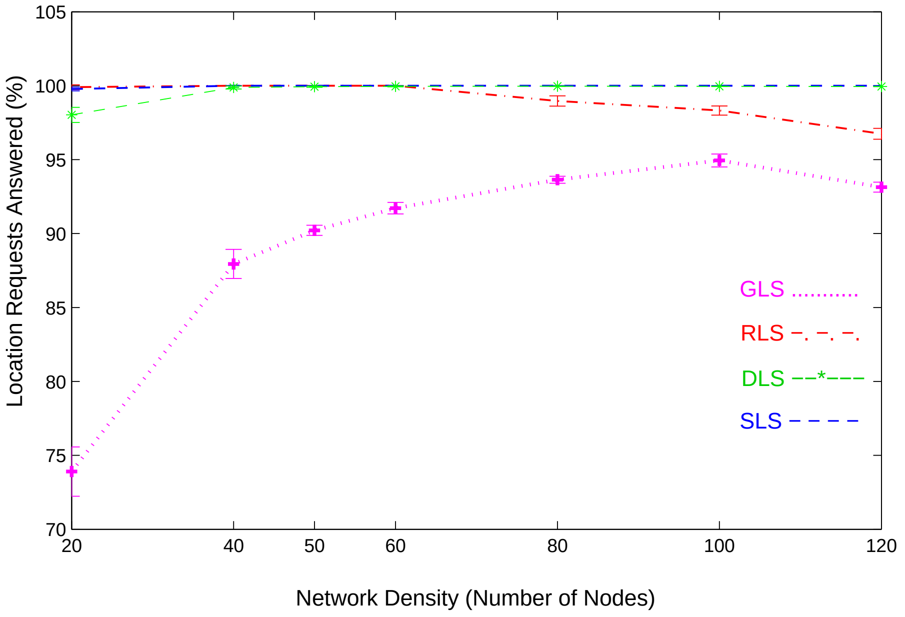


Fig. 3.6. Overhead Comparison of GLS, SLS, DLS, and RLS in a Network with 50 MNs

(a) Location requests answered vs. density.



(b) Error of location responses vs. density.

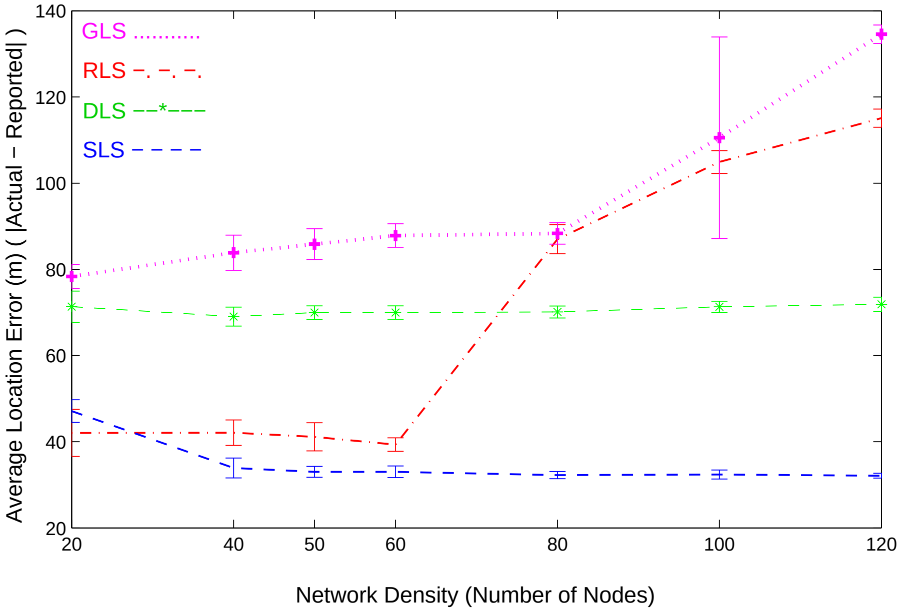
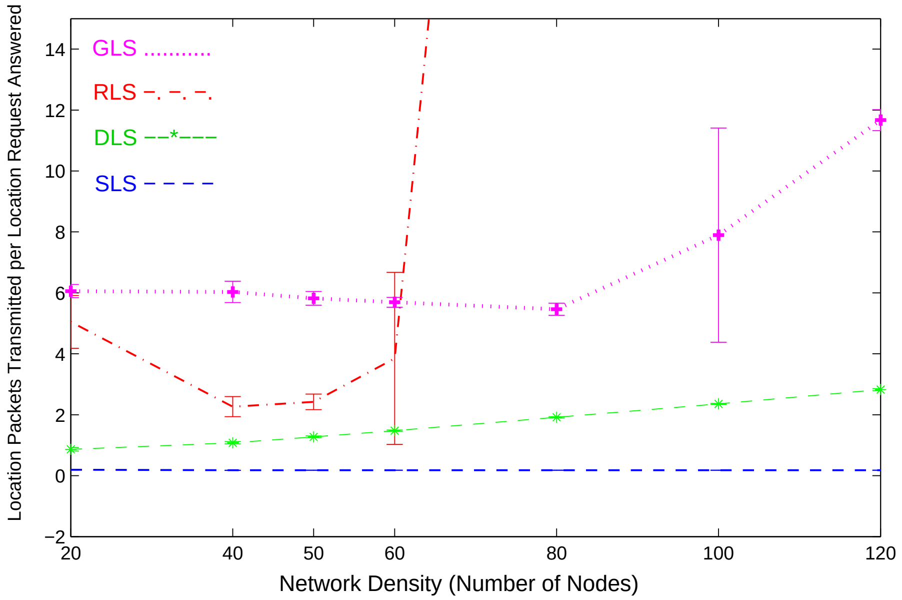


Fig. 3.7. Performance Comparison of GLS, SLS, DLS, and RLS in a Network with Speed 10 m/sec

(a) Location packet overhead vs. density.



(b) Location byte overhead vs. density.

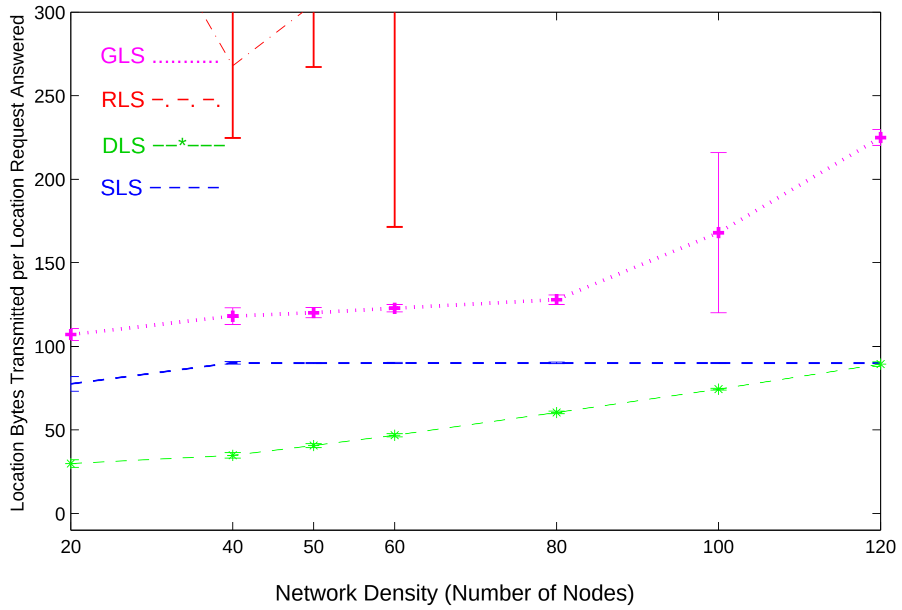


FIG. 3.8. Overhead Comparison of GLS, SLS, DLS, and RLS in a Network with Speed 10 m/sec

Chapter 4

PREDICTIVE LOCATION SERVICE (PLS)

4.1 Introduction

Four existing location services were compared in Chapter 3 and among them, SLS performed the best. All four of these protocols use location information previously saved in an MN's location table when needed. To improve upon previous methods, we predict where an MN currently is using previously stored information.

Using this motivation, we propose PLS (the Predictive Location Service). PLS is a proactive-reactive hybrid protocol. In PLS, location information is shared among the neighbors periodically, and an MN floods the network if the required location information is not found in the MN's location table. In PLS, an MN's previous state (location and velocity) is utilized to predict the MN's current state.

This chapter begins with an introduction of the PLS model, followed by the PLS implementation details. At the end of this chapter, simulation results comparing PLS to SLS are given.

4.2 The PLS Model

This section explains the basic model of PLS. The detailed selection of parameters is explained in the next section. PLS adheres to the following two design guidelines: *i*) this location service is predictive (i.e., the previously saved information is used to estimate an MN's current location), and *ii*) this location service is adaptive (i.e., several parameters adapt to the MN's speed).

There are four main activities in PLS:

1. Sending Location Updates to Neighbors

   Periodically, an MN using PLS sends an LP (Location Packet) to all of its 1-hop-neighbors (i.e., the MN transmits a broadcast packet with time-to-live, or TTL, set to one). Each LP contains the following information: the location information (the coordinates and velocity of the MN) of the source MN, plus location information about E MNs that are copied from the source MN's location table.

   The optimal number of location information entries to include in an LP depends on a lot of factors, e.g., the neighbors' knowledge of the network. In SLS, a constant E entries are chosen from the location table in a round-robin fashion. PLS uses an adaptive update scheme that has an MN transmit all of the updated information that the MN has received since its last update. If the number of updated entries, n, is less than E, $(E - n)$ more entries are chosen from the location table in a round-robin fashion. In this way, the updated information propagates faster.

   The conditions to send an LP are as follows:

   - If the MN is X_1 meters away from the predicted location calculated from its previous location update, where X_1 is a constant.
   - If the MN has moved X_2 meters away from the location where it sent the previous update, where X_2 is a constant.
   - If the last location update is $maxUpdateTime$ seconds ago. Each MN in PLS checks whether to send an LP every $sendLP$ seconds. $maxUpdateTime$ and $sendLP$ are constants.

2. Location Table Updating

   Every time an MN receives an LP, it updates its location table by comparing its own table to the location information in the LP. PLS is executed in "promiscuous mode", i.e., each MN using PLS updates its location table when a new location packet is overheard/received. Location entries that are older than t seconds are considered old and deleted from the location table. In SLS, time t is a constant. However, we noticed that slower MNs' location information should become invalid after a longer time than faster MNs. Also, slower MNs send update packets less frequently. Therefore, in PLS, time t is calculated by

   $$t = \max(locationGoodTime, \frac{entryRemoveFactor}{speed}),\ if\ speed \neq 0, \qquad (4.1)$$

   where $locationGoodTime$ and $entryRemoveFactor$ are constants, and $speed$ is the recorded speed of the requested MN in the location table.

3. Predicting the Location Queried

   In PLS, when a location query occurs in an MN and there is an entry for the queried MN in the MN's location table, the MN predicts the location of the queried MN using the previously saved location and velocity for the queried MN. The predicted location is calculated as

   $$\begin{gathered} location = location_{record} + (v \times T) \\ T = \max(predictionFactor/speed, (t_{now} - t_{record})), \end{gathered} \qquad (4.2)$$

   where v is the MN's velocity calculated from the location information in the last update, $predictionFactor$ is a constant, $speed$ is the recorded speed of the queried MN saved in the location table, t_{now} is the current time, and t_{record} is the time when the previous location information were saved.

The reason to use *predictionFactor* is to prevent over-predicting. That is, in PLS we assume an MN keeps moving at the same speed as the speed saved in the location table. If the MN has already changed its speed, over-predicting can happen.

We used $v = speed$ with the random waypoint mobility model [39] because, in this mobility model, an MN's velocity does not change until it reaches one of its destinations. The situation where an MN moves at a varying speed towards a destination is discussed in Chapter 5.

4. Requesting the Location of the Queried Node

An MN initiates a location request packet if the location information for a queried MN is either unknown or expired in the MN's location table. This location request packet is flooded in the network.

In RLS, any MN can reply to a location query and stop the propagation of the query if it has the queried information in the location table. This scheme has both an advantage and a disadvantage. The advantage is that it reduces the network traffic by controlling the number of flooding packets. The disadvantage is that outdated information may be returned. Since PLS is executed in "promiscuous mode", flooding might update every MN with an MN's current location information. Therefore, in PLS, when an MN receives a location request packet, it propagates the flooded location request if it is not the queried MN. If it is the queried MN, it transmits a reply packet with its current location information to the source of the location request.

4.3 Parameter Selection

Section 4.2 defined the PLS model and its parameters. This section discusses the parameter choices and the effects of those choices.

- E

  E controls the number of entries in an LP. It is set to 25, which is the value used in [13]. The overhead of PLS can be reduced by reducing E.

- X_1

  X_1 decides how far from its predicted location an MN can move before sending an update LP. X_1 is set to one because we used the random waypoint mobility model [39] in our simulation. Thus, an MN's velocity does not change before it reaches its destination. If an MN's path can be a curve, then X_1 should be a larger number.

- $maxUpdateTime$

  In SLS, the *maxUpdateTime* is set to 13 seconds to force the MNs to send location updates every 13 seconds [13]. We used the same value in PLS.

- $locationGoodTime$

  Location entries older than t seconds are considered old and deleted from the location table, where time t is given by Equation 4.1. According to SLS, location information is considered outdated and deleted if it is more than 46 seconds old [13]. We used the same value in PLS.

- *entryRemoveFactor*

  The variable *entryRemoveFactor* is used in PLS to keep the entries of slow MNs from being removed from the location table. *entryRemoveFactor* was set to be two times the transmission range. Thus, the information of MNs at high speeds (e.g., 15 m/sec or 20 m/sec) will be removed after *locationGoodTime* seconds, while the information of MNs at low speeds (e.g. 1 m/sec) will stay in the location table more than four times longer than *locationGoodTime* seconds.

- *predictionFactor*

  predictionFactor is used to prevent over-predicting. Figure 4.1 shows an example of PLS over-predicting an MN's location. As shown, the MN changes its velocity after it reaches Destination 1, and the predicted location of the MN is very far from its actual location. In this scenario, the best/closest prediction we can make is Destination 1. Thus, the goal is to predict the location of the MN's destination, i.e., we want to predict a path no longer than the distance between the position where the MN sent the last location update and its destination. If we assume these two locations are two random points in the network, then the *predictionFactor* would be the expected distance between two path endpoints within the network, which is 241.4316 according to equation 3.8. However, the results of many simulation trials indicate that 100 is the best value for *predictionFactor* instead of 241.4316. This difference arises because in the steady-state random waypoint mobility model [40], MNs are clustered in the center of the network. Consequently, the distance between these two locations, the point where the MN sent the last location update and the MN's destination, are shorter than the distance between two random points in the network.

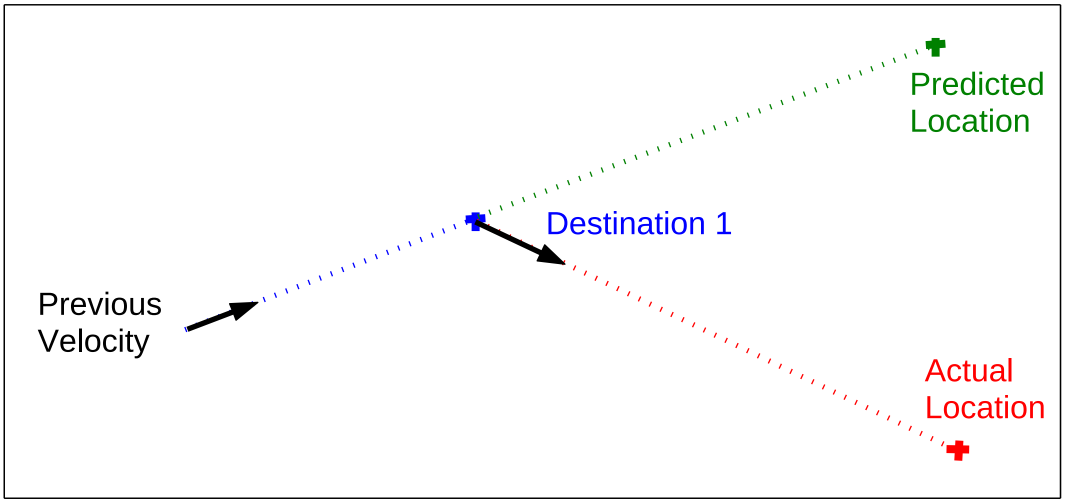


FIG. 4.1. An Example of Over-Predicting an MN's location

- X_2

  X_2 was decided through many simulations. It is set to be $\frac{100}{3}$.

4.4 Evaluation of PLS

Simulations were done in ns2 (version 2.1b8a) [37]. The input parameters are listed in Table 3.2. We evaluated three studies via simulation:

- Study 1: Comparison of SLS and PLS (NULL MAC)
  - Study 1.1 Network Density (10 m/sec)
  - Study 1.2 Speed (50 MNs in the network)
- Study 2: Comparison of PLS, SLS, DLS, and RLS (with 802.11 MAC)
  - Study 2.1 Network Density (10 m/sec)

  - Study 2.2 Speed (50 MNs in the network)

- Study 3: Network Complexity (with 802.11 MAC)

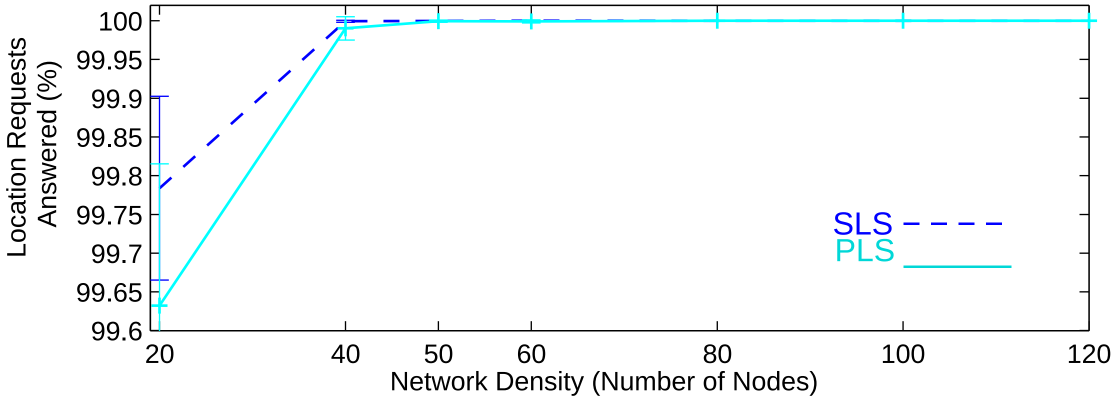
4.4.1 Study 1.1 – Network Density with NULL MAC

Because PLS is based on SLS, we first compared PLS to SLS with a NULL MAC layer. Therefore, in this study, SLS and PLS are compared theoretically. The results are plotted in Figure 4.2 and Figure 4.3.

Figure 4.2 shows the performance (the percentage of location requests answered and average location error) of SLS and PLS. Figure 4.2 (a) shows that both SLS and PLS were able to answer many of the location requests. Except for 20 MNs in the network, the percentage of location requests answered for PLS and SLS were both approximately 100%. Lower performance at fewer MNs can be expected because location information propagates slower in a sparse network where each MN has few neighbors. We note that at 20 MNs in the network, the confidence intervals of SLS and PLS overlap. Figure 4.2 (b) shows that PLS had lower location error than SLS. This result exists because PLS predicts the MNs' locations. The location errors of SLS and PLS at 20 MNs in the network were significantly higher than denser networks, which is also because of the slower propagation of location information in a sparse network.

Figure 4.3 shows the overhead of SLS and PLS, and we see that PLS has lower overhead than SLS. Figure 4.3 (a) shows the protocols' overhead in terms of number of location packets sent per node, per query answered. We find that PLS sent less packets than SLS per location request answered. We recall that the update conditions of SLS are every time an MN moves 25 meters away from where it sent its previous update, or at least every 13 seconds. The update conditions of PLS are every time

(a) Location requests answered vs. density, with NULL MAC.



(b) Error of location responses vs. density, with NULL MAC.

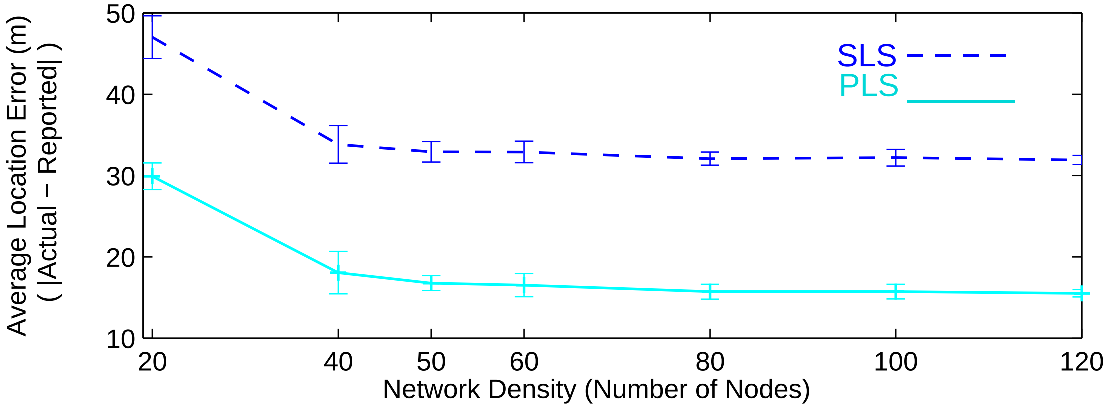
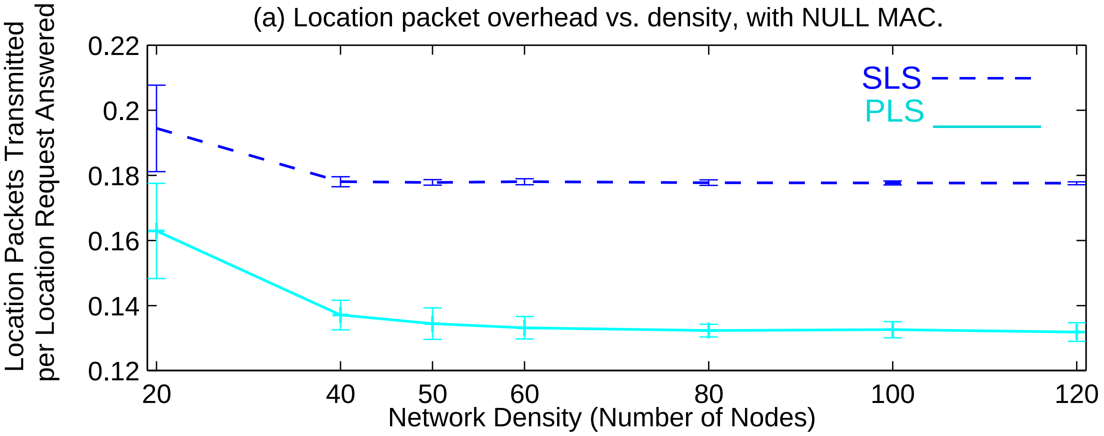


Fig. 4.2. Study 1.1 – Performance Comparison of PLS and SLS with NULL MAC in a Network with Speed 10 m/sec

an MN moves $\frac{100}{3}$ meters away from where it sent its previous update, every time an MN moves 1 meter away from its predicted location, or at least every 13 seconds. More LPs were transmitted in a network of 20 MNs because more flooding packets were sent due to the low location table hit ratio at low network density.

Figure 4.3 (b) shows the protocols' overhead in terms of bytes sent per node, per location request answered. Again, we find that PLS sent less bytes than SLS per location request answered. Fewer bytes were sent in a network of 20 MNs because


(a) Location packet overhead vs. density, with NULL MAC.

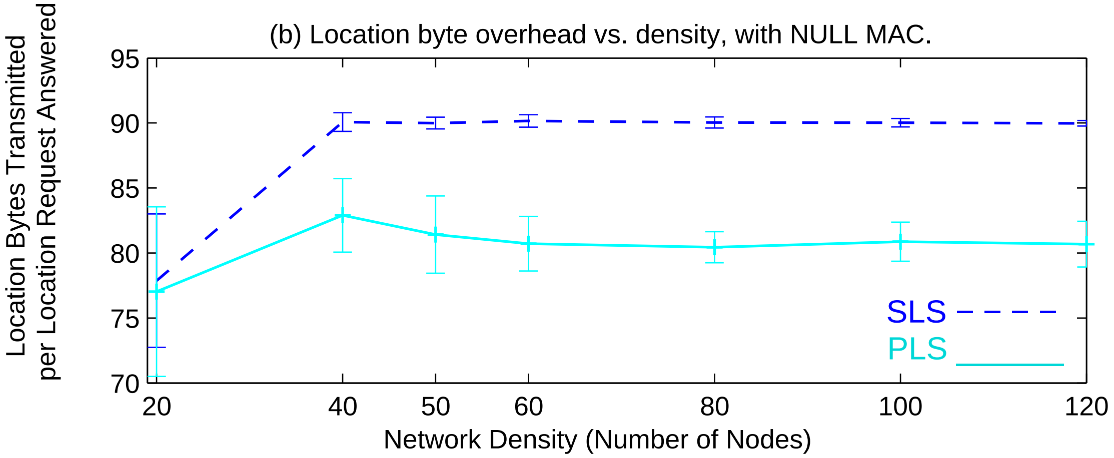
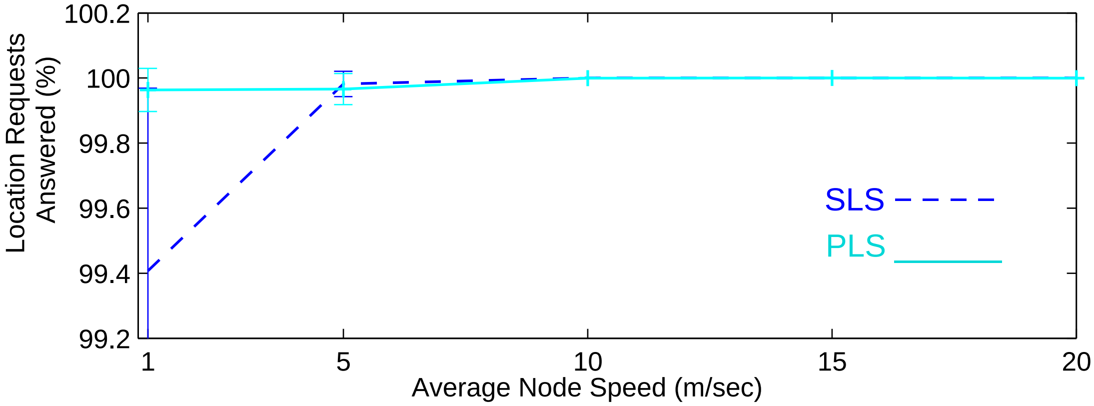

(b) Location byte overhead vs. density, with NULL MAC.

FIG. 4.3. Study 1.1 – Overhead Comparison of PLS and SLS with NULL MAC in a Network with Speed 10 m/sec

the LP size in a network of 20 MNs is smaller.

4.4.2 Study 1.2 – Speed with NULL MAC

The simulation results for SLS and PLS with a NULL MAC as speed increases are plotted in Figure 4.4 and Figure 4.5. The results show that PLS performed better (less location error, higher success rate, and lower overhead) than SLS most of the times.


(a) Location requests answered vs. speed, with NULL MAC.

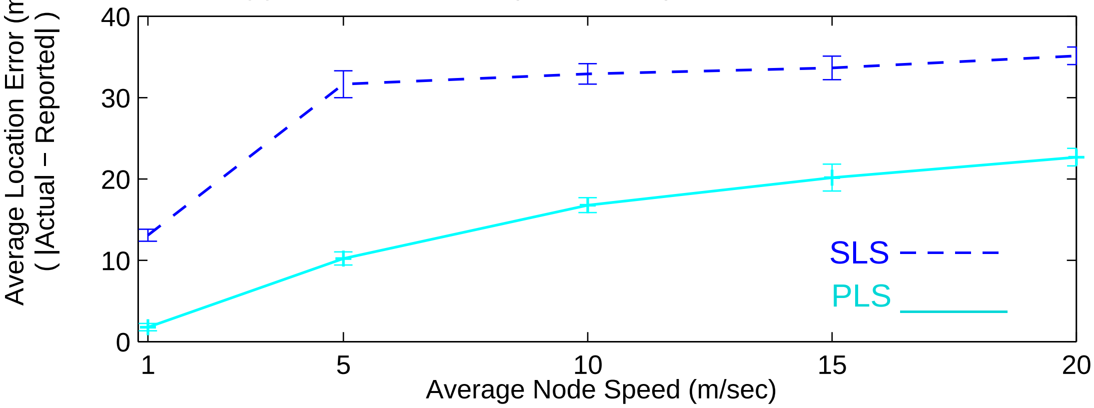

(b) Error of location responses vs. speed, with NULL MAC.

Fig. 4.4. Study 1.2 – Performance Comparison of PLS and SLS with NULL MAC in a Network with 50 MNs

Figure 4.4 shows the performance (the percentage of location requests answered and average location error) of SLS and PLS. Figure 4.4 (a) shows that except for 1 m/sec, the percentage of location requests answered for PLS and SLS are both approximately 100%. The percentage of location requests answered for SLS at 1 m/sec was much lower and the confidence intervals were large. At lower speed, a partitioned network gets re-connected after a longer period. PLS's performance was affected less by the partitioned network problem, because it keeps location information for slower MNs for a longer period of time. Figure 4.4 (b) shows that PLS has less location error than SLS.

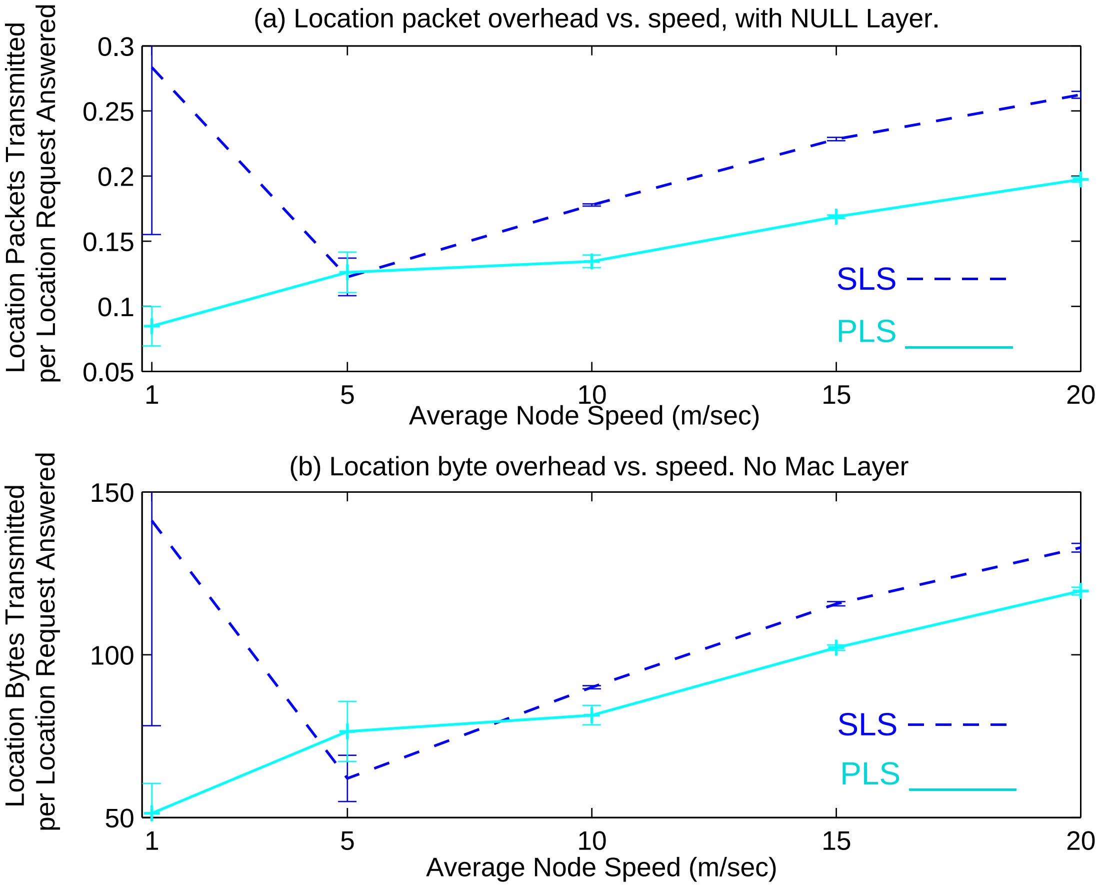


Fig. 4.5. Study 1.2 – Overhead Comparison of PLS and SLS with NULL MAC in a Network with 50 MNs

Figure 4.5 shows the overhead of SLS and PLS. We find that PLS has lower overhead (packets and bytes) than SLS. The overhead of SLS was much higher at 1 m/sec because of the partitioned network problem, as discussed in Figure 4.4 (a).

4.4.3 Study 2.1 – Network Density with 802.11 MAC

In this study we evaluate the performance of PLS with four other location services as the number of nodes increases. See Chapter 3 for details of SLS, DLS, GLS, and RLS.

In this study, we fixed the speed at 10 m/sec and increased the number of MNs (20, 40, 60, 80, 100, 120, 150) in an 802.11 MAC network. The results are plotted in Figure 4.6 and Figure 4.7. We found that a dense network of 150 MNs is very unstable; the confidence intervals for the results are huge because the results of each simulation highly depended on the movement of the MNs (mobility file). Therefore we only plotted the results up to 120 MNs in Figure 4.6 and Figure 4.7.

Figure 4.6 shows the performance (the percentage of location requests answered and average location error) of PLS and four other location services. Figure 4.6 (a) shows that the location requests answered for PLS is almost 100% with all node densities. The location requests answered of some other location services decreased when the density of the network increased, i.e., GLS and RLS. Therefore we conclude that PLS is more scalable than GLS and RLS. Figure 4.6 (b) shows that the location error of PLS is the lowest among the five compared location services. In other words, the performance of PLS is the best.

Figure 4.7 shows the overhead of PLS with four other location services. Because RLS's overhead is much higher than the other protocols, we decided not to plot all of the RLS results. Therefore, only parts of the RLS overhead can be seen in the figure.

PLS has the lowest number of location packets transmitted per location request answered (slightly lower than SLS). In networks of less than 120 MNs, the byte overhead of DLS is lower than PLS. However, DLS's byte overhead increased as the node density increased while PLS's byte overhead remained constant. At 120 MNs, PLS has slightly lower byte overhead than DLS.

4.4.4 Study 2.2 – Speed with 802.11 MAC

In this study we evaluate the performance of PLS with four other location services as the speed increases. See Chapter 3 for details of SLS, DLS, GLS, and RLS. The simulation results are plotted in Figure 4.8 and Figure 4.9.

Figure 4.8 shows that PLS provided the highest percentage of location requests answered and the lowest location error. These results match the results of study 2.1.

Figure 4.9 shows that PLS had the lowest overhead in terms of number of packets, and the second lowest overhead in terms of bytes. DLS had the lowest overhead in term of bytes because each DLS LP only contains location information for the sending MN, while each PLS LP also contains location information for other MNs.

Overall, PLS provided the best performance. The location error and packet overhead for PLS were the lowest. Therefore, PLS is preferred over the other location services.

4.4.5 Study 3 – Network Complexity with 802.11 MAC

Table 4.1 lists the input parameters for this study where we increased the network difficulty along the x-axis.

The results are plotted in Figure 4.10 and Figure 4.11.

- GLS

  We noticed from Figure 4.10 (a) that GLS's query success rate was low when the network density was either high or low. The reason why GLS does not perform well at low network density is due to the GLS "location server" mechanism: if an MN needs location information for another MN and it is not found in the

Table 4.1. Simulation Input Parameters for Trials

Trial number	Speed	Number of MNs
1	1 m/sec	40
2	5 m/sec	60
3	10 m/sec	80
4	15 m/sec	100
5	20 m/sec	120

querying MN's location table, the querying MN has to obtain the information by querying the MN's location servers. In networks of low density, there might be no route to the nearest location server. As simulated in [23], in a network of 100 nodes within an 2000 m × 2000 m area with transmission range of 250 m (which is equivalent to 28.125 nodes in our network settings), 60% of all query failures are due to "no route". In other words, GLS forwarding fails because there is no option for a next-hop MN at a forwarding MN. The reason why GLS does not perform well at high network density is due to the network congestion caused by the large amount of update packets, as shown in Figure 4.11. Figure 4.10 (b) shows that GLS had large location errors at each network complexity. Therefore, we conclude that GLS is not robust and overall performs poorly.

- RLS

We note in Figure 4.10 (b) that RLS's location error increased very quickly with the increasing network complexity. Figure 4.10 (a) shows that RLS's query success rate dropped with the increasing network complexity. Figure 4.11 shows that RLS had huge overhead (packets and bytes) in complex networks. In fact, RLS's byte overhead is high in non-complex networks. Therefore, we conclude that RLS should not be used in MANETs unless the data rate is very low.

- DLS

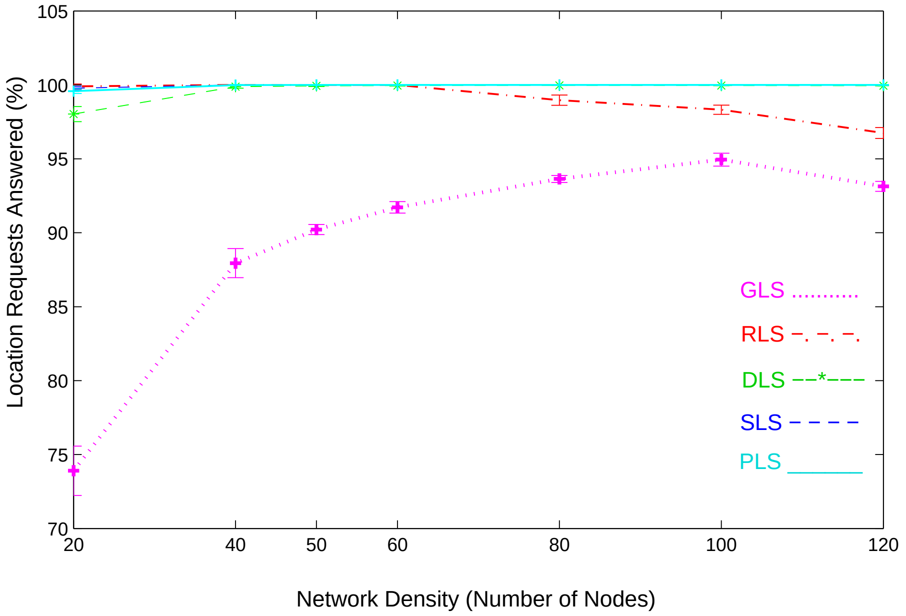
  As shown in Figure 4.10 and Figure 4.11, DLS had higher query success rate and lower location error and overhead (packets and bytes) than GLS and RLS in complex networks. In addition, DLS's location error increased slowly as the network complexity increased. Therefore, we conclude that DLS is relatively robust as the network complexity increases.

- SLS and PLS

  As shown in figure 4.10 (a) and Figure 4.11 (a), SLS and PLS had the highest query success rate and the lowest packet overhead. The packet overheads of PLS and SLS overlapped in all trials, except trial two; PLS had slightly lower overhead (packets and bytes) than SLS in trial two. As shown in Figure 4.10 (b), PLS had the lowest location error among all five protocols.

As a result, we conclude that PLS performed the best in Study 3.

(a) Location requests answered vs. density, with 802.11 MAC.



(b) Error of location responses vs. density, with 802.11 MAC.

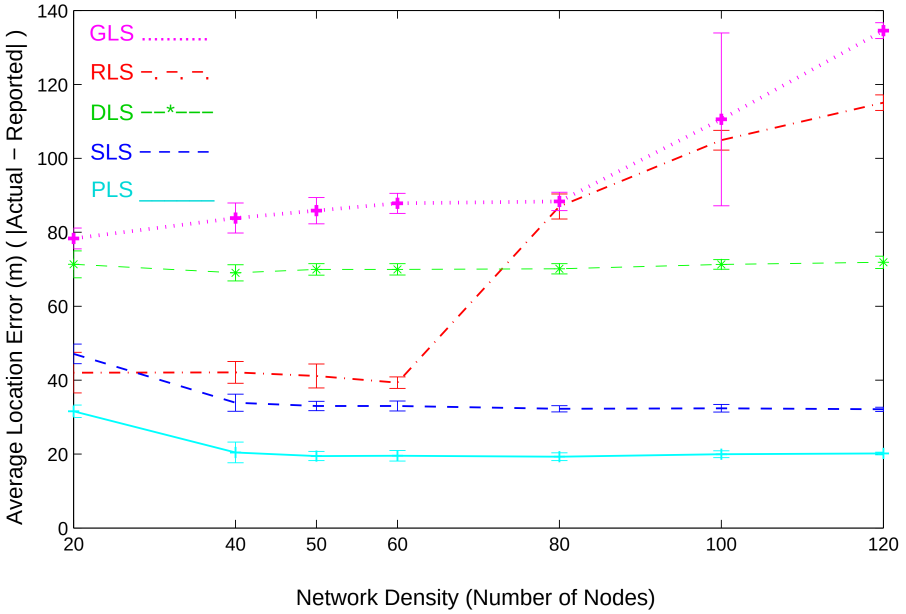
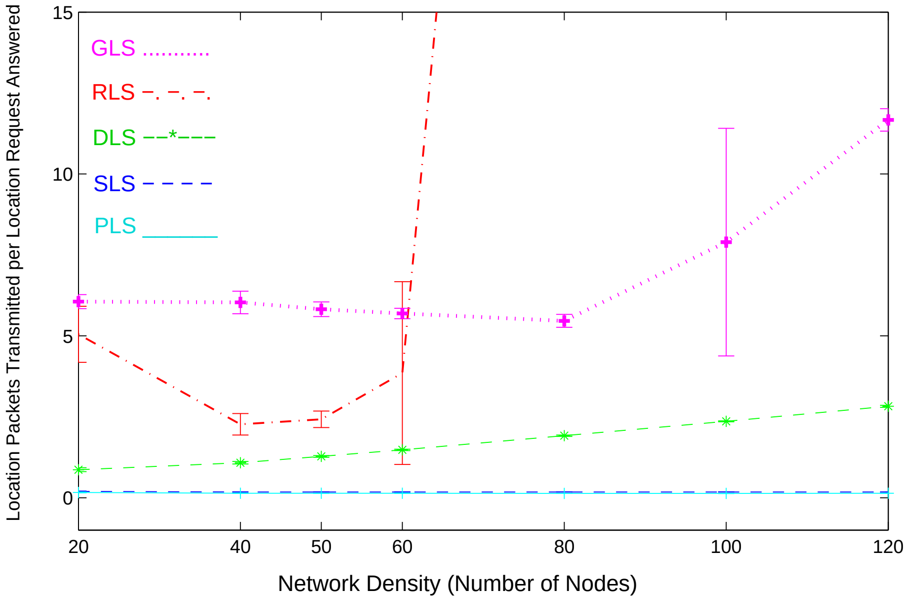


FIG. 4.6. Study 2.1 – Performance Comparison of PLS and Other Location Services in a Network with Speed 10 m/sec

(a) Location packet overhead vs. density, with 802.11 MAC.



(b) Location byte overhead vs. density, with 802.11 MAC.

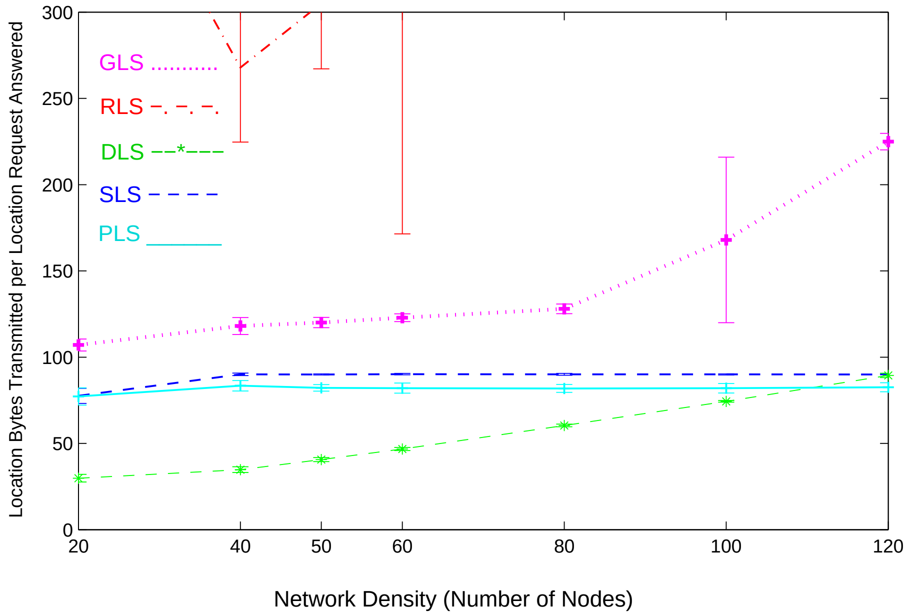
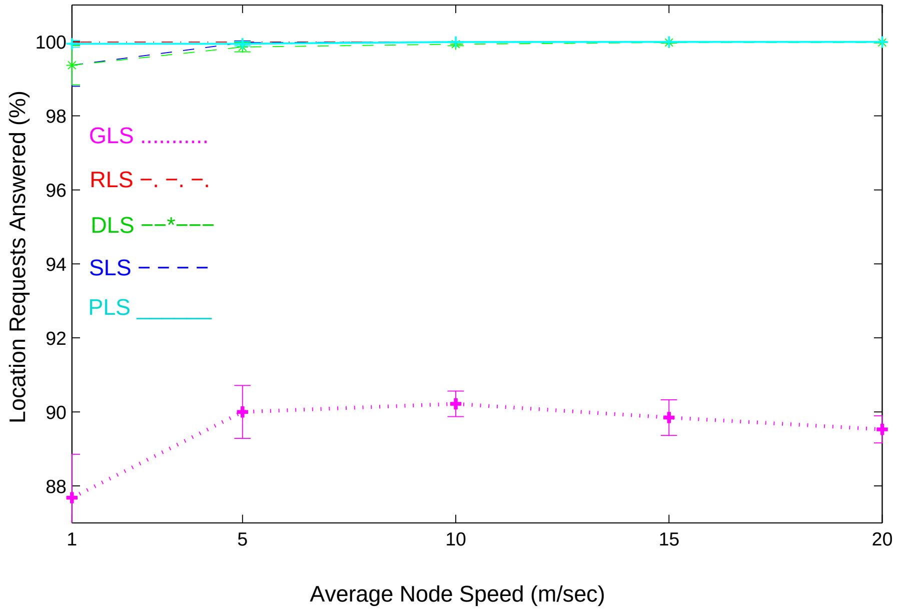


Fig. 4.7. Study 2.1 – Overhead Comparison of PLS and Other Location Services in a Network with Speed 10 m/sec


(a) Location requests answered vs. speed, with 802.11 MAC.

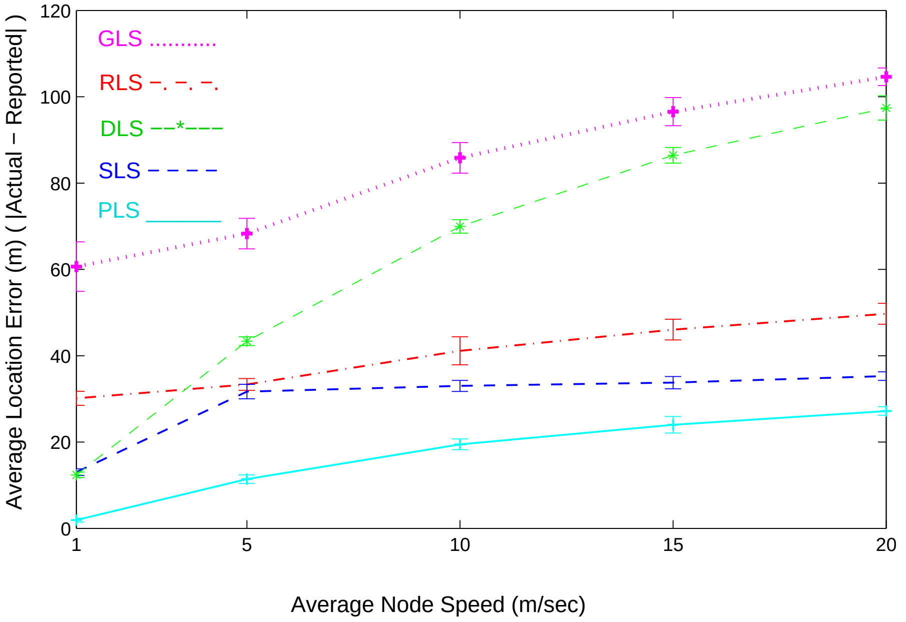
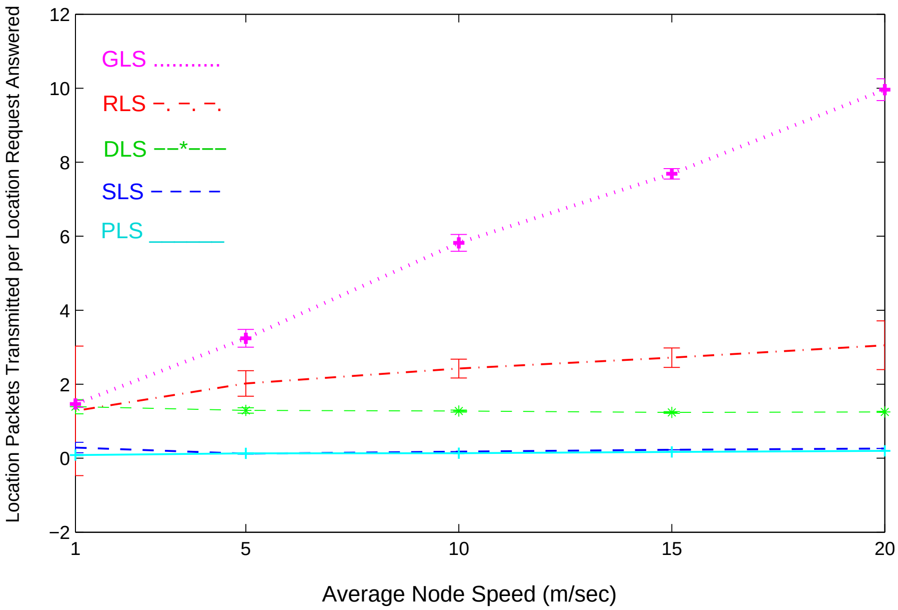
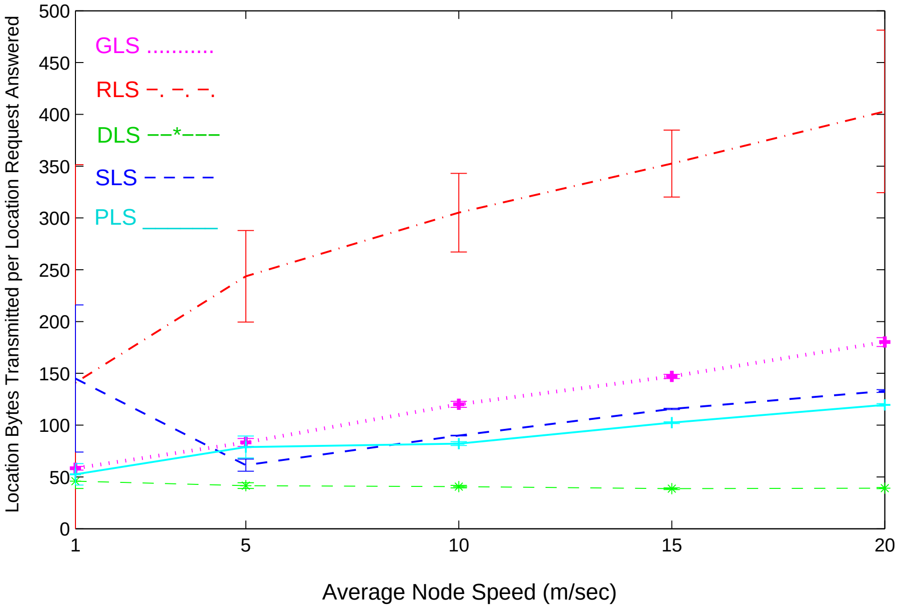

(b) Error of location responses vs. speed, with 802.11 MAC.

Fig. 4.8. Study 2.2 – Performance Comparison of PLS and Other Location Services in a Network with 50 MNs



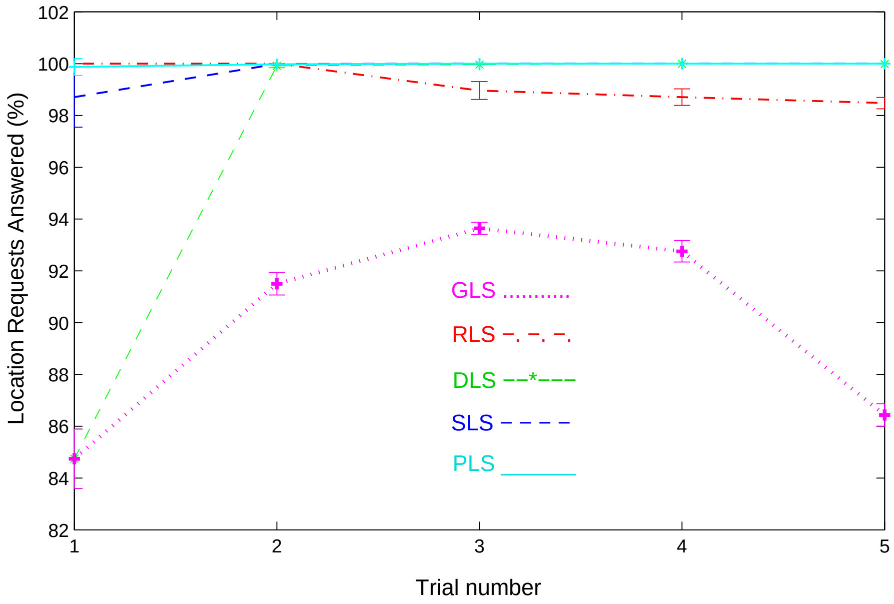
(a) Location packet overhead vs. speed, with 802.11 MAC.



(b) Location byte overhead vs. speed, with 802.11 MAC.

FIG. 4.9. Study 2.2 – Overhead Comparison of PLS and Other Location Services in a Network with 50 MNs

(a) Location requests answered vs. network complexity, with 802.11 MAC.



(b) Error of location responses vs. network complexity, with 802.11 MAC.

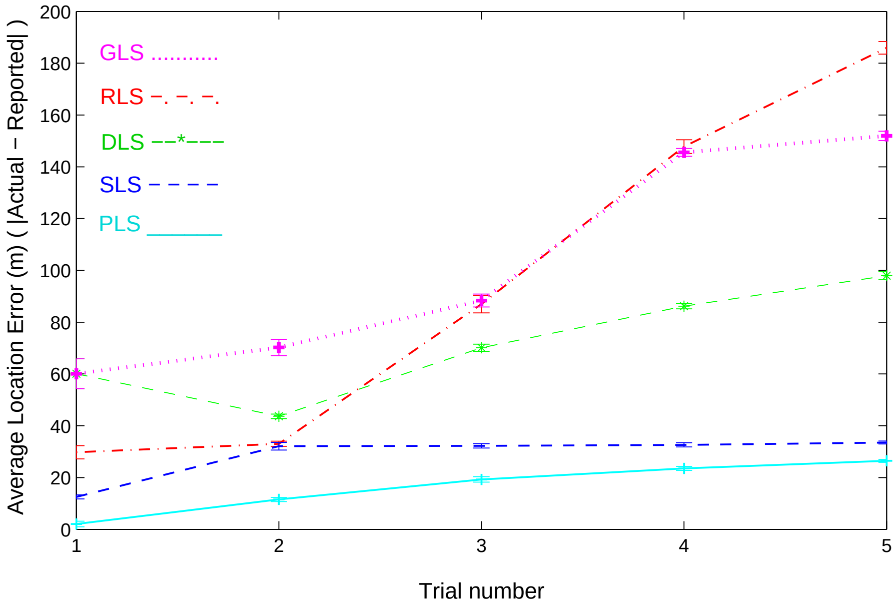
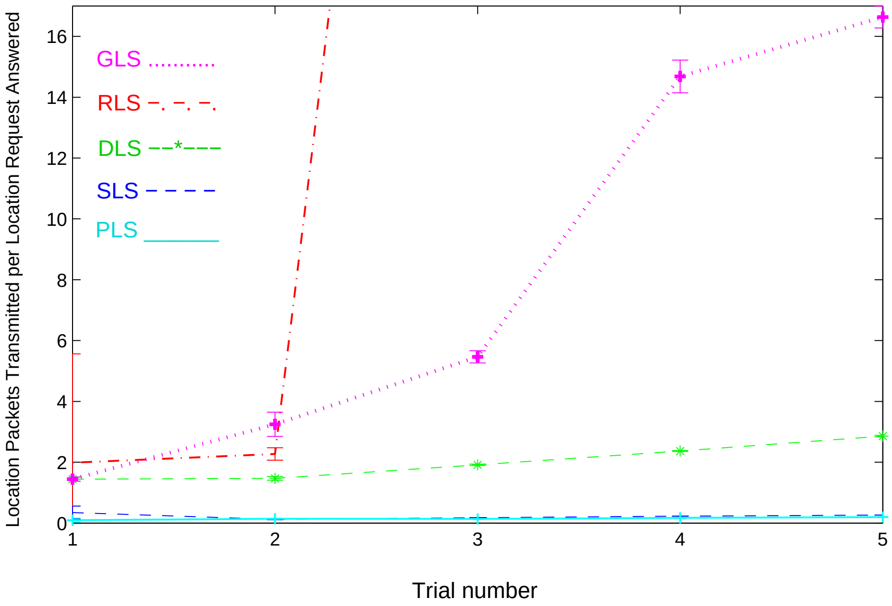


FIG. 4.10. Study 3 – Performance Comparison of PLS and Other Location Services in a Network with Increasing Network Difficulty

(a) Location packet overhead vs. network complexity, with 802.11 MAC.



(b) Location byte overhead vs. network difficulty.

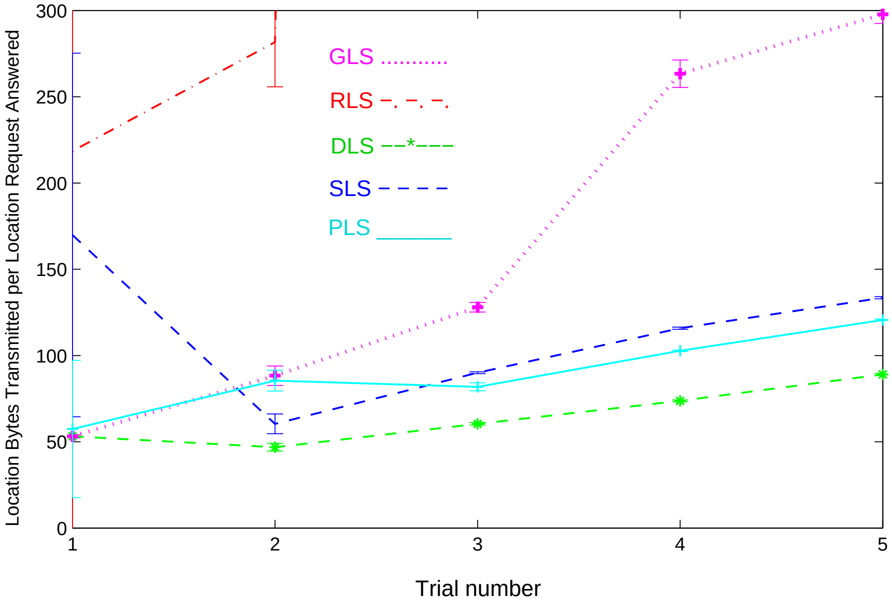


Fig. 4.11. Study 3 – Overhead Comparison of PLS and Other Location Services in a Network with Increasing Network Difficulty

Chapter 5

A METHOD TO PREDICT VELOCITY FOR PLS

5.1 Introduction

The goal of PLS is to use an MN's velocity to predict its location. The goal of this chapter is to predict velocity more accurately for use in PLS.

The most common mobility model used in a MANET is the random waypoint mobility model [39, 40]. One concern with this model is the straight movement pattern created by the MN to the next chosen destination [41]. In real life, an MN might not move in a straight line. Figure 5.1 illustrates an example: the black line is a straight line between the starting point and the destination, and the dotted curve is the path that an MN could take. It shows that an MN's average velocity can be very different than the velocity at one point on the MN's path.

Suppose the recorded previous velocity of the MN is V_1. If we use PLS and predict the MN's position by V_1, the result is marked as "prediction 1" in the figure. We note that "prediction 1" is very far from the MN's actual position. However, if the prediction uses the MN's average velocity then the prediction is marked as "prediction 2" in the figure, which is the same as the MN's actual position. In summary, making a prediction with only the latest updated location information may not be the best choice.

If we predict by taking the average of all the previous velocities, we might also obtain an inaccurate predictions. Figure 5.2 shows another example of an MN's movement. Suppose an MN moves from A to B, and then from B to C. Also suppose

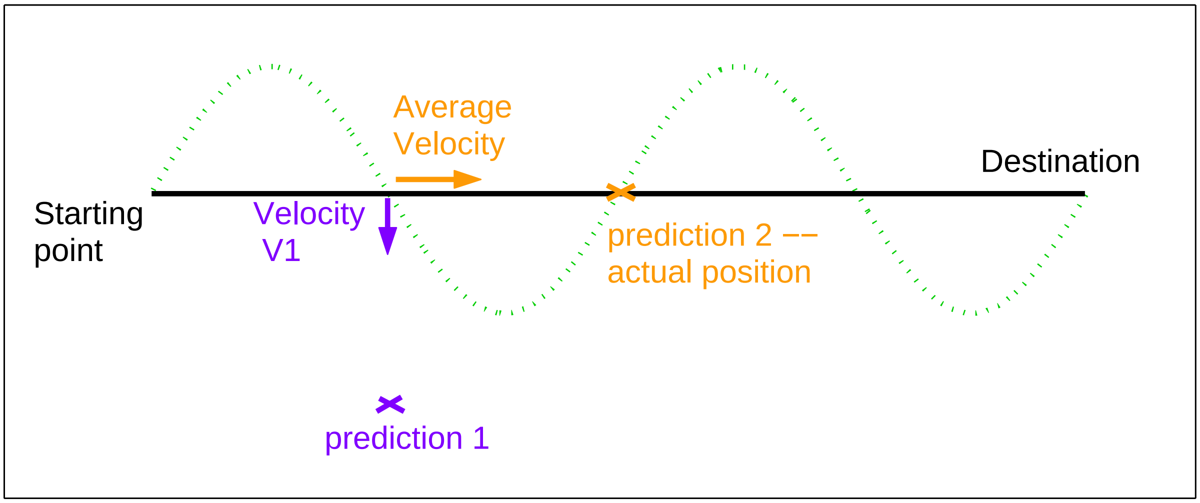


FIG. 5.1. Example of an MN's Movement Compared to the Random Waypoint Mobility Model

the MN transmits an update packet at time $t(n)$, $t(n-1)$, $t(n-2)$, $t(n-3)$ and $t(n-4)$. Clearly the velocities of the MN at time $t(n-3)$ and $t(n-4)$ do not help us predict the MN's velocity after t(n). Therefore, taking the average of all the previous velocities is not a good algorithm either. Furthermore, storing all the previous updates of other MNs takes a lot of memory space. This problem of predicting an MN's future velocity is discussed in this chapter, and a moving average method is proposed.

5.2 The Moving Average Velocity Prediction Model

The author of [42] presented a mobile tracking scheme that exploits the predictability of user mobility patterns in wireless PCS networks. The predictive scheme described in [42] uses the Gauss-Markov model which is complex to calculate and sensitive to violation of the mobility model assumptions.

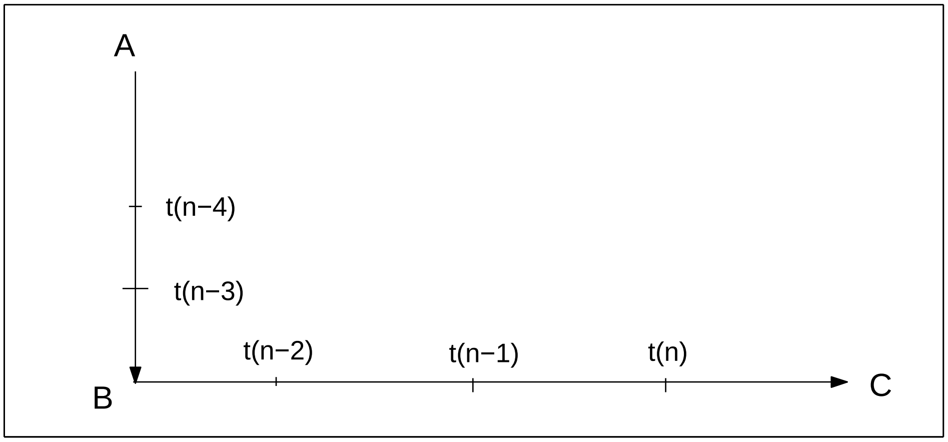


Fig. 5.2. Another Example of an MN's Movement

We propose a simple method, a moving average method, because an MN's current velocity is likely to be more correlated with recent past velocity. The predicted velocity of an MN after its last update time $t(n)$ is calculated as follows:

$$\widehat{v(n)} = \alpha v(n) + \alpha(1-\alpha)v(n-1) + \alpha(1-\alpha)^2 v(n-2) + \cdots + \alpha(1-\alpha)^n v(0), \quad (5.1)$$

where $v(0)$ is the first location information for an MN and $v(n)$ is the most recent location information; α is a parameter that is used to calculate the weights of the MN's previous updates and $0 < \alpha \leq 1$. Figure 5.3 shows an illustration of this velocity prediction model. The area of the grey squares are the weights that are multiplied with an MN's previous velocities. As shown, more recent velocities have larger weights than older velocities. For any value of α, the summation of the weights approaches 1 as n goes to infinity. If an MN moves at a constant speed, i.e., $v(0) = v(1) = v(2) = \cdots = v(n) = C$, then according to the equation above, $\widehat{v(n)} \approx C$. When the MNs' movements follow the random waypoint mobility model [39, 40], we use $\alpha = 1$ because we want to predict the MN's current velocity only by its last updated velocity.

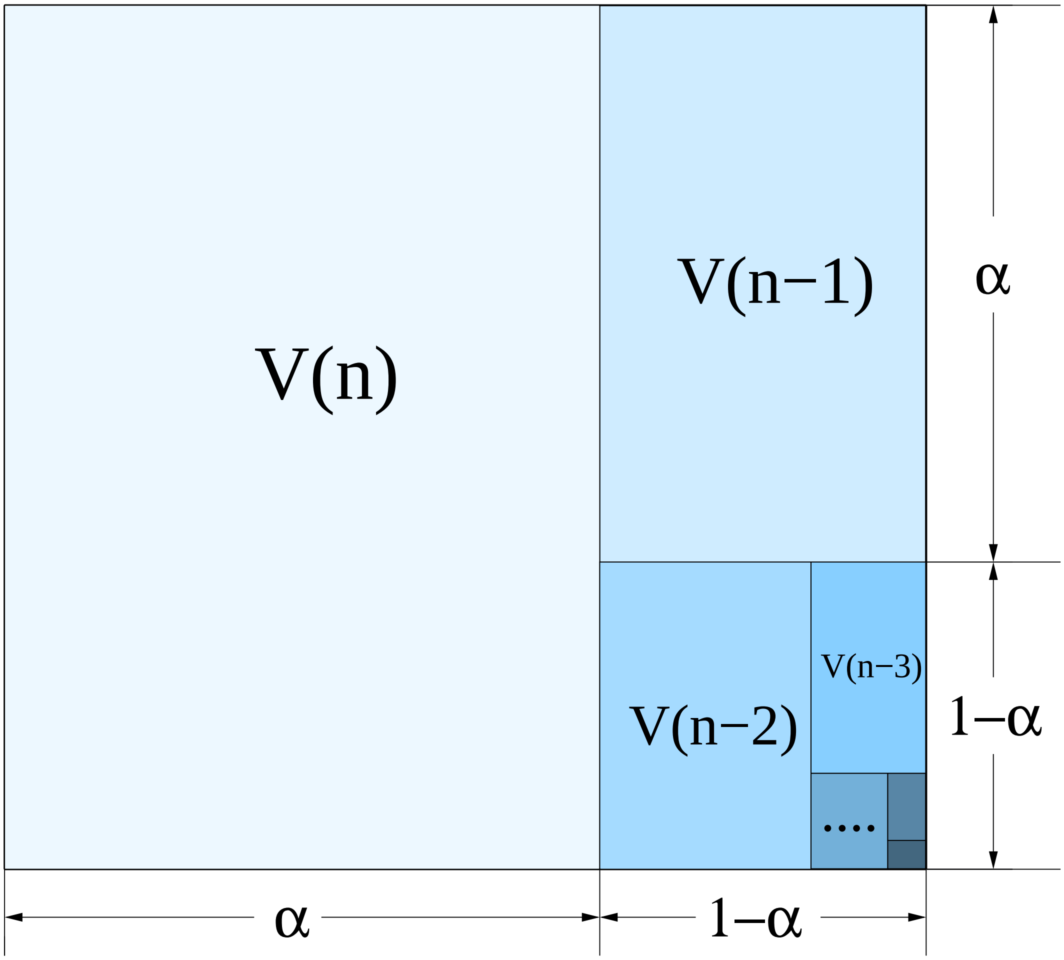


FIG. 5.3. Illustration of the Moving Average Velocity Prediction Model.

An MN's location at time $t(n) + \Delta t$ is calculated as follows:

$$\widehat{location(n} + \Delta t) = location(n) + \widehat{v(n)} \times \Delta t. \tag{5.2}$$

5.3 The Implementation Steps

The implementation steps for our moving average method are as follows:

- Each MN updates its current location information periodically.
- Each MN predicts other MNs' velocities by Equation 5.1 and predicts their locations by Equation 5.2 when needed. Equation 5.1 can be calculated in each step as $\widehat{v(k+1)} = \alpha\widehat{v(k)} + (1-\alpha)v(k+1)$. Accordingly, each time an MN receives a location packet, it calculates and stores the sender's current location and the sender's current predicted velocity in its location table. Thus, an MN need not save a sender's previous location information.
- Each MN updates its location table periodically.

We leave the comparison of PLS (which is described in Chapter 4) with this Moving Average Velocity Prediction Model as future work.

Chapter 6

CONCLUSIONS AND FUTURE WORK

Location information has recently been applied to MANET protocols. In recent years, many location based routing protocols have been developed for ad hoc networks [20]. Four existing location services were compared in this thesis: the Grid Location Service (GLS), the Simple Location Service (SLS), the DREAM Location Service (DLS), and the Reactive Location Service (RLS). All four of these location service protocols have relatively large location errors because they all use previously saved information when an MN's location is needed. Therefore, a new location service, the Predictive Location Service (PLS), was proposed.

A node using PLS predicts other nodes' states (location, velocity) by utilizing their previous states. The algorithm of PLS also adapts according to the nodes' speed. The simulation results show that PLS has lower overhead and lower location error than GLS, RLS, SLS and DLS. The results also demonstrated that PLS is robust, i.e., it can maintain good performance over a wide range of ad hoc networks with different densities and speeds. Hence, we conclude that our Predictive Location Service is preferred over GLS, SLS, RLS, and DLS.

A moving average method to predict an MN's velocity for PLS was also proposed. This method uses an MN's previous velocities to predict the MN's average velocity after its last update. This method can be used in real life, where MNs do not move in straight lines.

One avenue of future work is to probe PLS's scalability and the speed at which location information propagates by simulating PLS in larger networks. Also, the moving average velocity prediction method could be further analyzed by simulating with a mobility model that allows MNs to move in different patterns. Finally, the general idea of predicting MNs' location information could be applied to other location services protocols.

REFERENCES

[1] C. K. Toh. *Ad Hoc Mobile Wireless Networks*. Prentice Hall, Upper Saddle River, New Jersey, 2002.

[2] T. Camp, J. Boleng, B. Williams, L. Wilcox, and W. Navidi. Performance comparison of two location based routing protocols for ad hoc networks. In *Proceedings of IEEE INFOCOM*, pages 1678–1687, 2002.

[3] B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 243–254, 2000.

[4] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66, August 2001.

[5] E. Kaplan. *Understanding GPS*. Artech House, 1996.

[6] S. Čapkun, M. Hamdi, and J. P. Hubaux. GPS-free positioning in mobile ad-hoc networks. *Cluster Computing Journal*, 5(2):157–167, April 2002.

[7] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, 7(5):28–34, October 2000.

[8] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward. A distance routing effect algorithm for mobility (DREAM). In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 76–84, 1998.

[9] Y. Ko and N. H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 66–75, 1998.

[10] D. De Couto and R. Morris. Location proxies and intermediate node forwarding for practical geographic forwarding. Technical Report MIT-LCS-TR-824,, MIT Laboratory for Computer Science, June 2001.

[11] I. Stojmenovic, M. Russell, and B. Vukojevic. Depth first search and location based localized routing and QoS routing in wireless networks. In *Proceedings of the IEEE International Conference on Parallel Processing*, pages 173–180, 2000.

[12] J. Li, J. Jannotti, D. De Couto, D. Karger, and R. Morris. A scalable location service for geographic ad-hoc routing. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 120–130, 2000.

[13] T. Camp, J. Boleng, and L. Wilcox. Location information services in mobile ad hoc networks. In *Proceedings of the IEEE International Conference on Communications (ICC)*, pages 3318–3324, 2002.

[14] W. H. Liao, J. P. Sheu, and Y. C. Tseng. GRID: A fully location-aware routing protocol for mobile ad hoc networks. *Telecommunication Systems*, 18(1):37–60, 2001.

[15] J. P. Hubaux, J. Y. Le Boudec, S. Giordano, M. Hamdi, L. Blăzević, L. Buttyán, and M. Vojnović. Towards mobile ad-hoc WANs: Terminodes. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1052–1059, 2000.

[16] L. Blăzević, S. Giordano, and J. Y. Le Boudec. Self Organized Terminode Routing. *Cluster Computing Journal*, 5(3), April 2002.

[17] L. Blăzević, L. Buttyán, S. Čapkun, S. Giordano, J. P. Hubaux, and J. Y. Le Boudec. Self organization in mobile ad hoc networks: the approach of Terminodes. *IEEE Communications Magazine*, 39(6):166–174, June 2001.

[18] B. Karp. *Geographic Routing for Wireless Networks*. PhD thesis, Harvard University, 2000.

[19] I. Stojmenovic. Position based routing in ad hoc networks. *IEEE Communications Magazine*, 40(7):128–134, July 2002.

[20] M. Mauve, J. Widmer, and H. Hartenstein. A survey on position-based routing in mobile ad hoc networks. *IEEE Network*, 15(6):30–39, November/December 2001.

[21] S. Giordano, I. Stojmenovic, and L. Blăzević. Position based routing algorithms for ad hoc networks: A taxonomy. In X. Huang X. Cheng and D.Z. Du, editors, *Ad Hoc Wireless Networking*. Kluwer Academic Publishers, 2003.

[22] T. Camp. Location information services in mobile ad hoc networks. Technical report, Department of Math. and Computer Sciences, Colorado School of Mines, MCS-03-15, October 2003.

[23] H. Hartenstein, M. K*ä*semann, H. F*ü*ßler, and M. Mauve. A simulation study of a location service for position-based routing in mobile ad hoc networks. Technical report, Department of Science, University of Mannheim, TR-02-007, July 2002.

[24] M. K*ä*semann and H. Hartenstein and H. F*ü*ßler, and M. Mauve. Analysis of a location service for position-based routing in mobile ad hoc networks. In

Proceedings of the 1st German Workshop on Mobile Ad-Hoc Networks (WMAN), pages 121–133, 2002.

[25] N. Guba and T. Camp. Recent Work on GLS: a Location Service for an Ad Hoc Network. In *Proceedings of the Grace Hopper Celebration*, 2002.

[26] M. Käsemann, H. Füßler, H. Hartenstein, and M. Mauve. A reactive location service for mobile ad hoc networks. Technical report, Department of Science, University of Mannheim, TR-02-014, November 2002.

[27] J. Boleng. *Exploiting Location Information and Enabling Adaptive Mobile Ad Hoc Network Protocols.* Ph.D. dissertation, Colorado School of Mines, 2002.

[28] X. Jiang and T. Camp. An information dissemination protocol for an ad hoc network. In *Proceedings of the 23rd IEEE International Performance, Computing, and Communications Conference*, pages 337–345, 2004.

[29] Z. J. Haas and B. Liang. Ad hoc mobility management with randomized database groups. In *Proceedings of the IEEE International Conference on Communications (ICC)*, pages 1756–1762, 1999.

[30] Z. J. Haas and B. Liang. Ad hoc mobility management with uniform quorum systems. *IEEE/ACM Transactions on Networking*, 7(2):228–240, 1999.

[31] J. Luo, J. P. Hubaux, and P. T. Eugster. PAN: Providing reliable storage in mobile ad hoc networks with probabilistic quorum systems. In *Proceedings of the 4nd ACM Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 1–12, 2003.

[32] S. Giordano and M. Hamdi. Mobility management: the virtual home region. Technical Report SSC/1999/037, EPFL-ICA, October 1999.

[33] I. Stojmenovic. Home agent based location update and destination search schemes in ad hoc wireless networks. In A. Zemliak and N.E. Mastorakis, editors, *Advances in Information Science and Soft Computing*, pages 6–11. WSEAS Press, 2002. Also Technical Report, University of Ottawa, TR-99-10, September 1999.

[34] A. Agarwal and S.R. Das. Dead reckoning in mobile ad hoc networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1838–1843, 2003.

[35] V. Kumar and S.R. Das. Performance of dead reckoning-based location service for mobile ad hoc networks. *Wireless Communications and Mobile Computing*, 4(2):189–202, 2004.

[36] B. Ghosh. Random distances within a rectangle and between two rectangles. *Bulletin of the Calcutta Mathematical Society*, 43:17–24, 1951.

[37] M. Greis. Marc Greis' tutorial for the ucb/lbnl/vint network simulator "ns". http://www.isi.edu/nsnam/ns/tutorial/. Page accessed on April 13, 2004.

[38] The IEEE Working Group for Wireless Local Area Network Standards. IEEE wireless local area networks. http://grouper.ieee.org/groups/802/11/. Page accessed on April 13, 2004.

[39] W. Navidi and T. Camp. Stationary distributions for the random waypoint mobility model. *IEEE Transactions on Mobile Computing*, 3(1):99–108, January-March 2004.

[40] W. Navidi, T. Camp, and N. Bauer. Improving the accuracy of random waypoint simulations through steady-state initialization. In *Proceedings of the 15th*

International Conference on Modeling and Simulation, pages 319–326, March 2004.

[41] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communication* & *Mobile Computing* , 2(5):483–502, 2002.

[42] B. Liang and Z. J. Haas. Predictive distance-based mobility management for PCS networks. In *Proceedings of IEEE INFOCOM*, pages 1377–1384, 1999.

Appendix A

GLOSSARY OF TERMS

- DLS – Dream Location Service
- DREAM – Distance Routing Effect Algorithm For Mobility
- GLS – Grid Location Service
- GPS – Global Positioning System
- LAR – Location Aided Routing
- LP – Location Packet
- MANET – Mobile Ad hoc NETwork
- MN – Mobile Node
- PLS – Predictive Location Service
- RLS – Reactive Location Service
- SLS – Simple Location Service
- TTL – Time To Live