PREDICTION-BASED AUTHENTOCATION FOR MOBILE AD HOC NETWORKS

By

SALMAN LATIF

Bachelor of Electrical Engineering

N.E.D University of Engineering & Technology

Karachi, Pakistan

2000

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December 2006

PREDICTION-BASED AUTHENTOCATION FOR

MOBILE AD HOC NETWORKS

Thesis Approved:

Dr. Johnson Thomas
Thesis Advisor

Dr. John P. Chandler

Dr. Debao Chen

A. Gordon Emslie
Dean of the Graduate College

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to my thesis advisor, Dr. Johnson P. Thomas, for his intelligent supervision, constructive guidance, constant inspiration, assistance, and valuable time he gave me throughout my thesis work. I would also like to thank my other committee members, Dr. John P. Chandler and Dr. Debao Chen, for all the help and support they have given to me.

I would like to extend special appreciation and sincere gratitude to my parents, who have always been a great source of motivation for me throughout my life. I have no words to thank them for their love, confidence, support, and encouragement. I am also grateful to my friends specially Mustafa Hilal Qureshi and colleagues for their invaluable suggestions and support.

PREFACE

Ensuring that a path in a mobile ad hoc network is secure by means of authenticating every node in the route carries considerable overhead and results in packet loss. In this thesis we propose a prediction mechanism to determine a new link when the existing the link is fading. Once the predicted node has been determined it is authenticated. The objective of this research is to enable prediction and authentication to be completed before the current link breaks. Simulation results show that the proposed approach results in fewer packets being dropped, while ensuring a secure route. The proposed approach is compared to traditional protocols such as DSDV which do not employ any form of prediction. The prediction scheme we are using is based on the location based routing protocol LTR.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

LIST OF GRAPHS

Graph Page

CHAPTER I

INTRODUCTION

1.1 Introduction

Mobile Ad Hoc Networks (MANETs) are autonomous systems of mobile nodes (hand held user devices) interconnected by wireless links [10]. Intermediate mobile nodes act as mobile routers to support communication to other mobile nodes that are out of each others range. Although, originally designed for military purposes, the inherent flexibility of these networks is also appealing for various commercial applications such as convention meetings, electronic classrooms, and search and rescue operations.

In MANETs, nodes move in an arbitrary manner and can join or leave the network at any time. This host mobility may trigger unpredictable topology changes frequently. In order to facilitate communication within the network, a routing protocol is used to discover routes between nodes [8]. The primary goal of a routing protocol is to deliver messages with accuracy via the most efficient route so it all takes place in a timely manner.

There are many protocols which have already been proposed for MANET's efficient routing. In general, MANET protocols can be categorized into two types; on-demand and Pro-active protocols.

On-demand (Source-Initiated) protocols create routes only when desired by the source node [9]. When a node requires a route to a specific destination, a route exploring process is initiated within the network. Once a particular route has been established, it is maintained until the route is no longer required. The demand-driven routing protocols include: AODV, DSR, TORA, ABR, SSR [3][5][7]. These routing protocols do not need to maintain routing tables, but instead, have the overhead of route discovery.

Unlike on-demand protocols that create routes only when desired, the Pro-active protocols attempt at maintaining consistent and up-to-date routing information for all nodes in the network [9]. This is typically achieved through maintaining a set of routing tables. Alterations in network topology happen because of mobility or propagating updates throughout the network at periodic intervals to maintain a consistent view caters for node failures. These protocols include: DSDV, CGSR, WRP [6][7]. The major disadvantage of table-driven routing protocols is that each node needs to send messages to its neighborhood continuously to keep their routing tables updated. This may cause network traffic overload.

There exist algorithms and protocols that attempt to improve performance by using link state information. The simulation results reported in several papers [1][2][7] demonstrate that normally demand-driven routing protocols have higher packet delivery ratio and need less routing messages than table-driven routing protocols.

[8] suggested a new table driven routing protocol called Location Triggered Routing protocol (LTR). In LTR, instead of each node sending and receiving messages periodically to maintain its routing table, no messages need to be sent unless a node detected has a location change and has impacted at least one network route [8]. This results in a significant reduction in number of routing messages.

Most existing protocols do not take account of security. Some existing protocols do take care of security, but these protocols use encryption which does not solve all the problems. Although encryption may help in protecting messages, an intermediate node in the route may act maliciously and simply refuse to forward messages or start dropping packets. For highly sensitive applications, it may be necessary for each node in the route to be authenticated. Existing routing protocols do not deal with this issue. There are overheads associated with authentication. When a routing protocol starts authenticating new nodes in a route that has just been set up, packets are lost because it takes time to authenticate the other nodes.

In this thesis, we propose a lightweight authentication protocol to authenticate nodes in a route. To reduce the packet loss during the authentication process, we extend the LTR protocol. Our approach is to predict the next node in the path if a link is about to break and use the prediction information to authenticate and hence reduce the packet loss. The prediction scheme we are using is based on the location based routing protocol LTR.

We are using one-way hash chaining for authentication in our scheme to build a trusted route between nodes. One-way hash chaining scheme has recently become very popular in resource constrained mobile devices as they compute one-way hash functions within milliseconds [13] for the secured networks

In Chapter 2 we introduce the LTR protocol. We discuss objectives of this thesis in chapter 3. Chapter 4 outlines our prediction algorithm and suggested specific enhancements to the LTR protocol. Finally, we do simulations to analyze LTR extended with our proposed Prediction Algorithm and compare our proposed approach with a table driven routing protocol, particularly in terms of packet loss.

CHAPTER II

REVIEW OF LITERATURE

The MANET routing protocols may be generally categorized as table-driven and source initiated on-demand driven. The table-driven routing protocol consistently updates routing information from each node to every other node in the ad hoc network. On the other hand, on–demand routing protocol is always based on a query-reply approach.

2.1 Table-driven routing protocols

Table-driven protocols attempt to maintain consistent and up-to-date routing information for all nodes in the network [9]. This is typically achieved through maintaining a set of routing tables. Alterations in the network topology brought about by mobility or propagating updates throughout the entire network serve to maintain a live outlook for node failures so that new connections may be made instantly when a certain node fails.

These protocols include: DSDV, CGSR, WRP [6][7]. The major disadvantage of table-driven routing protocols is that each node needs to send messages to the entire network consistently to keep their routing tables updated which may cause network traffic overload.

2.1.1 Destination-Sequenced Distance-Vector Routing (DSDV)

Destination Sequenced Distance Vector (DSDV) protocol is a table driven (proactive) protocol. Each node maintains routing information for all known destinations inside ad hoc network. It updates routing information periodically by itself. In comparison with DSR, it also maintains routes which have not been used before.

Destination	Next	Metric	Seq. Nr	Install Time
A1	A1	0	A-550	001000
B8	B8	1	B-102	001200
C3	B8	3	C-588	001200
B2	B8	4	B-312	001200

Table 2.1: DSDV (Table Entries initialized at A1)

The DSDV driven (proactive) protocol serves as a means of quickly updating the network as well as storing up-to-date information about each node. Each node is to maintain routing information for all known destinations inside the ad hoc network. It will update routing information periodically by an intrinsic mechanism. This is achieved typically through maintaining a set of routing tables. The network is a dynamically changing entity due to mobility or propagating updates which allow early node failure detection.

2.2 Demand-driven routing protocols

Demand driven is based on a query-reply approach [11]. In on-demand routing protocols, packets have to wait until a route to the new destination is discovered. On-demand (Source-Initiated) protocols create routes only when desired by the source node [9]. When a node requires a route to a destination, a route discovery process is initiated within the network. Once a route has been established, it is maintained until the route is no longer required. The demand-driven routing protocols include: AODV, DSR, TORA, ABR, SSR [3][5][7]. The demand-driven routing protocols do not need maintain routing tables, but instead, have the overhead of route discovery.

2.2.1 Dynamic Source Routing (DSR)

One of the well-known On-demand (pro-active) routing protocols is Dynamic Source Routing (DSR). The DSR is a simple and efficient routing protocol designed specifically for use in MANETs. DSR allows the network to be completely independent and self-configuring. These types of routing protocols do not maintain a table of routes to all the nodes in the network. In DSR, a list of neighboring nodes and route information is stored at a node. Below is the overview of the DSR route discovery process [12]

- Source broadcasts route-request to Destination
- Each node forwards the request by adding its own address and broadcasts it again.
- Requests propagate outward until Target or an intermediate node in a route to destination is found.
- The destination node routes the packet containing route information back to the source.

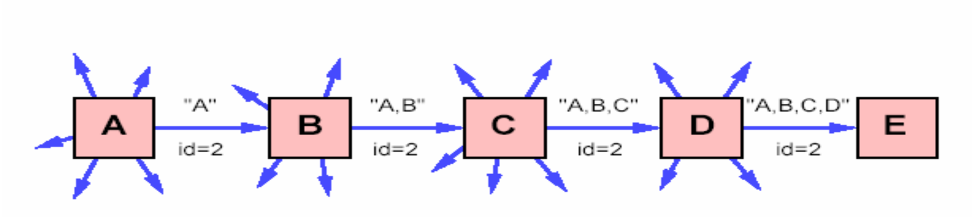


Fig 2.1: Route Discovery example: Node **A** is initiator, and node **E** is the target.

2.2.2 Secure Ad Hoc On-Demand Distance Vector Routing (SAODV)

SAODV is an extension of the AODV routing protocol that protects the route discovery mechanism mainly by using new extension messages. It provides security features like integrity and authentication. Alike AODV, because it typically minimizes the number of required broadcasts by creating routes on a demands basis, as opposed to maintaining a complete list of routes as in the DSDV algorithm. When a source node desires to establish a link to a destination link, for which it has no fresh enough route, it initiates a path discovery process to locate the destination node. It broadcasts a digitally signed route request message to its neighbors until the destination is located. During the route initiation process, Intermediate nodes record the details of the path in their routing table and sign the reply with their own key. In SAODV, collaboration is much heavier because of cryptographic signatures.

2.3 Position-based routing protocols

Position-aided routing protocols offer a significant improvement in performance as compared to traditional ad hoc routing protocols. Location based routing protocols use geographical information to make forwarding decisions, resulting in a significant reduction in the number of routing messages [14]. The recent availability of small, inexpensive low-power GPS receivers and techniques, provide justification for designing power-efficient and scalable MANETs [8]. LAR is one of the Position-Based Routing Protocols [4].

2.3.1 Location-Triggered Routing Protocol (LTR)

The primary goal of a routing protocol is to establish a reliable link for secure communication within a short time-frame. LTR utilizes geographical information to reduce message overhead and to provide link prediction to enhance performance. In comparison with table driven protocols, LTR only exchanges routing messages when a node changes its location and impacts at least one network route [8]. LTR uses the geographical information to predict movement between nodes. Thus, when a host node detects that the signal is getting weak because it is going out of range, it will automatically establish a new connection via link prediction.

In LTR location information will be used for

- Exchange route messages when a node changes its location.
- Predict the direction of a movement to establish a new route when a current communication has a potential to be broken [8].

LTR always compares the received message with the routing table to determine if there is any new information. In case there is new information, it will update the routing table and broadcast the new information to the network.

In figure 2.2, R is the signal coverage radius for node 1. Node 1 is traveling from point X to point Y but all its immediate neighbors are still within coverage range. In this case the node does not send any update message to the network.

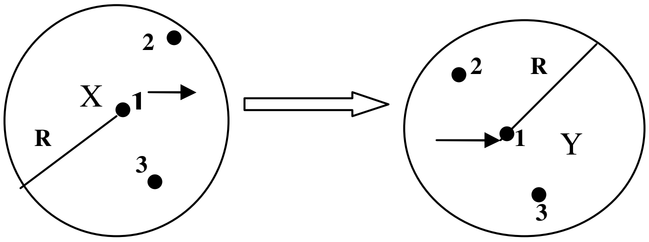


Fig 2.2: Network Snapshot

In Figure 2.3, the neighbors of node 1 have moved out of range and the route from node 1 to 2 is therefore broken. In this scenario, the route needs to be updated.

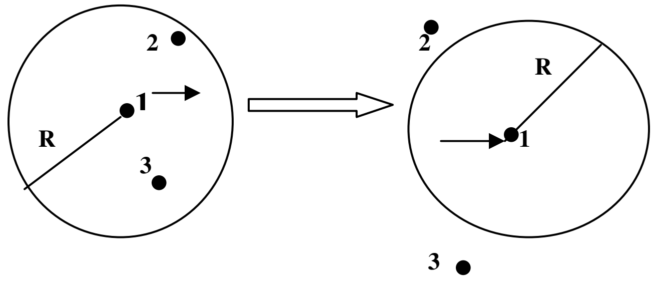


Fig 2.3: Network Snapshot

In Figure 2.4, Node 2 is within the signal range of node 1 and will become the immediate neighbor of node 1. Therefore, a new route should be established between node 1 and node 2.

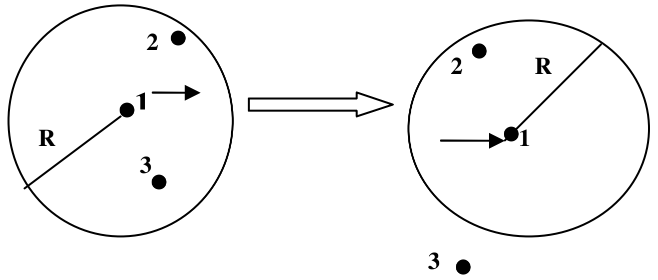


Fig 2.4: Network Snapshot

2.3.1.1 Algorithm for LTR

We assume that node MH1 and node MH2 currently have a communication link between each other. MH1 realizes that the signal from Node MH2 is becoming weak; it initiates the prediction process:

1. LTR calculates the link prediction time t_p, and based on that it will calculate the new predicted location for MH1 to be D [*X*p; *Y* p]. D is the new predicted location for MH1.
2. It will search its routing table to find all the nodes under its radius of communication and then will sort then in ascending order.
3. The node with the least distance from MH1 is the first choice.
4. MH1 checks the stability of the first node e.g. MH4; if MH4 is not trusted (i.e. stable), then MH1 chooses the next node (MH3) with smallest distance to the predicted location from the list.
5. If MH3 is stable, MH1 send a query message to it and MH3 do the same.
6. MH1 use the received routing information to establish a new path to the communication destination.

2.4 Authentication

Authentication mechanisms are used to ensue that the entity that supposedly sent a message to another party is indeed the legitimate entity [13]. It is important to make sure that any extruder doesn’t forge or alter the message sent by sender.

Different from fixed networks, the communication links in MANETs are more open for attacks from extruders. MANETs are characterize by absence of fixed infrastructure, rapid technology change and malicious alteration which determines that the authentication protocols used for routing and date packet delivery in MANETs, should be lightweight and scalable [13].

2.4.1 Time Defined Stream Loss-Tolerant Authentication (TESLA)

The idea of TESLA is proposed in [15]. TESLA uses one- way hashed chain to generate keys and delays disclosure of keys to guarantee that a node receives the packet before another node can forge the packet with already released keys [13]. The drawback of TESLA is, its security conditions requires clock synchronization, which is very difficult to achieve and maintain in MANETs, if not impossible.

2.4.2 Lightweight Hop-by-hop Authentication Protocol for Ad Hoc Networks (LHAP)

It is a lightweight hop-by-hop authentication protocol specifically designed for Ad hoc networks. LHAP uses two keys: TRAFFIC and TESLA key, one for authenticate packets and the other one to achieve trust maintenance by authenticating KEYUPDATE message. LHAP is not only a comprehensive authentication approach but also proved to be computationally efficient [16]. However, it require two keys which adds more complexity in authentication and also needs to periodically send key maintenance packages that themselves needs to be authenticated with TESLA keys. In addition, LHAP does not eliminate the delayed authentication in TESLA because the authenticity of the packets and the TRAFFIC key cannot be verified until TESLA key is authenticated [13].

CHAPTER III

THESIS OBJECTIVES

3.1 Thesis Objectives

The objectives of this thesis are:

1. To provide a secure route such that all nodes in the route are authenticated. Our suggested authentication algorithm will authenticate the new predicted node before a link breakage.
2. To minimize packet loss caused by link breakage.
3. To design and perform experiments to prove that our approach to LTR does indeed significantly reduce the number of dropped date packets.
4. To show that our suggested authentication algorithm will authenticate the new predicted node before the communication breakage.

CHAPTER IV

APPROACH

We assume that each node has a Global Positioning System (GPS) device attached to accurately identify its location. Each node has signal coverage radius R. In LTR, a node with will not send an update message until it discovers a location change. Let's assume that two nodes A and B are in communication and node A changes its location. Now, node B will calculate the new distance to node A and will check if node A is still under the signal coverage. If node A lies inside the signal coverage area of node B, no change will be made, otherwise route update message will be sent to the neighbors [8]. Neighbors will compare the receiving message with the information stored in the routing table, e.g. sequence number, to determine whether there is any new information. If the receiving message does hold new information, then the routing table will be updated accordingly. Otherwise, the receiving message is ignored.

First, we describe the Authentication algorithm with our assumptions and then we propose a link prediction algorithm using LTR that can predict the link breakage time between mobile nodes.

4.1 Assumptions for prediction algorithm

We make the following assumptions for our prediction algorithm:

- The Radius of communication is fixed and identical for all nodes.
- Communication is more expensive than computation.

- Each node has enough cache memory to hold the information in the routing table.
- If there is a broadcasting of location information, this information, will be received by all nodes that are within the signal coverage area.
- All nodes move in a two-dimensional plane.

4.2 Authentication scheme

We propose to protect routing and data packet transmission by authenticating all nodes in a route. Authentication process ensures that all the nodes in a route are legitimate.

We make the following assumptions for the authentication protocol:

1. Any new node that enters ad hoc network, must obtain public key of the Certificate Authority (CA) as well as the certificate of the node's own public key.
2. In certain instances, a node may not be able to communicate with the CA after it joins the network because it is difficult to provide and maintain the CA at all times especially since all the nodes are mobile.
3. The public key of the CA will be used to validate key certificates distributed by other nodes.

We will use the one-way hash chaining authentication protocol proposed by [13] with our prediction algorithm in this paper.

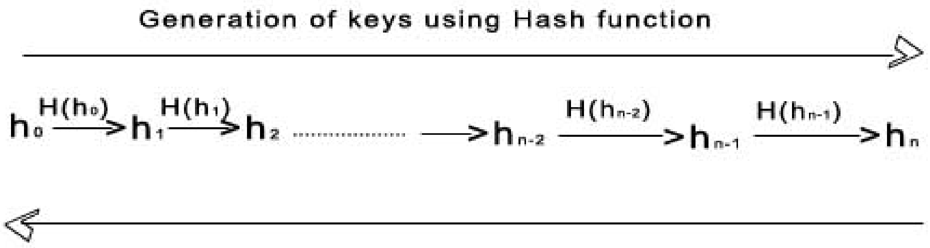


Figure 4.1: Generation of Hash Keys

In figure 4.1, we can see the construction, generation and utilization of keys in one-way hash chaining. To create a key chain of length n+1, the first element of the chain h_o is randomly picked and then the whole chain is generated by applying a one-way hash function denoted as H in the above figure [13]. In this case, the one-way hash function maps an input of any length to a fixed length bit string, which is defined as $H:\{0,1\} \rightarrow \{0,1\}^{Ø}$, where Ø is the length of the output of the hash function – the newly generated key [13]. Any key h_j can be verified from h_i ($0 \leq i < j < n$) to be certain element in the chain by applying H for j-i times, i.e**:**

$$h_j = H^{j-1} (h_i)$$

4.2.1 Algorithm for Authentication

One-way hash chaining scheme is efficient as it computes one-way hash functions within milliseconds for secured networks [13].

In this algorithm, a node distributes its authentic key h_n which is the first revealed key from the generated chain. This key commits to the whole key chain and helps validating the subsequent keys by applying hash functions to this key.

When a node tries to bootstrap trust with other nodes, it signs the message with its private key and broadcasts a JOIN message to the communicating node. In our case, we assume that node MH1 is communicating with a node MHX via node MH2 .The link between MH1 and MH2 is becoming weaker. MH1 needs to find another node (e.g. MH3) to replace MH2.

- MH1 sends a JOIN message to MH3. The JOIN message will be

  $$Y \rightarrow *: Cert_{Y,} \{Y \mid h^{Y}_{n} \mid H_Y \} , \text{Sign}_{Y} (Y, h^{Y}_{n} , H_Y)$$

  We are sending three things to node MH3 via JOIN message:

  $Cert_{Y,} \{Y \mid h^{Y}_{n} \mid H_Y \}$ and $\text{Sign}_{Y} (Y, h^{Y}_{n} , H_Y)$

Where *Cert*$_Y$ denotes the certificate of node MH1's public key that has been signed by CA's private key, Y denotes the id of node MH1, Sign $_Y$(Y, $h^Y{}_n$, H$_Y$) denotes the digital signature of message (Y, $h^Y{}_n$, H$_Y$), (Y, $h^Y{}_n$, H$_Y$) is the message based on the Mac key, H$_Y$ is the Hash function of node MH1 and $h^Y{}_n$ is the nth key in node MH1's one-way hash chain.

- Upon receiving the JOIN message, node MH3 will use CA's public key to verify the certificate of node A's public key. Once the node MH3's public key proved genuine, the key can be used to verify the digital signature on MH1's message.
- If the digital signature is validated to be authentic, the receiving node will record MH1's initial key $h^Y{}_n$ as well as its hash function H$_Y$.
- To bootstrap an authentic hash key to node MH1, node MH3 will reply back with an acknowledgement message ACK to node MH1:

  $$Z \rightarrow *: Cert_{Z,} \{Z \mid h^Z{}_m \mid H_Z\}, \text{Sign}_Z(Z, h^Z{}_m, H_Z)$$

  Where $h^Z{}_m$ denotes MH3's most recently released key and Z denotes the identity of node MH3. $h^Z{}_m$ can be verified by applying hash key function H as we mentioned earlier with its subsequent keys.
- Both MH1 and MH3 have therefore authenticated each other and can use the 0th key chain for encrypting their messages.

For trust maintenance, each node tries to build a relationship with its neighbors. Nodes send the most recent key used to compute the messages and neighbors verify the new released key with corresponding hash function. The key h_j can be authenticated by its neighbors based on previously release key $h^Y{}_{j+1}$; if it can be proved that

$$H_Y\,(\,h^Y{}_j) = h^Y{}_{j+1}\,,$$

The key h^{Y}_{j} is considered valid otherwise the key is invalid and receiving node MH3 issues an intrusion message to other nodes.

4.2.2 Security for Authentication Scheme

[13] shows that this protocol ensures data integrity and prevents 'man in the middle attacks'. This protocol uses digital signature in both initial trust establishment and subsequent trust reestablishment. The other schemes use asymmetric cryptography in only initial trust bootstrapping, [13] has guaranteed the genuineness of the key that commits to subsequent keys, and an "in-the-middle" attacker would not be able to use an already released key and forge packets with the obsolete key afterward [13]. This one-way hash chain key authentication scheme can effectively thwart the attacks of forging or maliciously alteration of packets. The delayed key disclosure property suggested in [13] can also prevent from in-the-middle attack in which an adversary may use an obsolete key to forge or alter packets.

4.3 Prediction Algorithm

The link state prediction in MANETs aims to reduce the number of dropped data packets as a consequence of link failure. The packet loss cause considerable degradation of both real time and non-real time data. In most existing protocols, nodes keep using the link until it breaks. However, our proposed prediction algorithm will foresee topological changes in order to perform a route rebuild prior to the link breakage.

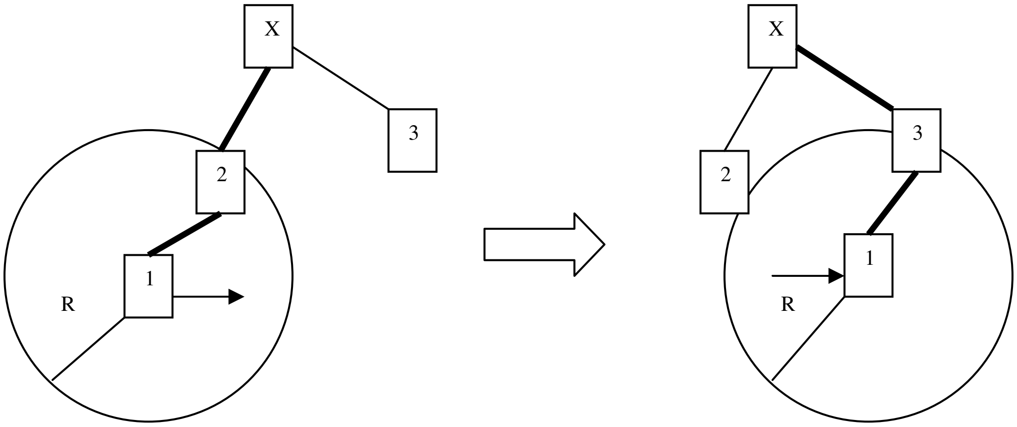


Figure 4.2: Routes from MH1 to MHX

In Figure 4.2 node MH1 currently has a communication link with node MHX using path 1->2->X. We assume that MH1 is traveling to another location and senses that the signal received from MH2 is becoming weak. This means MH1 is currently traveling away from MH2 and the link with MH2 can shortly be broken. At this point, it's not necessary to initiate a routing table update process; instead, it will be more efficient if MH1 discovers a new path to continue the communication with MHX. Thus, it can be done through MH3 instead of MH2 as MH3 will be in the vicinity of MH1. The rest of the routing table is updated after the communication is finished [8].

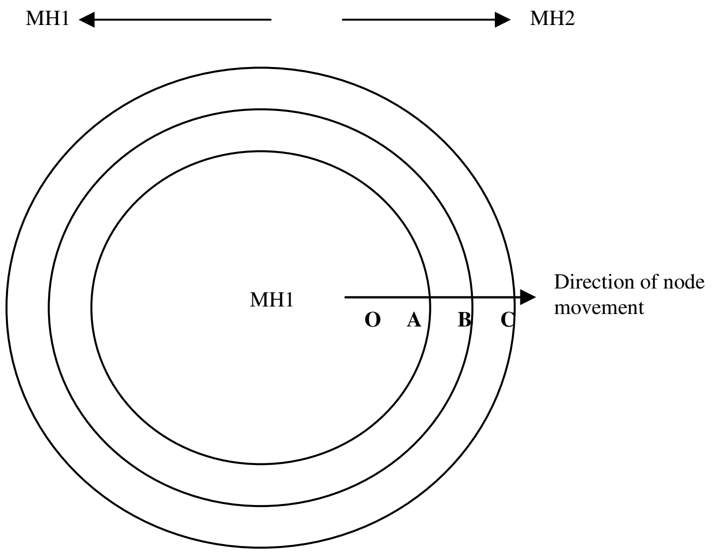


Figure 4.3: Nodes moving in opposite direction with maximum speed

In Figure 4.3, different circles are representing the different signal level strengths. As nodes are in continuous transition, signal strengths vary in different cases. Here are the cases studied in this thesis work:

- Nodes moving at maximum speed in opposite directions.
- Nodes moving at constant, but not maximum relative velocity
- Nodes are not changing direction, but changing speed
- Node are changing both speed and direction

Before we go into the details of each of the cases, we define the different notations and reference points as being used in the next few sections.

Point O – this is the point where we had the last location broadcast and has coordinates (x_O, y_O).

Point A – this is the point where we start the prediction and authentication process. Here we calculate the speed and direction of the moving node and has coordinates ($x_{A,}$ y_A).

Point B – this is the point that gives enough time to authenticate the new predicted node and has coordinates ($x_{B,}$ y_B).

Point C – this is the point where authentication must take place and will start communicating the new node. It has coordinates ($x_{C,}$ y_C).

Point D – this is the point where we decide whether to predict and authenticate the new node or not. It has coordinates ($x_{D,}$ y_D).

Therefore A-C is the minimum time required to predict and authenticate the new node. A-B is the minimum time to predict the new node and B-C is the minimum time to authenticate the node. From point O to point A, we determine the speed and direction of the moving node and assume that node moves with the same velocity and direction. It is from point A to point B, where we apply our approach to predict the new node for communication for MH1. As nodes are moving randomly, there can be cases where nodes are changing either speed or direction or sometimes both. In between point A and point B, a node might be moving slower or faster then the calculated speed, and there can be a scenario where this node is changing its direction altogether. Because of this random movement, our approach is going to record node's position from point A to point B and determine, if the node is changing direction or speed. The summary of these recorded positions from point A are:

Point A' – If a node is moving slower than expected, it reaches at point A' with coordinates ($x_{A'}$, $y_{A'}$) at which point the new node has been predicted. Therefore the signal at point A' will be stronger than the signal at B. There is therefore no need to authenticate immediately. This case is described in section 4.3.2.

Point X – If node is moving slower than expected or changing its direction, it reaches point A', but in some cases, it may be moving very slowly or changing direction and instead it reaches point X where it is closer to point A in comparison with point A'. This case is described in 4.3.3. Point X has coordinates (x_X, y_X).

Point Y – If node is moving faster than its expected speed as calculated from locations O-A, it should reach point A', instead it is moving faster than expected and reaches point Y. This is described in case 4.3.3. It has coordinates (x_Y, y_Y).

Point A" – If the node is changing its direction or speed or both and reaches either point Y or point X respectively, then we let the node move for a constant time (which is negligible), so it will maintain the new speed or direction or both from either point X or Y depending on the case. Point A" is the point that we derive from X or Y and gives sufficient time to execute our approach to LTR and has coordinates ($x_{A"}$, $y_{A"}$).

As node MH1 and MH2 may be moving in any direction with different or the same velocity, we will categorize these different scenarios below and will perform different simulations to determine the performance of the proposed scheme.

4.3.1 Nodes moving at maximum speed in opposite directions

In this case, Node MH1 and MH2 are moving away from each other at a maximum speed as depicted in the figure 4.4:

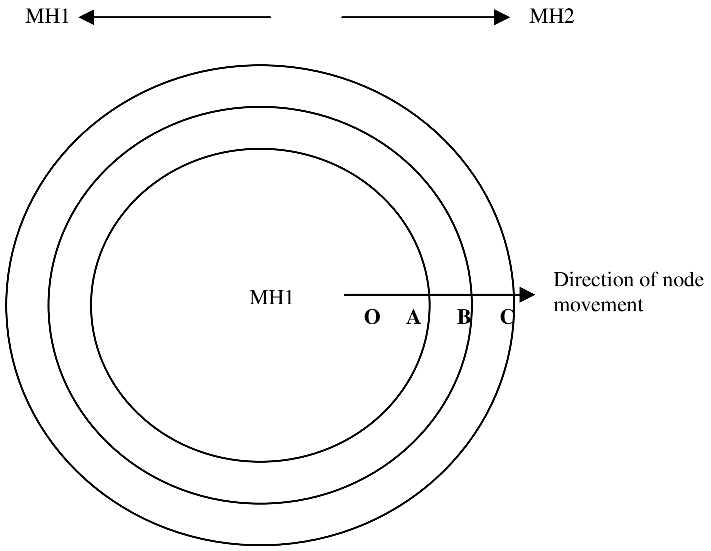


Figure 4.4: Nodes moving in opposite direction with maximum speed

Each circle in figure 4.4 is a signal strength level. In, this case:

- Point A is the current location of MH2. MH1 relative to MH2 is at the center of the circle.
- The last location broadcasted was at point O, so point O is not the center of the circle. Point A is fixed.
- A is defined as the point which allows sufficient time to execute prediction and authenticate even if the nodes are moving apart at the relative maximum velocity.
- Based on the location information at point O and A, the relative speed and direction can be determined (see appendix).
- If the relative speed and direction indicate that the nodes are moving away from each other at the maximum and relative velocity, execute our prediction mechanism.

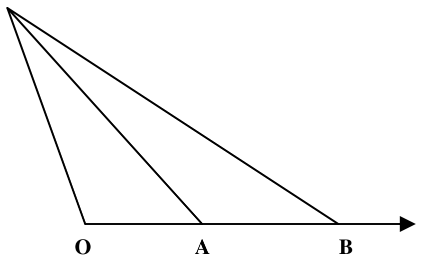


Figure 4.5: Node moving at maximum relative velocity [8]

- The assumption in this case is that there is no change in speed or direction. This means that both nodes will keep moving with maximum speed and in opposite directions.
- Here, Point A to point B is the time when our approach executes (figure 4.5).
- Our approach predicts node MH3 to be the next node.
- Point B to point C (fig 4.4) is the time for MH1 to authenticate MH3 and vice versa.

Point C is the time when break in link between MH1 and MH2 happens and MH1 will start communicating with MH3 after building trust by authenticating.

4.3.2 Nodes moving at constant, but not maximum relative velocity

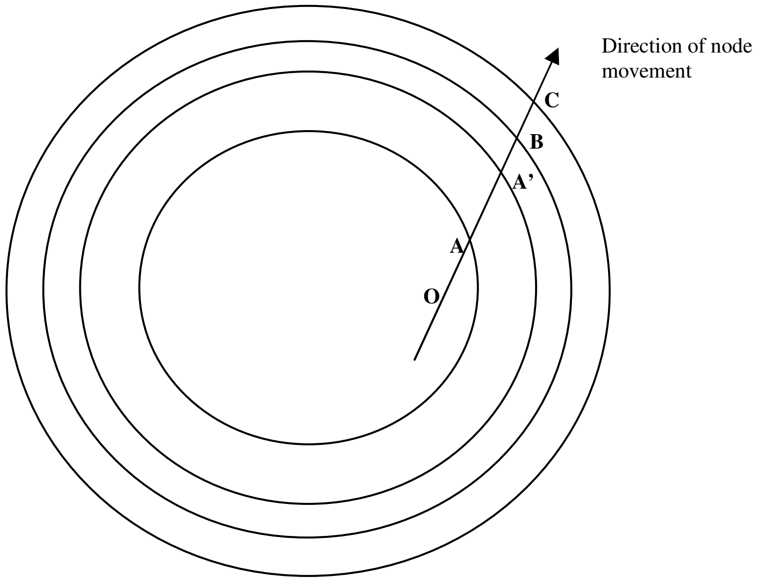


Figure 4.6: slow constant relative velocity

Here the assumption is that there is no change in direction or speed (figure 4.6). We are going to use A' as the reference point (Section 4.3). In this case:

- Point O to point A is the time to determine the relative speed and direction.
- The node is moving with a slower constant relative velocity, and A-C is too long for prediction and authentication. Instead as the node is moving at a slower speed, it is allowed to reach A' before the prediction and authentication process begins.
- A' is the point which gives enough time to execute our approach to LTR to predict a node as well as authenticate it.
- From point A' to C would give enough time to execute our suggested prediction mechanism to predict a node as well as authenticate it.

- We will start authenticating predicted node (MH3 in this case) at point B.
- Point C will be the end of authentication of predicted node
- Link between MH1 and MH2 break and MH1 will start communication with MH3 after building a trust.

This enables prediction and authentication to be delayed for as long as possible so that it is done only when it is necessary.

4.3.3 Nodes are not changing direction, but changing speed

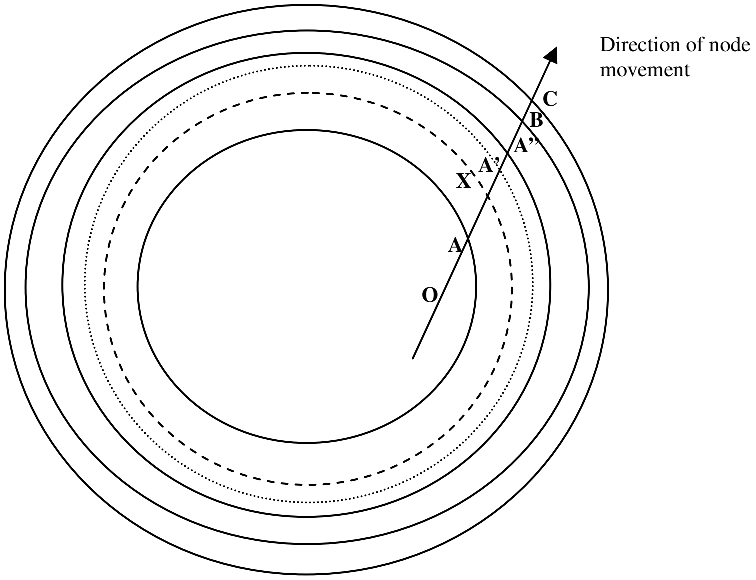


Figure 4.7(a): Change of speed

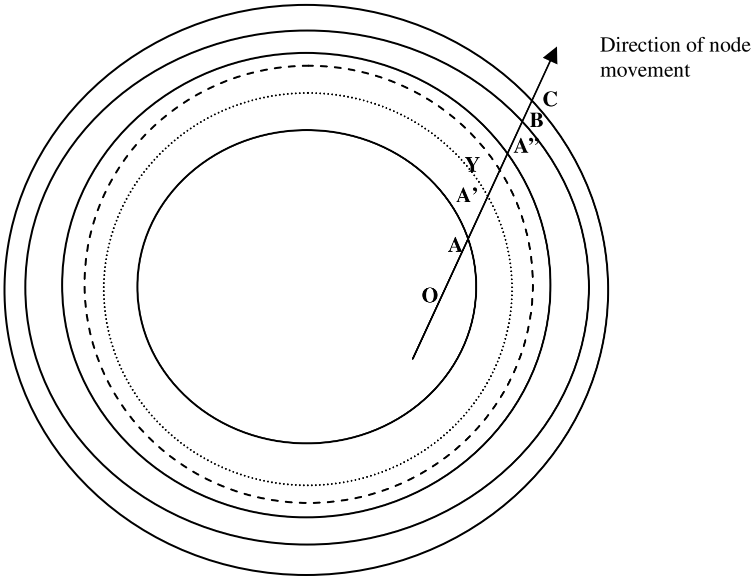


Figure 4.7(b): Change of speed

Here point O to point A is the time to determine the speed and direction of the node. We are going to use A', A", X and Y as the reference points (Section 4.3). In this case, if the node is moving slower than before:

- As the node is not changing its direction we will use information from point A to determine point A'.
- If the speed of this node drops down again, its only going to reach point X and not point A' (figure 4.7(a)).
- As there is a change in speed, we will calculate the speed from O to A and from A to X and compare the results of the two (See appendix). This allows us to determine the relative deceleration.
- As speed has changed, we let the node move for a constant time (which is negligible), so it will maintain a speed and then it reaches point A". Calculate new direction and speed (see

appendix) and determine point A” to execute the LTR prediction algorithm as well as authenticate the predicted node.

- At point A”, the prediction process starts.
- Start authenticating predicted node at point B.
- Point C is the end of authentication of predicted node and node MH1 will start communication with new node MH3 at point C.

If the node is determined to be moving faster, but no change in direction,

- Node should have reached point A’, instead it reached Y (figure 4.7(b)).
- By calculating the new speed, we can determine point A”.
- As there is change in speed, we will calculate the speed from O to A and from A to Y and compare the results of the two (See appendix.) to obtain the relative acceleration.
- As speed has changed, we let the node move for a constant time (which is negligible), from point Y so it will maintain a speed and then it reaches point A”. Calculate new direction and speed (see appendix) and determine point A” to execute the LTR prediction algorithm as well as authenticate the predicted node.
- At point A”, the prediction process starts.
- Start authenticating the predicted node at point B
- Point C is the end of authentication of predicted node and MH1 will start doing communication with MH3.

4.3.4 Nodes are changing both speed and direction

In this case, point O to point A is the time to determine speed and direction of the node (see appendix). We are going to use A’, A” and X as the reference points (Section 4.3).

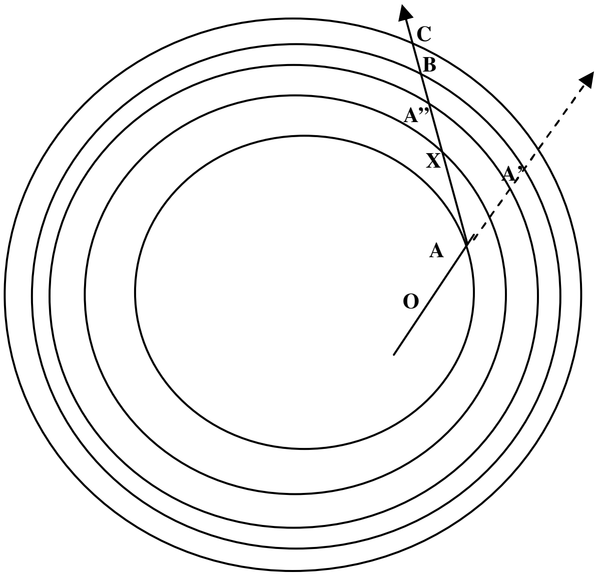


Figure 4.8: change of speed and direction

There are two cases to be considered. First, where the moving node changes the speed and direction and also stays outside the signal range of communicating node. This is shown in figure 4.9(a). In the second case, the moving node changes the speed and direction but it stays within the signal range of the communicating node. This is shown in figure 4.9(b).

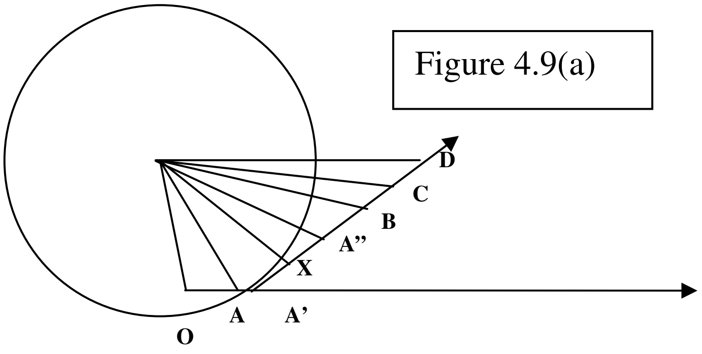


Figure 4.9(a): change of speed and direction

- Using this information determine point A’ to execute LTR to predict a node as well as authenticate it.
- Check location at each broadcast
- If location at a point indicates change of direction, but relative distance is the same or increasing (i.e. signal is not getting stronger) – the node should have reached point A’, instead it only reached X (figure 4.9 a).
- As direction and speed has changed, we let the node move for a constant time (which is negligible), so it will maintain a direction and speed and then it reaches point A”. Calculate new direction and speed (see appendix) and determine point A” to execute the LTR prediction algorithm as well as authenticate the predicted node.
- Calculate point B and point C.
- Start authenticating predicted node at point B
- Point C is the end of authentication of the predicted node and MH1 will start communicating to new node MH3 at this point.

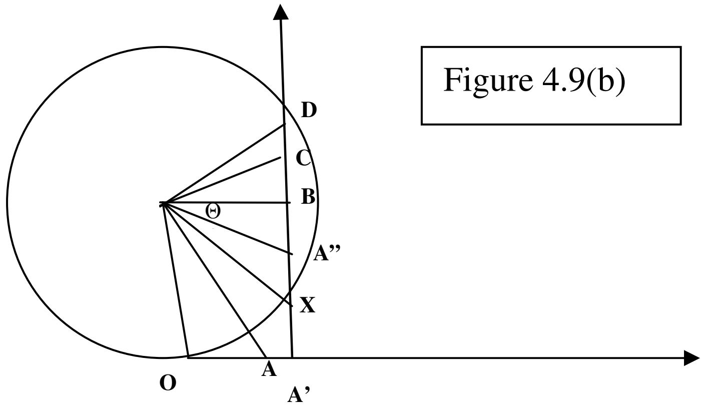


Figure 4.9(b): change of speed and direction

If signal is getting stronger, this indicates the node under discussion (i.e. MH1) is moving closer (figure 4.9(b)):

- Again, point O to point A is the time to determine speed and direction of node
- Using this information determine point A’ to execute our prediction mechanism to predict a node as well as authenticate it.
- If location is changed, that is, relative distance is decreased. Node has reached X instead of point A’.
- As direction and speed has changed, we let the node move for a constant time (which is negligible), so it will maintain a direction and speed and then it reaches point A”. Calculate new direction and speed (see appendix) and determine point A” to execute the LTR prediction algorithm as well as authenticate the predicted node.
- Determine point C, where C is the end of the signal range.
- Point C is the end of authentication of predicted node.

4.3.5 Algorithm

In this section we will describe our algorithms for prediction and path array mechanism. Path array mechanism is an algorithm that predicts the new link between any two nodes under consideration. The proposed algorithm for prediction is based on the modification of LTR and covers the details of our prediction scheme for link state prediction.

As the nodes are changing speed and direction, we have to deal with various calculations in order to get the coordinates (locations) of our nodes. This includes the distance ‘D’ ’, direction $\sin\theta$, speed ‘S’, minimum distance $d_{\min}$, and coordinates ‘ x_C ’ and ‘ y_C ‘.

In figure 4.2, MH1 is traveling away from MH2 and the link with MH2 is going to be broken shortly. As discussed earlier, at this point, it is more efficient to discover a new path to continue communicating with MHX instead of initiating a routing table update process.

In figure 4.3, each circle shows the signal strength of communicating nodes MH1 and MH2.

A → B is the time when MH1 predicts the new node for communication.

B → C is the time for MH1 to authenticate the new node.

C → Assumed as the maximum signal range of node MH2 and this is the point where authentication must take place and will start communicating the new node. It has coordinates ($x_{C,}$ y_C).

D → this is the point where we decide whether to predict and authenticate the new node or not. It has coordinates ($x_{D,}$ y_D).

We have already discussed four different scenarios in the previous section (4.3.1 – 4.3.4). For our algorithm, we are using the fourth case in which the node is changing both the speed and direction. In figure 4.9(b), MH2 is at point O. MH1 relative to MH2 is continuously traveling with variable speed and direction.

Point C is the end of authentication of the predicted node and this is the maximum signal range for node MH2. In our algorithm we define a point D where if point D, MH1's predicted location, lies inside the signal range then we don't need to predict the new communication node as it is inside MH2's signal range. For our prediction algorithm, we need to know if point D falls inside the signal range of node MH1 (or MH2). We will calculate point D and use it in our prediction algorithm (4.3.5.1). To determine point D in case 4 (Figure 4.9 (b)), the distance and direction traveled can calculated as,

$$D' = \sqrt{(x_B - x_{A''})^2 + (y_B - y_{A''})^2} \tag{1}$$

Where A” and B are reference points defined in section 4.3.

$$\sin\theta = \frac{y_B - y_{A''}}{D'} \tag{2}$$

We assume the time required for authentication is t_a. Time taken from traveling from A” to B and time spent for prediction mechanism is t_i . Now that we have the distance D’ and time t_i, speed s can be calculated as:

$$s = \frac{D'}{t_i} \tag{3}$$

The minimum distance d_{min} from B → D is

$$d_{\min} = s \times t_a \tag{4}$$

Knowing d_{min}, we can calculate the location of point D.

Therefore,

$$x_D = \cos\theta \times d_{\min} + x_B \tag{5}$$

$$y_D = \sin\theta \times d_{\min} + y_B \tag{6}$$

Where (x_D, y_D) are the coordinates of point D.. Let r be the communication radius of node MH2. Point D lies within the radius of MH2 if it satisfies the following condition:

$$(x_D - x_1)^2 + (y_D - y_1)^2 < r^2 \tag{7}$$

Where (x_1, y_1) are the coordinates of MH1. If above equation holds true**,** point D is in the circle (that is, within communication range) and there is no need for prediction as node MH1 is in the signal range of node MH2. In the next section, we will use these calculations for our prediction algorithm.

4.3.5.1 Prediction Algorithm

When a node is traveling from point A to point B (figure 4.9(b)) and there’s a possibility that it can go out of range with the other communicating node, we need to find another node so the communication continues between the two. This algorithm predicts the ‘new’ node by applying our approach presented in this thesis work.

Note: Points A’, A”, X, C, D are all defined in section 4.3.

Prediction Algorithm

```
Begin
Compute Distance and Direction
        if  No change in Direction or Speed
          if MH1 reached Point A’
              Determine B and D
              // where B and D defined in section 4.3
              if point D within the radius of MH2
             //from eq. (7)
                      Do nothing; Node is within signal range
                      No Prediction Needed
             else
               Execute prediction when point B is reached
                      Execute Path Array Algorithm *
                      if New Node exist within signal range of MH1
                              Do Authentication // (section 4.2.1)
                              if Authenticated
                                      Start Communication
                              else
                                              Find New Node
                                              Start Prediction Again
                              end if
                      end if
              end if
          end if
      else
              Reached X // (figure 4.9(a)) // there is change of direction or speed
              Calculate new Direction and Speed //from eq. (1) and eq. (3)
              Determine A” and D
              // where A’ and D defined in section 4.3
              if point D within the radius of MH2
              //from eq. (7)
                      Do nothing; Node is within signal range
```

```
                    No Prediction Needed
               else
                Execute prediction when point A" is reached
                    Execute Path Array Algorithm *
                    if New Node exist within signal range of MH1
                        Do Authentication (section 4.2.1)
                        if Authenticated
                            Start Communication
                        else
                                Find New Node
                                Start Prediction Again
                        end if
                    end if
               end if
           end if
       End
```

This algorithm is designed to work in situations where two nodes change their speed or direction just once. As the distances involved are small, we assume that a single detection is sufficient; the prediction algorithm can be easily modified to deal with multiple changes of speed and direction.

4.3.5.2 Path Array Algorithm*

Here is a brief description of the path array algorithm we have proposed and used in our simulation. The algorithm predicts the new link between any two nodes under consideration.

This Path Array mechanism uses our Path function where the path function finds a new link from the source to the destination node and is called from Prediction Algorithm (4.3.5.1). For instance, node A and node B are our source and destination nodes, respectively. In order to have continuous communication between the two nodes, we might have to use the nodes X, Y, so this algorithm will give us the link A->X->Y->B. In short, the path function is our routing algorithm.

We call this path function recursively to look for every single possible link to stay in communication with the destination node. If source and destination nodes are found to be neighbors, and thus within the signal range, we update this CheckNode array with a status 1. If the two nodes are not neighbors, then we update the CheckNode array with a status of 0. It starts by looking in the IsNeighbor array to determine if A and B are neighbors. If that's not the case, we look for all the neighbors of A and see if each of A's neighbor is within the communication range of B.

A temp array 'CheckNode' is used to store the intermediate nodes so as to ensure that we don't end up in a recursive loop. This algorithm either ends with a failure if a link cannot be found between two nodes or with a success and a path between the two nodes. The algorithm is outlined below:

Path Array Algorithm*

```
Begin
Public int Path (Source Node Src, Destination Node Dest)
   Initialize Flag with -1
   Initialize j with 0
        if IsNeighbor(Src, Dest)
             Call Check(Src, Dest, 1)*
              Set Flag as 1    //if link found
        else
             if (Check(Src,Dest,0) ==0) *
                 while (ClosestNode[Src][j] ! = -1)   // Whjle link not found
                        Path (ClosestNode[Src][j], Dest)
                        Increment j
             end if
        end if
 return Flag
 End  Path
 End

* Public int Check (Source Node Src, Destination Node Dest, Status Flag St)
      Initialize Temp with 0
            for i= 1 to GlobalCount
```

```
                if (CheckNode [i][0] == Src) && (CheckNode [i][1] == Dest)
                    Set Temp as 1
                endif

            if (Temp not equal to 1)
                CheckNode[GlobalCount][0] = Src
                CheckNode[GlobalCount][1] = Dest
                CheckNode[GlobalCount][2] = St
                Increment GlobalCount
            end if
 return Temp
end Check
End
```

CHAPTER V

SIMULATIONS

Different network scenarios are tested to note the performance and the behavior of the two Network Routing Protocols, LTR and table driven routing protocol DSDV. Both these routing protocols, maintains the routing information for each node but DSDV updates routing information periodically whereas our approach to LTR only updates the neighbors when find a topology change in the network. DSDV is been used widely as a routing protocol for MANETs. This program is implemented in Java using Net Beans IDE 5.0.

5.1 Simulation Program

In this simulation program, the number of network update messages, successful packet transfer between the communicating nodes and packet loss due to change in the topology, are tested for our suggested scheme in LTR and the table driven routing protocol DSDV. Network update messages are the messages which will update all the network nodes with the current location of each node. Network density (number of nodes) is the changing factor in this simulation. The simulation program takes input parameters as the number of nodes, radius of communication, transmission rate at which a node transfers packets and communication ratio.

Nodes moving speed, direction, and moving time are random factors in this simulation. All the simulations are performed with, 5, 10 and 20 nodes with 3 communication ratios of 25%, 50% and 75%. Communication ratio is the ratio which defines the percentage of communicating nodes in the network.

Nodes moving speed has been tested on 2 m/sec, 10m/sec and 20 m/sec. All these movements are random. After each movement, we assume that node rests for a time of 1 sec and then start the next move.

The transmission rate is 128 kilobytes per second. We assume that each data packet is of size 1 KB. The ad hoc network we have used in our simulation consists of an area of (400m x 1000m).

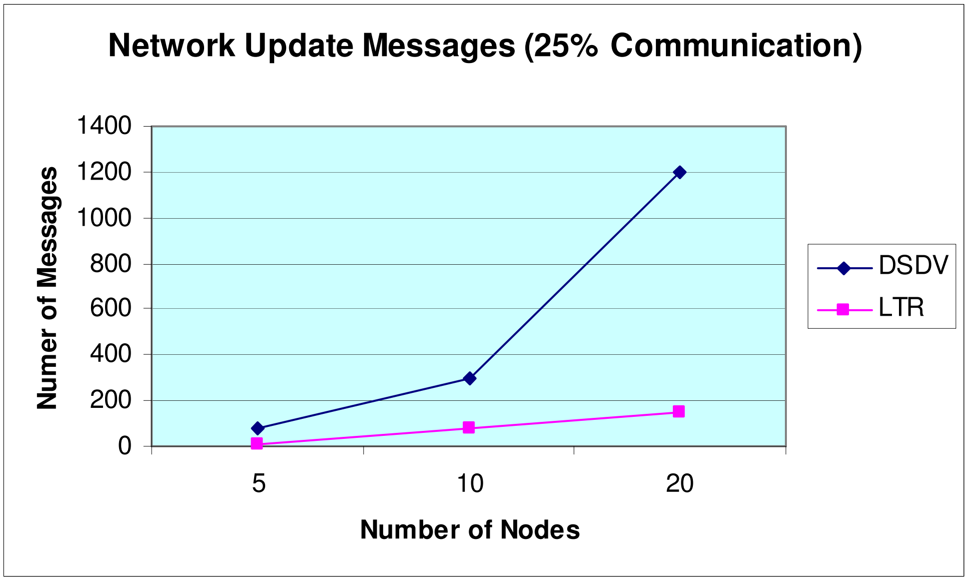
The simulation program generates the random direction, speed and time for each node and moves all the nodes simultaneously. It has been observed that most of the existing routing protocols are not very effective for random node movements. Our suggested algorithm in this simulation program would predict the topological changes in order to perform a route rebuild prior to the link breakage and we will compare the results with an existing table driven routing protocol DSDV.

We initialize the network with a distance of 5 m between each node to give enough distance between the nodes to move easily. Our simulation program will generate a communication array using a random number generator which defines the communicating nodes in the network. We calculate the direction and time by means of a random number generator for each node and move it dynamically with different velocities in the network. Since speed 20 m/sec would be much faster then speed 2 m/sec, the probability of node moving out of the networks becomes high.

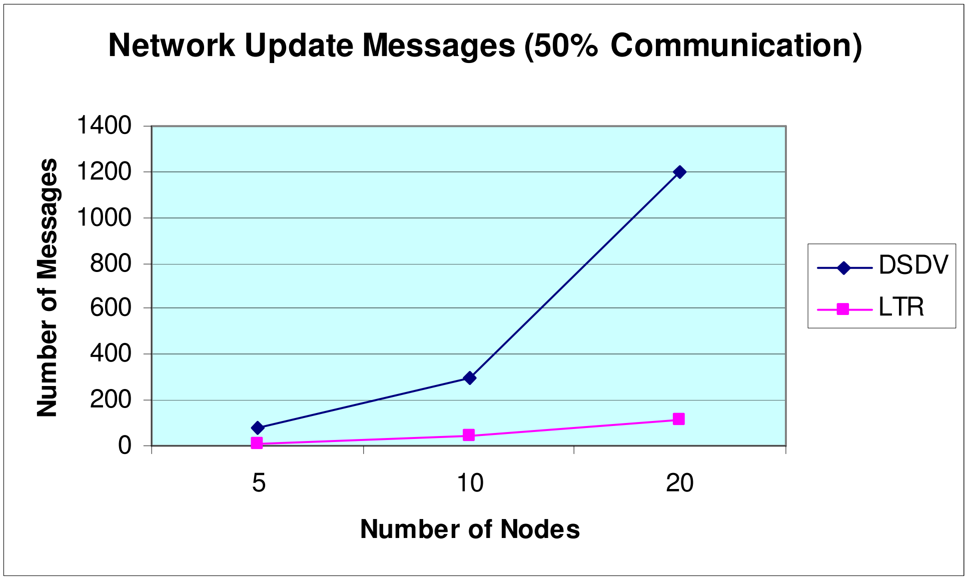
As nodes are moving randomly in this simulation program, our path array mechanism would predict and find a substitute link when a node moves out of radius from the communicating node. We have assumed that every time we find a new link between the communicating nodes, the program would consume a constant number of authentication packets in both protocols i.e. DSDV and LTR. We assume that each authentication packet is of size 1 KB. We assume that whenever a node finds a new link, it will consume a constant number of authentication packets i.e. two packets for each authentication process.

We perform experiments to evaluate our approach to LTR and examine if it does significantly reduce the number of dropped date packets in comparison with DSDV. This simulation program would calculate successful packet transfers (data packets), network update messages (messages to update the network), authentication packets and packets loss (data packets) due to change in the topology and link breakage for our algorithm and the existing table driven routing protocol DSDV. We have categorized different random scenarios which we have discussed in chapter 4, in our simulation results below. We assume that authentication will be performed before the existing link breaks in our approach whereas DSDV has no prediction mechanism. Hence, DSDV stops communicating if a node moves out of communication range of the other node. After each movement, each node takes a rest for a constant negligible time for 1 second, and then moves again, all randomly. The rest or pause time for each node is constant.

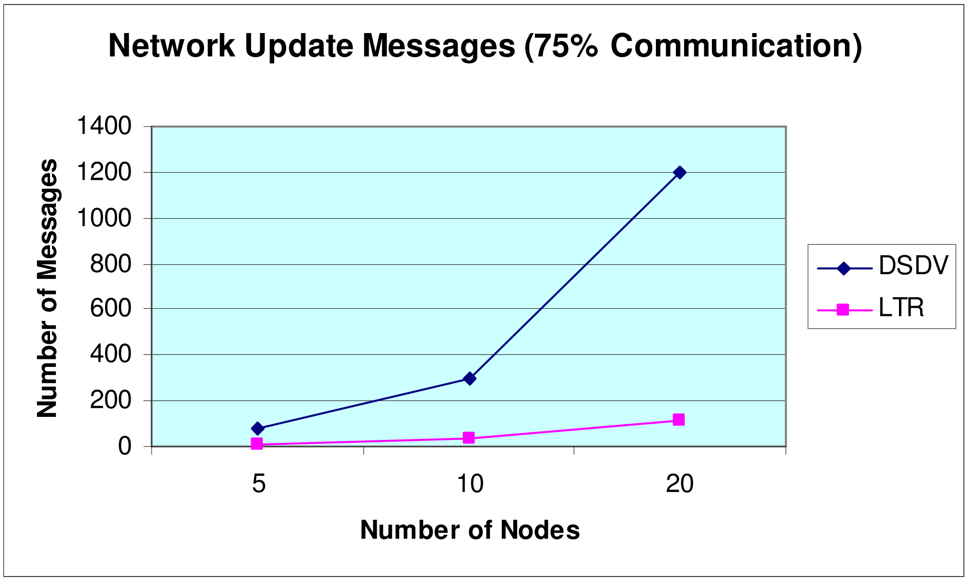
5.2 Results



Graph 5.1: No. of Network Update Messages vs. No. of Nodes (25% Communication)

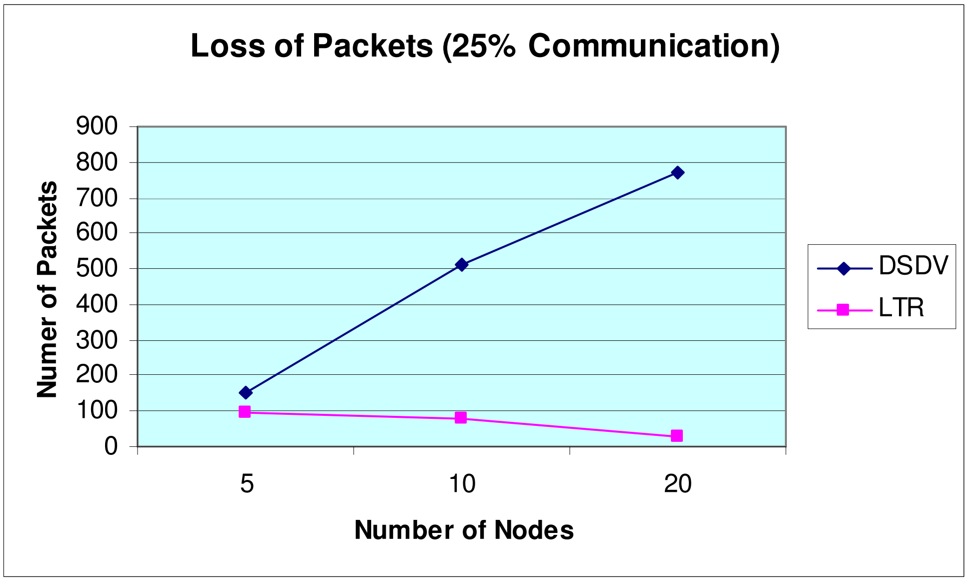


Graph 5.2: No. of Network Update Messages vs. No. of Nodes (50% Communication)

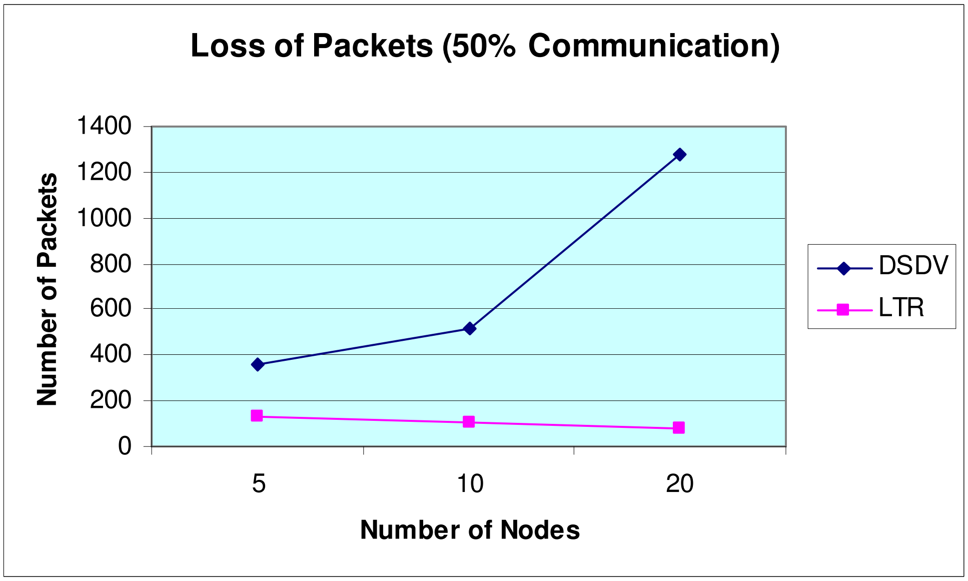


Graph 5.3: No. of Network Update Messages vs. No. of Nodes (75% Communication)

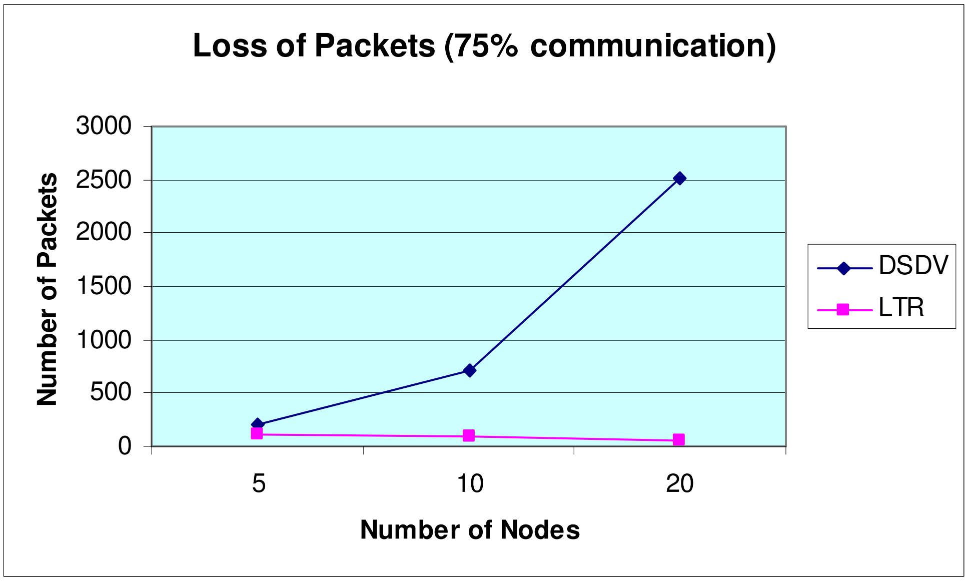
From graphs 5.1, 5.2 and 5.3 it can be seen that with the increase in the number of network nodes, our approach to LTR has less overhead compared with DSDV. We have run these simulations on different random speeds of 2, 10 and 20 m/sec i.e. Each simulation has 3 speeds. Each node would move at 2, take a rest, move at 10, take a rest, move at 2, take a rest, move at 20 etc. From the graphs, we can clearly see that when the network size is low, the differences in the network update messages are not very high between the two protocols. Graphs 5.1, 5.2 and 5.3 clearly indicate that the increase in the number of messages is very high in DSDV when we have a densely populated network in comparison with our approach. On the other hand our approach sends network update messages only when the algorithm finds a topology changed in the network. As the number of communicating nodes increase, DSDV generates more messages as expected and has more overhead. It can be seen from the graphs that communication ratio doesn’t affect the update messages in DSDV. DSDV updates routing information periodically and doesn't depend on the communicating nodes whereas our approach will only send the update message when it finds a topology change in the network.



Graph 5.4:.No. of Lost Packets vs. Number of Nodes (25% Communication)

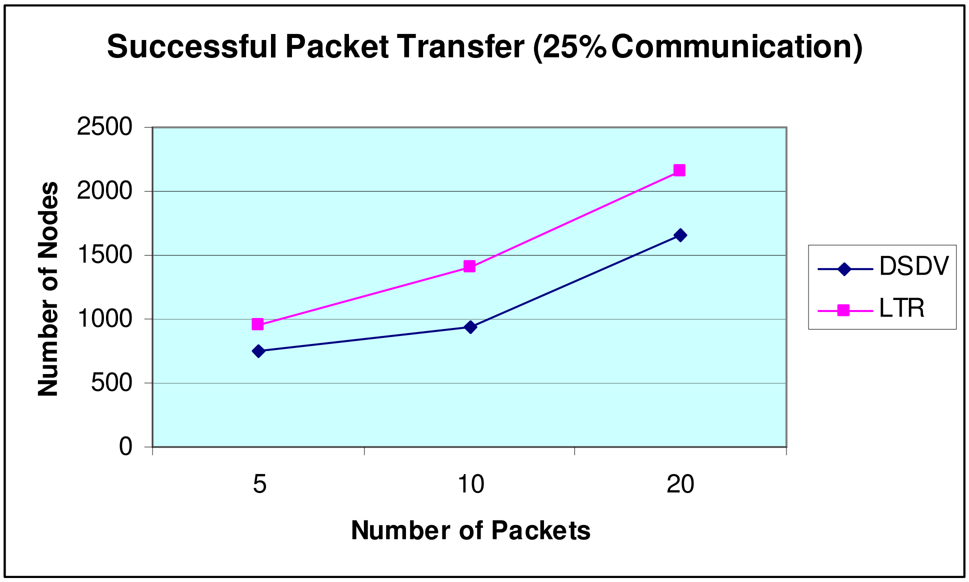


Graph 5.5: No. of Lost Packets vs. Number of Nodes (50% Communication)

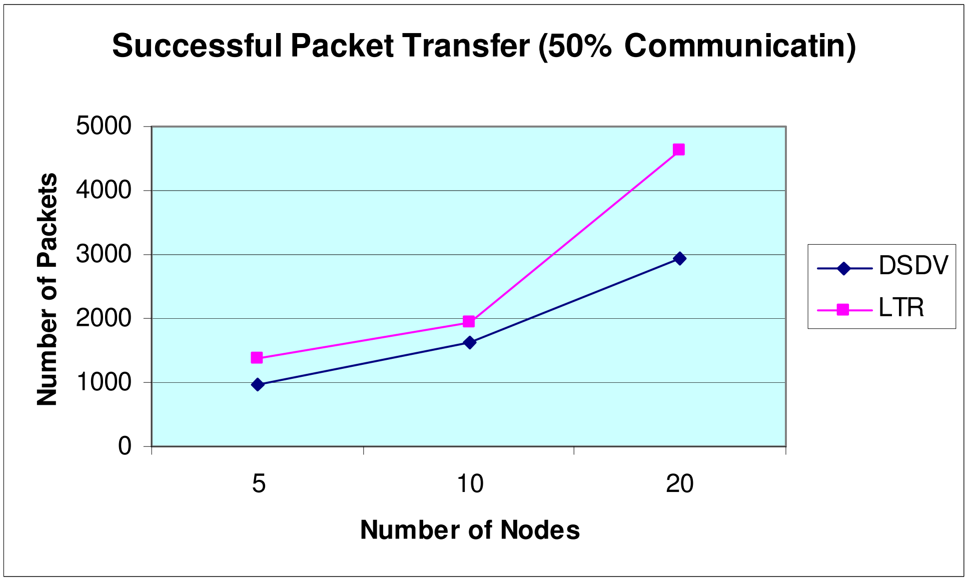


Graph 5.6: No. of Lost Packets vs. Number of Nodes (75% Communication)

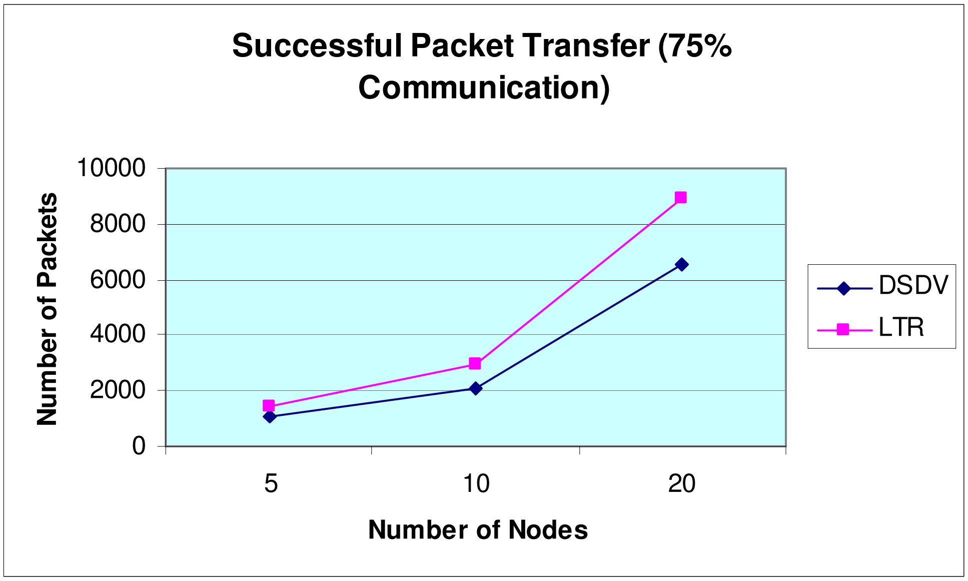
From Graph 5.4, 5.5 and 5.6 we can see our approach to LTR has a smaller packet loss than DSDV. The suggested algorithm which we have used in our simulation program will always look for some substitute link if a link has a potential to be broken and will predict the new link prior to the link breakage. With the increase in the number of nodes in the network, the suggested algorithm would have more options to search through to find the best link in the network and does significantly reduce the loss of packets between the communicating nodes. On the other hand, DSDV would not be able to predict any potential link breakage and would keep sending packets until the node gets out of range and would therefore lose packets. In DSDV the packet loss increases dramatically with the percentage of communications, whereas in our approach the rate of increase is far slower.



Graph 5.7: No. of Successfully Transferred Packets vs. No. of Nodes (25% Communication)

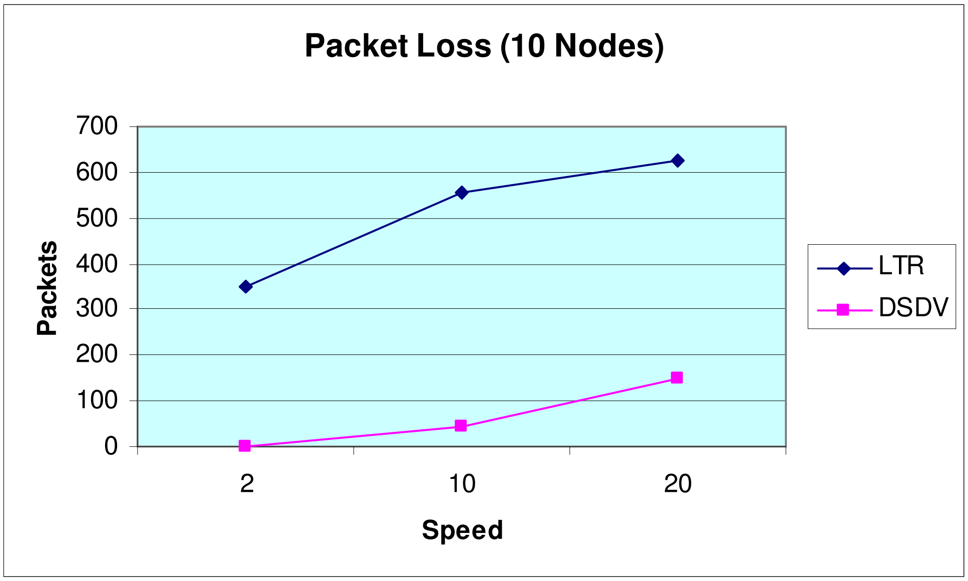


Graph 5.8: No. of Successfully Transferred Packets vs. No. of Nodes (50% Communication)

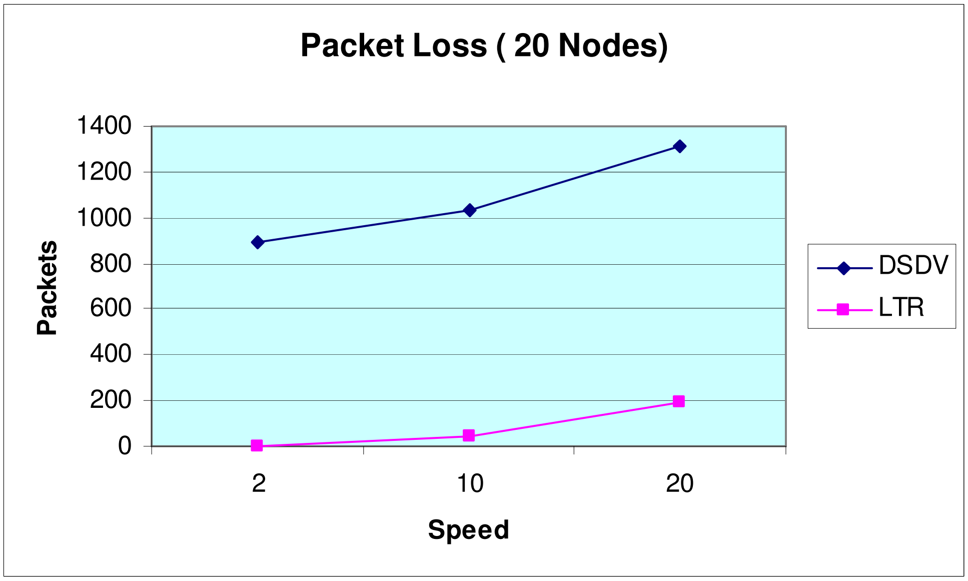


Graph 5.9: No. of Successfully Transferred Packets vs. No. of Nodes (75% Communication)

From Graphs 5.7, 5.8 and 5.9 we can see that with the increase in network size, our approach to LTR has a better ratio for successful packet transfer. When the network size is low, the results for DSDV and LTR are close to each other but with the increase in the number of nodes, DSDV significantly degrades in the successful transfer of packets. On the other hand, the proposed algorithm would find more options with the increase in the number of network nodes and would continue their communication through substitute routing links if needed.

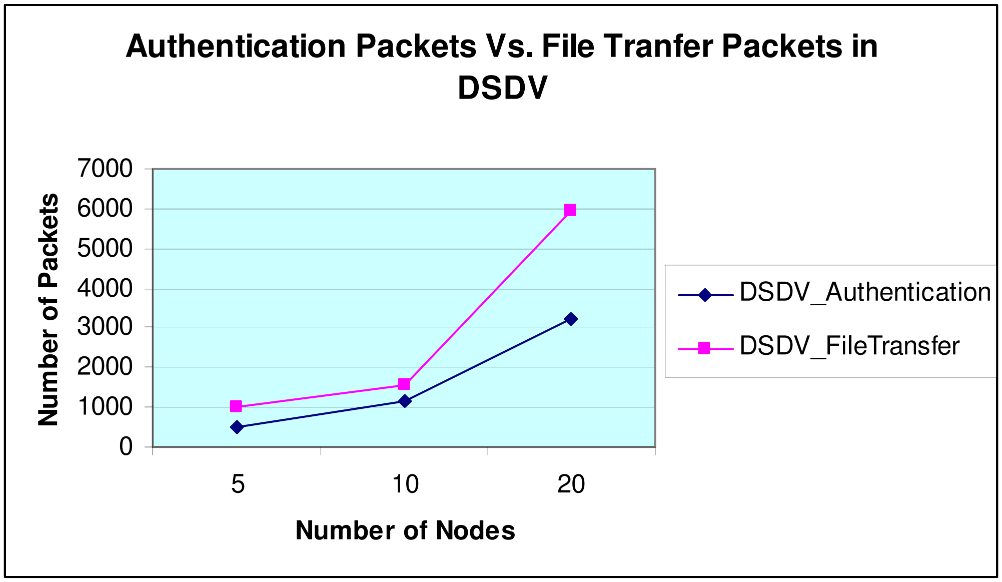


Graph 5.10: Lost Packets vs. Speed (10 Nodes, 50% Communication)

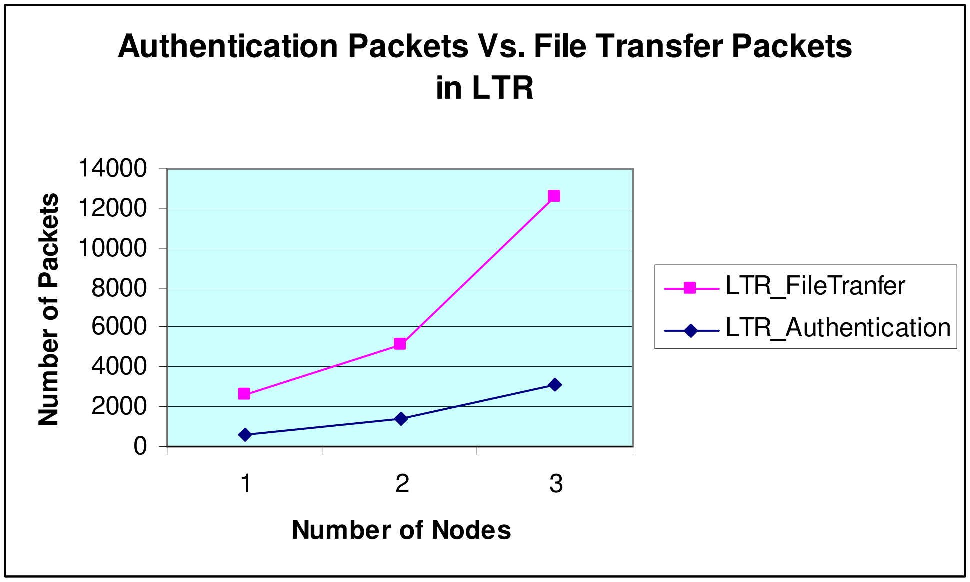


Graph 5.11: Lost Packets vs. Speed (20 Nodes, 50% Communication)

From Graphs 5.10 and 5.11, we can see our approach to LTR has smaller packet loss than DSDV. With the increase in speed, both protocols have dramatic increases in packet loss and it approximately doubles when the speed is increased from 10 m/sec to 20 m/sec. LTR still has less packets lost in comparison with DSDV as the proposed algorithm will always look for some substitute link if a link has the potential to be broken and will predict the new link prior to the link breakage. In DSDV the packet loss increases tremendously with the increase in speed, whereas in our approach to LTR, the rate of increase is far slower as expected.



Graph 5.12: Authenticated Packets vs. File Transfer Packets in DSDV



Graph 5.13: Authenticated Packets vs. File Transfer Packets in LTR

From Graphs 5.12 and 5.13, we can see the distinction between authentication packets and file transfer packets in both protocols. The difference in File transfer packets and authentication packets in DSDV are much smaller than the difference for LTR. LTR has approximately the same number of authentication packets as DSDV but the file transfer packets percentage is much higher in comparison to DSDV. Due to the random nature of the network nodes movement, the graphs show that nodes keep moving out of range of the communicating node. DSDV cannot handle such link breakages efficiently, whereas LTR because of its prediction mechanism manages it very efficiently.

5.3 Summary of results

The simulations were done for small and mid size networks, constant communication ratio, random directions, random node movements as well as low and high velocity scenarios. Based on the simulations, the following conclusions can be made:

- The proposed approach based on LTR can significantly reduce the number of lost packets compared with a no prediction routing protocol such as DSDV, especially in highly populated networks.
- Our approach does reduce the communication overhead among nodes by adding location information into routing tables, while on the other hand traditional table-driven routing protocol (DSDV) keep sending network update messages on a periodic basis incurring huge network overheads.

- Our approach has better packet transfer ratio between the communicating nodes than DSDV. As LTR can predict and find the substitute links prior to the link breakage, it successfully transfers the number of packets in randomly moving nodes with different velocities but on the other hand DSDV is not able to predict the pattern of randomly moving nodes and is deficient in the number of successful packet transfers.
- Our proposed approach has better successful packet transfer and fewer network overheads in densely populated networks.

CHAPTER VI

CONCLUSIONS

3.1 Conclusions

Our research shows that the prediction algorithm proposed here in this thesis does indeed significantly reduce the number of dropped date packets when nodes are moving randomly in the network and our prediction algorithm is efficient to find the substitute links in case of potential link breakage. We have compared our approach with a typical table-driven protocol DSDV. DSDV gives the best performance when we have low number of nodes in the network and transfers approximately the same number of successful packets as our proposed algorithm, but its performance significantly reduces with increased network density. On the other hand our proposed algorithm gives the best performance with less or highly populated networks where nodes are moving in a random pattern. Future work will involve more research on the overhead of the location prediction and the application of the proposed approach to other routing approaches such as AODV.

REFERENCES

[1] Z.J. Hass, R. Pearlman, "Zone routing protocol for ad-hoc networks", Internet Draft, draft-ietf-manet-zrp-02.txt, work in progress, 1999.

[2] Liang Qin, Thomas Kunz, "Increasing Packet Delivery Ratio in DSR by Link Prediction", Proc. 36th Hawaii International Conference on System Sciences, Track 9, Volume 9, pp. 300-306, 2003

[3] S. Das, C. Perkins, E. Royer, "Ad hoc on demand distance vector (AODV) routing", Internet Draft, draft-ietf-manetaodv-11.txt, work in progress, 2002.

[4] R. Ramanathan and J. Redi, "A brief overview of ad hoc networks: challenges and directions", IEEE Communications Magazine, Vol. 40 Issue. 5, pp. 20 –22, 2002

[5] Mehran Abolhasan, Tadeusz Wysocki, and Eryk Dutkiewicz, "A review of routing protocols for mobile ad hoc networks", Ad Hoc Networks, Vol. 2, No. 1, Pages 1-22, 2004

[6] T. W. Chen, M. Gerla, "Global state routing: a new routing scheme for ad-hoc wireless networks", Proc. IEEE International Conference on Communications (ICC), Vol. 1, pp 171 – 175, 1998.

[7] S. Murthy, J. J. Garcia Luna Aceves, "A routing protocol for packet radio networks", Proc. First Annual ACM Int. Conf. on Mobile Computing and Networking, Berkeley, CA, pp. 86–95, 1995.

[8] Zhengmin Shen, "Location-Triggered Routing Protocol in MANETS", Master's Thesis, Computer Science, Oklahoma State University, 2004.

[9] Petteri Kuosmanen, “Classification of Ad Hoc Routing Protocols”, Finnish Defence Forces, Naval Academy, www.netlab.tkk.fi/opetus/s38030/k02/Papers/12-Petteri.pdf.

[10] D. Dhillon, T. S. Randhawa, M. Wang, “Implementing a Fully distributed certificate Authority in an OLSR MANET”, RSA Security Inc. Vancouver, BC, Canada.

[11] Liang Qin, Thomas Qunz, “Pro-active Route Maintenance in DSR”, Proc. ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 6, Issue 3, pp 79 -89, 2002.

[12] Y. Zhang, W. Lee, “Intrusion detection in wireless ad hoc networks”, Proc. ACM International Conference on Mobile Computing and Networking, pp 275 -283, 2000.

[13] Bin Lu, Udo W. Pooch, “A Light Weight authentication Protocol in Manets”, Proc. International Conference on Information Technology: Coding and Computing (ITCC'05), Vol. 2, pp 546 – 551, 2005.

[14] Stephen Carter, Alec Yasinac, “Secure Position Aided Ad hoc Routing”, Proc. International Conference on Communications and Computer Networks (CCN02), pp 329-34, 2002.

[15] A. Perrig, R. Canetti, J. Tygar, D. Song. “Efficient authentication and signing of multicast streams over lossy channels”. Proc. of IEEE Symposium on Security and Privacy. pp. 56-62, May 2000.

[16] S. Zhu, S. Xu, S. Setia, and S. Jajodia. “LHAP: A Lightweight Hop-by-Hop Authentication Protocol for Ad-Hoc Networks”. ICDCS 2003 International Workshop on Mobile and Wireless Network (MWN 2003), Providence, pp. 749-755, Rhode Island, May 2003.

Appendix

To determine speed and direction given that we know the time *t* between two points.

Let A be point $x_A,y^{,}$

Let B be point x_B,y_B

Determining the distance and direction traveled

$$d = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

$$\sin\theta = \frac{y_B - y_A}{d}$$

Speed *s* is therefore *d/t* where *t* is the time taken to travel from A to B

θ is the direction .

VITA

Salman Latif

Candidate of the Degree of

Master of Science

Thesis: Prediction based authentication for Mobile Ad hoc Networks

Major Field: Computer Science

Biographical:

Personal Data: Born in Karachi, Pakistan, on Apr 26, 1978, son of Mr. and Mrs. Latif Ur Rehman.

Education: Received the Bachelor of Electrical Engineering Degree from N.E.D University of Engineering and Technology, Karachi, Pakistan in May 2000. Completed the requirements for the Master of Science Degree in Computer Science at the Computer Science Department at Oklahoma State University in December 2004.

Experience: Employed by Silicon Pak, Karachi, Pakistan, as Web Developer, June 1999 to March 2000; employed by Office Automation Services, Karachi, Pakistan, as Software Engineer, March 2000 to September 2000; employed by Multicultural Affairs, Oklahoma State University, Stillwater, OK, USA, as Web Developer, August 2003 to June 2004; employed by Oracle Inc, Denver, CO, USA, as a Business Software Developer, August 2004 to present.

Name: Salman Latif Date of Degree: December 2006

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: Prediction based Authentication for Mobile Ad hoc Networks

Pages in Study: 57 Candidate for the Degree of Master of Science

Major Field: Computer Science

Scope and Method of Study: This research focuses on the link state prediction in MANETs to reduce the data packets that are dropped because of link failure. The packet loss causes considerable degradation of both real time and non-real time data. In most existing protocols, nodes keep using the link in MANETs until it breaks. Instead the prediction algorithm described in this paper predicts the topological change in order to perform a route rebuild prior to the link disruption. We are not suggesting a new routing protocol but rather we are proposing a new prediction algorithm with some enhancements to the new table-driven routing protocol LTR.

Findings and Conclusion: Our research shows that prediction algorithm proposed here in our thesis significantly reduces the number of dropped data packets when nodes are moving randomly in the network. It is efficient enough to find the substitute links in case of potential link breakage. We have compared our approach with a typical table driven protocol DSDV. DSDV gives the best performance when there are lesser number of nodes in the network and transfer approximately the same number of successful packets as our proposed algorithm, but its performance significantly reduce with the increased network density. On the other hand our proposed algorithm gives the best performance with less or highly populated networks where nodes are moving in a random pattern.

ADVISOR'S APPROVAL: ______________________________

GLOSSARY

Mobile Ad Hoc Network: A complete wireless network with no wire connection at any stage in the network. A Mobile ad hoc network (MANET) is a collection of communication devices that wish to communicate. In MANET, no base stations exist and each mobile host (MH) acts as a router and a packet forwarder. Networks can be formed and fragmented on the fly without the intervention of a system administrator or the presence of fixed network devices. The bandwidth available for the exchange of routing information in ad hoc networks is much less than that available in a wired internet.

Global Positioning System (GPS): The Global Position System is a satellite system used in navigation that allows determining the position of any object/node any place of the globe and in any kind of weather. The GPS works with an error of between 15 to 100 meters.

Hop: One hop is defined as the transit through one router. Each router always adds 1 to account for itself.

Network: In information technology, a network is a series of points or nodes interconnected by communication paths. Networks can interconnect with other networks and contain sub networks.

Node: In a network, a node is a connection point, either a redistribution point or an end point for data transmissions. In general, a node has programmed or engineered capability to recognize and process or forward transmissions to other nodes.

Router: A router is a device that determines the next network point to which a packet should be forwarded toward its destination. They examine packets, calculate paths, and make intelligent routing decisions. The router is connected to at least two networks and decides which way to send each information packet based on its current understanding of the state of the networks it is connected to.

Packet Delivery Ratio (PDR): The number of data packets received by destinations over the number of data packets supposed to be received by destination nodes.

Route Discovery: Used only when a node attempts to send a packet to a destination node and does not already know a route to it.

Bandwidth: Bandwidth (the width of a band of electromagnetic frequencies) is used to mean (1) how fast data flows on a given transmission path, and (2), somewhat more technically, the width of the range of frequencies that an electronic signal occupies on a given transmission medium.