Topology Prediction in Mobile Ad hoc Networks

Mohammad Al-Hattab

Submitted in partial fulfilment of the requirements of the degree of

Doctor of Philosophy

Faculty of Engineering and Information Technology
UNIVERSITY OF TECHNLOLGY, SYDNEY

June 2011

Certificate of Authorship/Originality

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted a part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Mohammad Al-Hattab

Acknowledgments

I would like to express my gratitude to my supervisor Dr. Johnson Agbinya whose generosity and commitment are above and beyond the call of duty. I appreciate his vast knowledge and skills in many areas and his assistance in completing this thesis. I would like also to thank my co-supervisor Dr. Zenon Chaczko for his continuous assistance and support.

I must also thank the following people from the University of Technology, Sydney Dr. David Davis for his assistance and advices, Dr Kias Al-momani for his support and invaluable philosophical debates, Dr. Maen Takruri for his support and assistance in programming and the exchange of knowledge.

I would like to acknowledge and thank the members of Databases and Mobile Computing Laboratory at the University of Illinois at Chicago for the use of some of their data in part of this thesis.

I would like to acknowledge the support of the University of Technology, Sydney through the UTSD scholarship.

I am deeply grateful to my sisters Ala'a, Deya, Doa'a and Hana and my brothers Eyas and Ahmed and my family in law for their loving support. I owe my loving thanks to my beloved wife Nadia and my lovely kids Yarra and Khaled for supporting me in this journey. Without Nadia's love, patience and encouragements I would not finish this thesis. I wish to extend my deep and warm gratitude to my father Khaled and my mother Helaneh. They raised me, taught me and always supported and loved me.

To my parents and my wife

Contents

List of Figures

List of Tables

List of Abbreviations

CRT	Critical Transmitting Range
FIFO	First In first Out
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
LDT	Location Data Table
LTS	Location Time Series
PFL	Predicted Future Location
RSSI	Received Signal Strength Indicators
RTS	Recent section of Time Series
SNR	Signal to Noise Ratio
TC	Topology Control
TOA	Time Of Arrival

Table of definitions

Mobile Ad Hoc Network	A set of mobile node connected by wireless links in node to node configuration without central connector. Nodes have routing capability
Bandwidth	The overall data rate between nodes in the network
Link	A channel or physical medium connecting two nodes in a network
Topology	A layout pattern of interconnection of various nodes
Prediction	To find out a future data based on current and past data
Topology prediction	To find out the future layout of the network based on some metrics
Topology discovery	To find out the current topology status of a network
Reactive approach	To find required information when needed it is also called on-demand approach
Proactive approach	Maintaining the information periodically whether it was requested or not
Unicast	Transmitting the information to a unique address
Multicast	Transmitting the information to a multiple addresses
Location service	A service that provide nodes with the geographical location of other node(s).
Polynomial regression	Find the polynomial of degree n that link an independent variable x with the dependent variable y
Cross correlation	A measure of similarity between two time series

Abstract

A mobile ad hoc network is an autonomous system of nodes that is connected by wireless links. Nodes are connected in the absence of fixed infrastructure or central management. They can freely move, join or leave the network. Arbitrary and random motion of mobile ad hoc network nodes while communicating results in frequent topology changes and multiple disconnections of links. This dynamic environment challenges the delivery of data, especially in real time applications, and imposes the need for prediction models to track these changes, and subsequent determination of the future topology of the network. The prediction of network mobility into the future will reduce the frequency of location and route updates and improve the efficiency of the network.

In this thesis, we have reviewed existing prediction methods for the mobility of nodes and proposed three new schemes to predict the future topologies of the mobile network. The proposed schemes are simple, efficient, and applicable to many existing algorithms. We evaluate the presented schemes on simulated and real data obtained from the Databases and Mobile Computing Laboratory at the University of Illinois at Chicago. The results show that the schemes successfully predict the future topology of the dynamic networks with high accuracy. They also minimise the frequency of route and location updates and any associated delays.

The implementation of the proposed schemes on the upper layers makes them applicable to various routing algorithms including topology based routing algorithms and geographical routing algorithms, giving them the advantage of platform independence.

Chapter 1

1 Introduction

1.1 Background

Mobile ad hoc networks have recently emerged as an important research area which has stimulated substantial research activities in the past ten years. Many scholars have been attracted to investigate this domain for further research and learning [1]. This development has been encouraged by the dramatic advance in wireless communication technologies, the low manufacturing cost of wireless devices [2], and the ease of equipping any node with wireless cards.

A mobile ad hoc network is an autonomous system of nodes that is connected by wireless links in an arbitrary shape [3]. Two nodes are considered to be connected if they are within the transmitting range of each other and do not require any infrastructure or central management. A mobile ad hoc network can operate in stand-alone fashion or it can be connected to the Internet [4] [5], and it can be easily set up anywhere.

Mobile ad hoc networks have unique characteristics that distinguish them from other types of networks. They are self-organizing, self-configuring, multi-hop wireless networks capable of adaptive reconfiguration after any topology change[6] [7]. two nodes can communicate via other nodes if there is no direct link between them [8], This property makes them reliable, self healing and fault tolerant. Link failures due to node mobility or node failure can be replaced automatically. It is also energy efficient, because transmission using multi-hop is more energy efficient than single link [9].

Nodes are connected in the absence of fixed infrastructure; they can move, join or leave the network freely, and therefore the topology of a mobile ad hoc network is dynamic. Every node in the network acts as a router [8]. Packets can be routed through different routes depending on the current status of the topology of the network. The routing decision is performed by the node, which forwards the packet to the appropriate node independently from any management of the network.

The properties of mobile ad hoc networks make them a better solution for future networks having countless applications, such as vehicular networks, military and battlefield communications [10], disaster relief applications where the infrastructure might be destroyed, virtual classrooms and ad hoc communications during meetings or lectures, entertainment, sensor networks and many other applications.

On the downside, the security of the mobile node is acritical issue because the network is an open medium; any malicious node can join the network [1]. A minimum mutual trust must be available between nodes, especially for the purpose of routing, which makes the network prone to attack [11]. However, security and trust issues are not addressed in this thesis.

Although the mobility of the nodes is an advantage that provides flexibility and makes many applications possible, it affects the stability of the links and causes multiple link failures and unpredictable frequent changes in the topology, thus requiring a huge communication overhead to establish and maintain routes and reduce packet loss [12].

The unpredicted frequent changes of the topology of the network become a common problem in mobile ad hoc networks and make routing a challenging task [13]. My investigations and research into mobile ad hoc networks protocols found that the development of methods to overcome this problem is essential. This problem causes delay in data delivery because routes to the destination must be searched after any change; alternatively, if the routes become available, a large percentage of the available bandwidth will have been consumed through frequent route updates. This problem has not been addressed intensively in the literature and solutions have not been yet achieved.

In this thesis, I have presented a general and comprehensive framework to solve the problem of topology changes. The solutions address different prediction schemes, the prediction of the entire network mobility, and the prediction of the node's own mobility. This makes the framework presented in this thesis applicable to different networks with different scheduling methods.

To further illustrate the validity of the proposed topology prediction schemes, they were used in conjunction with real location data [14]. Moreover, a comparison with a well-known protocol was carried out which achieved better data delivery and less delay with the presence of the frequent change in the topology of the network[15].

1.2 Problem statement

Consider that a mobile ad hoc network consists of *n* nodes distributed randomly in a geographical area such as the one shown in Figure 1.1. These nodes are free to move in an arbitrary path. The topology of the network at time *i*, which is denoted by *TPi* can be determined by two factors: the location of each node at time *i* and the propagation model which depends on the environment of the network.

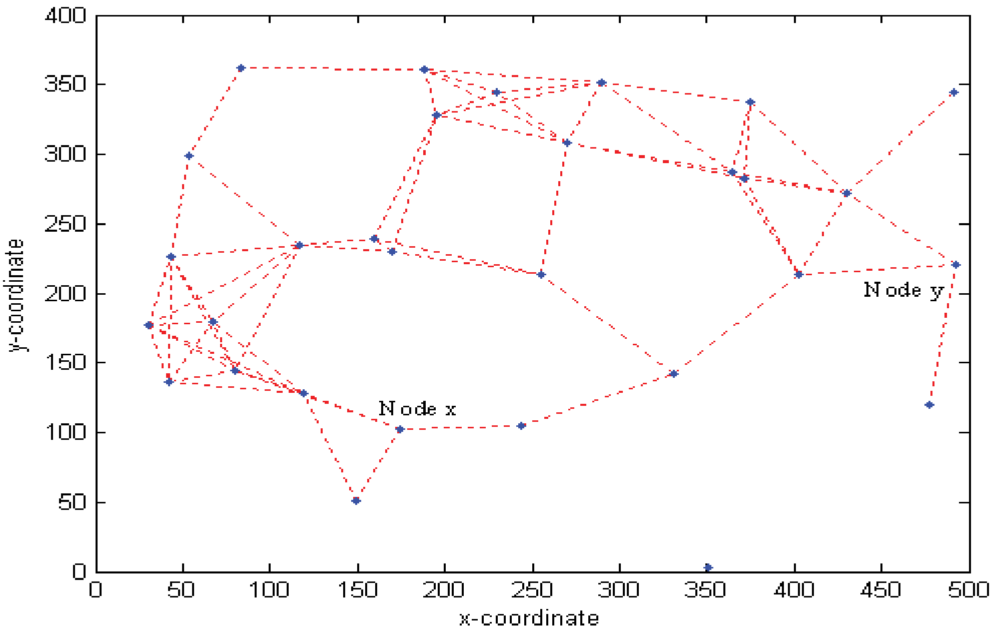

Figure 1.1: Mobile ad hoc network deployed in geographical region

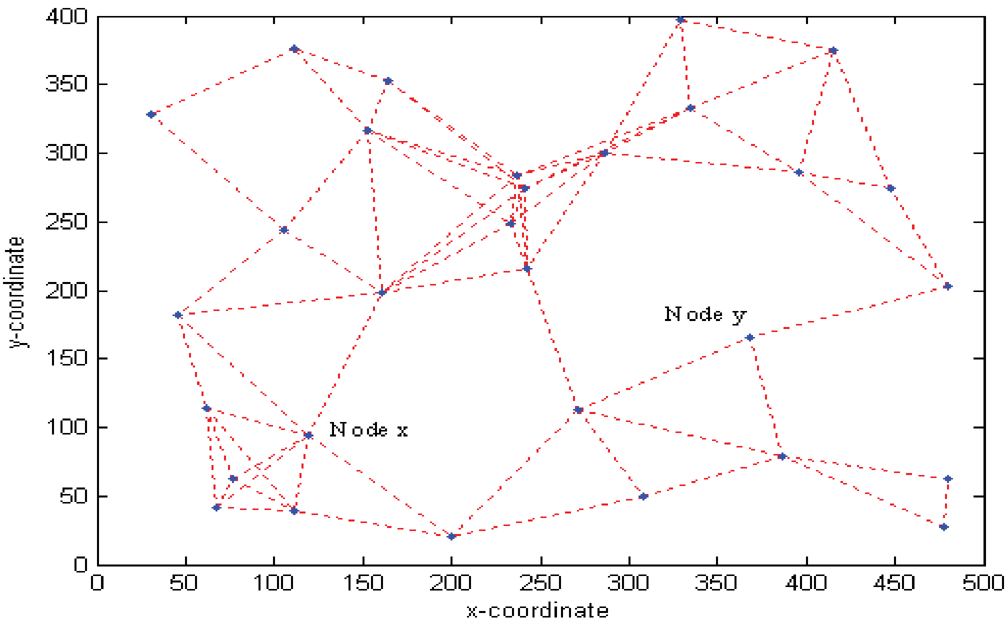

Figure 1.2: The effect of the mobility of nodes on the topology of the network

Assume that at time i the topology is known to node x and node x wants to send a packet to node y. Node x will send the packet to node y according to the current topology. As the time progresses, the topology at time $i+c$ would change due to mobility of nodes to become TP_{i+c} as it appears in Figure 1.2. At this time if node x has another packet to send to node y, or to any other destination. There will be three cases to consider:

1. Reactive case:
   If the network follows a reactive approach in scheduling and maintaining the routes, node x will search for a valid route to node y and then send the packet to the destination through that route. Node x performs the search because in reactive cases, the nodes do not receive a frequent routes update.
2. Proactive case:
   If the network follows a proactive approach in scheduling and maintaining the routes, node x will have obtained an available route to y, because every node receives frequent updates from the entire network.
3. Static case:
   Node x will send the packet using the same route because the topology is fixed.

The third case can not be considered because the network is mobile and the topology is dynamic, even if some nodes are not moving. The first case will experience a transmission delay because if the route to the destination is not available or not fresh enough, node x will issue a route request/ route reply process to obtain a route. This process is associated with delay, and the delay becomes greater in a high mobility environment. The second case requires a frequent update to be sent by each node. These frequent updates consume a significant percentage of the available bandwidth; moreover, the frequent updates may result in a congested network which may cause packet drops or more delay.

In order to mitigate this problem, each node in the network has to predict the changes in the network topology for some future time. By knowing the future topology, the node can know the future routes without searching for new routes through the route request/route reply phase. Moreover, the frequent route update will be reduced because information about future topology is available through prediction.

The topology prediction that we consider in this work is of two types. The first type is the prediction of the topology change of the entire network. Each node predicts the changes for all other nodes through current and/or past knowledge of the network status. The second type is the prediction of the node own future status. Each node predicts its own future changes based on information about current and previous status, and on the knowledge of the future trajectory of the node. The predicted status will be broadcast to the network in order to predict the future topology.

1.3 Importance of the research

Many applications in mobile ad hoc networks require real time data flow with timing constraints for the delivering of the data [16]. In such applications, any changes in the topology of the network due to the mobility of the nodes should not reflect on the quality of service or the routing of data. Hence, no delay should be associated with reactive route updates, nor should congestion in the available bandwidth be associated with proactive updates [7]. The reconstruction of the routes or the discovery of the topology after any change will be at a cost of congestion and/or delivery delay. If the future topology of the network can be predicted, the reconstruction of routes prior to the topology change will ensure a smooth flow of data with a minimal control overhead and without delay [16]. The prediction of the changes in the future topology of the network can achieve a better congestion control, better resource reservation and quality of service [17].

1.4 Thesis structure and contributions

The thesis consists of four major parts. The first part, Chapter 2, is an overview of mobile ad hoc networks, protocols, and a review of existing topology discovery and mobility prediction schemes. The second part, consisting of Chapter 3 and Chapter 4, presents the mobility model used in this work, the linear prediction method using the current status of the node motion, as well as the implementation and performance analysis. The third part, Chapter 5, presents a prediction method of mobility using polynomial regression, implementation details and performance analysis. The fourth part, Chapter 6, presents mobility prediction methods using pattern matching and correlation analysis, implementation and performance analysis.

The thesis also contains a comparison between all the above prediction methods which is presented in Chapter 7. Finally, Chapter 8 concludes the thesis and presents the future research direction.

Chapter 2 defines mobile ad hoc networks and articulates their characteristics, applications and challenges. It classifies mobile ad hoc networks protocols and reviews many protocols based on this classification. It reviews geographical protocols and shows how locations data is used in routing. An extensive survey of existing mobility prediction and their applications is given.

Chapter 3 defines network topology and shows how it can be represented and presenting a novel mobility model that represents the movement of the nodes. It addresses the problem of the dynamic changes of the topology and explains its effects on the network. It also shows how to benefit from the current mobility status of the node to predict the future status. The chapter argues that the prediction of the future topology of the network reduces the delay associated with obtaining routes to the destination and reserves a significant percentage of the bandwidth. It also argues the feasibility of employing GPS data in prediction. The contributions of the chapter are:

- Presenting a new mobility model based on the regularity of nodes movements and the fact of a non random behaviour of mobile networks.
- Introducing a simple prediction scheme which uses current location, speed and direction of nodes to predict future mobility.
- Validating the proposed scheme by showing how the use of a topology prediction scheme will reduce delay and reserve the bandwidth.
- Showing the viability of the proposed scheme in predicting future mobility with a rational error.

Chapter 4 modifies the algorithm in Chapter 3 to overcome its shortcomings. The algorithm in Chapter 3 assumes that the node whose mobility we are estimating maintains speed and direction until the end of the prediction period. It does not consider the sudden change in mobility status, and this drawback has been taken into consideration in this chapter. In this modification, the node continuously observes its own mobility status and responds to the sudden change by broadcasting an immediate update. The modification improves the performance of the scheme, but at the cost of extra updating in case of sudden change. The contributions of this chapter are:

- Introducing a simple and modified prediction scheme to improve the performance of the linear prediction model in Chapter 3. The model considers the sudden changes in the status of the mobility.
- Showing the viability of the proposed scheme in predicting future mobility with a rational error.
- Obtaining a longer prediction cycle concurrently with achieving a rational error in predicted topology.

Chapter 5 introduces a novel prediction scheme to predict the mobility of the node. Unlike the previous chapters, where the node predicts the mobility of all other nodes in the network, the model in this chapter predicts the own future mobility of the node. As the nodes are assumed to have a GPS receiver, they are able to find a set of future locations. The solution in this chapter lies in mapping the future locations of the mobile node with time. This mapping is achieved by polynomial regression. The scheme constructs a polynomial with degree *m* that maps the x and y coordinate of the points, and then finds the time stamp corresponding with each point. The predicted mobility is then exchanged between nodes to construct the future topologies of the network. The main contributions of this chapter are:

- Introducing the use of polynomial regression to predict the future topology of mobile ad hoc networks. This algorithm deals with sudden changes in mobility.
- Showing that the algorithm can achieve good prediction for future mobility with long prediction cycles.

Chapter 6 introduces a mobility prediction scheme that predicts the future mobility of nodes using pattern matching. The predicted mobility is exchanged between nodes to determine future topologies of the network as a discrete function of time. Each node monitors and stores its own locations as a time series for a sufficient length of time. When a node wants to predict its future mobility, the most recent part of the series is compared with the rest of the series to find a similar pattern. This pattern matching technique is used to find similarity between the recent part and the rest of the time series. When matching is found, the future locations will be similar to the time series where the matching was found. The main contributions of this chapter are:

- Introducing a prediction algorithm that uses the previous mobility status of the node to predict the future. This algorithm predicts its own future mobility of the node.
- Showing the viability of the proposed algorithm in predicting the future mobility with a rational error using real location data.
- Obtaining a longer prediction cycle concurrently with achieving a rational error in the predicted topologies.

1.5 Methodology

Arbitrary and random motion of mobile ad hoc network nodes while communicating results in frequent topology changes and multiple disconnections of links. This dynamic environment challenges the delivery of data, especially in real time applications, and imposes the need for prediction models to track these changes, and subsequent determination of the future topology of the network. The prediction of network mobility into the future will reduce the frequency of location and route updates and improve the efficiency of the network.

The methodology followed in this research can be summarised by:

- Obtaining trajectory data for mobile nodes.
- Use some of the trajectory points as an update of the algorithms.
- Predicting the future trajectories of the nodes using different prediction algorithms.
- Comparing the predicted trajectories to the actual trajectories.

Different metrics have been used in analysing the results including the update period length (T), the threshold direction, the threshold speed, the degree of the polynomial that represent the trajectory, the number of points in polynomial regression, and other metrics.

The data used in this work was obtained from two sources: firstly, computer generated data using mobility model. Secondly, GPS data that been collected from Databases and Mobile Computing Laboratory at the University of Illinois at Chicago during their daily commute.

In the mobility model, each node chooses a destination point at random in the area of interest; therefore, the destination of the mobile node is known. The node chooses a random direction $\theta \in [0, 2\pi]$ and checks the X and Y coordinates for a point of constant distance r from its current position in that direction. θ is obtained using a Gaussian distribution with an average value $\mu = 0$ and the standard deviation $\sigma = \frac{\pi}{4}$.

If the point is closer to the destination than the current position, the node will maintain travel in that direction at a given speed until it reaches the point. When the node reaches the point, it chooses a new valid direction and new speed. This process is continued until the node becomes close enough to the destination. Both collected and generated data contains node's X and Y coordinates, speed and direction of movement.

All tests have been implemented using MatLab on a given area and a certain number of nodes distributed randomly within this area.

In the first set of experiments, the simulation will test the ability of the algorithm to predict the future trajectory of the nodes for different values of the prediction cycle T. Moreover, it will test the effect of the threshold angel and threshold speed when a modification for the motion prediction algorithm is applied. The data in these set of experiments have been generated using the mobility model which was mentioned above.

The second set of experiments tested the ability of another proposed algorithm to predict the future trajectory of the nodes using polynomial regression. In this part, the purpose of the algorithm is to find a mapping between the coordinates on that path and the time, which means finding the x and y coordinates as a function of time, which in turn enables the prediction of future locations of the mobile nodes as a function of time. All variables including the length of the future prediction period, the number of points and the degree of the polynomial that represents the trajectory have been analysed. The data in these set of experiments will be obtained by the mobility model.

The third set of experiments uses the collected data to test the ability of the newly proposed algorithm to predict the future trajectory using pattern matching technique. Each node monitors and stores its past location to obtain a time series of locations for a given period of time. We refer to this series as the Location Time Series (*LTS*). When a node wants to predict its future locations, it searches the *LTS* for a pattern similar to the current situation. We refer to this pattern as the Recent section of Time Series (*RTS*) or 'the query'. It starts at the current point in time and goes back for *O* points. *O* is called the 'query order'. Several series can be stored and searched for best matching; searching more series will give a higher probability of pattern matching. All variables in these set of experiments including: RTS, the number of LTS and the length of required future trajectory have been varied to test their effect of the algorithm.

Finally, comparisons between all proposed algorithms have been carried out using different combinations of all variables to test their feasibility.
For all the three set of simulations, the future topologies of the network were determined based on the predicted trajectories and the propagation model used by the network.

1.6 Publications developed from this thesis

Several publications were produced based on the research described in this thesis; these publications which were peer reviewed include a book chapter, two journal papers, and four conference papers. Details of these publications are listed below:

Journal papers

- M. Al-Hattab and J. I. Agbinya, "Trajectory estimation for wireless mobile networks using polynomial regression," *International Journal of Electronics and Telecommunications,* vol. 56, pp. 451-456, 2010.

- M. Al-hattab, J. Agbinya and M. Takruri "Mobility prediction using pattern matching", Accepted at *International Journal of Electrical and Computer Engineering*.

Book chapter

- M. Al-Hattab and J. I. Agbinya, "Planning of wireless mesh networks," in *Planning and Optimisation of 3G and 4G Wireless Networks*. vol. 8, J. I. Agbinya, Ed.: River, Aalborg, 2010, pp. 201-222.

Conference papers

- M. Al-Hattab and J. I. Agbinya, "Self trajectory estimation for wireless mobile networks," presented at the 4th International Conference on Broadband Communications, Information Technology and Biomedical Applications, Wroclaw, Poland, 2009.
- M. Al-Hattab and J. I. Agbinya, "Use of street maps to aid node localization in mobile wireless networks," in *Proceedings of International Symposium on Parallel and Distributed Processing with Applications* 2008, pp. 471-476.

- M. Al-Hattab and J. I. Agbinya, "Topology prediction and convergence for networks on mobile vehicles," in *Proceedings of International Conference on Computer and Communication Engineering*, 2008, pp. 266-269.

- M. Al-hattab and J. Agbinya, ”Localization for dynamic networks nodes carried by mobile vehicles” *Engineering showcase 2008, Faculty of Engineering, UTS.* University of Technology, Sydney, 2008, pp. 42-43.

Chapter 2

2 Literature review

2.1 Introduction

A mobile ad hoc network is a wireless multi hop network that consists of many nodes, connected through radio communication. Nodes move in a random or determined manner which results in a variable topology network. Ad hoc networks are infrastructure-less networks [18]; that is, they do not have fixed routers and all nodes are capable of moving and being connected in an arbitrary manner whereby they can act as routers and regular nodes at the same time [19]. Data and traffic generated by mobile ad hoc nodes cannot be forwarded using wired network protocols because of frequent topology changes. The development of protocols that satisfy the requirements for mobile ad hoc networks have therefore become inevitable [20].

Many applications are implemented in ad hoc networks such as military applications, disaster areas, on road transport communications, wireless mesh networks and many others. The most common desired characteristics for all applications in ad hoc networks are to reduce network traffic caused by control traffic and packet overheads and to minimise any delay associated with route search.

To develop a protocol that satisfies the frequent topology change and improves the efficacy of the network, it is necessary to understand the current protocols.

2.2 Classification of Ad Hoc Networks Protocols

Routing protocols can be classified according to different criteria. These classifications include scheduling, communication model, state information, structure and type of cast [21].

2.2.1 Scheduling

Scheduling means to determine how topology information is acquired and maintained by mobile nodes. It can be obtained by a periodic-based process known as *proactive*, or it can be obtained on demand-based known also as *reactive* [2]. Scheduling is the most desirable and spread classification criterion.

Reactive protocols (on demand)

Reactive protocols do not maintain up-to-date routing information. Information about the route is requested and obtained when required [22] [23]. This can be achieved by route discovery/route maintenance messaging. The nodes initiate a route request and get a route reply; once the route is established, it is maintained until the route is no longer needed or it fails to satisfy a connection between the source and the destination. Reactive protocols have some limitations which include the fact that route discovery is required, because the route is maintained while in use. Route request/route reply leads to delay in transmitting the first packet. Despite route maintenance being performed only for routes in use, it may still generate traffic due to frequent topology changes. Moreover, some packets may be lost if the route to the destination changes[24]. Classic examples of these protocols include DSR[25] and AODV[26].

Proactive protocols

Proactive protocols keep up-to-date routing information about the entire network in every node[22] [23]. Nodes periodically propagate route updates to the entire network. Nodes also propagate updates as a response to any topology change in the network. The main drawback of proactive protocols is that the maintenance of unused paths may occupy a significant part of the available bandwidth, especially if the network is on the move with frequent topology changes[27]. Examples of proactive protocols are DSDV[28] , OLSR[29] and TBRPF[30].

2.2.2 Communication model

Wireless network protocols can be classified according to the communication model into two types: single channel and multiple channels. Single channel protocols are Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) oriented; they rely on specific link layer behaviours [31]. Multi channel protocols are low-level routing protocols that combine channel assignment and routing functionality; such protocols are used in time division multiple access (TDMA) or code division multiple access (CDMA)-based networks[19].

2.2.3 State information

Protocols can be classified in terms of state information into topology-based and destination-based. In topology-based protocols such as DSVR [28], DSR[25], AODV [26] and ZPR [32], each node maintains large scale topology information about the entire network [31]. Nodes know a complete and least-cost path to the destinations, and each node has a routing table that contains paths for all possible destinations in the network. Destination-based protocols do not maintain a large scale topology information[31], nodes have topology information about neighbours. During routing decision a node determine which of its neighbours is the closest to the destination (i.e. the next hop). Examples of destination-based protocols are: GPSR [33] and LAR [34]

2.2.4 Structure

Protocols can be classified in terms of structure into two types: uniform and non-uniform protocols [2]. This classification determines the method of treating nodes during routing and which of the nodes have a routing scheme. In uniform protocols, all nodes play the same role and have the same functionality in routing; they can respond and send routing control messages. DSR and AODV are classified as uniform routing protocols.

By contrast, non-uniform protocols distinguish between nodes because there is a hierarchy structure in the protocols. Some nodes carry out distinct management and routing roles. Zone Routing Protocol ZPR and Clustered Gateway Switch Routing (CGSR) [35] are classified as non-uniform routing protocols.

2.2.5 Type of cast

Protocols can be classified by the type of cast. This includes unicast and multicast [2]. In unicast, a source node sends packet(s) to one destination; this is the most common means of communication in any network. In multicast, the destination is more than one node. Multicast protocols create a mesh or tree from the source to several destinations. Multicast routing protocols, such as Distance Vector Multicast Routing Protocol (DVMRP) and Multicast Open Shortest Path First (MOSPF) have been widely used in wired networks [2].

2.3 Related work

2.3.1 Overview of selected mobile ad hoc networks protocols

This section reviews the most relevant protocols used in mobile ad hoc networks and shows their advantages and shortcoming. In the literature, many of the proposed protocols are analogical in terms of discovering the topology of the network and the scheduling of the updates of the topology. Therefore, selected protocols that use different topology discovery techniques will be reviewed.

The dynamic source routing protocol (DSR) is an on-demand, topology aware, uniform routing protocol that uses source routing instead of relying on a routing table in each intermediate node [25]. DSR has two phases: route discovery and route maintenance. The DSR route discovery process is a typical query response cycle: if the source's routing cache does not contain a path to the destination, the source creates a Route Request (RR) packet and broadcasts it into the network. The Route Request packet has several fields: source ID, destination ID, sequence number generated by the source node and a record of all nodes the packet has traversed. When a node receives a RR packet, it checks the RR fields and takes one of the following actions:

1. Discards the packet if it has seen this packet before.
2. Replies with a route reply if the node itself is the destination or a route to the destination is available in its cache.
3. Adds its own identity to the route record in the RR packet and rebroadcasts it.

This process continues until the RR packet reaches the destination or reaches a node with a route to the destination.

When the destination node receives the RR packet, it replies to the source node though the reverse path traversed by the Route Request.

Dynamic source routing protocol (DSR) is an on-demand protocol that restricts the bandwidth consumed by control packets in ad hoc wireless networks. It eliminates the periodic messages required in the table-driven approach. Moreover, it is beacon-less and hence does not require periodic Hello packet (beacon) transmissions, which are used by a node to inform its neighbours of its presence.

Stale route information in the node may result in an incorrect path to the destination, because there is no expiration time for the route. The route error message may not return to the source node due to the mobility of nodes, because the path to the source node could change and become different from the stored path in the route error packet. As in most reactive algorithms, DSR suffers from excessive delay when establishing the route in a high mobility environment [36]. This delay is extensive because DSR uses source routing.

The Optimized Link State Routing Protocol (OLSR) [29] [37] is a proactive, non-uniform, topology aware link state routing protocol. Every node periodically broadcasts the "Hello" message with information to specific nodes in the network to exchange information about the neighbourhood. Each node selects a set of its neighbour nodes as "multi point relays" (MPR). In OLSR, only nodes selected as such MPR are responsible for forwarding control traffic. MPR provides an efficient mechanism for flooding control traffic by reducing the number of transmissions required. The information in the Hello message includes node IP, sequence number, a list of neighbours with links information and the node's MPR.

When a node receives the periodic Hello message it checks the sequence number of the message and update its own routing table and rebroadcasts the Hello messages to its MPR nodes. Each node broadcasts a partial topology into the network, resulting in the entire network topology being known by all nodes. All destinations within the networks are known and maintained before use; there is no route discovery delay associated with finding a new route.

OLSR does not have any mechanism for link existence and detection; it simply assumes that the link is exists if it has received a recent Hello message. In a high mobility environment when the packet loss is high, the use of MPRs to forward the control packet could lead to a problem. The traditional proactive method in OLSR does not suit a high mobile environment because routes converge poorly due to mobility [10]. Similar to all pure proactive protocols, OLSR node's power and network resources are wasted on possible unused routes.

The spread of GPS-enabled devices and the reduced cost of implementing GPS receivers into mobile nodes encourage researchers to use location information in routing decisions. The delivery of data to a destination identified by its location is called geographical routing or geocasting. Dommety and Jain suggested the use of location information for routing decisions [38]. The location of nodes can be obtained by Global Positioning System GPS [39] or any type of positioning service [40] [41].

In most geographical routing protocols, nodes do not need to maintain routing tables to route packets. Instead, they need to know the location of the destination and the location of their neighbour that is closest to the destination node [24]. The location of the destination node can be obtained by different methods depending on the protocols; some protocols maintain a table of locations for all other nodes in the network through exchanging periodic control messages such as DREAM [42], whereas other protocols use a rendezvous-based method whereby a rendezvous server provides the network with the required location of nodes. SLURP [43] divides the network into sub-areas in which the location of nodes can be obtained by a special node in each area.

A Distance Routing Effect Algorithm for Mobility (DREAM) is a geographical routing protocol where each node in the network maintains a table of locations for all other nodes in the network. Each node periodically floods the control packet that contains the node's location, ID and life time (distance). A node can control the frequency of the control packet based on its own mobility; nodes with higher mobility propagate updates more frequently than nodes with less mobility. Moreover, this can adjust the distance 'life time' of the packet, which guarantees more position accuracy updates for neighbour nodes and fewer for far away nodes. This is due to the distance effect, as described by Basagni et al.[42]

> *The greater the distance separating two nodes, the slower they appear to be moving with respect to each other*.

When a node receives a control packet, it checks the distance in the packet and calculates the distance between itself and the sender of the packet. If the distance in the packet is higher than the one that has been calculated, the node rebroadcasts the packet, otherwise it discards it.

Control packets are broadcast more frequently with short distance and less frequently with longer distance, resulting in frequent location updates for neighbours and less frequent updates for distant nodes [44].

When sender node *S* needs to send a message to destination *D*, it checks the location and the direction of *D* and chooses one or more nodes of its directed neighbours who are in the same direction as *D*. The neighbours do the same thing until the message reaches *D*.

DREAM is bandwidth and energy efficient because the control packet has minimal size; also, bandwidth is saved by distance effect and the use of a less frequent control packet with higher distance. It is loop free as the packet travels away from the source. Packets can follow independent routes and have adaptive mobility.

On the other hand, this version of DREAM always assumes that at least one of the sender's neighbours is in the direction of the destination, which might not be always the case. Moreover, as the size and area of the network increase, these are more likely to have fewer updates for far nodes which increase the uncertainty of the nodes' location, even when the distance effect is considered. The proactive nature of DREAM makes it unsuitable for large scale networks.

Greedy Perimeter Stateless Routing (GPSR) [33] is a geographical protocol that uses a periodic beacon to update its location by propagating it to one-hop neighbours. For efficiency, GPSR piggybacks the local sending node's position in all data packets and resets the time of the periodic beacon. Thus, a node sends a beacon if it has not sent any data for a certain time. If a node has not received a beacon from a neighbour for a certain time, it removes that neighbour from its table.

Nodes in GPSR perform greedy forwarding based on the immediate neighbours. If the greedy forwarding fails, the algorithm recovers by routing around the perimeter of the region where the greedy forwarding failed.

When a node needs to send a message to a destination node, it obtains the coordinate of the destination from the location service and checks which of its neighbours is the closest to the destination. The intermediate nodes do the same thing until the message reaches the destination.

If the greedy forwarding fails, which means that the sender is the closest node to the destination within the neighbourhood, the algorithm uses the right hand rules to send the packet around the region where the greedy forwarding failed. Krap and Kung explained this rule in detail [33].

GPSR assumes that a node can obtain the location of the destination from a location service. The assumption of the availability of the service with no cost in terms of delay is the main drawback of GPSR. Moreover, the void region concept assumes all nodes are distributed in two dimensional spaces which is not always true. The right hand rule in perimeter mode does not always give an optimal path to the destination.

Scalable Location Update-based Routing Protocol (SLURP) [43] is a routing protocol that uses geographic location management strategy to keep the overhead of the packet small. This protocol has its own location management. The location management maintains the locations of all nodes in the network and the algorithm divides the network into regions. Each node is assigned a home region which maintains the location information of the node. The algorithm has a mapping function that maps the node's ID to its home region. When a node moves into new region, it informs all the nodes present in its home region about the new region.

When a source node S wants to send a message to a destination node D, S sends a location discovery packet to D's home region, obtained by the mapping function. This packet contains the destination's ID; in addition, it contains the source's ID, current location and a sequence number. Upon receiving the location discovery packet, the neighbours update their tables about the location of the source, and the neighbour which is the closest to the destination home region will forward the packet. This process continues until the packet reaches the home region. The first node in the home region will respond with a location reply containing the destination's current location.

During data delivery, a data packet contains source ID, the source location, destination ID and destination location. If a data failure occurs, an error message will be sent to the source and the source will send the location discovery packet again.

The mapping function between the node and the home region add complexity to the protocol; moreover, the delivery of data fails if the home region has no node at the time of communication.

Location aided routing (LAR) in a mobile ad hoc network [34] is a geographical routing algorithm proposed to reduce the number of routing messages by limiting the area of the search for a new route to a region in which the destination is expected to exist, the *Expected Zone*, and the region separate from the source and destination, the *Request Zone*. These regions can be estimated based on information obtained when the source and destination were last communicated [45].

LAR proposes two routing schemes: LAR Scheme 1 and LAR Scheme 2. In LAR Scheme 1 the request zone is a rectangle and the expected zone is a circle centre at the destination coordinates corresponding to time t_0, with radius $R=v(t_1-t_0)$ where v is the maximum speed and t_1 is the request time. Figure 2.1 illustrates the concept of the request zone and expected zone.

When the source node sends a route request message, the message will contain the coordinates of the request zone. Any node within this zone will forward the message to its neighbours, and any node outside the zone will discard it. When the request reaches the destination node, it will reply with a route reply that includes the coordinates of the destination, the time stamp and the destination speed.

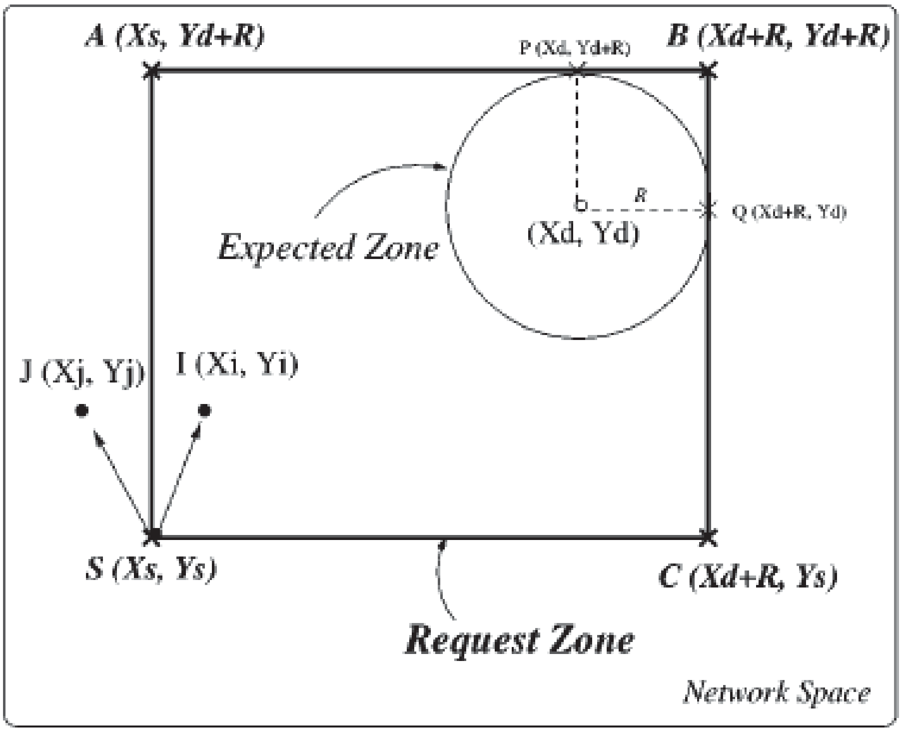

Figure 2.1: An illustration for the concept of request zone and expected zone in LAR Scheme 1[34]

In LAR Scheme 2 the route request includes extra two pieces of information added to LAR 1scheme: the location of the destination and the distance to the destination. When any intermediate node receives the request, it computes the distance to the destination and compares it with the distance in the route request. If the distance between the intermediate node and the destination is less than the distance in the packet plus a certain value, then the intermediate node will forward the packet; otherwise, it will discard the packet. This means the request will travel toward the destination only.

This improvement may cause a problem in a case where no neighbours of the intermediate node are closer to the destination. In this case, the request will be discarded and the algorithm will fail LAR is considered to be a route search rather than a routing protocol because there is no clear policy when a route search fails due to stale information. Because the location information is obtained from ordinary communication between nodes and the last time the nodes were in communication, this information could be stale if the two nodes do not communicate very often [46].

2.3.2 Topology discovery and mobility prediction

Topology discovery is a task of providing the node with the required routing information which includes the status of the network topology or the locations of nodes. The continuous discovery of network topology consumes a large portion of network resources while the on demand discovery causes extended delay. The conventional reactive and proactive topology discovery schemes are therefore not suitable for highly dynamic mobile networks [47].

The authors in [48] [49] [50] use mobile agents to discover the topology of the network. Agents circulate the network and obtain the local topology of each node, and when they meet they exchange this information. Topology discovery using mobile agent does not provide instantaneous topology for the network. It is intensive in time and messages to discover the topology of the whole network [51], which does not work efficiently in high mobility environments.

Link state routing protocols discover the topology of the networks through route requests and route replay messages. These messages are propagated in the networks by different methods. Proactive topology discovery in proactive protocols such as DSDV[28] OLSR [29] and TBRPF [30] send periodic messages to update the state of the network topology. This conventional proactive discovery method does not suit networks with high mobility because unnecessary high traffic in the network may be generated for possibly unused routes.

DSR [25] and AODV [26] discover the network topology using on demand route request/route reply messages. As in all on-demand routing protocols, an extended delay is expected to obtain the topology update which will affect the delivery of data, especially in real time applications. Moreover, the frequent mobility of nodes may invalidate the obtained routes causing high traffic in the network.

Geographical routing protocols discover the topology of the network by disseminating the location of nodes in the network. The location information can be used to construct global topology such as in DREAM or a local topology such as GPSR.

Regardless of the method of topology discovery, nodes use this topology to deliver data from the source to the destination. If the topology of the network changes due to the nodes' mobility, or any other reason, then one or more routes becomes invalid. To address this change, the topology should be re-discovered. The discovery of the new topology will delay the delivery of data, or it might be invalid again due to the high mobility of the nodes.

If the changes can be predicted, it may be possible to establish a stable route, or known future changes in the topology. This can be achieved by predicting the future links quality between nodes or by predicting the future mobility of the nodes.

The prediction of the mobility of nodes has been extensively addressed in the literature. In cellular networks, many mobility prediction schemes have been proposed [52] [53] [54] to improve the handover between cells. An accurate mobility prediction can offer smaller call-dropping probability and reduce the handover latency [55]. The authors in [54] proposed that the mobile node should store its movement history as a means of predicting future movement. A mobile node stores a record containing the current location, and the next visited location when the mobile node is at this location. The most visited location will be chosen as the next predicted location. The visited locations are also used to form a segment of movement between locations. A search for similarity between the current segment and past segments is then used to predict future locations. The prediction relies on the history of the individual node which may not lead to any prediction in new locations.

The proposed method does not predict the location of the node; rather, it predicts the future cell which the node is going to visit, but this prediction does not provide enough information to enable data delivery in mobile ad hoc networks.

The mobility motion prediction algorithm [52] uses a Markov model to predict the mobility of a node in cellular networks. Movements are stored as a sequence of states in an itinerary–pattern base. The future movement of the node can be predicted by comparing the present itinerary with the stored itinerary-pattern base using correlation analysis.

The regular path recognition algorithm [53] employs the behaviour of humans to predict the mobility of nodes. It records the cells and the period of time the node stays in each cell as paths for this node. The current path is compared with previously stored paths to predict the movement of the node.

The algorithms in [52] and [53] rely only on the history of movement of individual nodes which may not lead to a correct prediction if the node visits a cell for the first time. A small random variation in the mobility of the node could lead to the failure of these methods.

The dead reckoning technique [46] uses location information to predict the mobility of the node in mobile ad hoc networks. A mobile node samples its own location periodically and constructs a model of its movement. Two successive location samples are taken at known times to calculate the speed in both X and Y directions, then the calculated speed is used to estimate the next location. The distance travelled is calculated and compared to a threshold value. This threshold determines the allowable error in the system. The complexity of the model depends on the predictive ability of the mobile node. The node disseminates its current location as well as this model into the network. Every other node in the network uses this information to track the location of this node. The model does not consider any sudden change in the mobility of the node, which may end with a high rate of prediction error.

The 'Sectorized ad hoc mobility prediction scheme' [56] [16] uses network clustering [44] [57] to predict the mobility of the nodes. The network is divided into clusters and each node belongs to one cluster. To achieve a higher prediction the cluster is divided into three regions with the probability of cluster change is defined by three cases:

No cluster change region, Low cluster change region and High cluster change region.

In 'No cluster change region', nodes do not satisfy the requirements of membership of any other neighbouring cluster, therefore, the change is not possible.

In 'Low cluster change region', nodes are reachable by the No cluster change nodes and the probability of change is low. In 'High cluster change region', the nodes are reachable only by Low cluster region nodes and the probability of change is high. The cluster-sector numbering scheme is used to predict the next cluster change based on the position of the node and its direction of movement in the High cluster change region[16].

These schemes predict the next cell that the mobile node might visit, not the exact location for the mobile node that is required for geographical routing, which is the most suitable routing technique in high mobility environment. Predicting the next cell is also not enough to build the future topology of the network in link state routing protocols. Moreover, the inaccurate predictions in these schemes could affect the quality of service (QoS), because a certain percentage of the bandwidth will be reserved in the incorrectly predicted cell that the mobile node will not visit [58].

The authors in [13] proposed an algorithm to predict the expiration time of the link between two adjacent nodes, and the routes are then reconfigured before the expiration time of the link is reached in order to prevent disconnection. The algorithm calculates the expiration time for the link using the speed, direction and coordinates for both nodes. This algorithm obtains the information about the location and the speed by piggybacking them during live connections. If the nodes do not communicate very often, the information is not accurate and may lead to inaccurate prediction. Similar to the dead reckoning technique in [46] this prediction algorithm does not consider any sudden change in the mobility of the node.

Location Prediction Based Routing (LPBR) [59] has been proposed to reduce the number of route discovery in the network. It uses a flooding-based route request to find a route to the destination. The route request packet gathers the location information for each node traversed and this information is stored in the destination. The route reply contains a path to the destination which leads to the source node. If the destination node does not receive a data packet, it constructs a global topology of the network using the recent location information gathered and sends a route to the source to use in delivering data.

LPBR does not broadcast any location update; however, locations are obtained during the route search which may result in stale information, especially in high mobility networks. The algorithm also does not respond to any sudden change in the mobility of the node.

An autoregressive model using Kalman filter was used by several authors to estimate the mobility of wireless networks [44] [60] [61]. The Yule-Walker equations [62] and Kalman filter [63] are used to estimate the mobility of a wireless network in [60]. A given mobile state at time n can be used to estimate the mobile state at time n+1. States n, n-1, n-2,… are used to calculate the state variables. These estimation schemes are limited to position tracking in cellular networks because the use of the received signal strength indicators (RSSI) or the time of arrival (TOA) for mobility estimation must be analysed based on the fixed source of the signal, such as the base station. This estimation scheme therefore works poorly in mobile ad hoc networks.

In [64] the authors proposed a prediction model called the hierarchal location-prediction model (HLP) to predict the future trajectory of the mobile node in ATM networks cells. HLP reserves resources in advance for the mobile node in the expected cell based on the prediction; part of the resource in the cell could be reserved for a node which may not visit this cell.

Cross correlation and pattern matching are used to predict the future state of the link in wireless networks [65]. The node monitors and stores a series of signal to noise ratio (SNR) measurements for its neighbours. These measurements are cross-correlated with certain query values to predict the future value of the link's SNR.

2.4 Topology control

Topology control is a technique used in ad hoc and wireless sensor networks to reduce energy consumption and to control the transmitting range of each node in the network [66]. Reducing energy consumption in any node will prolong its life time, therefore prolonging the life time of the network. Moreover, controlling the transmitting range enables more simultaneous communications between nodes.

The main goal of *topology control* is to change the transmitting range of nodes dynamically while maintaining some of the network properties (e.g., connectivity). Moreover, it reduces the energy consumed by the node's transceiver which is the primary factor of energy consumption in the wireless unit. Ideally, the transmitting range the nodes should be set to the minimum value such that the graph that represents the communication links between units is still connected [67].

Topology control approach can be classified into homogenous and non-homogeneous. In the homogenous approach, all nodes have the same transmitting range. This value is the minimum value of the transmitting range that satisfies the connectivity condition of the network and is called the critical transmitting range. In the non-homogeneous approach, different values of transmitting range are used by different nodes. A more complex approach is used to determine the transmitting range; for example, nodes can exchange information about their location to determine the transmitting range.

Topology control concerns finding a minimum value of the transmitting range called the Critical Transmitting Range (CTR) such that the graph which represents the network topology is connected. By reducing the transmitting range, more nodes can transmit simultaneously with less interference which improves the capacity of the networks [67] [66].

The surrounding environment of the networks is another good reason for existence of topology control. In the case of most WMN networks, nodes are designed to work in harsh environments where the access of the node and the replacement of battery are economically inconvenient. Nodes also work in poor communication environments with time varying topology due to node failure or node mobility. Moreover, the number of nodes can grow to thousands in some networks.

2.4.1 Energy conservation through topology control

In ad hoc networks and wireless sensor networks, the computation and communication capabilities of the network are bounded by battery power nodes [68]. The greatest amount of energy is consumed through communication between nodes, so the energy consumption becomes a critical issue, especially in wireless sensor networks. The computational capability of nodes in wireless sensor networks is limited; therefore the collected data should be transmitted to a remote computational unit for processing. Communication between nodes which consumes most of the power of the node is very frequent. Considering the fact that the WSN node is a simple micro sensor with a small size battery, the battery should be utilized efficiently to maximize its life time. In most cases, accessing nodes to replace batteries or to replace failed nodes is economically inconvenient.

Topology control reduces the transmitting power which reduces the coverage of the transmitted signal. This reduction can affect the connectivity condition for the network, so there is a minimum value of transmitting power needed to satisfy the connectivity of the network.

Figure 2.2 shows three nodes *a*, *b* and *c* that have the same receiver sensitivity p_r. To transmit a packet from node *a* to node *c* in a direct link, the transmitted power will be p_{t1}. However, to transmit the same packet through two stages by sending it to node *b* then to node *c*, the transmitted power of nodes *a* and *b* are p_{t2}, p_{t3} respectively. Since all nodes have the same receiver sensitivity, and the received power p_r is inversely proportional to the distance between the transmitter and the receiver [69]:

$$p_r = c.\frac{p_t}{d^n} \qquad (2.1)$$

where *c* is a constant and *n* is path loss exponent, $n > 2$

Because $d_3 \le d_1 + d_2$

Then:

$$(d_3)^n > (d_1)^n + (d_2)^n \qquad (2.2)$$

Therefore:

$$p_{t1} > (p_{t2} + p_{t3}) \qquad (2.3)$$

Topology control reduces the transmitting range of node *a* which changes the topology of the network and eliminates the link between node *a* and node *c*. The packet is therefore transmitted through a multi-hop path to conserve the power in the network.

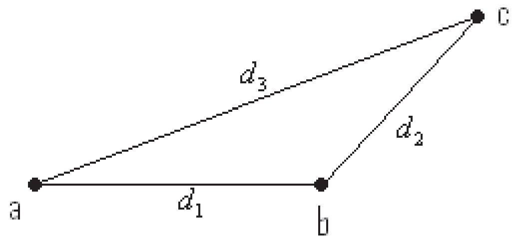

Figure 2.2: a single link vs. multi-links transmission: transmission in one stage through link d_3 consumes more power than transmission in two separated stages through link d_1 then d_2

2.4.2 Bandwidth conservation through topology control

Links between nodes in wireless networks are determined by number of factors, such as the distances between nodes, the transmission power assigned to transceivers, signal interference, and the direction of antenna [70]. A link between two nodes exists if the received signal strength is above the level of the receiver sensitivity.

Radio interference reduces the available bandwidth of the network, especially for dense networks, which is the case for ad hoc networks. Topology control can increase the availability of bandwidth by varying the transmitting range of the node. When the transmitting range is reduce to minimum value while the network still connected, the interference between nodes will be reduced so that more simultaneous communications become possible.

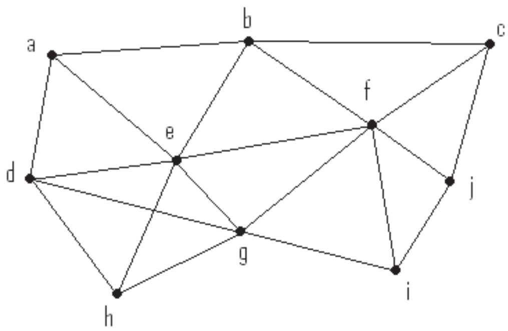

Figure 2.3a: longer transmitting range

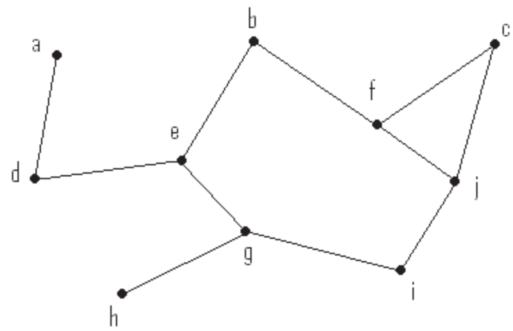

Figure 2.3b: shorter transmitting range

Figures 2.3a and 2.3b show the same part of a dense network with two different values of transmitting power (transmitting range). In Figure 2.3a, the pairs d-e and h-g can not communicate simultaneously because of radio interference. Topology control is used to reduce the transmitting range while the network is still connected; therefore, simultaneous communications between pair d-e and pair h-g are possible.

2.4.3 Mobility prediction and Topology control

Mobility prediction can achieve better topology control. By predicting the future mobility of nodes, the level of transmitted power can be determined based on the distances between nodes. A node predicts its own future location by knowing the current location and speed. Neighbouring nodes can exchange their future predicted locations to build a predicted topology using a suitable propagation model that suits the environment of the network. The minimum transmitting range is then calculated based on the predicted topology [71].

2.5 Location service

All protocols and algorithms that use location information in routing need to know the location of some or all of the nodes in the network. A node may obtains its own location information through Global Positioning System (GPS) capability, which at the time of writing this thesis is the only fully functional Global Navigation Satellite System (GNSS) [72]. The system utilizes a constellation of at least 24 Medium Earth Orbit satellites that transmit precise microwave signals, enabling a GPS receiver to determine its location, speed, direction, and time using a mathematical process called Trilateration [73] [74]. A node can also obtain its own location through different localisation methods such as the localization methods presented in [75] [40] [41].

The source node needs to know the location of the destination: this information could be stored in the node or somewhere in the network in a service called 'location service'. Each node must inform other nodes, or the location service in the network, about its location, to make geographical routing possible.

The dissemination of location information in the network can be done by two different methods, the flooding-based method or the rendezvous-based method [76]. In the flooding-based method, the node propagates the location information periodically to the network such as DREAM [42], GRSS [77] and SLURP [43] which disseminates the update when the node moves outside a pre-defined region [78]. In the rendezvous method, such as in [79] [80] [81], all nodes in the network agree upon a mapping that maps each node in the network to one or more nodes. The mapped-to nodes are the location server in which the location update is stored and retrieved.

2.6 Conclusion

This chapter has reviewed recent topology discovery and prediction techniques used in many mobile ad hoc protocols. It shows how the discovery of the topology in a high mobility environment causes delays in route construction and floods the network with route updates and route request packets. The review shows that the existing mobility prediction schemes are either not suitable for high mobility environment or have drawbacks that reduce their efficiency in such environments.

In the following chapter, three new topology prediction schemes will be proposed for mobile ad hoc networks to improve these drawbacks.

Chapter 3

3 Prediction of network topology using motion of nodes

A mobile ad hoc network is an autonomous system of nodes that move in an arbitrary manner and communicate via wireless links with the absence of any infrastructure. Each node acts as a router for forwarding and receiving packets to and from other nodes. Two nodes communicate directly if they are within the transmitting ranges of one another, or through multi-hop if there is no direct link between them.

The first direct impact of the mobility of nodes on the network is the frequent changes of the topology; it creates a dynamic network topology [82] that makes routing and the delivery of data in mobile ad hoc networks a non trivial task, especially in real time applications which require a strict timing constraints.

To ensure the delivery of data and to keep the network functional, routing tables must be maintained and reconstructed after any change in the status of the network. This maintenance can be done in a frequent manner or on demand. Accurate and up-to-date information about the changes in the topology of the network must be available to achieve this aim.

Route updates and reconstructions should be done with a minimum of delay, a minimum of traffic overhead and a minimum bandwidth consumption [10]. If nodes can predict the changes in the topology, data can be delivered more efficiently with minimum overheads and minimum bandwidth consumption. Recent work has shown that topology prediction achieves better performance in terms of maximizing data delivery and minimizing transmission delay [15].

The focus in this chapter is to give an overview of the prediction of the future topology of the network, to familiarise the reader with the effects of change of the topology on the network, to highlight the importance of finding a solution for this, and to show how nodes can use this prediction method to predict the future topology of the network.

Topology prediction can be applied to topology-based and non-topology-based networks. In a topology-based network, the algorithm predicts the future topology of the network to provide the nodes with a future vision that enables the nodes to update their routing tables concurrently with the dynamic changes to the topology caused by the mobility of the nodes. In non-topology-based algorithms such as geographical routing algorithms, nodes predict the location of the destination rather than consulting the location service for the delivery of every packet.

3.1 Advantages of topology prediction

In any mobile ad hoc network the mobility of the nodes imposes regular status updates which can be implemented on demand or periodically. The updates could be route updates in topology-based routing algorithms or they could be location updates in geographical routing algorithms.

In the case of periodic update, routes will always be available, however, network traffic will be high and a large percentage of the available bandwidth will be consumed.

In contrast to periodic updates, on demand updates may not consume same amount of bandwidth but it is causing delay in data delivery because routes must be obtained prior to transmission through the route request/route reply phase.

Topology prediction will improve the performance of the network by reducing the volume of traffic and reducing the data delivery delay associated with the route request/route reply phase.

3.1.1 Reducing control traffic using topology prediction

In traditional proactive updates, a node sends a periodic update every T_1 second to keep all nodes informed about its up-to-date status. As the mobility of the nodes increases, the topology of the network changes more frequently which require to send the update more frequently to ensure an accurate update. Therefore the value of T1 should be decreased to fulfil the accurate update requirements, which implies more traffic into the network. If a prediction algorithm which predict the change in the mobility for a future time say T_2 second is considered, the update is then required every T_2 seconds, where $T_2>T_1$.(i.e. $T_2=nT_1$). In this case, nodes predict the topology n times during this period before they receive a real update again at the end of T_2. These predicted topologies will be used to update the status of the routing tables between two consecutive real updates. Therefore, no update is propagated into the network during this period because updates are predicted by the node itself.

3.1.2 Reducing transmission delay using topology prediction

In traditional reactive updates, where on demand updates are used to reduce control traffic in the network, routes to the destination may not be available or not fresh enough. Nodes will issue a route request packet and will wait for the route reply to obtain a route to the destination. Similarly, in a geographical routing algorithm, the location of the destination may not be available, which requires sending a request to the location service to obtain the location of the destination node. In both geographical and topology-based routing, route request/ location requests and replies are associated with time delay which will affect the delivery of data.

When topology prediction is considered, updates will always be available; therefore no delay is associated with route or location updates.

3.2 Mobility model

In general, nodes in mobile ad hoc networks are free to move in an arbitrary manner. However, most applications in mobile ad hoc networks impose a non-random behaviour on the mobility of the nodes because these nodes are carried by humans or controlled by humans. Many examples, such as vehicular networks, soldiers on the battlefield and disaster relief aid personnel are all moving in non-random manner, and therefore complete randomness is not assumed, at least in the short term.

Based on a review of many studies [46] [83] [15] [84], a mobile node maintains its mobility status for a certain period of time and exhibit some degree of regularity in its mobility patterns [13] [83]. This regularity allows the use of a metrics such as speed and position to construct a mobility model. Many mobility models have been proposed in the literature [85] based on these metrics. The Random Waypoint model has been introduced by the authors of [25] to simulate their proposed DSR algorithm. In this model, each node chooses a random point as a destination and moves toward it at a randomly chosen speed. When arriving at the destination, the node holds for some time before choosing another destination. The simplicity of the model causes it's spreading, however the lack of providing a movement pattern close to reality and the existence of border effects are the main drawback of this model. Details of the border effect are available in [86]. Random direction model [87] was developed to avoid the border effect. In this model, the node chooses a direction from [0-2π] and moves in that direction for a certain distance then chooses a new direction, and so on. This model does not simulate a real movement because of the random behaviour of its mobility.

In most practical cases, nodes travel with a destination in mind [46], this fact denies the randomness and confirms the regularity of the behaviour of nodes movement. Previous work have shown that cars exhibit high regularity in their mobility [15] by analysing statistical traffic count data obtained from the Roads and Traffic Authority (RTA) report [88] for vehicles movement Sydney region. The analysis concluded that the probability that a vehicle will continue travelling in a straight line as it reaches a street junction is higher than the probability of the vehicle turning right or left. This probability increases as the vehicle travels on main streets. The Analysis results confirmed the proposed regularity of behaviour.

Based on this regularity of behaviour and the fact that a node always has a destination in mind, I have developed a mobility model that follows the regularity of behaviour and movement toward a known destination.

In this model, each node chooses a destination point at random in the area of interest; therefore, the destination of the mobile node is known. The node chooses a random direction $\theta \in [0, 2\pi]$ and check the X and Y coordinate for a point of constant distance r from its current position in that direction. θ is obtained using a Gaussian distribution with an average value $\mu = 0$ and the standard deviation $\sigma = \frac{\pi}{4}$. If the

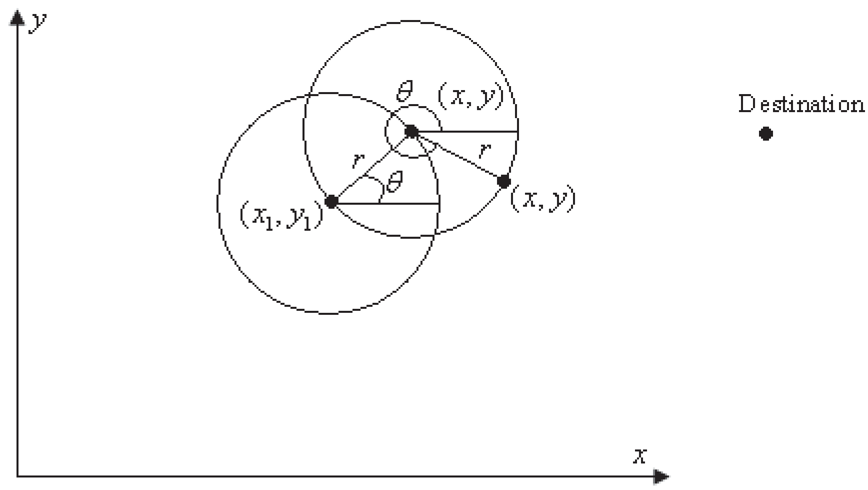

Figure 3.1: the choice of the direction in the mobility model

point is closer to the destination than the current position, the node will maintain travel in that direction at a given speed until it reaches the point. When the node reaches the point, it chooses a new valid direction and new speed. This process is continued until the node becomes close enough to the destination. The model is expressed in Figure 3.1.

To find a point (x,y) with distance *r* from the current position (x_1,y_1) the new point would be on a locus of a circle centred at (x_1,y_1) [89] according to the following equation:

$$(x-x_1)^2+(y-y_1)^2=r^2 \tag{3.1}$$

A random direction $\theta \in [0,2\pi]$ is chosen and applied to the following two equations to find the point (x,y) such that the distance between (x,y) and the destination is less than the distance between (x_1,y_1) and the destination.

$$x = x_1 + r\cos\theta$$

$$y = y_1 + r\sin\theta$$

When the direction is chosen, the node moves at a given speed obtained using a Gaussian distribution with an average value 50km/h and the standard deviation 20km/h until it reaches the point (x,y) where a new direction and speed are assigned. Figure 3.2 shows a flowchart of the proposed mobility model

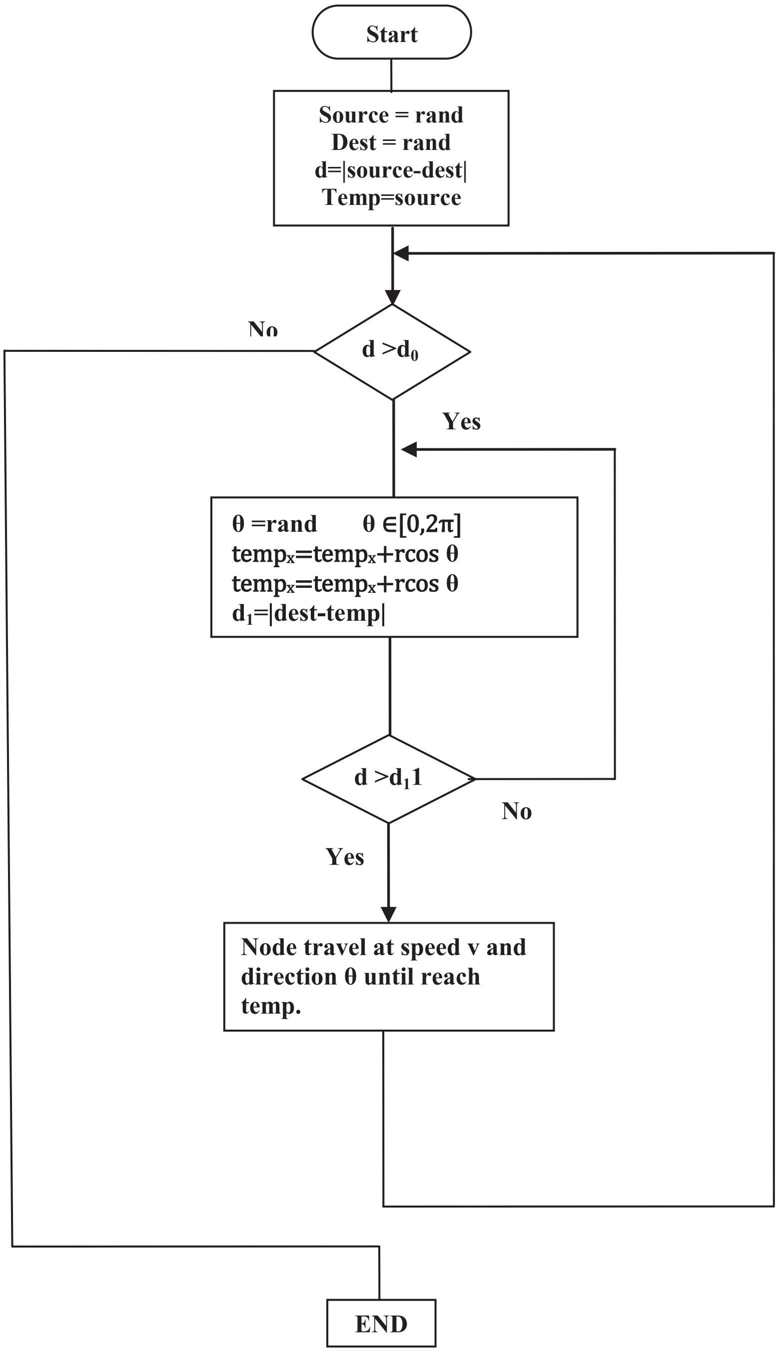

Figure 3.2: A flowchart for the proposed mobility model

3.3 Mobility status of mobile nodes

In this work, it is assumed that every single node is aware of its current location, speed and direction of movement using GPS technology. Nodes may also be aware of their location through different localization techniques. Many localization techniques have been proposed. Some of these techniques use GPS technology to localize some nodes (seeds) and use the seeds as references points; for example, the Monte Carlo localization algorithm [90]. Other techniques use localization methods without relying on GPS; for example, the localization techniques in [75] [40] do not rely on GPS technology.

The use of GPS technology for location estimation is justified by several factors. Firstly, GPS devices can provide extra information to produce a better prediction; for example, besides mobility status information, GPS-enabled devices along with information about the terrain can improve the node's prediction ability [91]. Secondly, GPS can help the node to save power because when the node knows the location of the next hop, it will transmit using directional antenna which consumes less power than transmitting in all directions [92]. Thirdly, the widespread availability of GPS devices, the ease of their installation with any node and their low cost make GPS data a good choice for this algorithm. Finally, since Wi-Fi localization routers know their GPS location, a node can, with the help of Wi-Fi technology, obtain its location indoors and/or in dense building areas where the GPS signal is weak [93]. The problem of unavailability of GPS in densely built urban areas is becoming non-existent.

3.4 Topology expression

The topology of the network $\boldsymbol{TP_k}$ at time t_k can be express in $n \times n$ matrix. l_{ij} represents a link between node i and node j. This entry has a value of 1 if the link exists and 0 if the link does not exist. The main diagonal of the topology matrix is 1 and the matrix is symmetric about the main diagonal because link l_{ij} is equal to link l_{ji} .

$$TP_k = \begin{bmatrix} l_{11} & l_{12} & & \cdots & l_{1n} \\ l_{21} & l_{22} & & \cdots & l_{2n} \\ \vdots & & \ddots & & \vdots \\ l_{n1} & l_{n2} & & \cdots & l_{nn} \end{bmatrix} \tag{3.2}$$

$$\text{Where } l_{ij} = \begin{cases} 1 & \text{, if link exist} \\ 0 & \text{, if link does not exist} \end{cases}$$

Because ***TP*** is a symmetrical matrix, all topology information can be obtained from the upper diagonal, therefore we can rewrite the topology in equation (2.5) as a vector TP_k to express the topology of the network at t_i

$$TP_k = \left[l_{12} \quad l_{13} \ldots \quad l_{1n} \quad l_{23} \quad l_{24} \quad \ldots \quad l_{2n} \quad \ldots \quad l_{(n-1)n} \right] \tag{3.3}$$

During a period of time T, the topology of the network will change due to mobility. For the period T the topology can be expressed at discrete time topologies from *t* to T+*t* by the matrix **TP**(t) as:

$$TP(t) = \left[TP_1 \quad TP_2 \quad TP_3 \quad \cdots \quad TP_m \right]^T \tag{3.4}$$

TP(t) is $m \times n(n-1)/2$ matrix that express the topology of the network for $t_1, t_2, \ldots t_m$

For the purpose of comparison between two consecutive topologies, say T_k and T_{k+1}, an XOR operation is applied to these two vectors. The comparison is carried out on all corresponding links at both topologies, for example, link l_{qs} from T_k and link l_{qs} from T_{k+1}. If the resulting value of the comparison between the two links is 1, it indicates that the link has changed. However, the 0 indicates no change to that link.

3.5 Topology prediction scheme using motion of nodes

The method presented in this chapter is rather simple. It assumes that the nodes are aware of their current mobility status including location, speed, and direction of movement. It also assumes that nodes exchange mobility information periodically to enable each node to build a global topology of the network. In principle, therefore, each node can predict the future mobility of all other nodes for a short period of time into the future. However, to fulfil the main objective of the prediction scheme, which is to achieve better performance with less delay and less traffic, the exchange of mobility information between the nodes must be less frequent than traditional proactive updates.

Consider n mobile nodes distributed randomly in a geographical area as in Figure 3.3. Each node moves in a given direction and at a certain speed, as illustrated by the arrow on each node.

Every single node is aware of its current location, speed and direction of movement through GPS technology.

The main objective of topology prediction is to provide the network with accurate status updates, reduce the delay associated with route updates and minimize the frequency of the status update broadcast by each node. To achieve all these aims, each node broadcasts a status update every T seconds. The value of T which must be long enough to achieve the proposed aims depends on the allowable prediction error and the speed of nodes. Details of the value of T will be discussed later in error analysis section. During T, each node will receive an update from all nodes in random order. During the period of time between receiving two consecutive updates from the same node, a number of predicted updates will be forecasted by the receiving nodes. These predicted values will be used to update the routing table of the receiving node during this period of time.

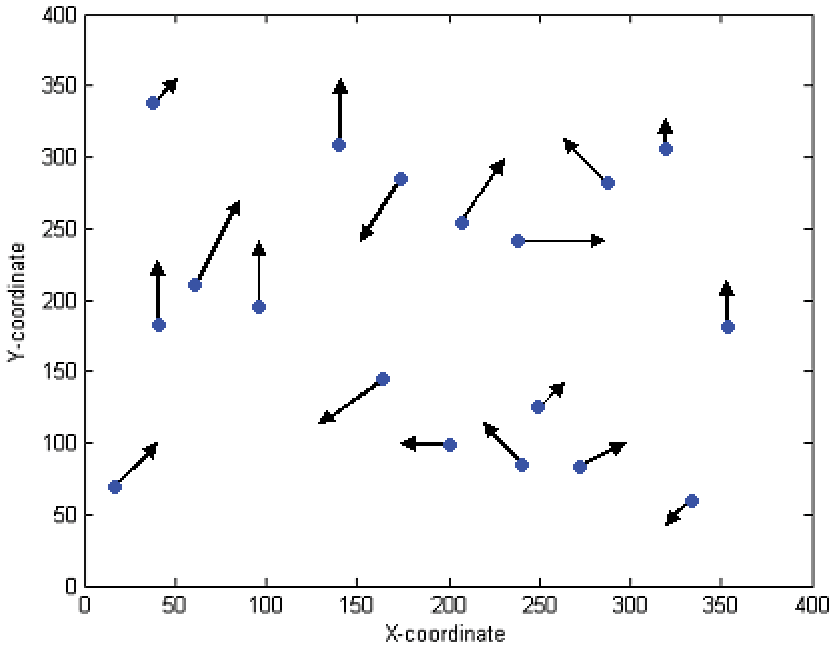

Figure 3.3: A set of mobile nodes with different mobility status distributed in geographical region

All nodes disseminate their current location, speed and direction every T seconds. Recipient nodes update their location data tables once they receive an update. These updates include actual location, actual speed and actual direction of movement of the node that sends the update. The format of the update packet is shown in Table 3.1

Table 3.1: General format of update packet broadcast every T seconds

Node ID	Time of update	Location	Speed	Direction

Every node uses the received update from a particular node to estimate a set of future consecutive locations for that node. The number of estimated locations during the prediction cycle is m=T/t, where t is the time between two estimates.

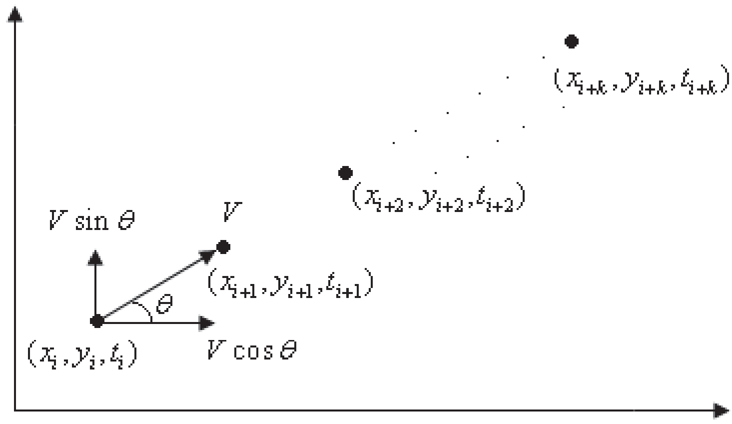

Figure 3.4: Node's movement along a path

Consider node X in Figure 3.4 as a mobile node travelling in a two-dimensional plane. Node X disseminates its current mobility status into the network through a status update packet. The packet contains: speed, direction and location of node X. Nodes in the network receive this update packet. Consider node Y as one of the nodes which receives the packet from node X. Node Y will predict m future locations for node X according to the following mobility model:

$$x_{i+1} = x_i + vt\cos(\theta) \tag{3.5}$$

$$y_{i+1} = y_i + vt\sin(\theta) \tag{3.6}$$

Where $t = t_{i+1} - t_i$ is the time for node X to travel from location (x_i, y_i) to location (x_{i+1}, y_{i+1}) at speed v m/s. θ is the direction of movement measured with positive X-axis at point (x_i, y_i).

Node Y will have received updates from all other nodes. It estimates the locations for all other nodes in the networks using the same model; thus, node Y has a set of future locations for every node. Similarly every single node will predict the same future locations for all nodes in the network.

Although a single estimate will result in m future points for every node in the network and gives the future locations for T seconds ahead, nevertheless, the node must estimate the mobility of the network every t seconds, not every T seconds. The reason for this is to enable the node to keep its own location data table up-to-date in case of receiving any update from the network during T, because updates from different nodes will be received at different times and not all nodes broadcast their updates at the same time.

At any instant of time, every node will obtain m predicted locations separated by t seconds from one another. These locations are used to build *m* connectivity matrices and *m* topologies for the network in the future. Details of estimated connectivity matrices and topologies are presented in the next sections.

3.6 Definitions of Tables

Before proceeding to the details of predicting future connectivity at every time and the future topologies of the network, several tables must be defined to simplify the concepts in the algorithm. These tables are the location data table, distance matrix, connectivity matrix, updates buffer and routing.

3.6.1 Location data table (LDT)

The location data table is a table that contains actual and predicted mobility information about nodes in the network. Every entry in the LDT contains a node's ID, the time of receiving the latest update, X and Y coordinates, the speed and direction of movement, and the predicted future locations for that node.

The algorithm uses the actual data to predict a series of future locations for the node. The predicted points are stored in the table with a time index for each point. The predicted points for all nodes are used to construct the future connectivity matrix for the network at each time index.

$P_1(X,Y,t)$, $P_2(X,Y,t)$,…, $P_m(X,Y,t)$ are the estimated future locations for a certain node for time t_1, t_2,…,t_m

Each node updates its LDT table every t seconds. By 'updating', we mean rerunning the algorithm to ensure that any received update in the buffer is considered.

Table 3.2: The format of LDT

Node ID	Time	Received update (X,Y,S,θ)	$P_1(X,Y,t)$	$P_2(X,Y,t)$	…	$P_m(X,Y,t)$
1						
2						
⋮						
n						

The time of receiving the latest update in the LDT table is used to determine whether or not the node is still connected to the network. If the time of a particular node has not changed for a given time, this node will be deleted from the LDT. The format of LDT is shown in Table 3.2.

3.6.2 Distance Matrix

The distance matrix is a matrix that contains the distances between all nodes in the networks at a given time. The entries of this matrix are obtained using equation (3.7)

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{3.7}$$

where, d_{ij} is the distance between nodes i and j. The format of the distance matrix is expressed by:

$$D = \begin{bmatrix} d_{11} & d_{12} & & \cdots & d_{1n} \\ d_{21} & d_{22} & & \cdots & d_{2n} \\ \vdots & & \ddots & & \vdots \\ d_{n1} & d_{n2} & & \cdots & d_{nn} \end{bmatrix} \tag{3.8}$$

The values of the X and Y coordinates used in equation 3.7 are obtained from the predicted location stored in the LDT using equations 3.5 and 3.6.

3.6.3 Connectivity matrix

The connectivity matrix is $n \times n$ matrix constructed and updated using the LDT every t seconds and the propagation model used by the algorithm. The entries of the connectivity matrix might be one or zero. An entry of one indicates a direct link between the two nodes corresponding to that link. An entry of zero means no direct link between the two nodes.

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix} \tag{3.9}$$

$$\text{Where } c_{ij} = \begin{cases} 1 & \text{, if link exist} \\ 0 & \text{, if link does not exist} \end{cases}$$

3.6.4 Updates buffer

The updates buffer is a FIFO stack that stores the received updates. Each node in the network broadcasts an update packet every T seconds. Nodes broadcast these updates at different times. The update packets might be received while a current cycle of the prediction is running. The algorithm will check the buffer after the prediction cycle is complete; if the buffer is not empty, the algorithm will update the LDT and reproduce the future estimates. The importance of the buffer will be shown clearly in the next section.

3.6.5 Routing table

The routing table contains routes to all possible destinations. In our work, we use the shortest path algorithm to construct the route table.

3.7 Constructing and updating the connectivity matrix

After the completion of mobility prediction phase which produces a set of future points for each node, the algorithm calculates the distance matrix for each value of time. The distance matrix contains the distance between any two nodes in the network at any given time.

Free space propagation model [69] can be used to construct the connectivity matrix for the network. The model estimates the received signal power at distance *d* from the transmitter using the following formula:

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \tag{3.10}$$

where P_r is the received power at a distance *d* from the transmitter, P_t is the transmitted power, G_t is the transmitter gain, G_r is the receiver gain, *L* is the system loss factor and λ is the wavelength. In practice, free space propagation model is not suitable for an environment with many obstacles. This model works better in free space where a line-of-sight is available between the transmitter and the receiver.

In this study, a log-distance path loss model is used to estimate the received signal power, because it is more practical in an environment with obstacles with no line-of-sight path between the transmitter and the receiver for the signal propagation. The path loss for an ordinary transmitter-receiver pair is proportional to nth power of the distance between them [69], as expressed by the following formula :

$$\overline{PL} \propto \left(\frac{d}{d_0}\right)^n$$

$$\overline{PL}(dB) = \overline{PL}(d_0) + 10n\log\left(\frac{d}{d_0}\right) \tag{3.11}$$

where *n* is the pass loss exponent. It indicates the rate of path loss increase with the distance. d_0 is a reference close in distance to the transmitter and *d* is the distance between the transmitter and the receiver.

The value of *n* depends on the propagation environment. Table 3.3 lists typical values of path loss exponent for different environments.

The received power at distance *d* can be expressed by equation (3.8):

$$P_r(d)[dBm] = P_t[dBm] - PL(d)[dB] \tag{3.12}$$

Table 3.3: Typical path loss exponents for different environments, quoted [69]

Environment	Path loss exponent, n
Free space	2
Urban area	2.7-3.5
Shadowed area	3-5
In building line-of-sight	1.6-1.8
Obstructed in buildings	4-6
Obstructed in factories	2-3

The received signal is inversely proportional to the distance between the transmitter and the receiver. If all nodes in the network transmit at the same power level and have the same receiver sensitivity, and if the value of the path loss exponent is known, then the links depend only on the distance between nodes. If the distance separating two adjacent nodes is less than a given value, the link exists; therefore, the connectivity matrix can be constructed from the distance matrix between nodes as follows:

$$D = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & & \ddots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{nn} \end{bmatrix}$$

The distance between two nodes can be found using equation 3.9:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{3.13}$$

where d_{ij} is the distance between nodes i and j

The distance matrix can be used to generate the connectivity matrix which represents the topology of the network at a given time t_i. The connectivity matrix is expressed by:

$$C_{ti} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix} \tag{3.14}$$

$$\text{Where } c_{ij} = \begin{cases} 1 & \text{, if link exist} \\ 0 & \text{, if link does not exist} \end{cases}$$

3.8 Updating the routing table

Updating the routing table takes place after the construction of the connectivity matrices. For each constructed matrix indexed with time t_1 to t_m we will have a routing table. Dijkstra’s algorithm is used to find the shortest path from the node that performs this topology prediction algorithm to all other nodes in the network. Packets can be delivered using current and future topologies.

3.9 Dijkstra’s algorithm

Dijkstra’s algorithm is a link state routing algorithm named after its inventor E. W. Dijkstra in 1959. The algorithm computes the shortest path from the node that runs the algorithm to every other node in the network.

In our topology prediction algorithm, the topology of the entire network is known by each node, therefore Dijkstra’s routing algorithm is preferred over distance-vector routing [94] [95].

The algorithm computes the least cost path from the source node A to all other nodes. Dijkstra algorithm is an iterative algorithm that starts with nodes connected directly to the source node and assigns them the value of the link cost. All other nodes will have a cost of ∞. The table starts with the source node A and a cost to all other nodes. At the next iteration, another node is added to the table and the cost is recalculated (updated). This iteration is continued until all nodes are added to the table.

The Dijkstra algorithm can be expressed by the following notation [95]:

- c(i,j) is the link cost from node i to node j. If no direct link between i and j then c(i,j)= ∞
- D(v) is the cost of the path from the source node to node v.
- P(v) is a previous node of v along the current path with least cost from source node to v
- N is a set of nodes whose least cost path is known.

Pseudo code of Dijkstra algorithm:

```
// initialization
      N=[A]
      For all nodes v
```

If v is adjacent to A

then D(v) is c(A,v)

else D(v)= ∞

Loop

Find node w not in N such that D(w) is min.

Add w to N

Update D(v) for all v adjacent to w and not in N

D(v)=min(D(v), D(w)+c(w,v))

Continue until all nodes are added to N

3.10 How does the prediction algorithm work?

Each node broadcasts a registration packet to join the network. This packet contains the node's identification, current location, speed, time of sending the packet and direction of movement. The format of the registration packet is similar to entries in the LDT, but without any prediction points. Figure 3.5 shows a flowchart of the algorithm that represents a single node predicting *m* future locations for all *n* nodes in the network.

The recipient node checks the ID of the received packet and adds its contents to the location data table by creating a new entry if the node is not listed in the LDT, or by updating the LDT if the node already exists. The LDT of the node that has recently joined the network will build up gradually by receiving data updates from other nodes.

Each node sends a periodic update packet similar to the registration packet every T seconds. This period is chosen to be large enough to reduce the number of periodic updates or periodic beacon in traditional algorithms. The choice of T depends on the allowable error in the system.

When a node broadcasts the registration packet or the update packet, it starts a timer. This timer is regularly compared with the prediction cycle T. The node broadcasts its update after the timer reaches T and resets the timer.

The node/algorithm performs a loop of calculations to estimate a set of future locations for every node in its LDT. The estimated future points are separated by time *t* and the number of points is m=T/t. Prior to each prediction, the algorithm checks the time field for each entry to ensure the validity of the entry. If the time difference between the current time and the time field for a certain entry is greater than a given value, this indicates that no update has been received from the node that corresponds to this LDT entry for a long time. Therefore, the node is considered not to be connected to the network and is removed from the LDT table.

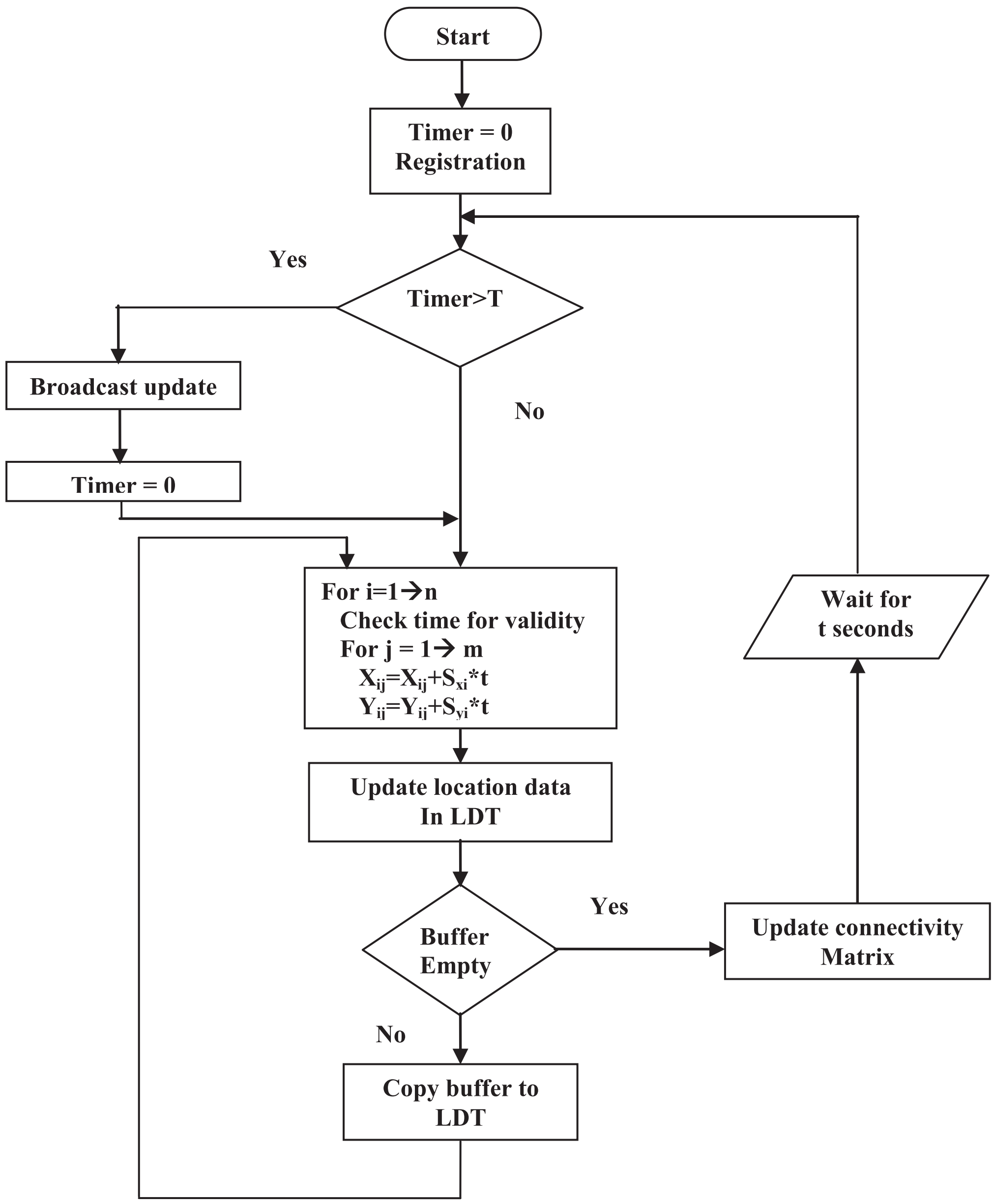

Figure 3.5: A flowchart for topology prediction algorithm using the motion of node.

The predicted future points for each node in the table are stored in the LDT. The X and Y coordinate and a time index for each X-Y pair are updated, resulting in m pairs of predicted X-Y coordinates as in Table 3.4. The actual data of the received update

Table 3.4: The format of LDT

Node ID	Time	Received update (X,Y,S,θ)	$P_1(X,Y,t)$	$P_2(X,Y,t)$	…	$P_m(X,Y,t)$
1						
2						
⋮						
n						

packet is unchanged during T which includes the original location, original speed and direction.

As mentioned above, each node broadcasts an update every T seconds containing its current location, speed, and direction. This update is stored in the updates buffer of the receiving node. The algorithm checks the update buffer after the prediction; if the buffer is empty it updates the connectivity matrix based on the estimated data obtained. If the buffer is not empty, the content of the buffer is copied to LDT and the prediction is rerun before updating the connectivity matrix. This step ensures more accurate estimates because the data in the buffer is actual and recently received.

After the first cycle of prediction, the algorithm waits for t seconds before executing the next cycle. After this time, the algorithm will produce a new set of points for the nodes that have broadcast their updates. The other nodes which have not sent updates will have the same future points shifted *t* seconds to the left in addition to an additional point for the last estimated location (i.e., $P_m(X,Y,t)$).

3.11 Error in mobility prediction

Similar to the prediction of any phenomenon, the prediction of the mobility of nodes may result in prediction error which will be a cumulative error. The algorithm is linear; it uses the predicted value as the initial value for the next prediction cycle and assumes constant speed and constant direction for the mobile node during the period T. The error is defined as the difference between the actual and estimated locations. Consider node B as a recipient node which receives a status update from node A at time t_0. The error of the i^{th} estimate in X-coordinate at the LDT of node A can be expressed by:

$$ex_i = (x'_{i+1} - x'_i) - (x_{i+1} - x_i) = v'_x t - v_{x,i} t \tag{3.15}$$

Where x'_i is the i[th] predicted X coordinate and x_i is an i[th] actual X coordinate, v'_x is the speed of node X at the beginning of the period [t_0, t_0+T], $v_{x,i}$ is the instantaneous speed at point (x_i, y_i) in X direction

The error at the first prediction cycle would be the difference between the predicted and the actual locations and can be expressed by:

$$ex_1 = (x'_2 - x'_1) - (x_2 - x_1) = v'_x t - v_{x,1} t$$

The error at the second prediction cycle which occur t seconds after the first cycle would be:

$$ex_2 = (x'_3 - x'_2) - (x_3 - x_2) = v'_x t - v_{x,2} t$$

The accumulated error at the end of the second prediction cycle will be the summation of ex_1 and ex_2.

The accumulated error at the end of the period T will be the summation of all error along this period, it can be expressed by:

$$ex_{\max} = ex_1 + ex_2 + \ldots + ex_m \tag{3.16}$$

$$ex_{\max} = \sum_{i=1}^{m} (x'_{i+1} - x'_i) - (x_{i+1} - x_i)$$

$$ex_{\max} = \sum_{i=1}^{m} v'_x t - v_{x,i} t$$

v'_x is constant for over the period T, therefore:

$$ex_{\max} = v'_x mt - \sum_{i=1}^{m} v_{x,i} t \tag{3.17}$$

$v_{x,i}$ is the instantaneous speed of the node in X-direction at point(x_i,y_i).

Equation 3.17 cab be written as:

$$ex_{\max} = v'_x mt - \bar{v}_x mt$$

$$ex_{\max} = T(v'_x - \bar{v}_x) \tag{3.18}$$

Where $\bar{v}_x$ is the average speed of node X on [t_0,t_0+T].

Similarly, the error of the i[th] estimate in Y-coordinate at the LDT of node A can be expressed by:

$$ey_i = (y'_{i+1} - y'_i) - (y_{i+1} - y_i) = v'_y t - v_{y,i} t \quad (3.19)$$

Where y'_i is the estimated Y coordinate and y_i is an actual Y coordinate, v'_y is the speed of node X at the beginning of the period [t_0, t_0+T], $v_{y,i}$ is the instantaneous speed at point (x_i, y_i) at y direction

The error at the first prediction cycle is:

$$ey_1 = (y'_2 - y'_1) - (y_2 - y_1) = v'_y t - v_{y,1} t$$

And the error at the second prediction cycle is:

$$ey_2 = (y'_3 - y'_2) - (y_3 - y_2) = v'_y t - v_{y,2} t$$

The accumulated error at the second prediction cycle will be ex_1+ex_2.

The accumulated error at the end of the period T will be the summation of all errors along the period [t_0,t_0+T]

$$ey_{\max} = ey_1 + ey_2 + \ldots + ey_k \quad (3.20)$$

$$ey_{\max} = \sum_{i=1}^{m} (y'_{i+1} - y'_i) - (y_{i+1} - y_i)$$

$$ey_{\max} = \sum_{i=1}^{m} v'_y t - v_{y,i+1} t$$

$$ey_{\max} = v'_y mt - \sum_{i=1}^{m} v_{y,i+1} t$$

$$ey_{\max} = v'_y mt - \overline{v}_y mt$$

$$ey_{\max} = T(v'_y - \overline{v}_y) \quad (3.21)$$

Where v'_y is the speed at t_0 and $\overline{v}_y$ is the average speed on over the period [t_0,t_0+T]

Equations 3.18 and 3.21 clearly show that the prediction error depends on the length of the prediction update period T. The values of v' and $\overline{v}$ in both direction are non-controllable variable. T is the only variable to control the value of the error during the prediction period.

The values of the error (allowable error) along with the average speed of the node are used as design variables to determine the length of the update period T.

3.12 Algorithm performance

The algorithm was evaluated to test its ability to predict the topology of mobile ad hoc networks. The test was implemented using MatLab on an area of 1 km by 1 km; the number of nodes in the experiment was 50 nodes. The simulation tested the algorithm for different values of the prediction cycle T for the same set of data. The data was generated using the mobility model which was presented in section 3.2, in which regularity of movement for nodes was assumed. The data includes both x and y coordinates for each point, the speed of the mobile node at that point and the direction of movements of the mobile node at that point.

To briefly describe the mobility model: a node is assigned a random start point and destination point. It chooses a certain direction and travels for a given distance in that direction. The direction should be chosen to take the node closer to the destination. The node travels at a speed chosen randomly from 15 to 20 m/s and travels at the chosen speed for a given period of time before changing speed again. The X-Y coordinates along the trajectory of each node were observed and indexed with time.

The simulation tests the ability of the algorithm to predict the future mobility of each node and compares the resultant predicted trajectories, including the time index for each predicted location, with the trajectory obtained by the mobility model.

Different measurements for different values of T were obtained and, a comparison between data was carried out.

A random node out of the 50 nodes in the simulation was chosen to compare its actual and predicted data. The comparison includes node trajectory, X-coordinates and Y-coordinates. The number of topology errors for each value of the update cycle T was also observed.

It is worth mentioning that for the simulation purpose, a node is assumed to receive the broadcasted packet from other nodes instantly. This assumption is justified by the purpose of this simulation which is to prove the ability of the node to predict the trajectory of other nodes in the network. Therefore, the channel characteristics, packet lost rate and other issues are beyond the scope of the simulation.

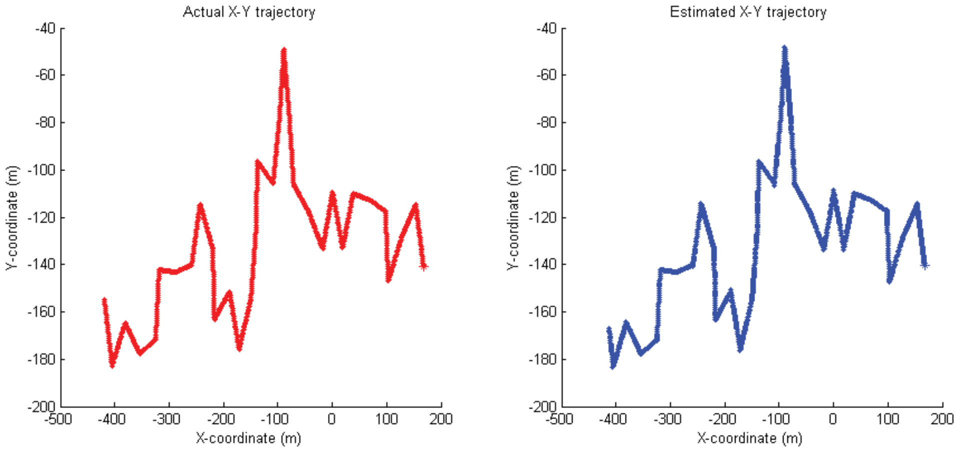

Figure 3.6: Real and estimated trajectory for node 1 with the update period T = 1s

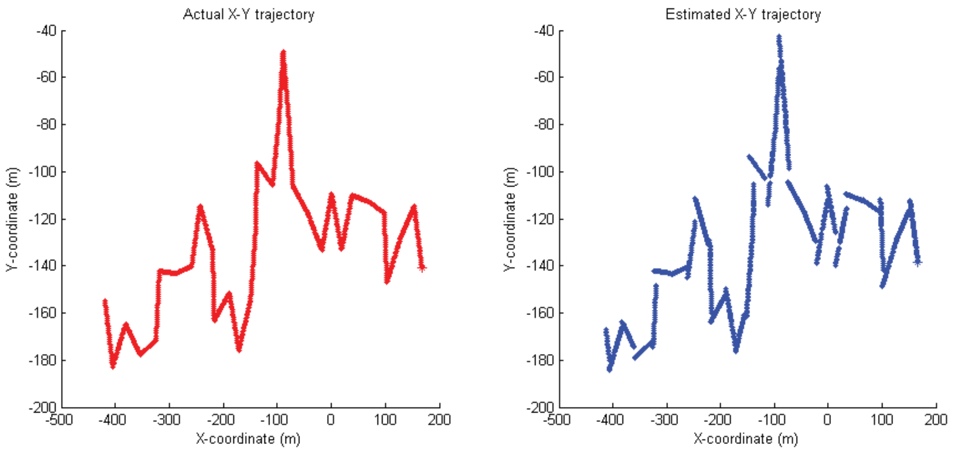

Figure 3.7: Real and estimated trajectory for node 1 with the update period T = 10s

The simulation results confirm the theoretical error analysis in the previous section. The value of the update period T has a direct impact on the predicted trajectory and location.

A comparison between Figures 3.6 and 3.7 clarifies the effect of T on the predicted trajectory for node 1. When the value of T was 1 second, the predicted trajectory of node 1 was very similar to the actual trajectory. However, when T was given a higher value, T=10, the difference between predicted and actual trajectory was clear.

In contrast, higher values for T reduced the frequency of the periodic updates. For example, the number of updates reduced by 10 times when T increased from 1 second to 10 seconds, while the effect of this increase of T on the connectivity of the network was not high. The simulation shows that the total number of topology prediction errors for the network during the whole simulation period using T=1s is 584 links, which forms a total percentage of 0.1930% of all links during the simulation period. This value increases to 6004 links with a percentage of 1.988% of the total number of links for T=10s, and 12835 (4.277%) if T=100s.

The total number of errors was obtained by applying an XOR between the real and predicted topology at each time step.

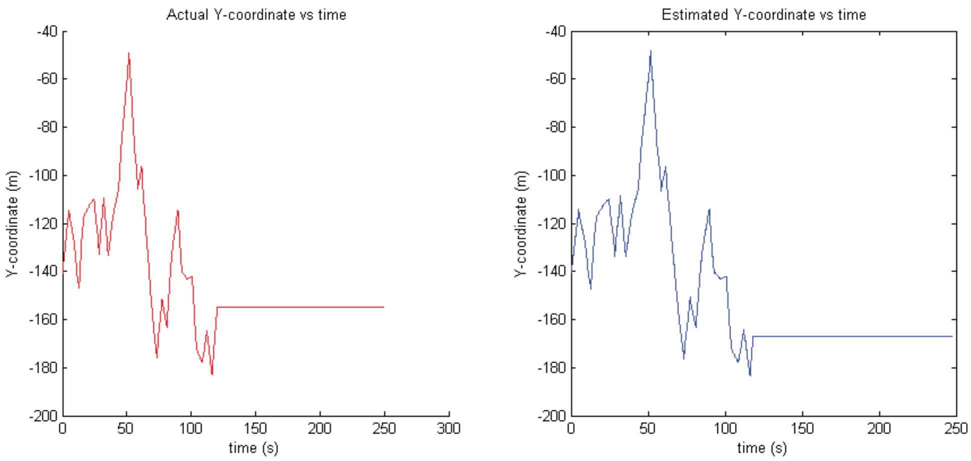

Figure 3.8: A comparison between real and estimated Y- coordinate for node 1 using the value of the update period T= 0.1s

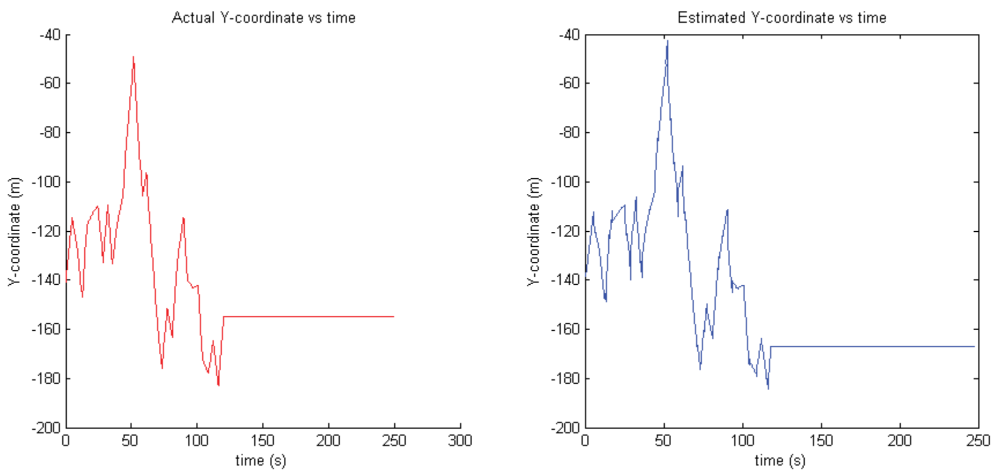

Figure 3.9: A comparison between real and estimated Y- coordinate for node 1 using the value of the update period T= 1s

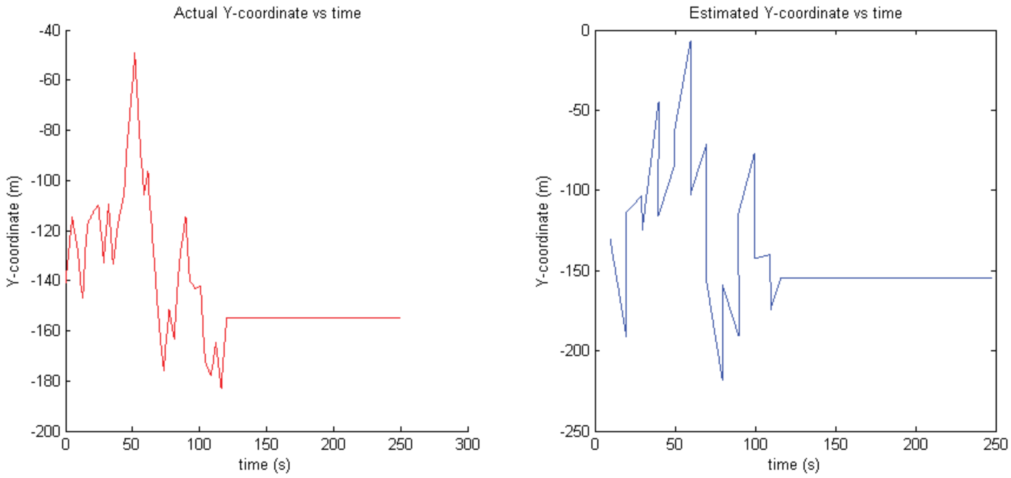

Figure 3.10: a comparison between real and estimated Y- coordinates for node 1 using the value of the update period T= 10s

Figures 3.8 - 3.10 show comparisons between a real and estimated Y-coordinate. There is no particular reason to choose the Y-coordinates over X-coordinates; it is purely for the purpose of comparison. The comparison emphasises that the error increases as the value of T increases. The accumulated error reaches its maximum value at the end of the update period (T) before being eliminated after receiving the update packet. For example, in Figure 3.10, where the value of T is 10s, the actual value of the Y-coordinate at t =100s is -170, whereas the predicted value for the Y-coordinate at the same time is -80. At t=101, the node would have received the update and eliminated the error; therefore, the value of the y-coordinate becomes -160 which is similar to the actual value at this time.

Similarly, in Figure 3.9, where the value of T is 1s, the actual value of the Y-coordinate at t =100s is -170, whereas the predicted value is -160. At t=101, the node would have received the update and eliminated the error; therefore, the value of the y-coordinate becomes -160 which is similar to the actual value at this time.

The simulation results clarify that the prediction model is affected by the sudden change of the mobility of the node. The actual and predicted y-coordinates of the node in Figure 3.10 shows that when the node changed its mobility status, the algorithm did not detect the changes until the end of T, i.e. every 10 seconds in the case of Figure 3.10. At t=15s for example, the actual y-coordinate was-148, then the node changed its direction of movement sharply causing the Y-coordinate to increase. The algorithm did not detect the change and the predicted y-coordinate reached -178 at t=20s which was the end of the update cycle; the update was received and the error was eliminated. When the value of T is smaller, the error due to sudden mobility change becomes smaller, as in Figure 3.9 where the predicted y-coordinate reaches -153 and the y-coordinate starts to increase because the update is received every one second.

To achieve a longer update cycle with accurate results, a modification to the algorithm will be discussed in the next chapter. The modified algorithm will deal with the sudden changes in the mobility of the node immediately.

3.13 Conclusion

We have addressed the problem of the mobility of nodes on the topology of the mobile ad hoc network and explained its negative effects on data delivery and route updates.

We introduced a simple and effective prediction model to predict the future topology changes. We also argued that GPS data is a good choice for this purpose.

The simulation result showed that the model produces reliable future predicted topologies for mobile networks in which nodes can use them to route data efficiently.

We have noted that although this model predicts the future topology of the network with simple computations, it induces some error that accumulates until the node receives the next update from the other nodes. The solution for this problem will be introduced in the next chapter.

Chapter 4

4 Modified prediction of network topology using motion of nodes

In the previous chapter, we defined a topology prediction algorithm for a mobile ad hoc network using motion model. We have shown how a node can predict the mobility of all other nodes through the use of the motion model of a particle in two-dimensional space. The node then constructs the future topologies of the network. We have shown that each node disseminates its current mobility status to all nodes in the network every T seconds through a status update packet. The algorithm assumes that the value of the speed and the direction of movement in the update packet received from any node remain constant until that node sends the next update packet after T seconds. The algorithm therefore uses these values to predict a series of future locations for that node.

4.1 Importance of the modification

Along the path to the destination, the mobile node changes its mobility status. The change could be an increase or decrease of the speed of the node, or it could be a change in the direction and/or speed when the node reaches a road junction. In some cases, the terrain imposes a change in the direction and/or speed. Figures 4.1 and 4.2 show some possible changes in the mobility status for a mobile node.

To clarify the purpose of this chapter, consider node A as a mobile node travelling along a given route on the map of Figure 4.1. The actual trajectory of the node is represented by the black dashed line. It is obvious that the node changes its direction of movement and possibly its speed along the route. Based on the assumption of constant mobility, each node predicts a set of future locations for node A assuming that node A is travelling at the same speed and direction during the whole period T. The resulting predicted trajectory will be the trajectory represented by the grey dotted-dashed line. Although the mobility status of node A has changed, and the real trajectory of node A is the trajectory represented by the black line, not all nodes detected that change, because the only way for all nodes to know the changes is by receiving the update packet from node A.

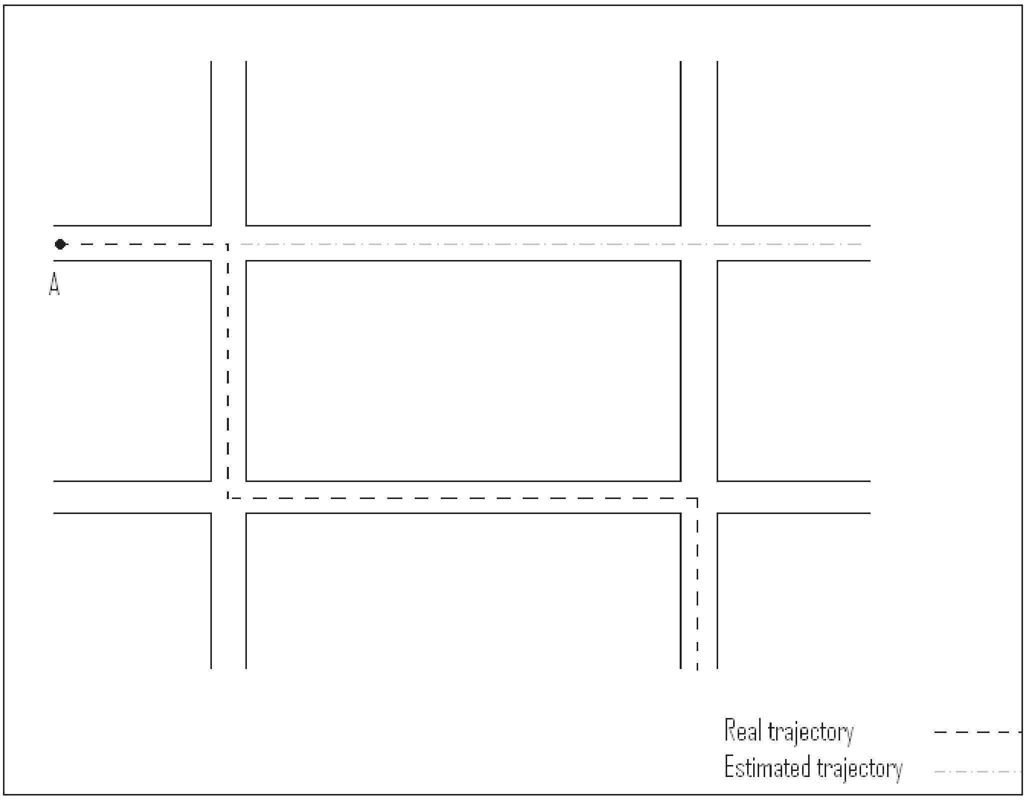

Figure 4.1: An illustration of how a node could change its mobility status when reaching a road junction. The black dashed line is the actual trajectory of the node on the street while the dashed grey line is the predicted trajectory for this node by other nodes.

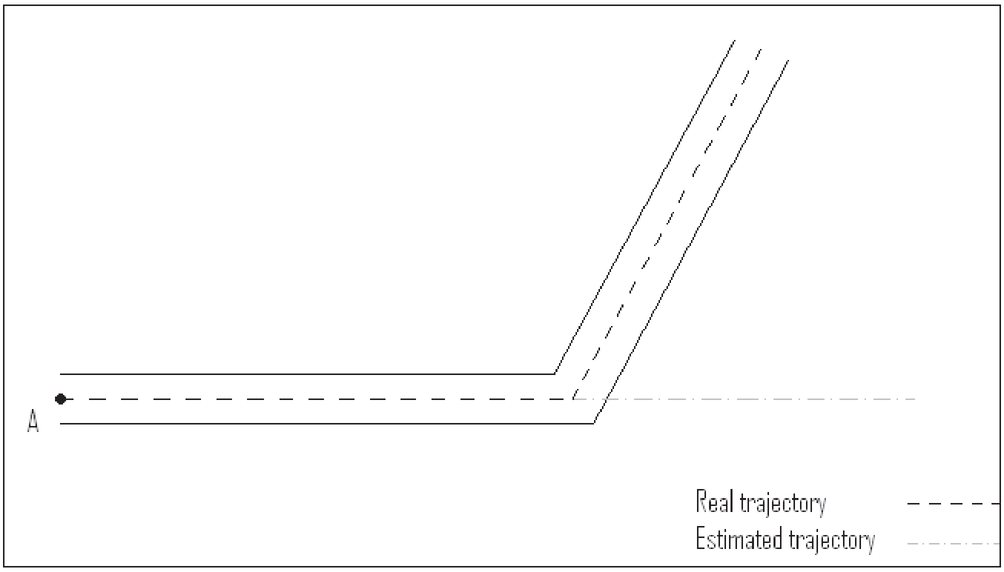

Figure 4.2: A node could change its mobility status when the terrain changes. The black dashed line is the actual trajectory of the node on the street while the dashed grey line is the predicted trajectory for this node by other nodes.

When node A broadcasts the next scheduled update, all nodes will update their LDT and eliminate the error in the predicted topology. During the time between the change of the mobility status of node A and the scheduled update, all nodes in the network would have estimated a set of locations for node A with a possibility of higher than usual prediction error. Moreover, if node A has a high mobility, then a higher possibility of error in the prediction of the topology is expected.

To resolve this problem, all nodes must have accurate updates about one another. These accurate updates can be achieved by increasing the frequency of the status updates sent by each node. In other words, by reducing the value of the update period T. the reduction of the value of T in the network will ensure that nodes will receive an update before the value of the error accumulates to higher value, even with high mobility.

In the previous chapter, we showed how varying the value of T affects the accuracy of the prediction model. We have proved that the cumulative error is proportional to the value of T; therefore, reducing T will reduce the prediction error.

Figure 4.3 shows how different values of T affect the prediction of the mobility of a node. Consider node A as a mobile node travelling along the street on the map of Figure 4.3. Let node A broadcast its update at time t_1 and let points ***a*** and ***a'*** be the real and estimated locations respectively for node A at time t_1. Let node B be a node in the network which receives the update from node A. Points ***a*** and ***a'*** would be the same at time t_1. Node B will predict a set of locations for node A along the grey dashed/dotted trajectory once it has received the update packet from node A.

Firstly, let us consider a small value of T in the model, $T = t_2 - t_1$. Point ***b*** will be the real location for node A at time t_2 while point ***b'*** will be the predicted location for point A at time t_2 as predicted by node B. The predicted points (location) between the corner (point X) and point ***b'*** will be incorrect. At t_2 the algorithm will eliminate the error because node B will receive an update from node A, therefore the maximum accumulated error will be the distance between point ***b*** and point ***b'***.

Secondly, consider a higher value of T, say $T=t_3-t_1$, where, $t_3-t_1 > t_2-t_1$ then node A will travel a greater distance before broadcasting the next update. The real location for node A at t_3 will be ***c*** while the predicted location for node A at t_3 will be point ***c'***. Therefore, the maximum accumulated error in this case will be higher. It would be the distance between point ***c*** and point ***c'***. This distance is greater in value than the distance between point ***b*** and point ***b'*** as it appears in the Figure. In other words, when the value of T is high, there will be a lack of information about the status of node A for a longer period, which leads to a higher prediction error in node B and the result is a higher probability of error in the predicted topologies.

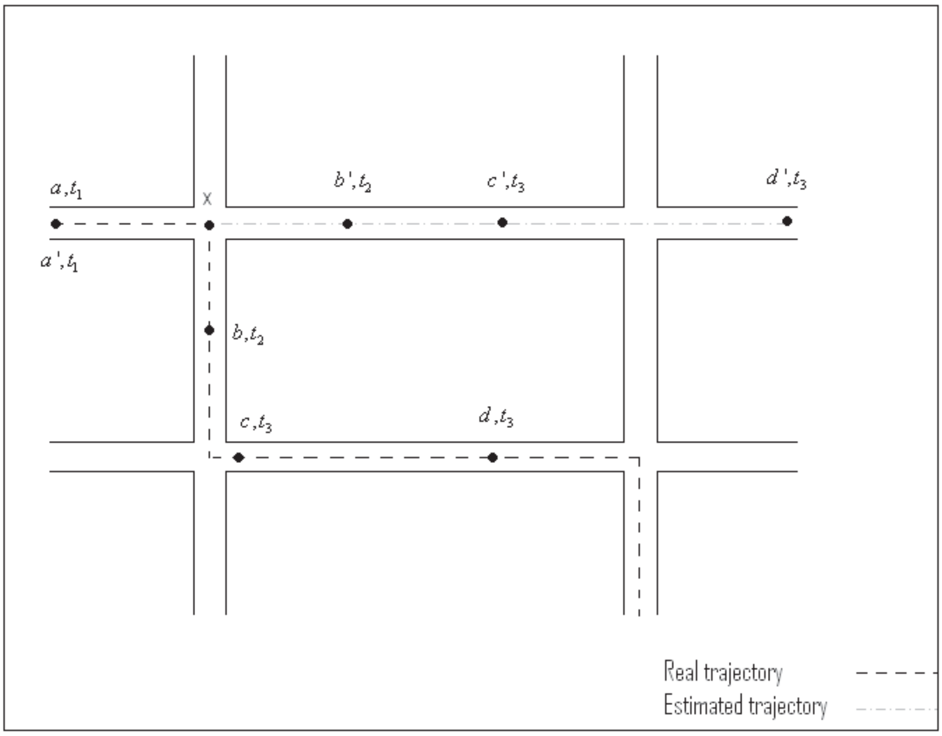

Figure 4.3: A comparison between actual and predicted trajectory as seen by observing nodes for different values of T

Reducing the value of T increases the accuracy of the prediction model and reduces the accumulated error, but it is not a practical solution because, while providing the network with extra status update, a higher percentage of the bandwidth is wasted which may cause congestion in the network, leading to higher packet loss and more delay. Our aim in this chapter is to provide accurate updates to the network simultaneously with setting T to a high value.

The solution is inspired by the concept of 'on demand' update but from different point of view. The on demand concept is used in the reactive routing protocol where the algorithm issues an on demand route request when necessary; 'necessary' here means that the data is ready to be transmitted and a valid route is unavailable. Similarly, in our modified algorithm, the node will send an immediate update when it detects a sudden change to its own mobility status, making this update necessary at this moment to prevent incorrect prediction. Therefore, the algorithm will not only send the usual update every T seconds, but will also send an immediate update when required. During the period between two consecutive updates, the mobile node could change the status of its mobility so that an update at this moment becomes inevitable to provide the other nodes with accurate information that will lead to accurate prediction.

As per the previous assumption, the mobile node is aware of its current mobility status, and it is also able to detect any change in this status. To determine whether the detected change requires an immediate update or whether the update can be done normally as scheduled at the end of T, the mobile node periodically compares changes in the speed and direction to threshold values. If the change is greater than these values, an immediate update is sent, and the following regular update will be sent T seconds after this moment. On the other hand, if the changes are less than the threshold values then the update will be sent normally as scheduled, at the end of the current update cycle.

4.2 How does the prediction algorithm work?

Similar to the algorithm in Chapter 3, the modified version of the algorithm runs on each node in the network. Each node broadcasts a registration packet to join the network. This packet contains the node's unique identification, current location, speed, direction of movement and time of broadcasting the packet. The format of the registration packet is similar to the format of the location data update table, but without any prediction points. It is represented by Table 4.1

The recipient node checks the ID of the received packet and adds its contents to the location data table by creating a new entry if the node is not listed in its LDT, or it updates the LDT if the node already exists. The LDT of the node that recently joined the network will build up gradually through receiving data updates from other nodes.

Each node sends a periodic update packet similar to the registration packet every T seconds. This period is chosen to be large enough to reduce the number of periodic updates or periodic beacon in traditional algorithms. The choice of T depends on the allowable error in the system

Table4.1: format of the registration packet

Node ID	Time of update	Location	Speed	Direction

When a node broadcasts the registration packet or the update packet, it starts a timer. This timer is regularly compared with the prediction cycle T. The node broadcasts the next update after the timer reaches T and resets the timer. The algorithm has been modified in this chapter such that the node continuously checks its own status in terms of speed and direction. If a sudden change in the speed or in the direction of movement is detected, the node does not wait for the timer to reach T to send the next update: it broadcasts an immediate update packet containing the current status information, and then the timer will be set again to zero. The detection of the changes is achieved by comparing the current status of the mobility at the t_i and a previous value of the mobility status at t_{i-q}. q can be any value depending on the required detection accuracy.

Figure 4.4 shows a flowchart of the modified algorithm, which contains two extra blocks to represent a periodic comparison between the current and a previous state of mobility to detect any sudden changes.

The first block periodically checks the change in direction and compares the change $\theta c=|\theta_i-\theta_{i-q}|$ with a threshold value θ_t where θ_i is the current direction of movement and θ_{i-q} is the direction at a previous moment. Similarly, $vc=|v_i-v_{i-q}|$ is compared to a threshold speed v_t where v_i and v_{i-q} are the current and a previous speeds respectively.

The algorithm performs a loop of calculations to predict a set of future locations for every node in its LDT. The estimated future points are separated by time t and the number of points is m=T/t. Prior to each prediction, the algorithm checks the time field for each entry to ensure the validity of the entry. If the time difference between the current time and the time field for a certain entry is greater than a given value, this indicates that no update has been received from the node that corresponds to this LDT entry for a long time. Therefore, the node is considered not to be connected to the network and is removed from the LDT table.

When a node receives an update it stores that update in the buffer. The algorithm checks the update buffer after the end of each prediction cycle; if the buffer is empty, it updates the connectivity matrix based on the predicted points obtained. If the buffer

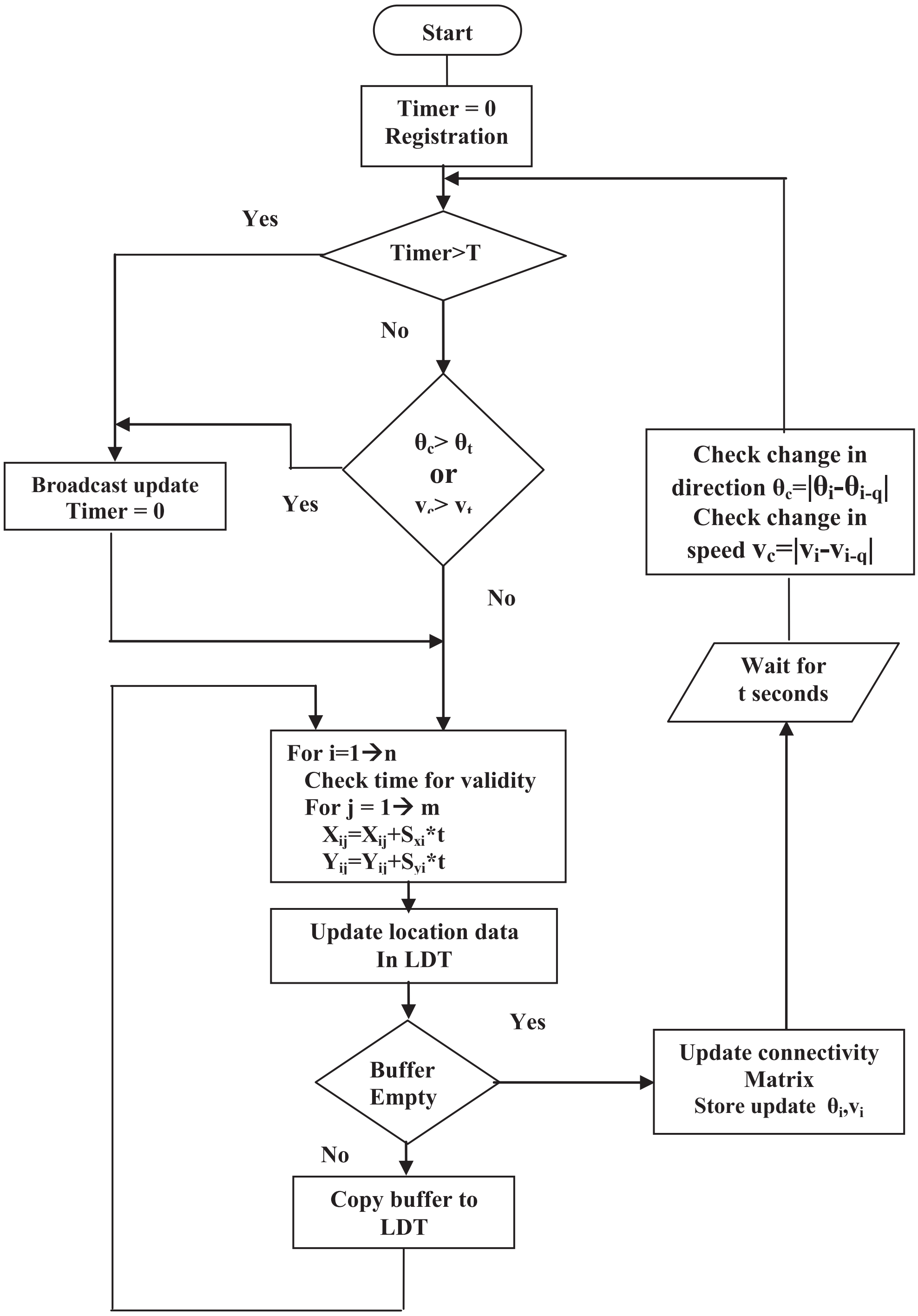

Figure 4.4: Flowchart of the modified topology prediction algorithm

is not empty, the content of the buffer is copied to the specified LDT entry or entries and the algorithm runs the prediction again before updating the connectivity matrix. This step ensures more accurate information because the data in the buffer is actual and accurate.

After the execution of a prediction cycle, the algorithm holds for *t* seconds before executing the next cycle because each cycle is separated from the previous one by *t* seconds. In the next cycle, the algorithm will predict a set of points for each node. The predicted points will be a new set of future locations for the nodes that send an update. For those nodes which have not sent an update, the predicted points will be similar to the previous cycle except that an extra point at the m_{th} location on the LDT entry with shifting all other elements to the left (i.e $E_m(X,Y,t)$) will be obtained.It is worth mentioning here that when a mobile node, for example, node X, is exposed to a sudden change, the change does not affect the prediction made by node X; rather, it affects the prediction made by other nodes about node X, which affects the whole predicted topology.

4.3 Algorithm performance

Our aim is to evaluate the ability of the proposed algorithm to predict the future topology of mobile ad hoc networks with minimum error in both X and Y coordinates. We are also interested in comparing the performance of the modified version of the algorithm with the non-modified version, which was defined in the previous chapter.

We simulated a network that consists of 50 mobile nodes deployed randomly in an area of 1000m *1000m. We compared the results of the algorithm with the algorithm in the previous chapter.

The data used in this simulation was generated using the mobility model in Chapter 3. It includes both *x* and *y* coordinates for each point, the speed of the mobile node at that point and the direction of movements of the mobile node at that point.

Figures 4.5 and 4.6 compare the predicted trajectory of a randomly chosen node with its actual trajectory. The comparison between the two Figures shows a noticeable improvement in the estimated trajectory using the modified algorithm. We can see how the modified version in Figure 4.6 eliminates the error that occurs due to a sudden change in the mobility of the node which appears in Figure 4.5, given that the same value of T still the same in both cases.

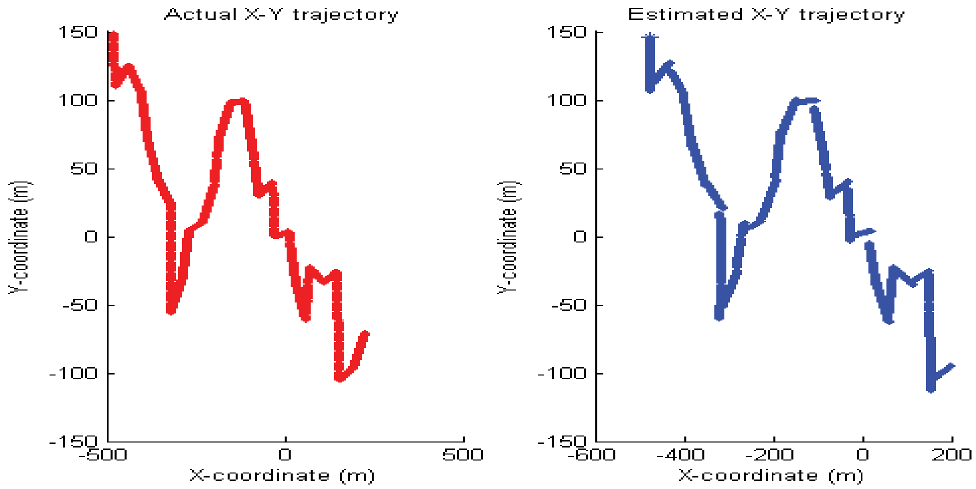

Figure 4.5: A comparison between real and predicted trajectory for a node using non-modified topology prediction algorithm, T=10s

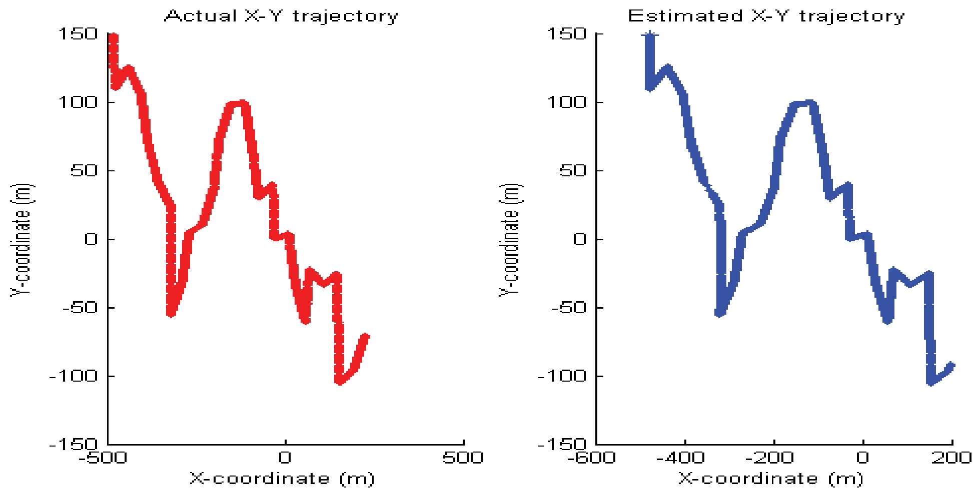

Figure 4.6: A comparison between real and predicted trajectory for the same node in Figure 4.5 using modified topology prediction algorithm, T=10s

The plots of the predicted trajectory of the node in Figure 4.6 and Figure 4.7 show how the modified algorithm reduces the effect of the update period T on the accuracy of the prediction. A quick look at Figures 4.6 and 4.7 shows the predicted trajectory for the node in both Figures are almost the same although two different values of T were used. The value of T in Figure 4.6 was 10s, while it was 100s in Figure 4.7.

The simulation results prove that the modified version of the algorithm can efficiently reduce the effect of T on the predicted locations through the broadcasting of the immediate update when necessary.

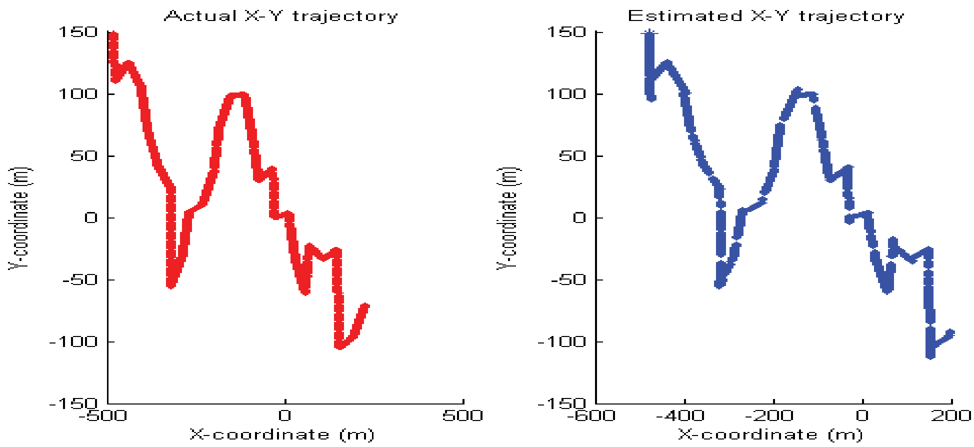

Figure 4.7: A comparison between real and estimated trajectory for node 1 using modified topology prediction algorithm T=100s

To further illustrate the improvement achieved by the modified algorithm consider Figure 4.8 which show the actual and the predicted trajectories for a mobile node using the modified and none modified version of the prediction algorithm with the value of T=50. The red curve in Figure 4.8 represents the actual trajectory of the node while the blue and the black curves represent the predicted trajectories using the modified and non-modified version of the algorithm respectively.

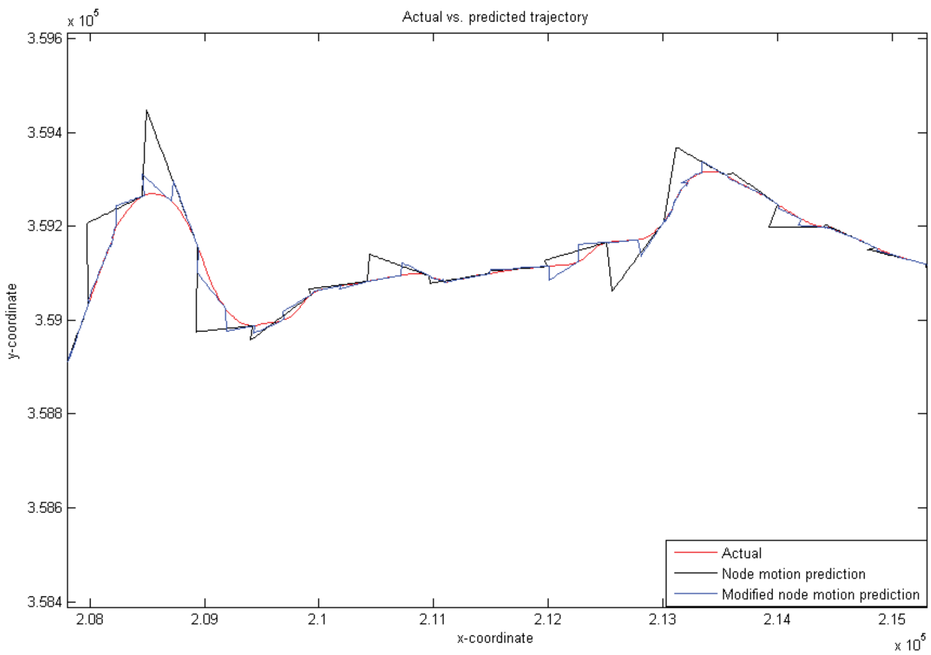

Figure 4.8: a comparison between node motion prediction and modified node motion prediction with prediction update period length T-50.

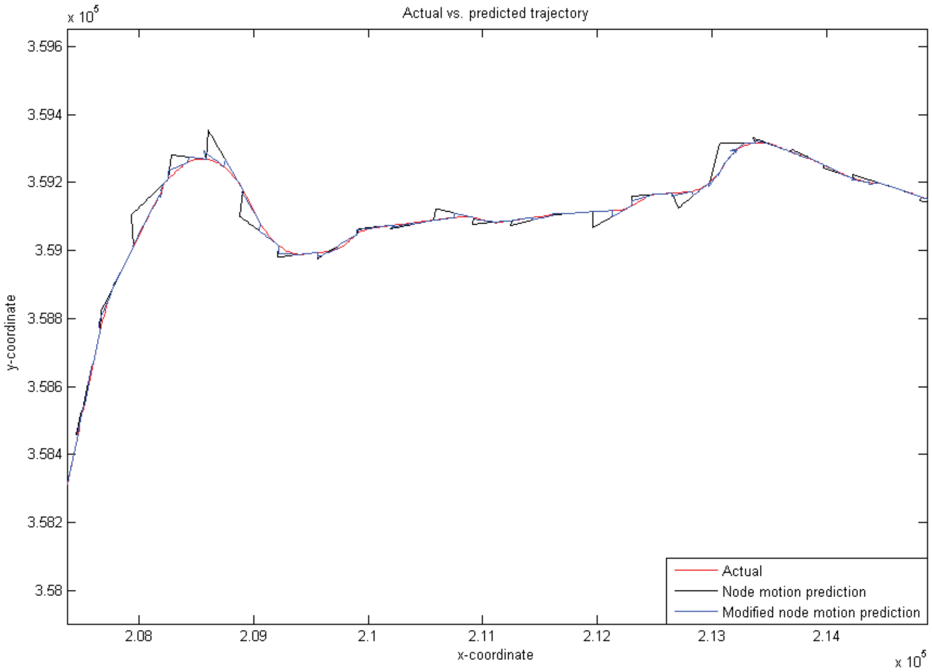

Figure 4.9: a comparison between node motion prediction and modified node motion prediction with prediction update period length T-20.

Similarly Figure 4.9 shows how the modified prediction algorithm achieves a better prediction than the non-modified version when the value of T was set to 20.

Since node motion prediction algorithm eliminates the cumulative error only after receiving the update at the end of T, all predicted points between two consecutive updates may have a higher cumulative error than the modified algorithm. Node motion prediction algorithm uses the same values of speed and direction during the whole period T, theses values are the instant values of speed and direction of the node when it broadcasting the update. Resulting in a set of predicted points forming a straight line. When the node receives the next update, then the first predicted point will be on the actual trajectory curve. So the shape of the predicted trajectory will be close to saw-tooth, especially when the trajectory of the node is often tortuous. This shape is less appearing in the modified node motion prediction algorithm because the mobile node broadcasts an update more often when it travelling in tortuous path.

The analysis of the simulation result for the predicted trajectories in Figures 4.5 and 4.6 show that the modified algorithm decrease the mean absolute error in the predicted locations to less than 2 m from 46 m in the case of non-modified algorithm with the value of T set to 100 in both cases.

To clarify the results, the absolute errors in the predicted X and Y coordinates were studied. The comparison between actual and predicted Y-coordinate which appears in Figures 4.10 and 4.11 shows how the prediction error in the modified version is reduced compared to the non-modified version, setting the value of T to 10s. The number of points with zero error in the modified prediction algorithm is more than in the non-modified algorithm; this is due to the immediate update packets in the modified algorithm. The update of mobility status eliminates the error in the predicted trajectory, whether it was a regular or immediate update.

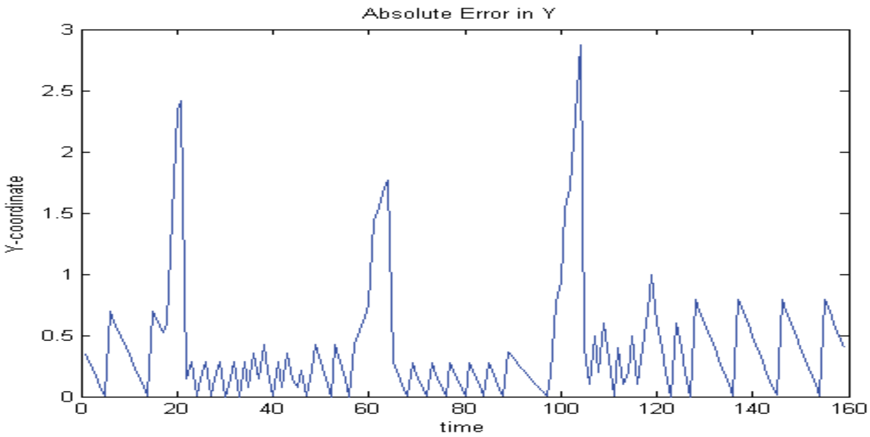

Figure 4.10: The absolute Y-coordinate error in the non-modified algorithm

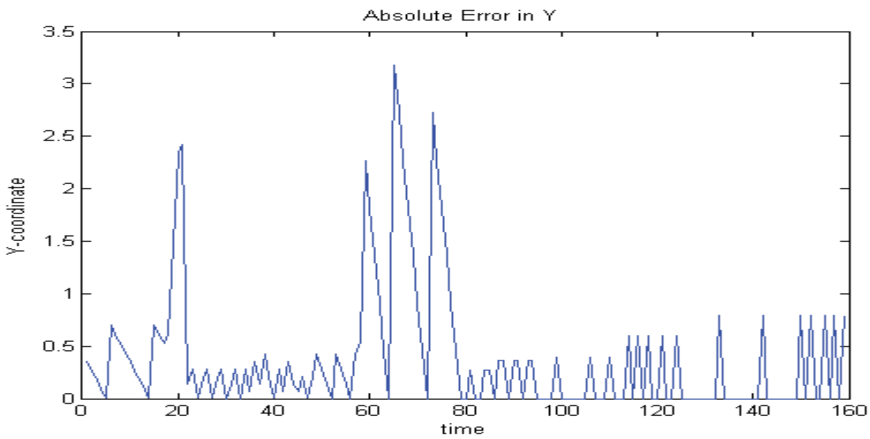

Figure 4.11: The absolute Y-coordinate error in the modified algorithm

The efficiency of the modified algorithm becomes very clear when the value of T is high. For example, in the previous two Figures the value of T was 10s, therefore the difference between the prediction error in Y coordinate was small. However, a comparison between Figure 4.12 and Figure 4.13 shows how the error in the non-modified algorithm is very high compared to the modified algorithm when the value of T is 100s. Moreover, the range of the prediction error in Figure 4.13 where the value of T is 100s is still in the same range of the error as Figure 4.11 where the value of T is 10s.

In Figure 4.12 the error accumulates to about 80m and the number of points with zero error is very few. Moreover, the error will continue accumulating for the period of T, which is 100s, and will not be removed until the update is received at the end of T. By contrast, Figure 4.13 shows that due to the immediate updates, the error in the modified algorithm does not reach high values, and did not accumulate to more than 6m using the same value of T.

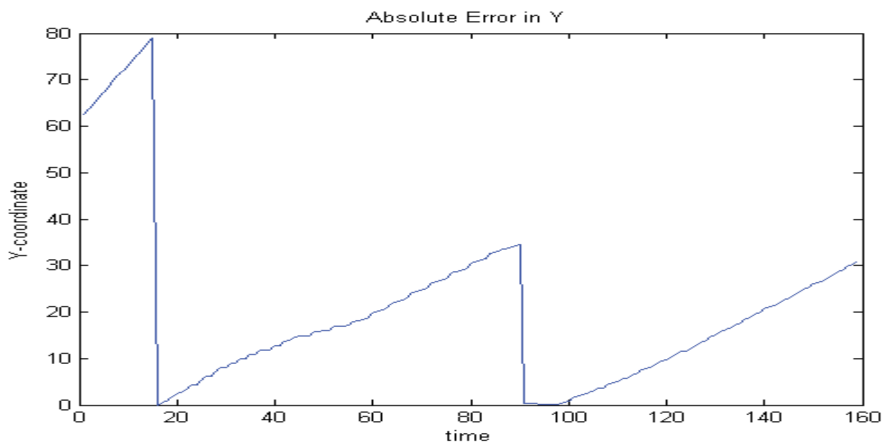

Figure 4.12: The error in Y-coordinate using the non-modified algorithm

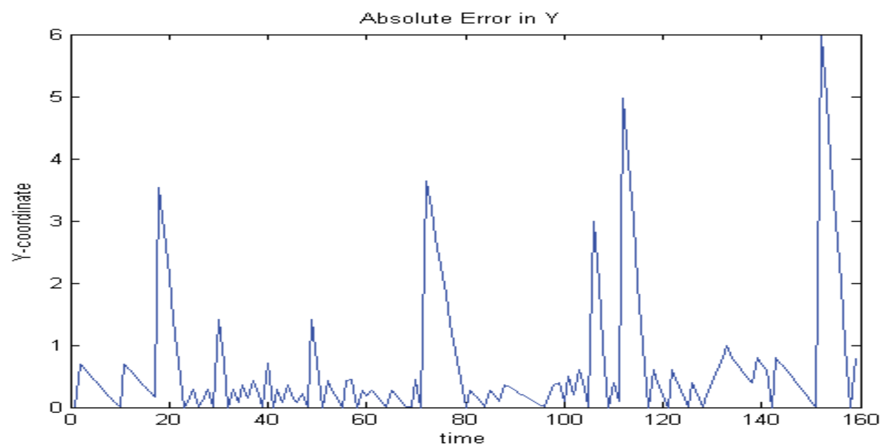

Figure 4.13: The error in Y-coordinate using the modified algorithm

The simulation measures the effect of the modification which was made on the ability of the algorithm to predict the topology. The results show that the effects of varying T are negligible on the predicted topology obtained by the modified algorithm. For example, the total number of topology prediction errors for the network during the whole simulation period using T=1s is 1051 links, which is 0.4009% of the total links during the simulation period. The number of incorrect links and their percentage in both the modified and non-modified algorithm are almost the same, because at T=1s the updates are received very often and the sudden change will be updated quickly.

As the value of T increases, the importance of the modification becomes noticeable. For example, when the value of T was 10s, the number of incorrect estimated links in the non-modified version of the algorithm was 13440, which is 6.144%. However, with the modified algorithm, the number of incorrect links is still 1120 links, which is 0.5012% of the total number of links. When T was set to 100s, the number of errors in the predicted links was 18830 (8.7183%) for the non-modified algorithm, whereas this number is still 1149 with a percentage of 0.5141% using the modified algorithm.

Finally, the value of the threshold angle was studied too. The results of the simulation while varying the value of the threshold angle confirm that it's strong influence on the accuracy of the algorithm, because it determines the immediate update. The accuracy of the prediction is inversely proportional to the threshold angle, for example, Figure 4.14 shows two predicted trajectory for the same node using the modified algorithm with different values of threshold angle. Setting the threshold angle to $\theta_t=30^0$ degree produces the blue curve and when the threshold angle was set to $\theta_t=5^0$ degree the predicted trajectory was more accurate which is the one represented by the green curve in the same Figure.

As mentioned earlier the modified algorithm comes with a cost, which is represented by the extra updates. The number of the extra updates depends on the tortuosity of the path, the value of T and the required accuracy. For the case of the path in figure 4.8 when the required accuracy was 3^0 the percentage of the extra update was 91.3% for T=50s and 80.96% for T=20 s. However, when the required accuracy is decreased the percentage of the extra update decreases. For the same experiment above when the required accuracy was 8^0 the percentage of the extra update was 84.5% for T=50 and 68.37% for T=20.

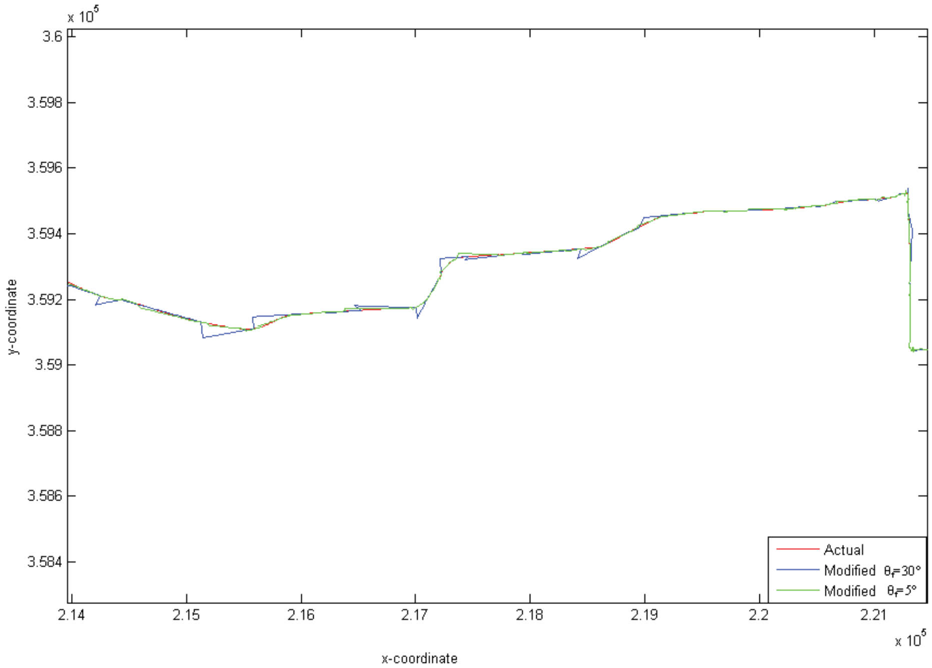

Figure 4.14: the effect of the threshold angle on the accuracy of the prediction

4.4 Conclusion

In this chapter, we proposed a modification to the algorithm presented in Chapter 3 to reduce the error in the predicted topologies caused by the sudden change in the mobility of nodes. The solution is to allow the node which detects a sudden change in its mobility to broadcast an immediate update packet. We have demonstrated that the modification enables the algorithm to eliminate the prediction error caused by sudden change and to predict the future topologies of the network with a dramatic decrease in the number of incorrect topologies. The solution also enables the use of a longer prediction update period T without affecting the accuracy of the prediction model.

Chapter 5

5 Mobility prediction using polynomial regression

Many applications in wireless sensor networks and mobile ad hoc networks require knowledge of the exact location of nodes [96]. In navigation, for example, a node must know its location to calculate a valid route to the destination using digital maps. The location of nodes can also be used in routing [38] where routing based on the location of nodes is called geographic routing. Many algorithms and protocols have been proposed for geographic routing, such as DREAM [42], SLURP [97], LAR [34], and GPSR [98].

In geographical routing, a node must determine the location of the destination to be able to deliver data to that destination. Information about the location of a node can be stored in the node itself or in a special server in the network called a location service. A mobile node must inform other nodes and/or the location service about its new location to make geographical routing possible. If a node can provide the location service/nodes with information not only about its current location but also about future locations, this will result in a reduction in the frequency of the periodic location update. This chapter presents a novel prediction method to predict the future locations of nodes to produce a set of time and location pairs for the mobile node which can be used in geographical routing and topology-based routing.

The prediction method presented in this chapter is different from the method discussed in the previous two chapters, in which a node predicts the mobility of all other nodes by employing the periodic update information received from them to find a set of future locations for each one. The mobility prediction therefore depends

directly on the other nodes. The node then uses the predicted locations to construct a set of time varying future topologies of the network.
In contrast, the mobile node in this chapter predicts its own mobility, instead of predicting the status of other nodes, by predicting a set of future locations for itself and then broadcasting them to the network. In this case, the mobility prediction depends only on the node itself and not on any other node. All such nodes that receive this set of predicted locations from another node, and other sets from different nodes, construct a time varying future topology. In other words, the nodes do not predict the mobility of each other: they predict the future topologies of the network.

When a mobile node predicts its own mobility, the accuracy of the prediction is expected to be higher than predicting the mobility of other nodes. This expectation is reasonable because the variables that the mobile node uses in the prediction are available with the node at all times, unlike the methods used previously for prediction where the variables are received once every update cycle T.

The algorithm proposed in this chapter does not require an immediate update in the case of sudden mobility change, because the mobile node predicts its own mobility and therefore realises sudden changes immediately. Moreover, the algorithm assumes that the node knows the trajectory during the period of the prediction, as we will explain in the next section. This period is similar to the update period in the previous chapters; therefore, we will call it T. The next section will explain how the assumption of knowing the trajectory makes the algorithm consider the sudden change in mobility.

5.1 Prediction of node's location

Any mobile node equipped with a positioning system can use a navigation software to calculate a possible path to the destination [39] and determine the coordinates of many points along that path. The positioning system can be a global satellite navigation system (GNSS) such as the U.S. NAVSTAR global positioning system (GPS), or it could be a local positioning system such as the Skyhook Wi-Fi positioning system [93].

The purpose of this chapter is to find a mapping between the coordinates on that path and the time, which means finding the *x* and *y* coordinates as a function of time, which in turn enables the prediction of future locations of the mobile node as a function of time. Nodes exchange their predicted future locations to use in predicting the future topology of the network.

Navigation capability enables the node to determine a set of valid points where the node is going to travel along that path. Figure 5.1 shows an illustration of a mobile node travelling along a path; the path in this case is a map. The mobile node can navigate its way along the path and determine the coordinates of many points along this path using digital mapping.

The coordinates of these points are known to the node; therefore we assume that the mobile node will pass through these nodes; however the time in which the node will pass these points is unknown.

Let point A with coordinate (x_1,y_1) be the location of the node at time t_1 and let point B with coordinates (x_n,y_n) be the point where the mobile node is going to reach at time t_n.

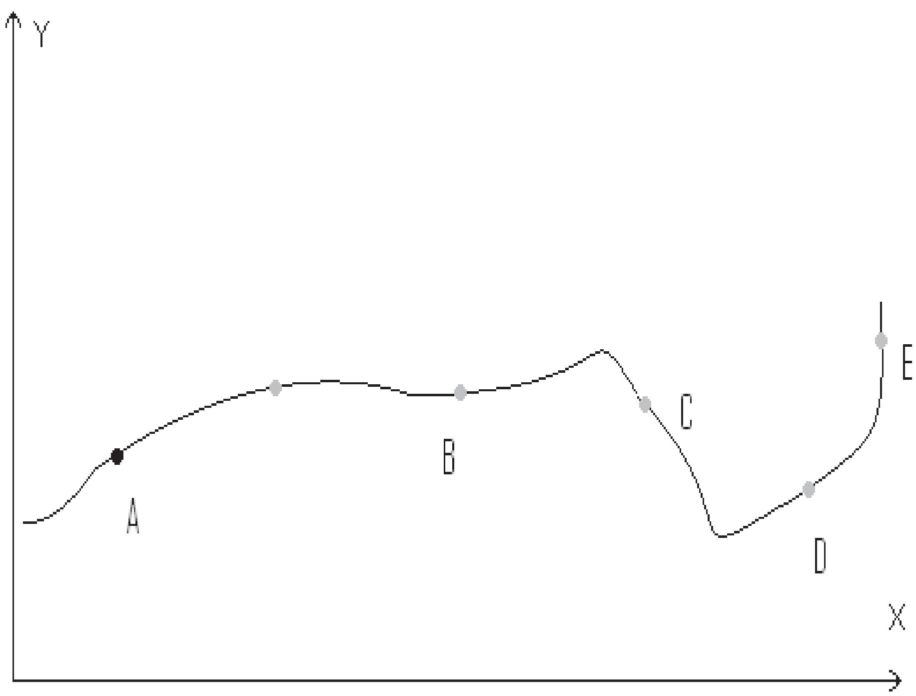

Figure 5.1: A mobile node travelling along a path

The first aim is to find the equation for the locus formed by movement of the node along the path. This equation must satisfy the following assumptions which must be held during the prediction period T:

The relation between X and Y coordinates must be a function; i.e. each X coordinate value has a unique Y value. Y=F(X) or X=F(Y). This condition must be held for the small section of the path between say point A and point B. It is not necessary for this condition to hold for the whole path.

In the previous chapters, based on the regular behaviour of the node, we assumed that the speed and direction of movement are fixed during the period between two consecutive updates. In this chapter, only the speed is assumed to be constant during the prediction period (T).

The value of the speed is taken as the instantaneous value for the speed at point (x_1,y_1) which is the speed at the beginning of the prediction period. This value has two components as expressed in equation 5.1.

$$\vec{V}_0 = \vec{V}_x + \vec{V}_y \tag{5.1}$$

The magnitude of the speed $\left|\vec{V}_0\right|$ is assumed to be constant on $\left[t_1, t_1 + T\right]$ however, the X and Y components of the velocity are variable. Therefore, the direction of the movement of the node is not constant.

For simplicity, we consider a two-dimension movement in the X-Y plane; a three-dimension case is possible, but with more calculations.

Let us look in more depth for the section of the path between point A and point B, which is represented by Figure 5.2. The X and Y coordinates for many points along the path between points A and B can be obtained and represented in *n* pairs of X and Y coordinates $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$. Our aim is to construct a function $y = f(x)$ of degree *m* that fits all points $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$.

Using polynomial regression, the general form of the polynomial that connects all points is expressed by:

$$y_i = a_0 + a_1 x_i^1 + a_2 x_i^2 + ... + a_m x_i^m + \varepsilon_i \tag{5.2}$$

Where (i=1, 2,…, n) . In matrix notation, equation 5.2 can be written as:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^m \\ 1 & x_2 & x_2^2 & \cdots & x_2^m \\ \vdots & & \ddots & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^m \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix} \tag{5.3}$$

$Y = X\vec{a} + \varepsilon$, where ε is a zero mean random error.

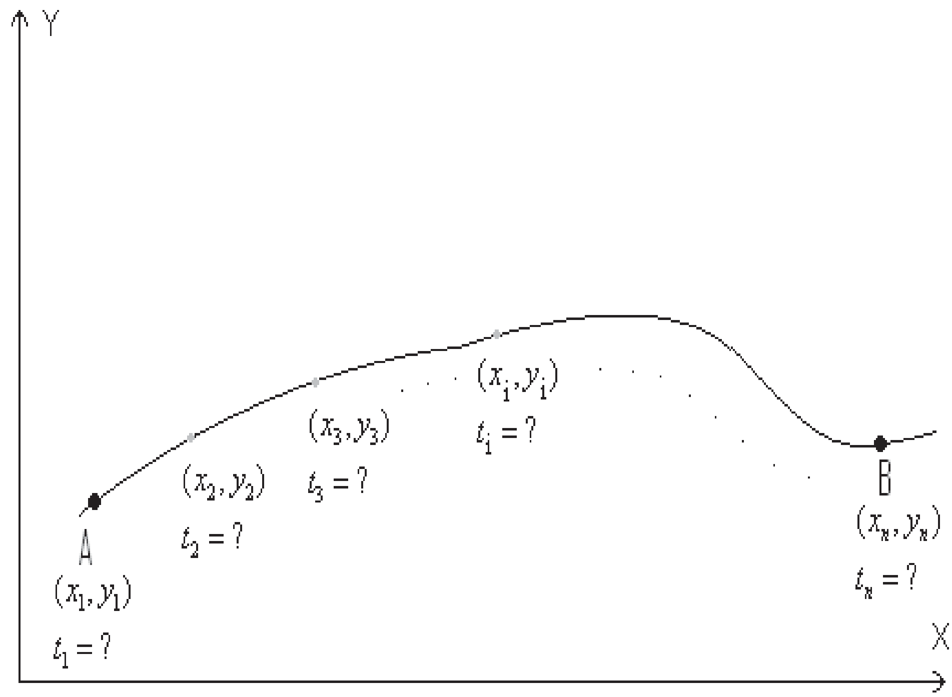

Figure 5.2: An illustration of many points between point A and point B that the mobile node knows their coordinates.

By substituting the points $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ into equation 5.3 and setting the value of the random error ε to zero, we will obtain a set of linear equations which can be solved to find the values of the constants $a_0, a_1, a_2, ...a_m$

$$\hat{a} = (X^T X)^{-1} X^T Y$$

By substituting the values of the constants $a_0, a_1, a_2, ...a_m$ into the general form of equation 5.2, we will obtain the polynomial that is the best curve to fit all points $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$. This polynomial is expressed in equation 5.4

$$y = a_0 + a_1 x^1 + a_2 x^2 + ... + a_m x^m \tag{5.4}$$

Equation 5.4 represents the change of the node's location along the path between point A and point B. To find out how this location changes with time, let us consider the second assumption we have made, which states that the speed is constant during the period of prediction T.

Figure 5.3 shows a mobile node moving at a constant speed along the path between point A and point B.

The node is moving along the path at constant speed $\left|\vec{V}_0\right|$ and variable direction θ, where θ is measured with respect to the X-axis. The velocity of a mobile node describes the change in its speed or direction of movement or both. The best way to describe the velocity of an object is to find the rate of change of its position with time. The relation between the X and Y coordinates is formed clearly in equation 5.4. The derivative of this equation gives us a view of how one variable changes with respect to the other.

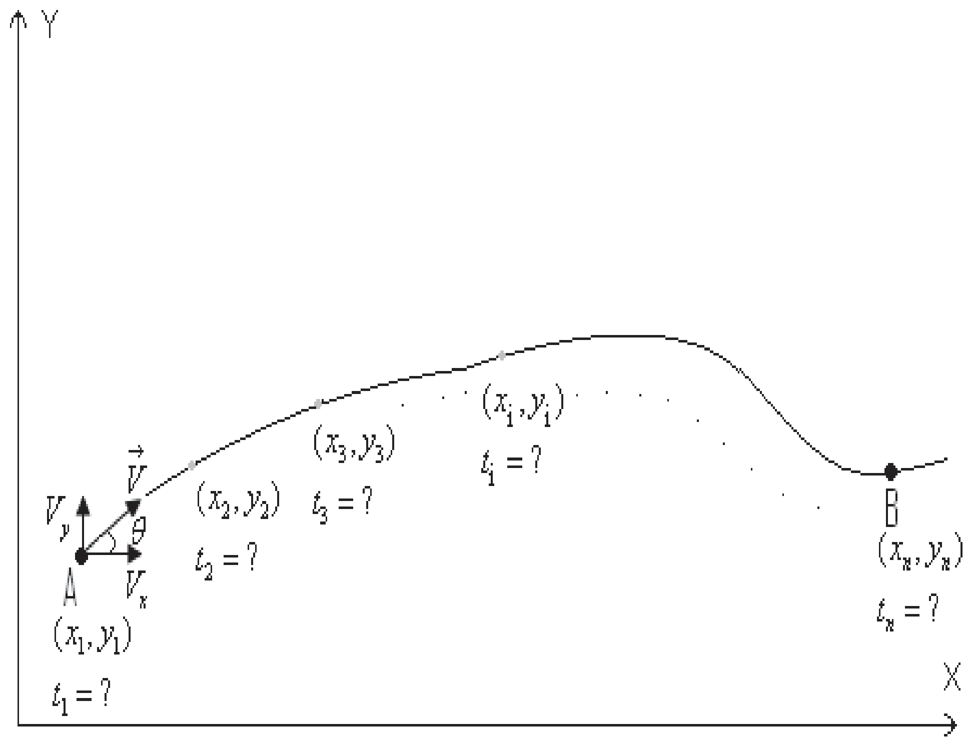

Figure 5.3: A node moving at a constant speed $\left|\vec{V}_0\right|$, and initial direction θ

To find the relation between the X coordinate of a node's location and time, let us find the derivative of equation 5.4. The derivative can be expressed as the slope of the tangent along the curve.

$$\tan\theta = \frac{dy}{dx} \tag{5.5}$$

The direction of the movement of the node can be expressed in terms of the tangent of the curve at each point as it appears in equation 5.6.

$$\Rightarrow \theta = \tan^{-1}\left(\frac{dy}{dx}\right) \tag{5.6}$$

Recall, $\vec{V_0} = \vec{V_x} + \vec{V_y}$.

Since $\left|\vec{V_0}\right|$ is constant on $[t_1, t_1 + T]$, therefore, the change is either by $\vec{V_x}$ or $\vec{V_y}$.

Since the speed at a certain direction is the change of position in that direction with respect to time; we can express the X-component of the velocity by:

$\vec{V_x} = \frac{dx}{dt}$

We can also express the X coordinate of the location by:

$$\Rightarrow x = \int_{t_0}^{t} v_x dt \tag{5.7}$$

$$x = \int_{t_1}^{t} V_0 \cos(\tan^{-1}\frac{dy}{dx})dt \tag{5.8}$$

Equation5.8 gives a relation between X and *t* during the period of prediction.

The values of Y can be obtained by substituting the values of X into the original coordination equation or by following the same steps for Y to get:

$$y = \int_{t_1}^{t} V_0 \sin(\tan^{-1}\frac{dy}{dx})dt \tag{5.9}$$

5.2 The Effect of prediction period (T) on the algorithm

To clarify the effect of T on the prediction, let us first find the relation between T and the points $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ which were used in constructing the polynomial y=F(x).

The derivative of the polynomial gives the value of the angle θ at each point along the polynomial y.

$$\theta = \tan^{-1} \frac{dy}{dx}$$

The value of θ at the point (x_i, y_i) is:

$$\theta\Big|_{x_i} = \tan^{-1} \frac{dy}{dx}\Big|_{x_i}$$

The component of the speed in x direction can be expressed equation 5.10:

$$v_x = v_0 \cos\theta = \frac{\Delta x}{\Delta t} \quad (5.10)$$

And the speed in x direction is the change of the x position with respect to time, therefore $\frac{\Delta x}{\Delta t}$ can be written as:

$$\frac{\Delta x}{\Delta t} = \frac{x_i - x_{i-1}}{t_i - t_{i-1}} \quad (5.11)$$

Substituting equation 5.11 in equation 5.10 results in the formula

$$t_i = \frac{x_i - x_{i-1}}{v_0 \cos\theta\big|_{x_i}} + t_{i-1} \quad (5.12)$$

Equation 5.12 can be used recursively to find the values of time for every point between (x_1, y_1) and (x_n, y_n), which are the points that the polynomial regression uses to find the polynomial y=f(x).

Knowing the values of x_i, t_{i-1}, T and the time step tp = $(t_i - t_{i-1})$, which is a design variable, gives us the number of points between the current location and the location corresponding to T.

When the value of T is too small, the number of points to construct the polynomial will be small and it could be two points only, therefore the resulting polynomial will be a straight line. This line does not give any future predicted points.

On the other hand, as T increases, the number of points to construct the polynomial will increase, which increases the number of future predicted points but with low accuracy. The choice of T should therefore be considered carefully.

To further explain the effects of T on the accuracy of the prediction, consider the diagram in Figure 5.4. To find a polynomial of degree *m* that fits the blue points on the Figure above, we use polynomial regression. There are three cases to consider. The first case is to consider all points between point A and point C, which is equivalent to choosing large T. The result will be poly1, the polynomial shown in red. It is very clear that the resulting polynomial does not fit all points; therefore the prediction will not be as accurate as required.

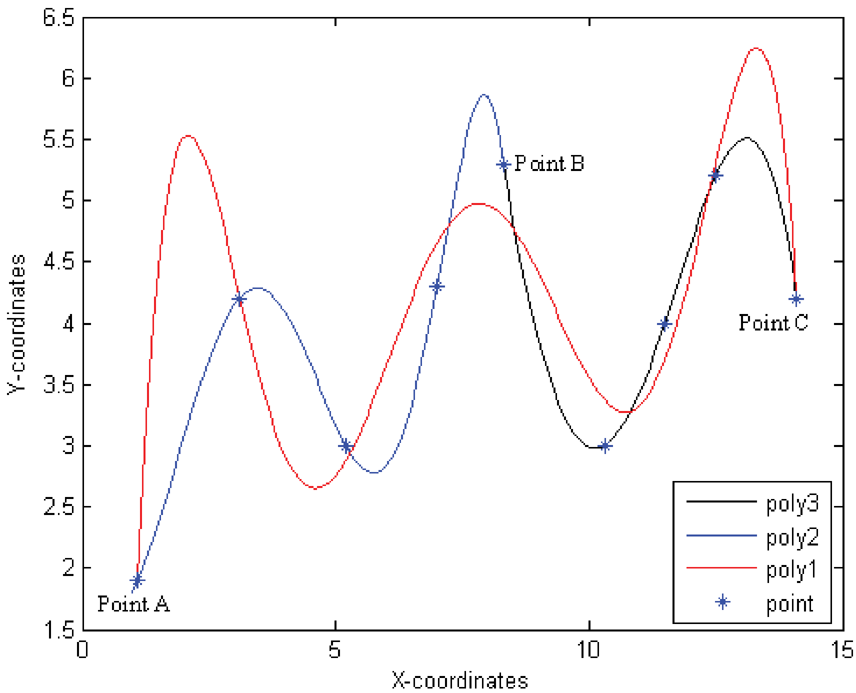

Figure 5.4: The effect of the adjacency of the points on finding a better polynomial fit

The second case is to choose the points between point A and point B and find a polynomial which is poly2, the polynomial in blue. This choice is equivalent to a smaller value of T. The resulting polynomial fits all points between point A and point B. A new polynomial should be obtained for the points between points B and C. It is obvious that a smaller value of T gives fewer predicted points, with higher accuracy,

The third case is to consider too small a value of T, which will give a straight line. This choice is trivial because it does not give any predicted points.

Table 5.1: Format of update packet

Node ID	Control fields …	x_1	x_2	$\cdots$	x_k
		y_1	y_2	$\cdots$	y_k
		t_1	t_2	$\cdots$	$t_k = t_1 + 2T$

5.3 Topology Prediction

The algorithm predicts a set of future locations for the mobile node with the corresponding time for each predicted location every T seconds. The predicted locations are for a period of 2T ahead. Each node which executes the prediction process disseminates the predicted location values into the network every T seconds through an update packet. This packet is addressed to the location service in geographical routing or to the entire network in topology-based routing. Table 5.1 shows the format of the update packet.

The reason for sending the packet every T for a length of 2T is simple. Let us assume that the prediction of the future locations of the node cover a period of T and not 2T. All nodes receive the update packets from one another in random order, therefore a time will come when the information about some nodes is not long enough to predict the topologies for T seconds ahead. To explain this in more detail, consider the update packets received by node A in Figures 5.5 and 5.6.

In Figure 5.5, for simplicity, let the number of predicted locations in each update packet be 6 locations. Node A has the predicted locations for the other nodes with different timing, received from these corresponding nodes. At the current time, node A cannot predict a complete topology of the network because the future locations for node F are unavailable and the future locations for other nodes are insufficient for the prediction of 6 future topologies.

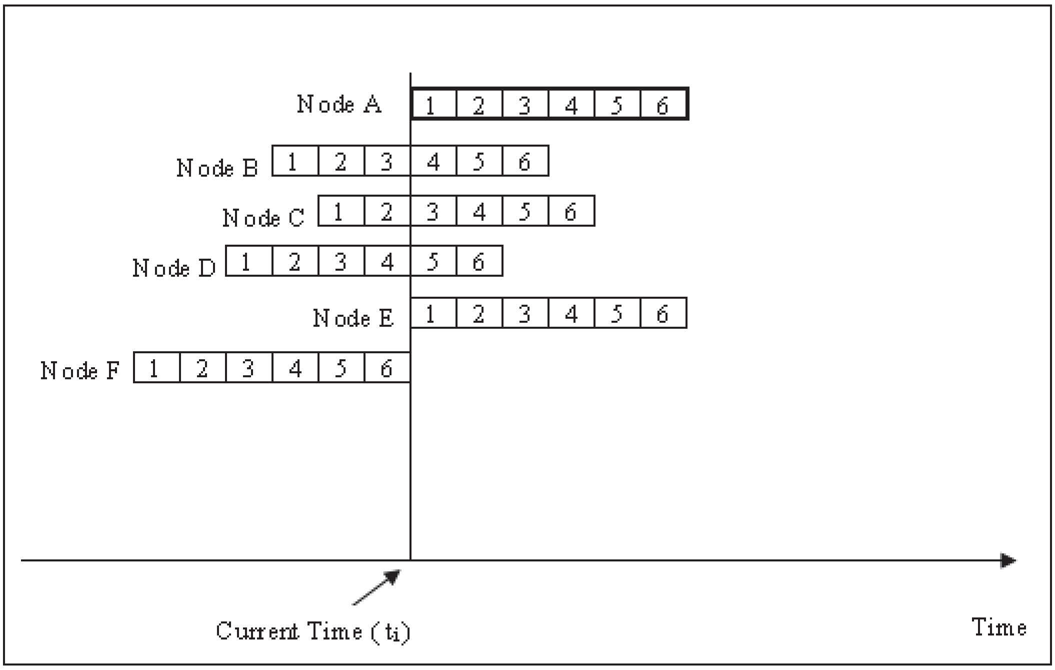

Figure 5.5: The received update packets at node A from other nodes. The period that the update packets cover is T seconds.

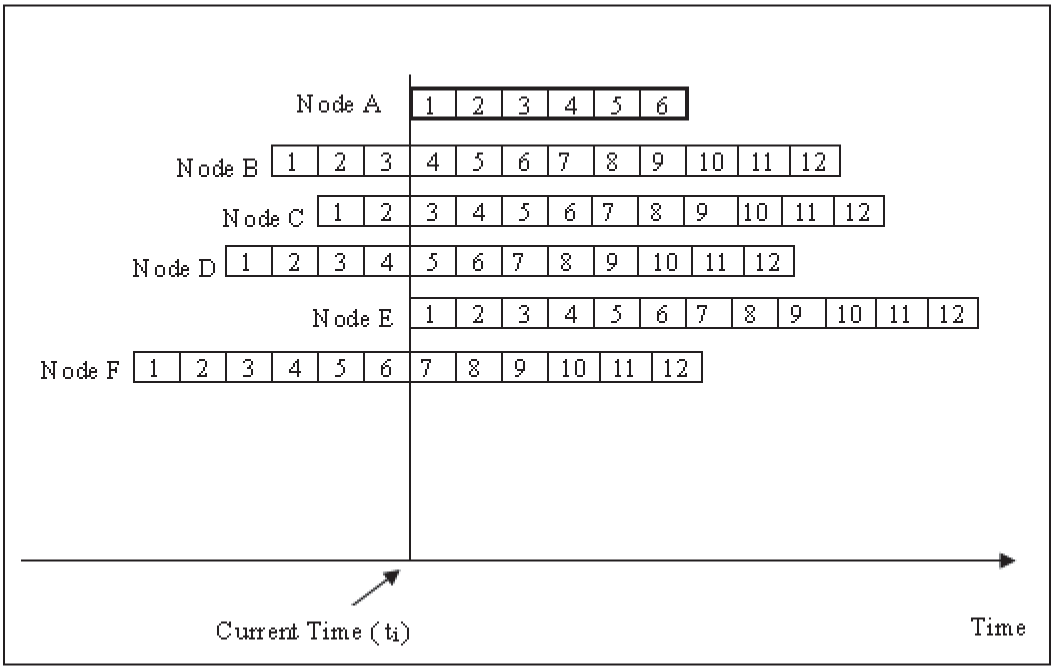

Figure 5.6: The received update packets at node A from other nodes. The period that the update packets cover is 2T seconds.

Now consider the length of the future prediction period is 2T and each node predicts the future of 12 locations, as appears in Figure 5.6. At the current time, the information is sufficient to predict the topologies of the period T.

To keep the prediction as accurate as it must be, each node will broadcast a new set of predicted locations at the end of its own prediction period T, therefore node A will receive a new set of points from node F, which will produce a more accurate prediction.

The received points from node F will overwrite the predicted locations (7-12) which are more recent and give an accurate prediction for node F’s future locations.

5.4 Evaluation of self trajectory prediction performance

The algorithm was evaluated to test its ability to predict the mobility of nodes in mobile ad hoc networks. A computer simulation was carried out using MATLAB. The area of the network was set to 2000m*2000m. The first phase of the simulation was to generate a route from the source to a random destination using our own mobility model described earlier in Chapter 3. The data includes both *x* and *y* coordinates for each point, the speed and the direction of movements of the mobile node at each of these points. The model monitors and stores the coordinates and the time for the node along the path between the source and the destination to use as the actual trajectory. The second phase of the algorithm was to predict the trajectory of the node using the polynomial regression prediction technique, and then to compare the predicted and actual trajectories. This comparison was necessary to test the accuracy of the algorithm and to measure the error during the prediction cycle. The effect of the prediction cycle (T) was also studied by predicting the same trajectory for different values of T. The algorithm predicts a set of future locations along with the expected time for each estimated location. These points were used in predicting the topology of the network at different future times.

Figures 5.7 and 5.8 illustrate a comparison between the actual and the predicted trajectories for a mobile node using two different prediction cycles. The values of T were 35 seconds and 5 seconds respectively.

From these two Figures, we note that the accuracy of the algorithm is inversely proportional to the prediction cycle T. The trajectory in Figure 5.7 is clearly non-identical because the high value of T induces a noticeable error in the predicted trajectory. However, the actual and the predicted trajectories in Figure 5.8 are almost identical. Therefore, a small value of T results in accurate prediction and vice versa.

Choosing a high value of T makes it difficult to find a polynomial that fits all points between the current location and the future location corresponding to T.

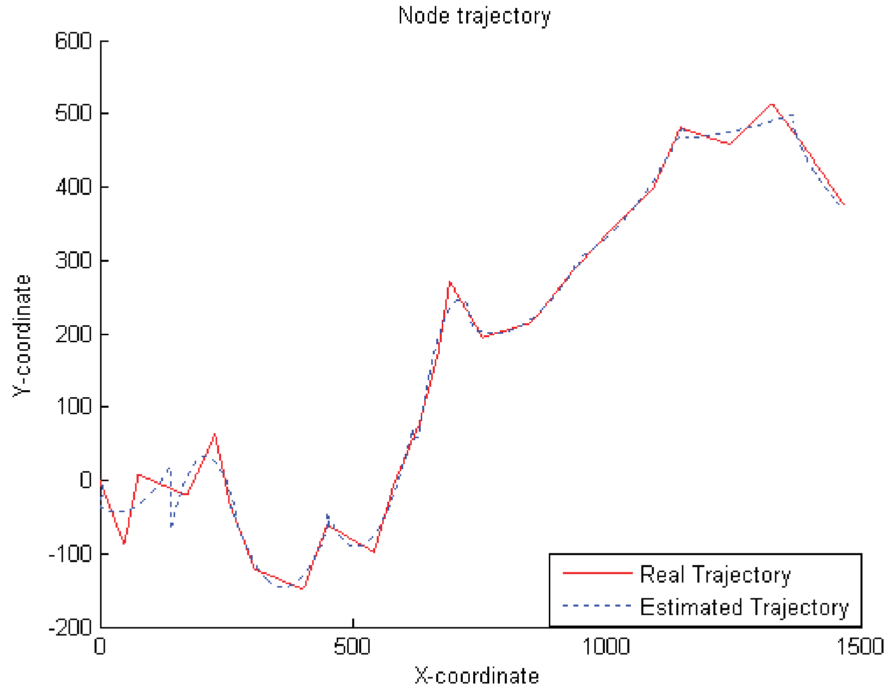

Figure 5.7: A comparison between real and estimated trajectory with a prediction cycle T=35 seconds.

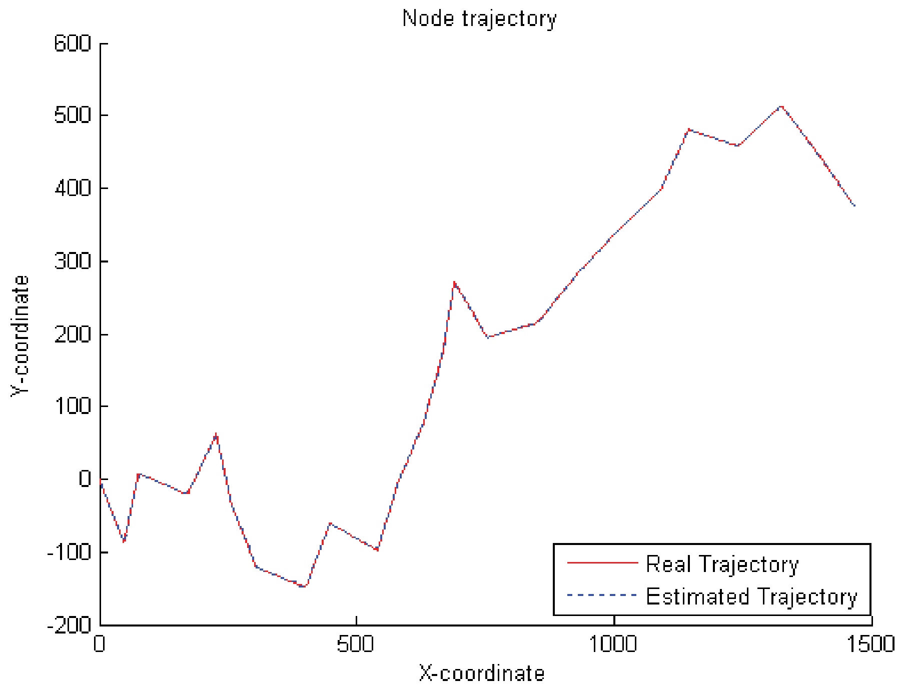

Figure 5.8: A comparison between real and estimated trajectory with a prediction cycle T=5 seconds.

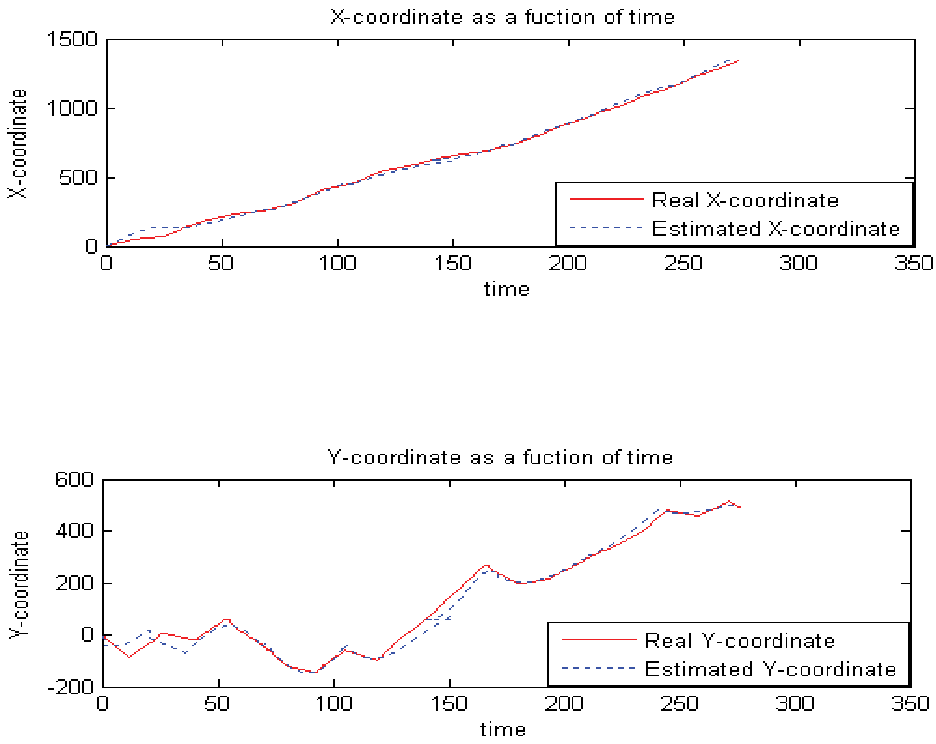

Figure 5.9: Real and predicted coordinates for a mobile node with prediction cycle T=35

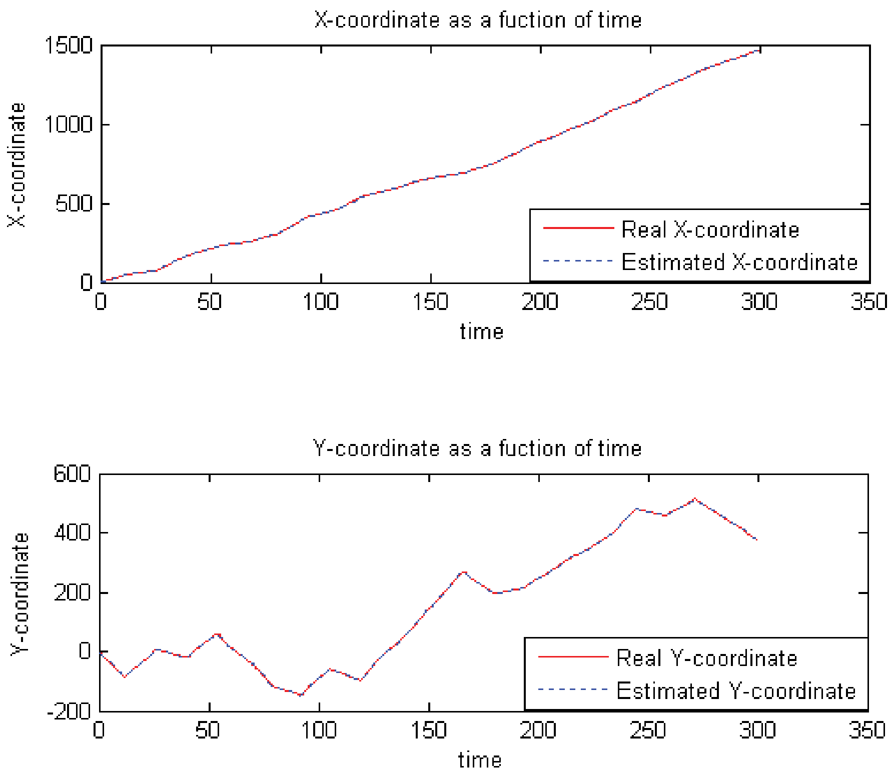

Figure 5.10: Real and predicted coordinates for a mobile node with prediction cycle T=5

The real and predicted values of x and y coordinates for a mobile node with different values of T are shown in Figures 5.9 and 5.10. In Figure 5.9 where the value of T was 35 seconds, the actual and predicted values of x and y coordinates are much less close to each other than the Figure predicted and the actual values in Figure 5.10 where the value of T was 5 seconds.

Comparing the same coordinates in both actual and predicted cases gives a clearer picture of the effect of T on the prediction.

One of the advantages of this prediction algorithm is the non-accumulation of error. A comparison between the actual and predicted x-coordinates for the node does not show any accumulation of error. Similarly, the same results are obtained when comparing the actual and the predicted y-coordinates.

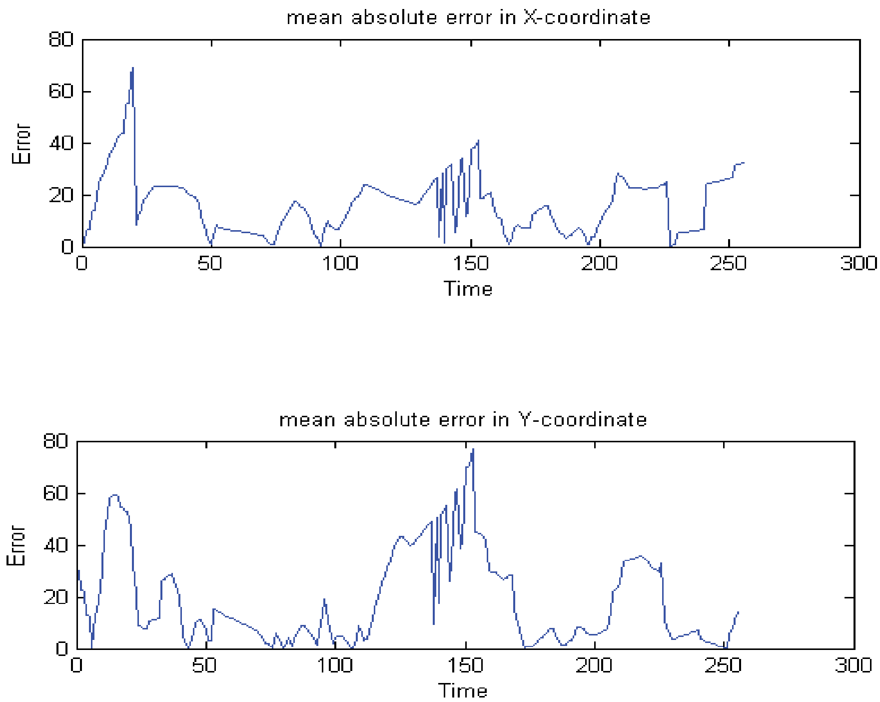

Figure 5.11: Absolute errors in both X and Y coordinates for T= 35 seconds

Figures 5.11 and 5.12 show the absolute errors in both X and Y coordinates for two different values of T. It is clear that the error in any of these Figures is not accumulated because the predicted value of the location at a given time does not directly depend on the previous value. The causes of errors are the assumption of constant speed during the prediction cycle and the process of finding the polynomial.

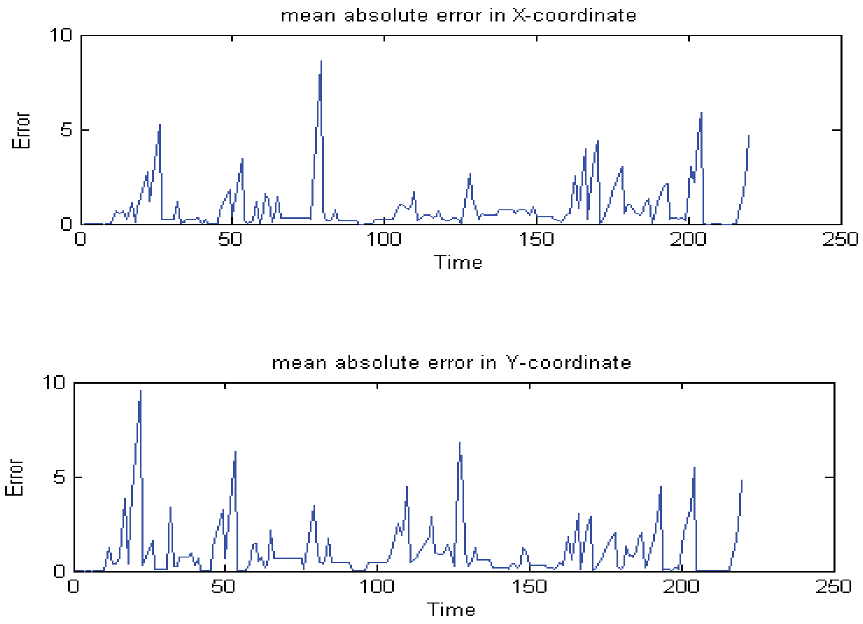

Figure 5.12: Absolute errors in both X and Y coordinates for T= 5 seconds

In the polynomial regression, the choice of the points, the number of these points and the degree of the polynomial affect finding the perfect polynomial. Moreover, after constructing the polynomial, the values of T and t_p which are the variables to mapping the predicted locations to the time, affect the errors. The mean absolute errors in both x and y coordinates, which appear in Figures 5.11 and 5.12, confirm these results.

The mean absolute error in Figure 5.11 where T=35 seconds measures an average value of 20m for both X and Y. this value is reduced to less than 1m in Figure 5.12 when the cycle T=5 seconds.

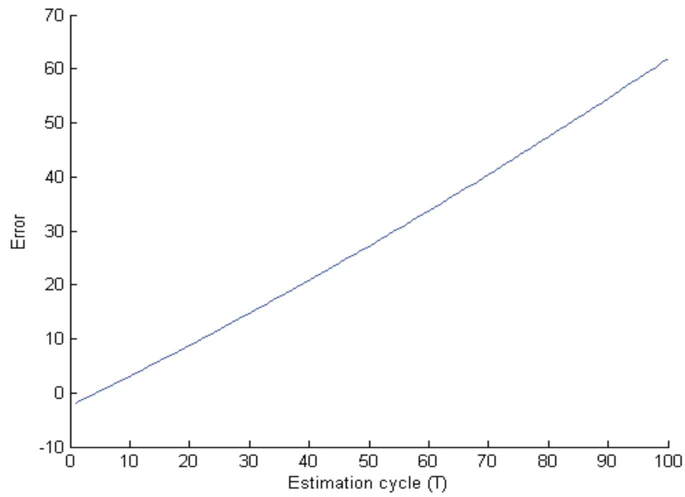

Figure 5.13: Average absolute error in X as a function of prediction cycle (T)

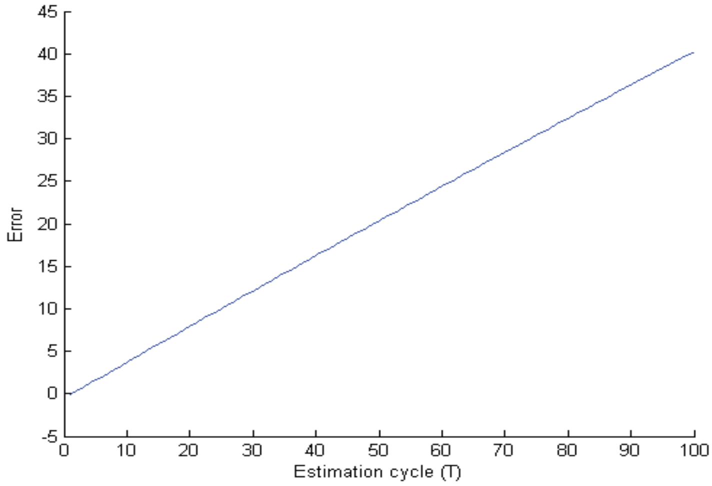

Figure 5.14: Average absolute error in Y as a function of prediction cycle (T)

Figures 5.13 and 15.14 illustrate the relation between the average absolute error and the prediction cycle (T). The mean absolute error increases by the power of two with the prediction cycle T

5.5 Conclusion

In this chapter, we introduced the polynomial regression trajectory prediction to use in estimating the future mobility of a dynamic network. We developed a mathematical model based on polynomial regression to find the relation between X and Y coordinates for a mobile node. We derived the relation between both X and Y coordinates, and the time, for the period of prediction

The method developed in this chapter produces good localization results, and is capable of predicting the future location as a function of time which will be used in predicting the future topology of the network. The sudden change in the direction of movement of the mobile node was solved in this prediction algorithm without broadcasting any extra update.

We have derived several equations which have been used in the design of the algorithm and the choice of the prediction parameters.

Chapter 6

6 Mobility prediction using pattern matching

In this chapter, we introduce a prediction scheme to predict the future mobility of the node based on its past behaviour. Similar to the previous chapter, the node in this scheme predicts its own future mobility and broadcasts it to the network. Once the nodes receive the mobility prediction, they use it to predict the future topologies of the network.

As stated in previous chapters, nodes exhibit some degree of regularity in their movement [13]. They also exhibit similarities in their current and past behaviour; for example, if a node followed a certain route to reach a given destination it will usually follow the same route to go to the same destination later, starting from the same point. People go to work at almost the same time daily. They use the same route. They also stay at work in a similar manner daily. Therefore, it is possible to draw on the past to predict the future.

The movement history of a node which is being observed and recorded on a daily and weekly basis creates a movement pattern for the node [99]. This pattern is used to predict future movement by looking for similarity between the pattern and a small recent section that ends at the current time, as appears in Figure 6.1.

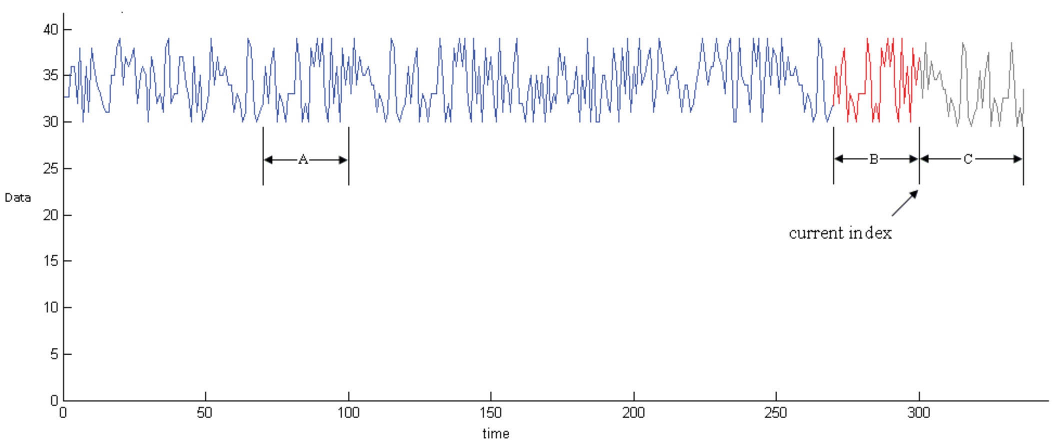

Figure 6.1: An Illustration for the change of data during a certain period.

Figure 6.1 shows an illustrated example of the concept of pattern matching in a given time series of data. Section B of the time series is a repetition of the data that occurred sometime in the past at section A of the time series. The algorithm searches the whole time series to find a section similar to section B. When a match is found, the node's future data will be similar to the section of the time series starting at the point where the match is found (i.e. at the end of section A on the series) and ends at a point which is determined by the required prediction length (the end of section C).

Prediction using cross correlation has been extensively used in the literature for applications such as the prediction of rainfall [100], the stock market predictions [101] [102] and many other applications.

The prediction of movement based on a node's history has previously been proposed for cellular mobile networks. In [54], the movement history of the node was reported as being an effective means to decrease the call-dropping probability and to shorten handover latency. A cross correlation prediction method was used to predict future link quality using a time series of signal–to–noise ratio between the node and its neighbours [12]. Inspired by this work, which uses pattern matching to predict the signal-to-noise ratio for the node's neighbours, we propose a prediction scheme to predict the future locations of the mobile networks using pattern matching and correlation analysis.

Each node monitors and stores its past location to obtain a time series of locations for a given period of time. We refer to this series as the Location Time Series (*LTS*). When a node wants to predict its future locations, it searches the *LTS* for a pattern similar to the current situation. We refer to this pattern as the Recent section of Time Series (*RTS*) or 'the query'. It starts at the current point in time and goes back for O points. O is called the 'query order'. Several series can be stored and searched for best matching; searching more series will give a higher probability of pattern matching. Details of the proposed scheme are presented in the following sections.

To perform the search, the node computes the normalized cross correlation between the current query and the *LTS*. The normalized cross correlation is used to limit the value of the cross correlation between 1 and -1.

6.1 Cross correlation

Cross correlation is a method to find how two series are correlated. It works by shifting one signal by *d* and multiplying it by the other series.

Consider two series *x(i)* and *y(i)* where *i=1,2...N*. The cross correlation *cor* at delay d_1 is defined as:

$$cor = \frac{\sum_i [(x(i)-\bar{x})*(y(i-d_1)-\bar{y})}{\sqrt{\sum_i (x(i)-\bar{x})^2}\sqrt{\sum_i (y(i-d_1)-\bar{y})^2}} \quad (6.1)$$

where $\bar{x}$ and $\bar{y}$ are the means of the corresponding series. If the cross correlation is computed for all delays *d=1,2,...N* then it results in a cross correlation series that has a value in the range [-1 ,1]. Equation 6.2 represents the cross correlation series for all values of delay *d*. The resulting cross correlation time series will peak at several values of delay *d* indicating high similarity between *x(i)* and *y(i)* at the corresponding values of shift *d*. The global peak will then represent the best match between the two time series

$$cor(d) = \frac{\sum_i [(x(i)-\bar{x})*(y(i-d)-\bar{y})}{\sqrt{\sum_i (x(i)-\bar{x})^2}\sqrt{\sum_i (y(i-d)-\bar{y})^2}} \quad (6.2)$$

To further explain the concept of correlation; let us consider the two rectangular pulses shown in Figure 6.2. The rectangle pulses are shown in blue and green, whereas the correlation series is shown in red. The maximum correlation is achieved at a delay of 3 when both pulses are similar, the second series is slid past the first. At each shift, the sum of the product of the newly lined-up terms in the series is computed. The sum will be large when the shift (delay) is such that a similar structure lines up. This is essentially the same as the so-called convolution except for the normalisation terms in the denominator [103]

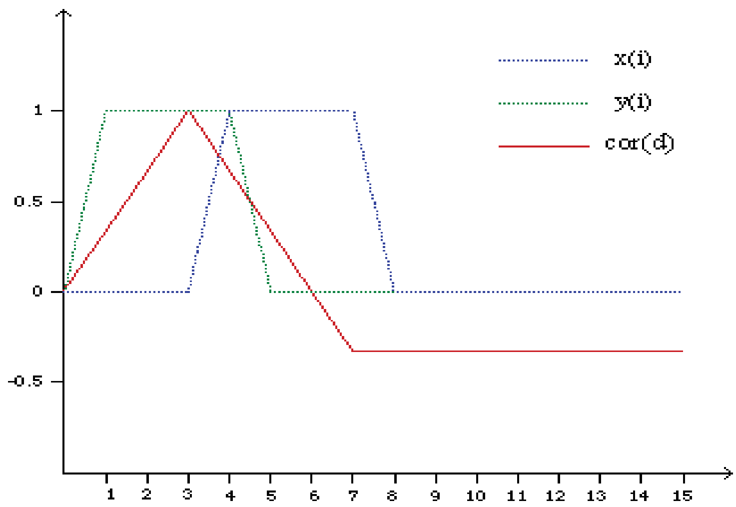

Figure 6.2: A simplified concept of the correlation between two rectangular pulses [103]

6.2 Algorithm assumptions

As with all our mobility prediction methods used in previous chapters, Nodes are assumed to exhibit some regularity of behavior This regularity generates a pattern of the mobility of the nodes; therefore, the cross correlation is used to search for the similarity between patterns. In other words, the regularity generates patterns and the patterns are used to predict future mobility.

We also assume that nodes are aware of their current location by using GPS technology. The locations are stored to form a time series of locations which are used by algorithm to predict the topology through pattern matching.

6.3 Algorithm description

Each node stores its own location as a time series of locations called *LTS* in a circular buffer. As discussed earlier, the recent part of the location time series, also referred to as the query, is denoted by *RTS*. The *LTS* contains *N* past measurements for the node own location spanning from the current time location back to *N* past locations. These measurements are periodically obtained by the node and stored in a circular buffer. *LTS* can is represented by $LTS = D_{i-N} ... D_i$ with D_i denoting the current location. On the other hand, *RTS*, which is a set of recent measurements of *LTS* spanning from current time back to *O* past measurements is represented by $RTS = D_{i-O} ... D_i$ with *O* representing the order of *RTS*. The choice of *O* in *RTS* and the effect of *O* will be discussed in detail in the next section. Figure 6.3 illustrates the difference between *LTS* and *RTS*.

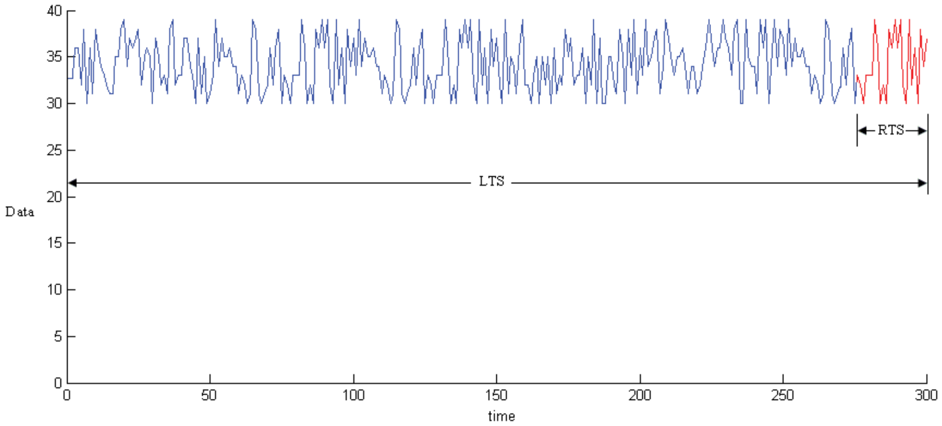

Figure 6.3: *LTS* is a time series of data for the node; it includes all data from the current time and back to *N* points, while *RTS* is a recent part of *LTS* from the current data and back *O* points.

When a node wants to predict its future mobility, it calculates the normalized cross correlation between *LTS* and *RTS* for all values of delay $d = 1, 2, \ldots N$ using equation 6.3:

$$cor(d) = \frac{\sum_{i=1}^{O} (RTS(i) - \overline{RTS}) * (LTS(i+d) - \overline{LTS})}{\sqrt{\sum_{i=1}^{O} (RTS(i) - \overline{RTS})^2 \sum_{i=1}^{O} (LTS(i+d) - \overline{LTS})^2}} \tag{6.3}$$

Equation 6.3 is basically the same as equation 6.2 with *LTS (i)* replacing *y(i)* and *RTS(i)* replacing *x(i)*. However, since *RTS* is shifted to the left, the delay *d* is given a negative sign. The resulting series is the normalized cross correlation for all values of the delay *d*. The global peak at the normalized cross correlation that occurs at delay *m* is the best match. Therefore, the predicted future locations (*PFL*) of the node will be similar to the values of *LTS*, $PFL = D_{i-m} \ldots D_{i-m+p}$, starting at *i-m* and ending at *i-m+p* with *p* denoting the length of the prediction series. Similar to the algorithm in Chapter 5, the *PFL* for X and Y coordinates is calculated every T seconds and broadcast to the network to enable the nodes to predict the future topologies of the network.

6.4 Evaluation of the algorithm

Our aim is to evaluate the proposed algorithm's ability to predict the future trajectory of a mobile node based on its past trajectories. We simulate the trajectory of a mobile node using actual GPS data. The trajectory data used in this simulation was obtained by GPS reading systems carried by lab members from the DBMC at the University of Illinois at Chicago during their daily commute [14]. The trajectory was sampled every second. X and Y coordinates for the sample points along the trajectory were obtained using the projection of NAD 1983 HARN State Plane Illinois East.

For simulation purposes, one of the mobile node trajectories on a certain day was taken as a reference representing the path followed by the node under consideration. The reference is divided into a number of slots, each representing a query (RTS). For each query, the algorithm using pattern matching is used to predict future location *PFL*s (future trajectory of the node). The predicted trajectory is then compared with the actual trajectory in order to evaluate the performance of the algorithm. The plot of the actual and predicted trajectories of the node will be similar to Figure 6.4.

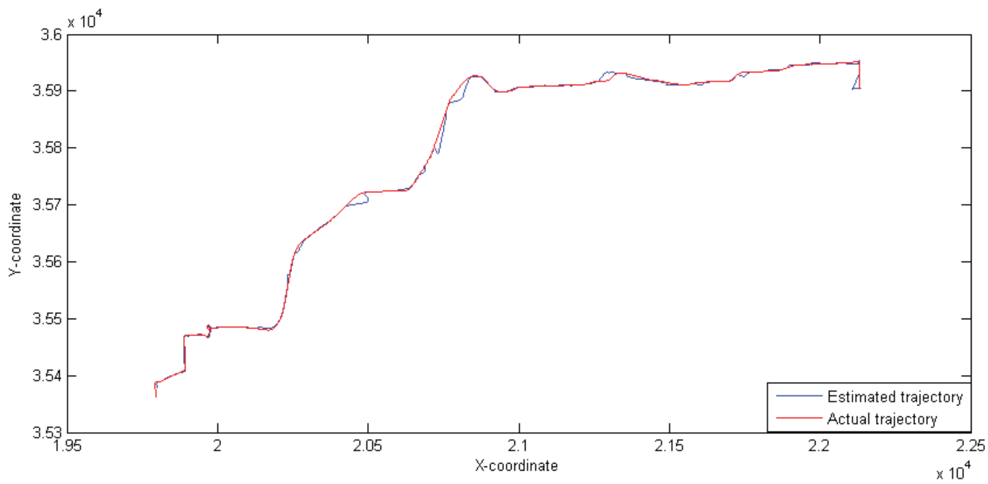

Figure 6.4: Actual vs. predicted node trajectory

The number of data sets used in this evaluation was ten. The algorithm therefore searched for a matching between the current query and all ten data sets to find the best matched pattern. The algorithm was run for different lengths of the predicted future locations (*PFL*). The length of the *PFL* series, *p*, affects the accuracy of the prediction. The shorter the length *p*, the more frequently *RTS* is obtained. As *RTS* is actual data, the prediction will become more accurate. As a result of the new query at the end of *PFL* the estimation error will be slashed to zero. When the length *p* is chosen as small, the resulting mean absolute error is small. This means the algorithm will search for matched patterns more frequently than when *p* is high.

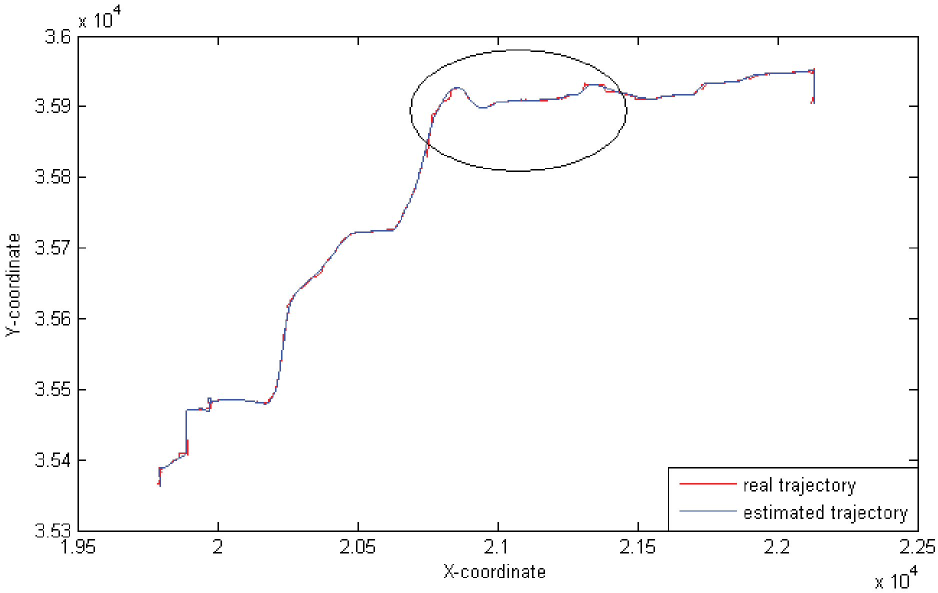

Figure 6.5: Actual and estimated trajectory for a node with *PFL*=25

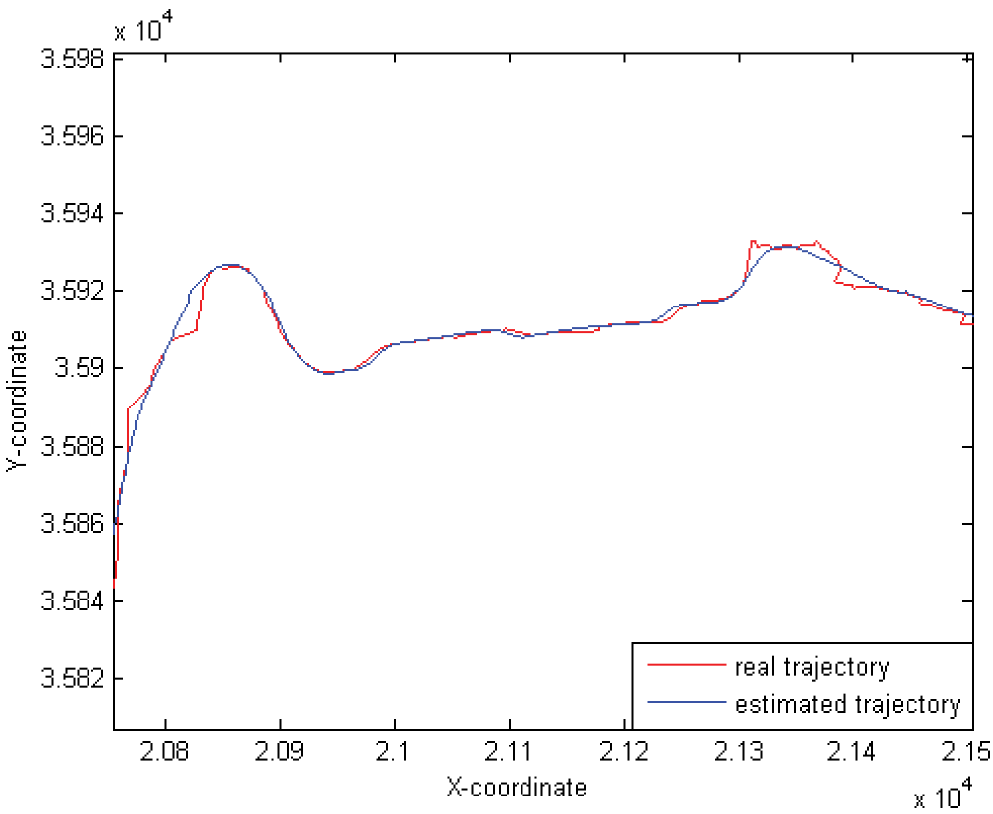

Figure 6.5A: Magnification of the circled area of figure 6.5

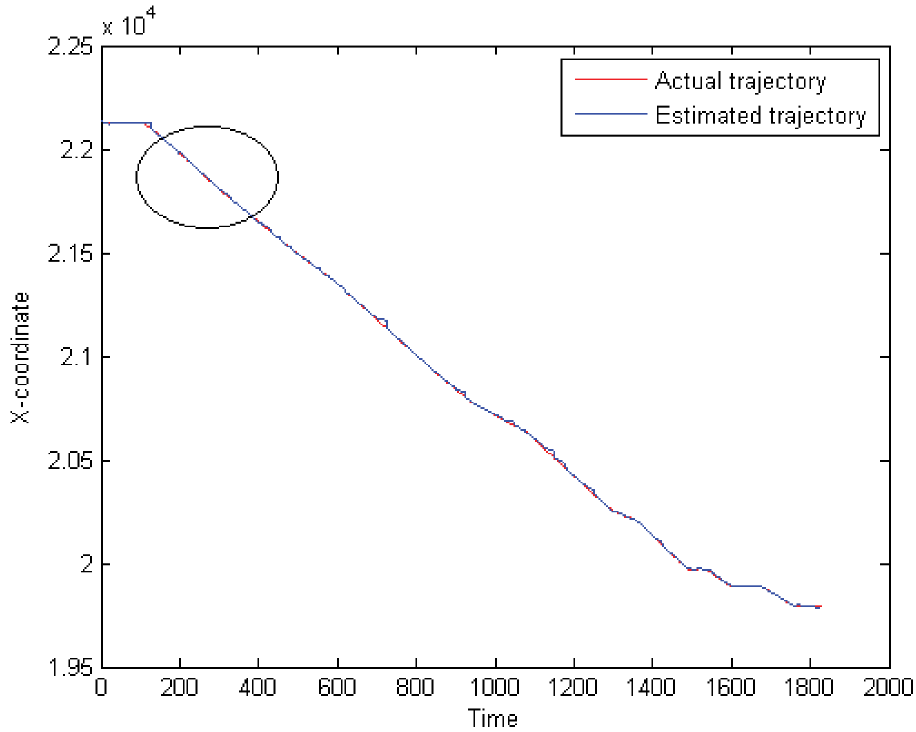

Figure 6.6: Actual and estimated X-coordinate varying with time for a node with *PFL*=25

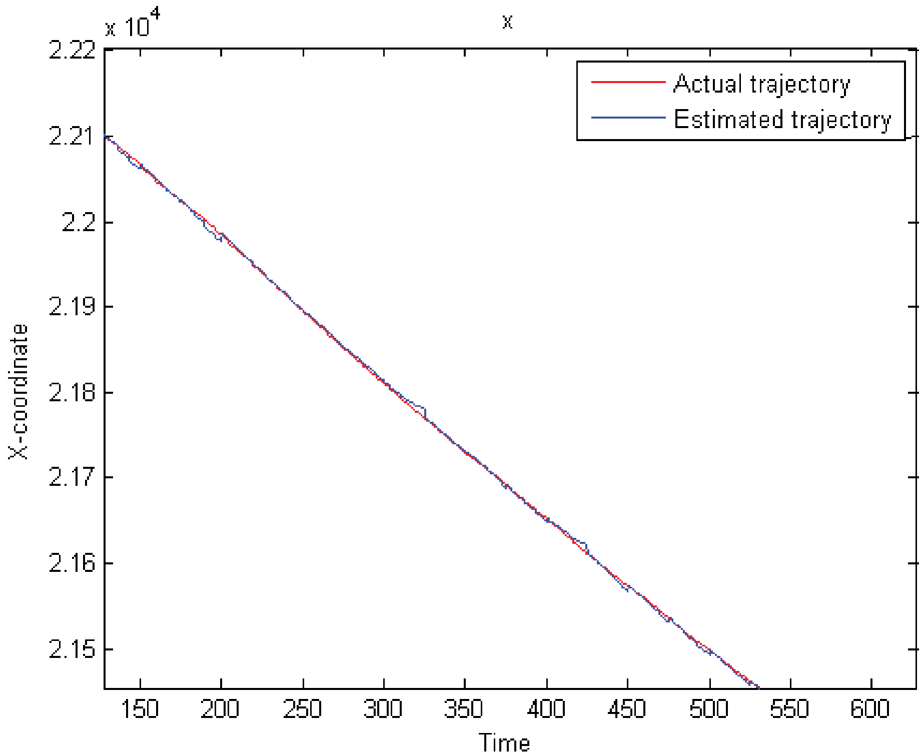

Figure 6.6A: Magnification of the circled area of figure 6.6

To illustrate the effect of *p* on the accuracy of the prediction, let us consider two values of *p*, namely, 25 and 150. Firstly, let us consider the case when the value of *PFL* was set to 25 points. The mean absolute error in the x-coordinate was 3.2m and the mean absolute error in the y-coordinate was 1.42m. Setting p to 25 means the algorithm estimates the future mobility of the node for 25 seconds into the future. This is a very reasonable value for the error; however, the use of a small value for *p* requires more computations to find the future locations compared to a high value of *p*. Figures 6.5, 6.6 and 6.7 show the result of the simulation using *p*=25 and figures 6.5A, 6.6A and 6.7A a magnification of the circled area of figures 6.5, 6.6 and 6.7 respectively.

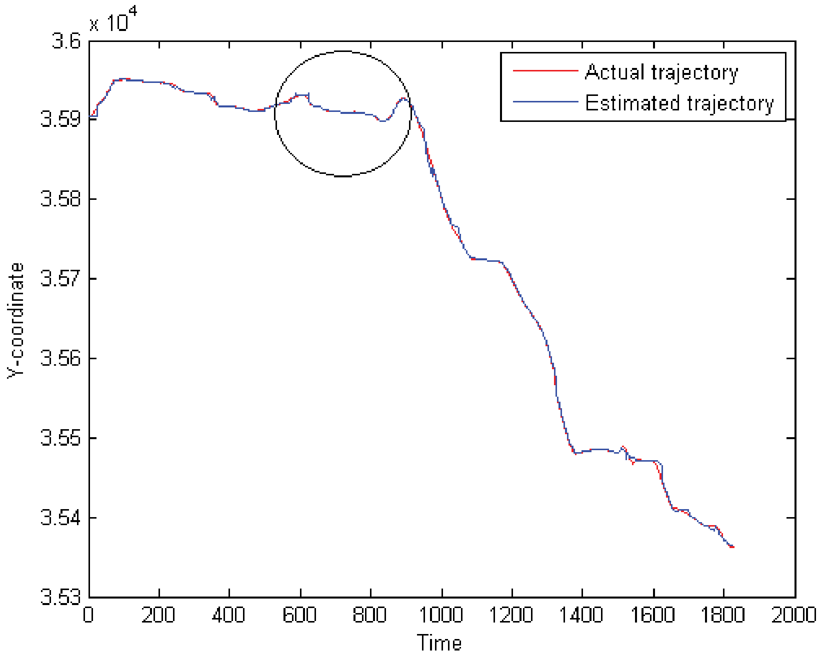

Figure 6.7: Actual and estimated Y-coordinate varying with time for a node with *PFL*=25

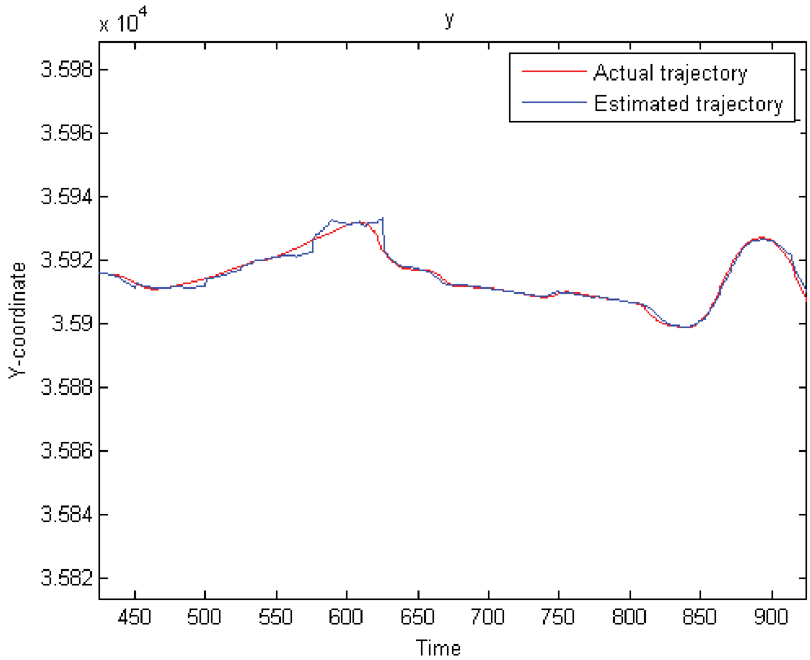

Figure 6.7A: Magnification of the circled area of figure 6.7

On the other hand, when the value of *p* is chosen to be 150, the mean absolute error in x-coordinate is 24.1m and the mean absolutes error in y-coordinate is 7.96. Comparing the mean error in *x* and *y* coordinates in both cases of *p* clearly shows how varying the *PFL* affects the accuracy and complexity of the algorithm. Figures 6.8, 6.9 and 6.10 show the results for *p* =150. The results for this value of *p* give a higher value of error with less computational complexity. Figures 6.8A, 6.9A and 6.10A show details plots for the circled areas in figures 6.8, 6.9 and 6.10 respectively.
The maximum number of loops required to find the correlation that best matches the query can be expressed by:

$$M = (N / PFL) \times N \times N$$

where, *N* is the length of *LTS*, *N/p* is the total number of *p* sections. The complexity of the algorithm is inversely proportional to the value of *p*.

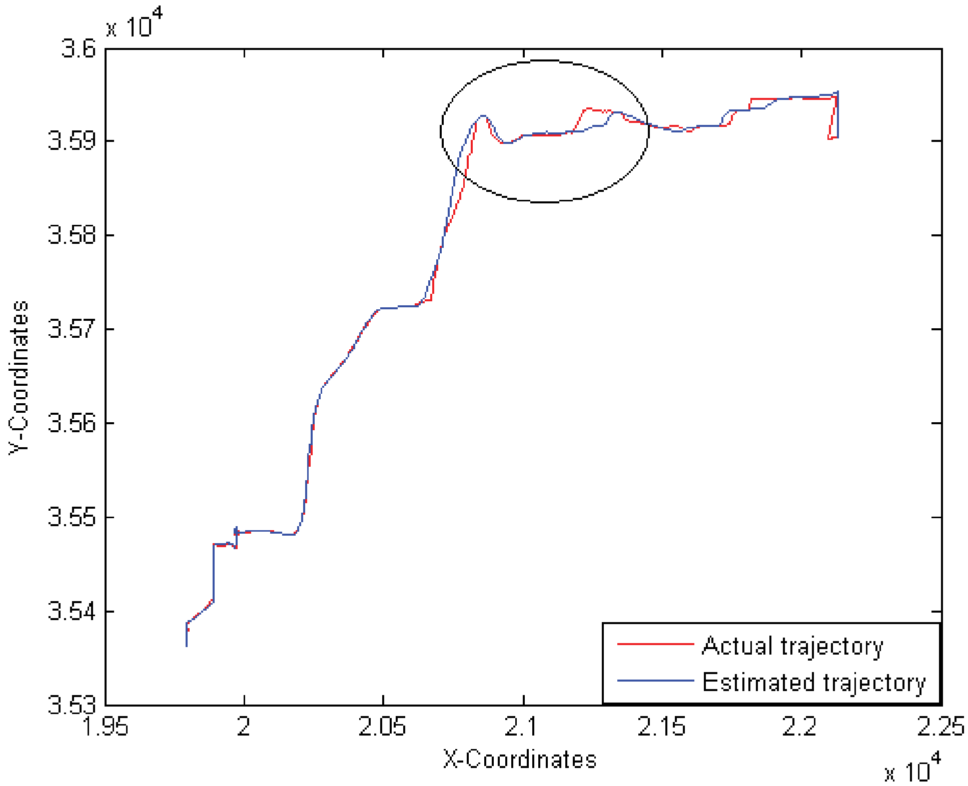

Figure 6.8: Actual and estimated trajectory for a node with *PFL*=150

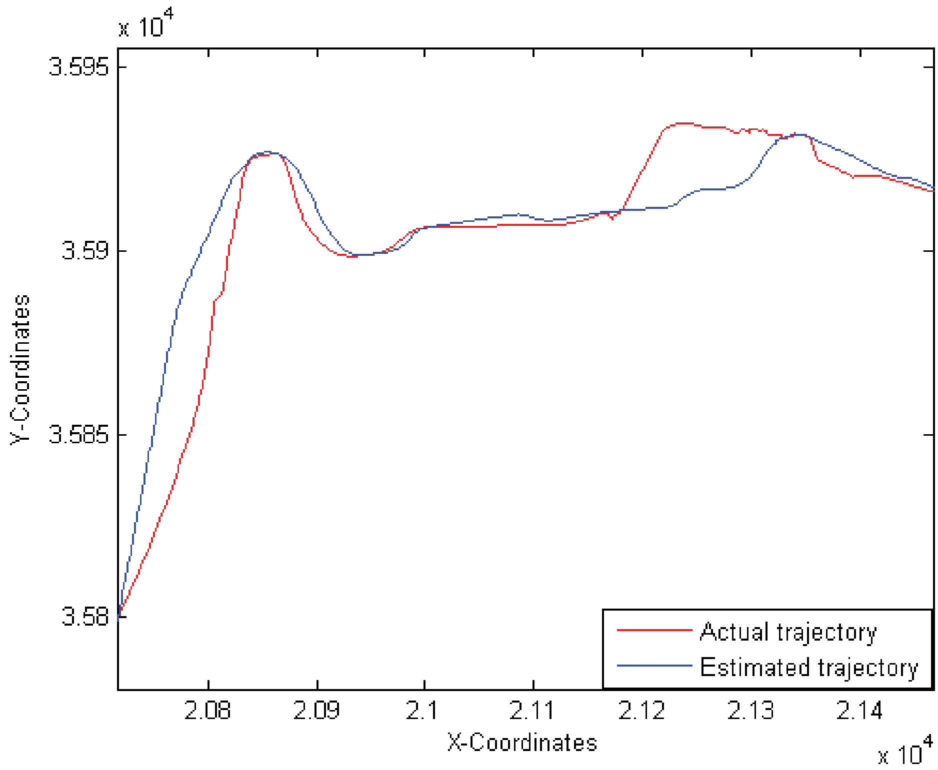

Figure 6.8A: Magnification of the circled area of figure 6.8

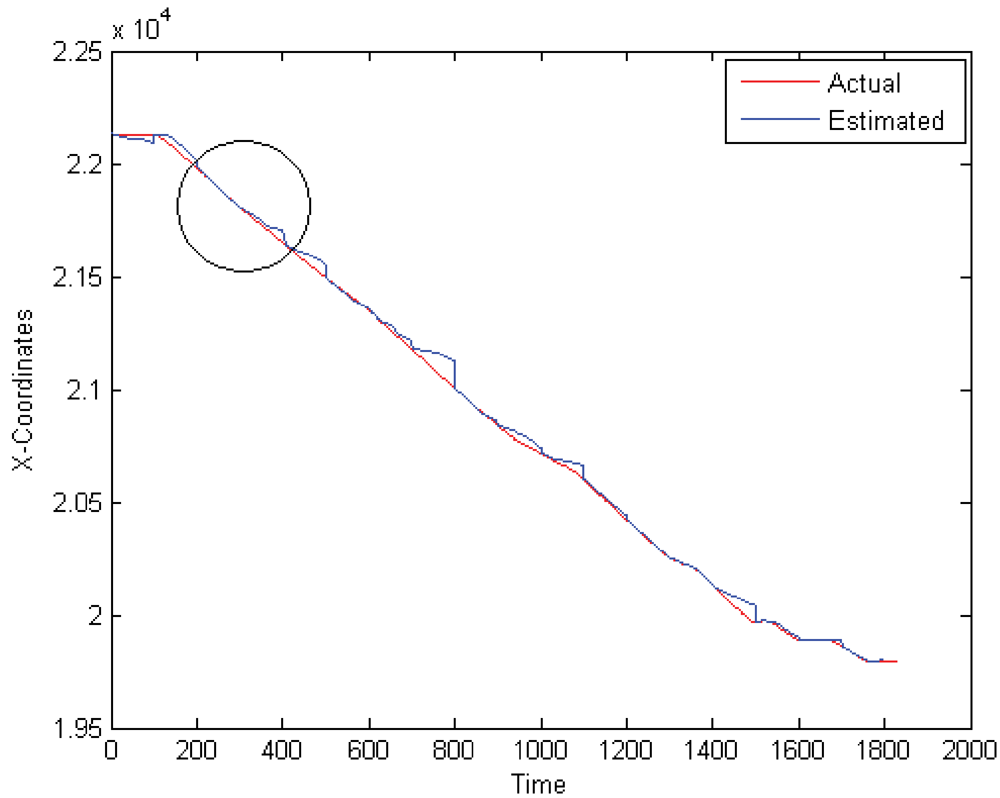

Figure 6.9: Actual and estimated X-coordinate varying with time for a node with *PFL*=150

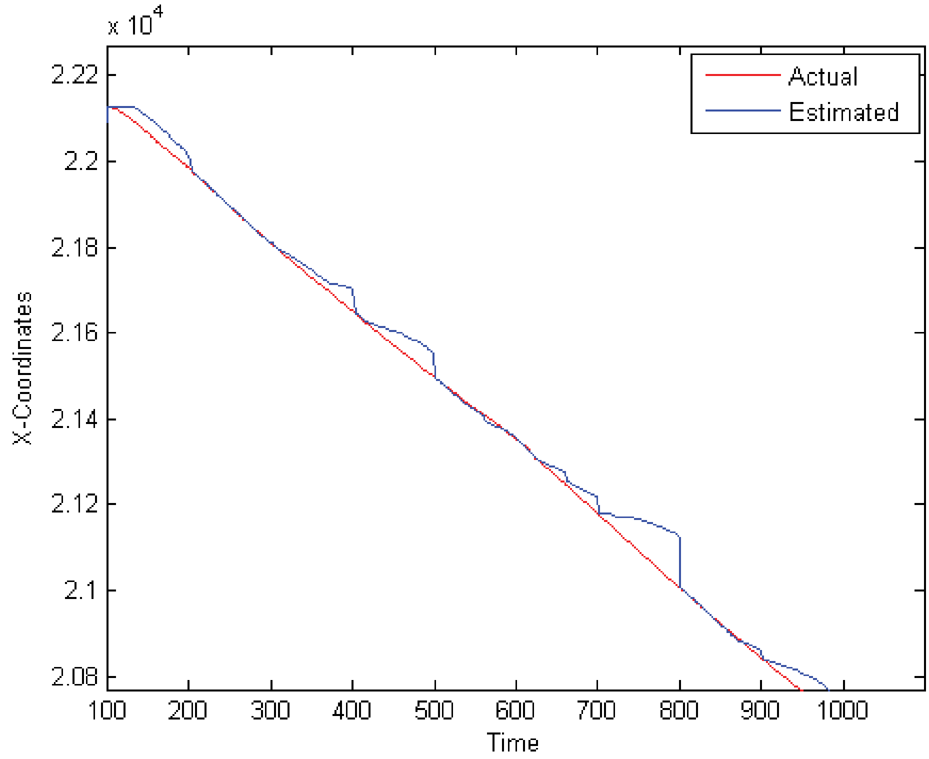

Figure 6.9A: Magnification of the circled area of figure 6.9

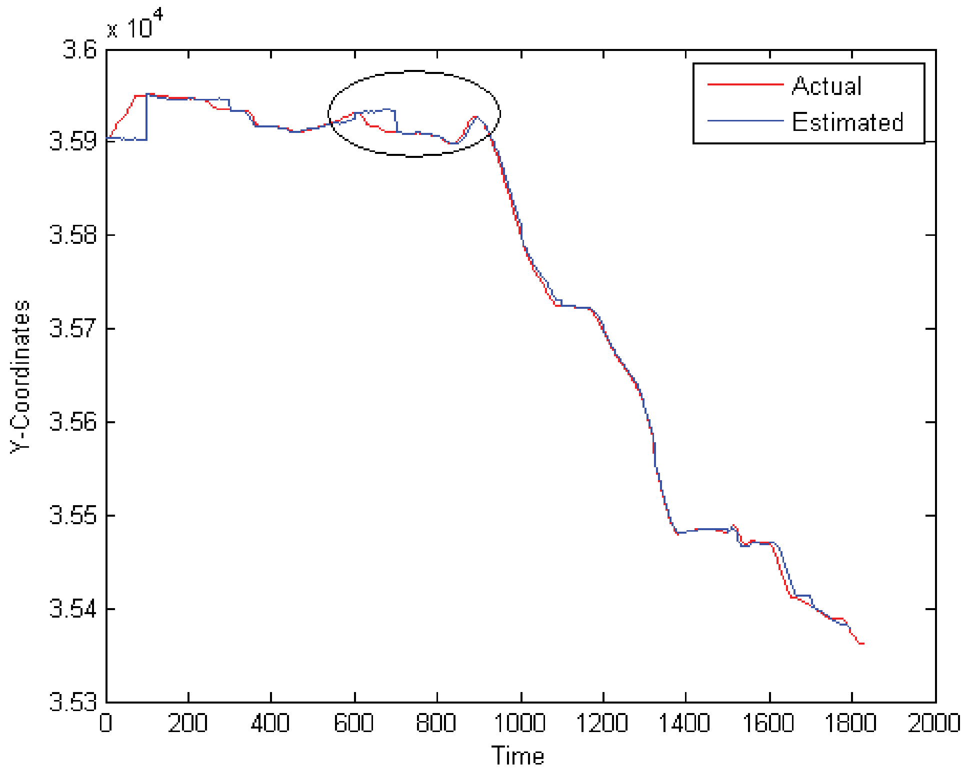

Figure 6.6: Actual and estimated Y-coordinate varying with time for a node with *PFL*=150

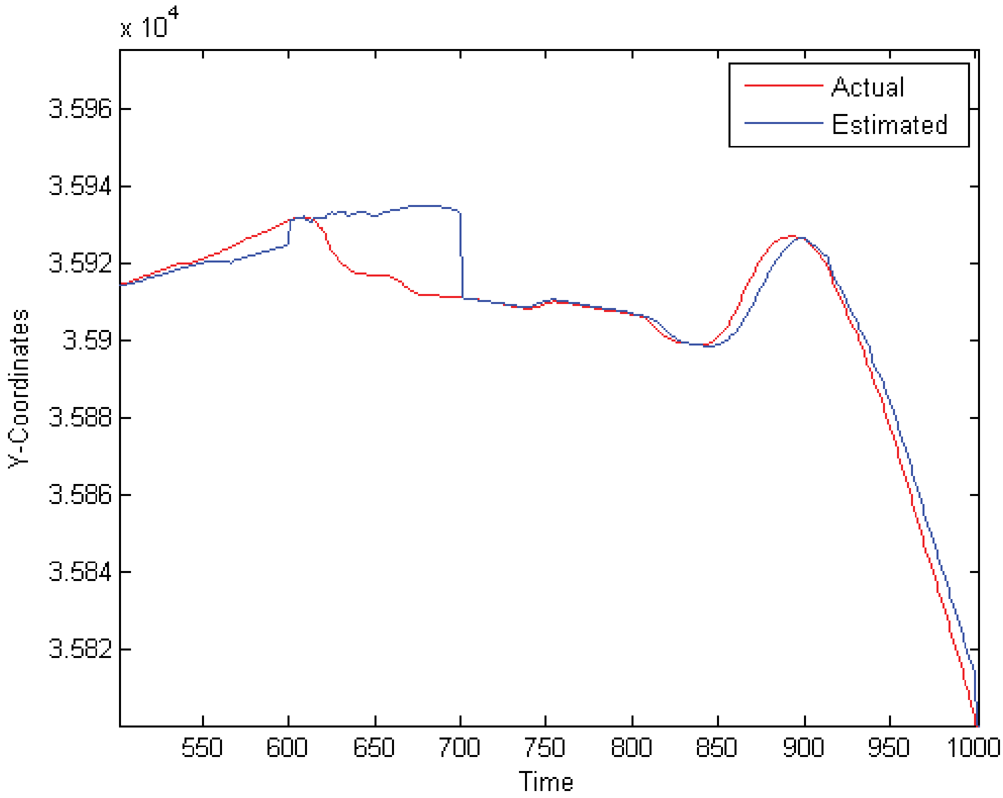

Figure 6.7A: Magnification of the circled area of figure 6.10

In our simulation, we vary the value of p and run the simulation using the same reference data set and the same data sets for the search. We conclude that the value of prediction error is proportional to the value of p. Figures 6.11 and 6.12 show how the mean absolute error in estimated x and y related to the length of the future predicted locations.

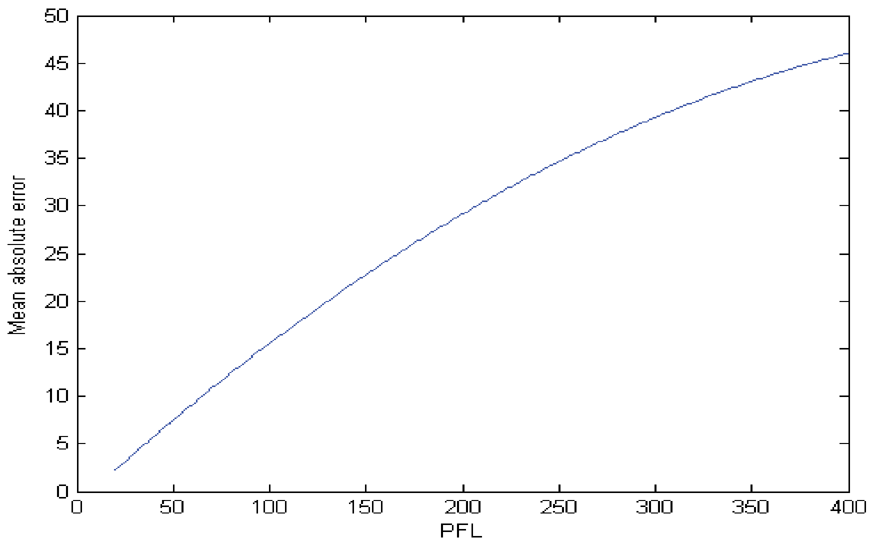

Figure 6.8: Mean absolute error in estimated x

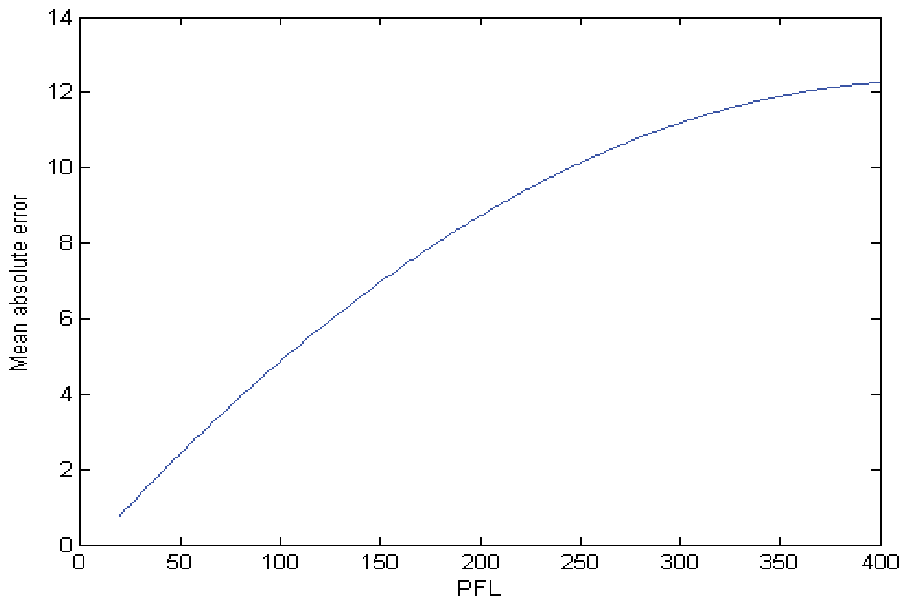

Figure 6.9: Mean absolute error in estimated *y*

It is important here to highlight the effect of *O*, the order of the recent time series on the matching between *RTS* and *LTS*. It has been noticed that having a small value for *O* increases the probability of finding matching patterns that do not necessarily lead to the required prediction. It is not necessarily true that having a small matching period on a certain data set means that the following period will match the actual future path of *PFL*. On the other hand, having high value *O* will make the matching process very hard to obtain and sometimes impossible leading to no prediction at all.

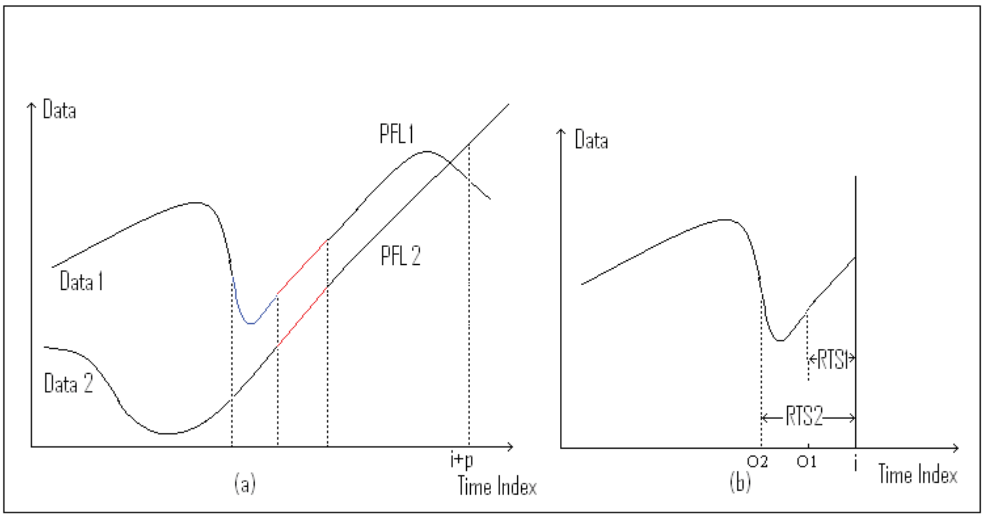

Figure 6.10: The effect of the order of *RTS O* on finding a correct matching between *RTS* and the *LTS*s

To further illustrate the effect of the value of *O*, consider Figure 6.13 which shows the actual trajectory followed by the node (Figure 6.13 b) and two data sets previously obtained for that trajectory(Figure 6.13 a). The node is moving along the path as appears in Figure 6.13 b. This movement is very similar to the past movement stored at data1 as illustrated in Figure 6.13 a. At time *i* the algorithm is to predict new PFL. If the value of *O* was small as for *O1*, it will lead to the recent pattern *RTS1*. There are two possible matches for this pattern, in data1 and data2 (the part coloured red Figure 6.13a). The algorithm may choose a *PFL* from data 2, resulting in the future predicted values for the trajectory (path) of the node in Figure 6.13b being *PFL2*, which is an incorrect prediction. Alternatively, if the value of *O* is *O2*, this will lead to the recent pattern RTS2; the matched data will be only from data1 (the red and blue section together) and the *PFL* will be *PFL1***.**

6.5 Conclusion

We have proposed and tested a prediction scheme using pattern matching to predict the future mobility and topology of mobile networks. We have assumed that the node exhibits a periodicity in its behaviour in terms of mobility. Real location data was used to test the validity of the proposed algorithm, which gave promising results. This solution can produce a longer prediction period than all other methods but it requires the use of memory to store the past locations and needs more computations.

The solution presented in this chapter allows the node to predict its own mobility.

The resulting error is very small compared to other solutions and does not accumulate because the predicted value at time t_{i+1} does not depend directly on the predicted value at time t_i. Rather, it depends only on the length of the predicted future locations.

Chapter 7

7 Comparison analysis

7.1 Comparison description

In previous chapters, four different prediction schemes were proposed and tested. These schemes are:

1. prediction of node mobility using the motion of nodes
2. modified prediction of node mobility using the motion of nodes
3. prediction of node mobility using polynomial regression
4. prediction of node mobility using pattern matching

In the evaluation of each scheme, different sets of location data were used in testing and validating the proposed schemes. For the first three schemes, data was obtained by the proposed mobility model which was presented in Chapter 3, while in the evaluation of the fourth prediction scheme, real location data was obtained by members of the Databases and Mobile Computing Laboratory at the University of Illinois at Chicago during their daily commute [14].

The evaluation method in each scheme included a comparison between the actual and predicted trajectory of the mobile node of interest, comparing the predicted and the actual x and y coordinates and finding the absolute error in each of the predicted coordinates. The effects of varying several variables in each scheme were also observed.

In this chapter, the same set of data is used to test all prediction schemes and to compare them. The use of the same data set will test the ability of each prediction method of working under the same environment and the same mobility conditions. The comparison concentrates on the predicted trajectory and will not go into detail about any of the schemes because each prediction scheme has been covered in the previous four chapters, and the purpose of this comparison is to discover how each prediction scheme behaves under the same environment and mobility conditions.

To compare these different schemes, we have evaluated them using the same set of location data while varying the parameters that affect each prediction model such as the prediction period T, the threshold angle, the length of *PFL*.

7.2 Evaluation

The result of the evaluation confirms that the accuracy of the prediction in all techniques is inversely proportional to the length of the prediction period, but with various extents in each scheme. In other words, increasing the length of the prediction period will decrease the accuracy of the prediction scheme. Figures 7.1, 7.2 and 7.3 show the actual trajectory along with the predicted trajectories for the four prediction schemes, using three different values of the prediction period T. These Figures show how increasing the length of the prediction period decreases the accuracy of the predicted trajectory. For example, when T was set to 10 seconds as in Figure 7.1, the difference between the actual and the predicted trajectories is not noticeable, however, it became very clear when T was set to 75 seconds or 150 seconds as illustrated in Figures 7.2 and 7.3 respectively.

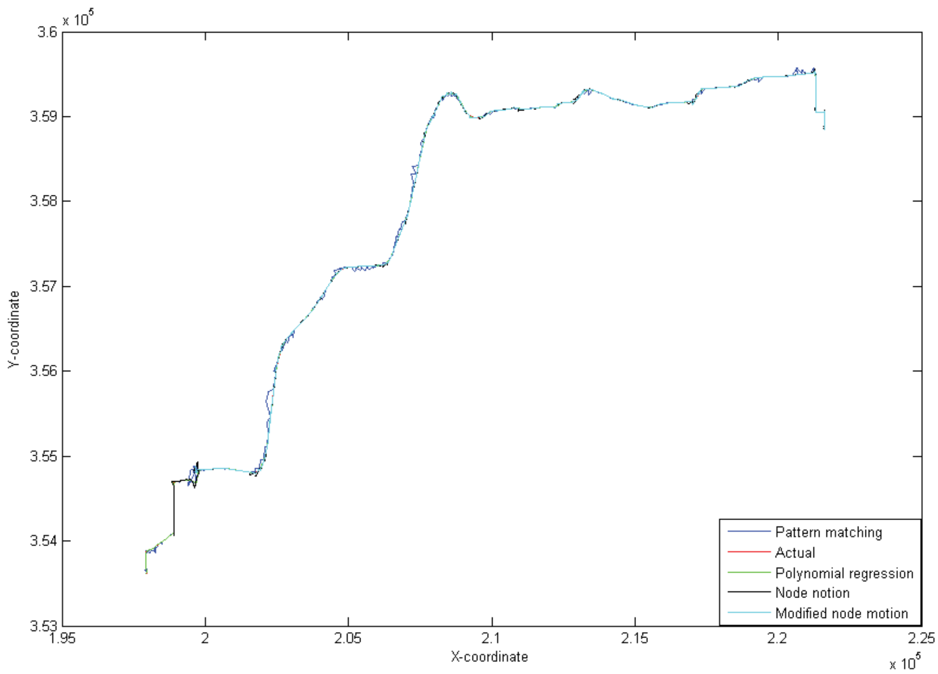

Figure 7.1: Actual vs. predicted trajectories for a node using different prediction schemes with the value of the prediction period T equal to 10 seconds.

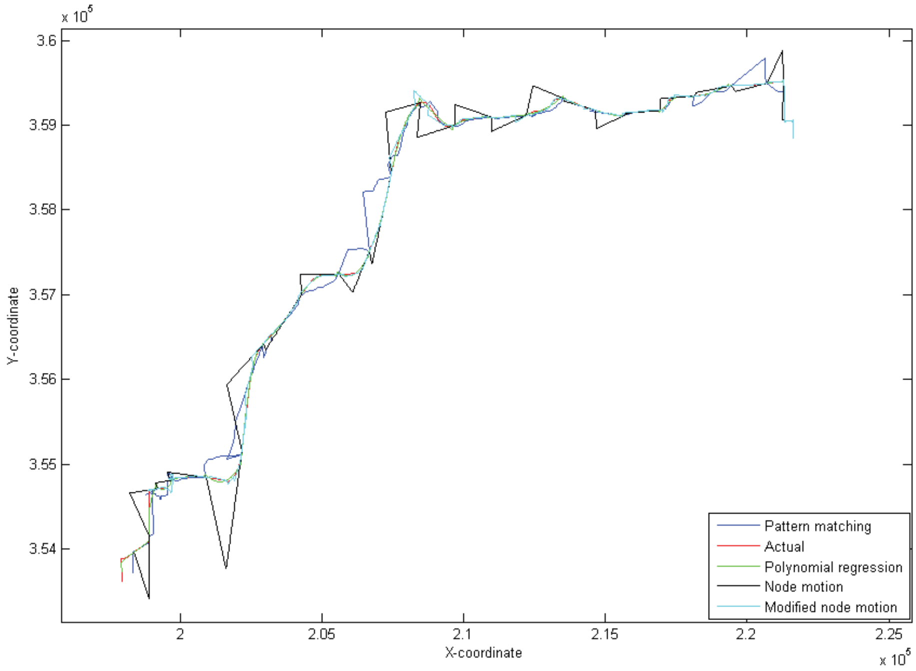

Figure 7.2: Actual vs predicted trajectories for a node using different prediction schemes with the value of the prediction period T equal to 75 seconds.

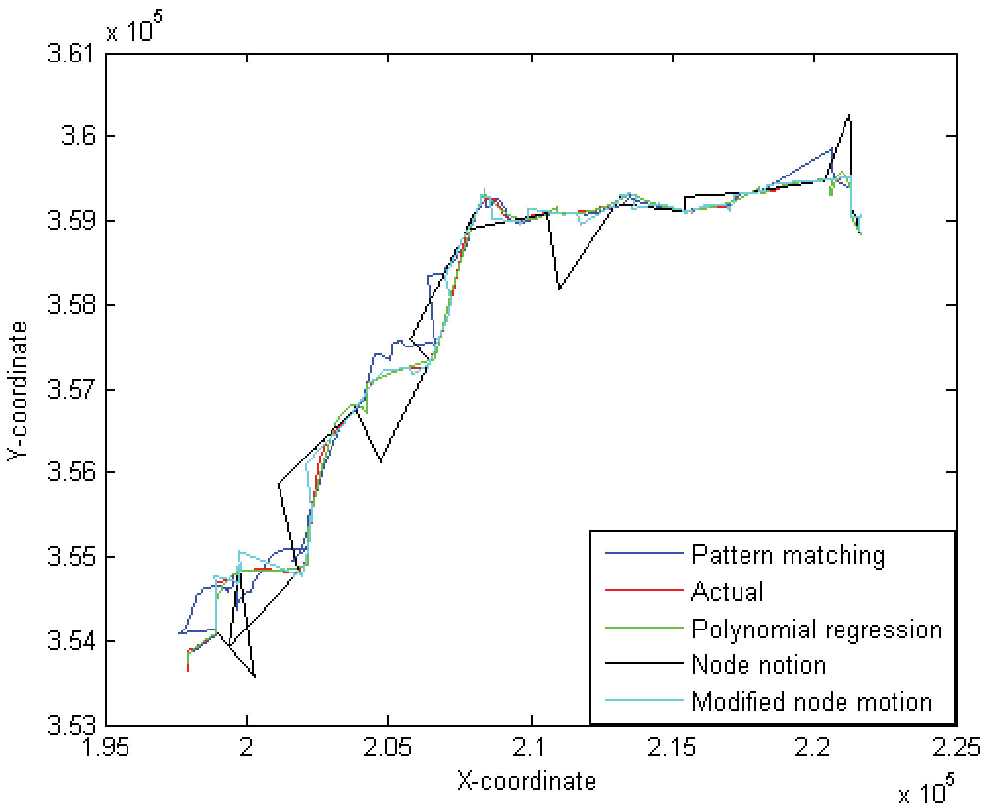

Figure 7.3: Actual vs predicted trajectories for a node using different prediction schemes with the value of the prediction period T equal to 150 seconds.

Since Figures 7.1, 7.2 and 7.3 show the complete path for the node, which covers a large area, a section of each path has been chosen and magnified to give a clear view. Figures 7.4, 7.5 and 7.6 are magnified sections for Figures 7.1, 7.2 and 7.3 respectively which give more details for the predicted trajectories for each prediction scheme.

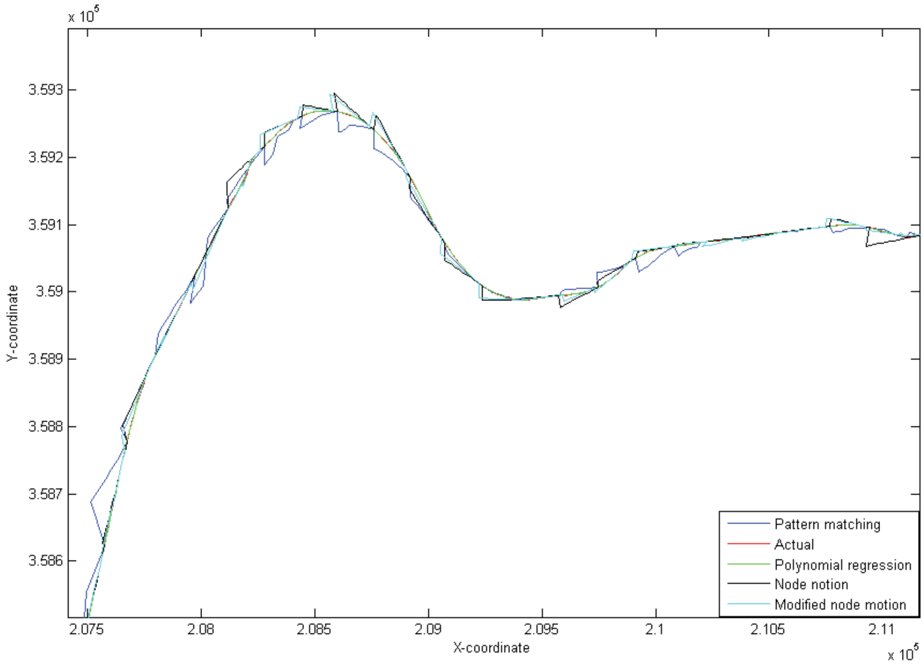

Figure 7.4: a magnified section of Figure 7.1 which gives more detail of the comparison between the actual and the predicted trajectories using T=10 seconds.

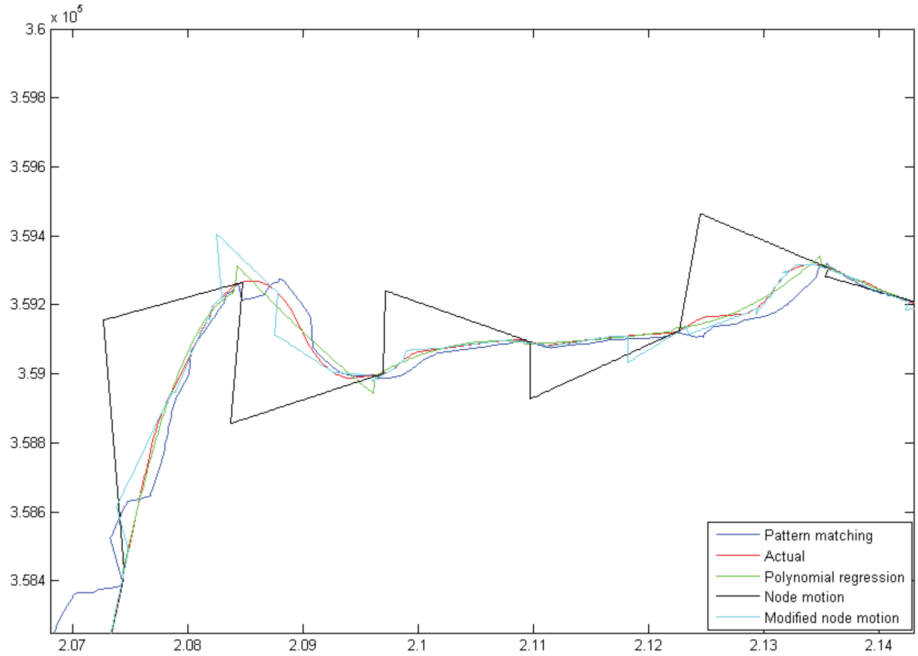

Figure 7.5: a magnified section of Figure 7.2 which gives more detail of the comparison between the actual and the predicted trajectories using T=75 seconds.

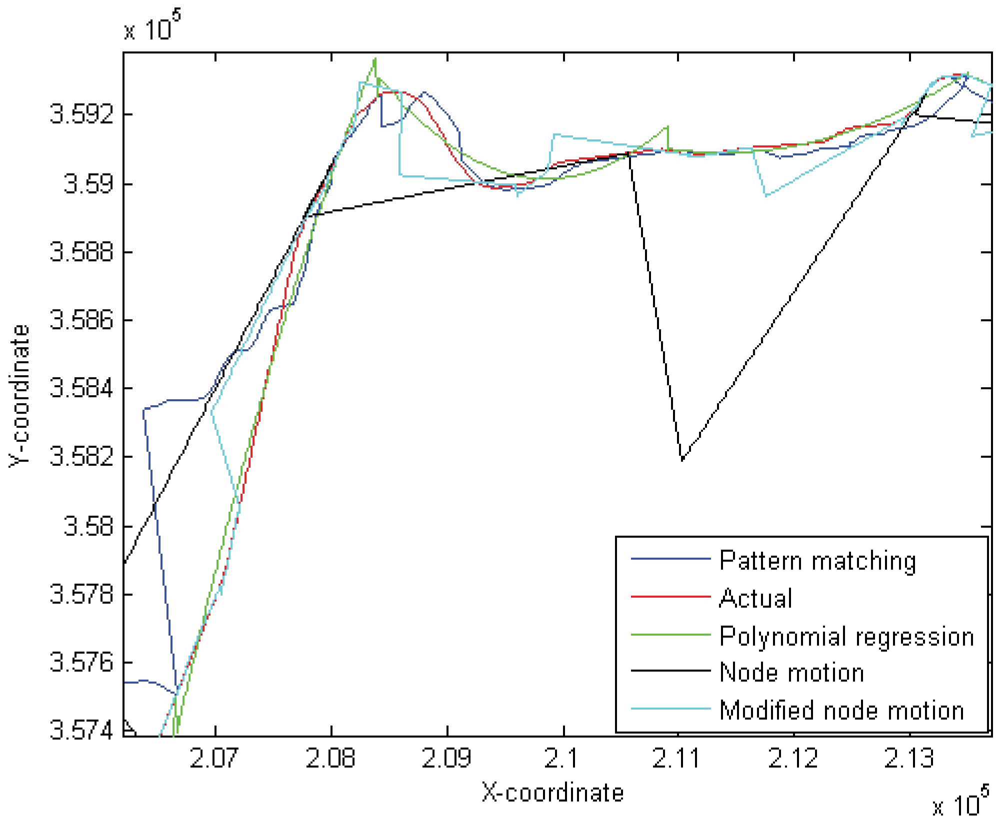

Figure 7.6: a magnified section of Figure 7.3 which gives more details of the comparison between the actual and the predicted trajectories using T=150 seconds.

A quick comparison for the prediction schemes in the above Figures, in addition to many other simulation results obtained through this research are confirming that the polynomial regression prediction scheme and pattern matching prediction scheme are less affected by varying the value of T than the node motion and modified node motion schemes. Moreover, comparing the prediction scheme to the modified prediction scheme shows that the modified prediction scheme is less affected by the increase of the prediction period than the motion prediction scheme; because the modified prediction scheme has two other parameters that positively affect the accuracy of the prediction. These parameters are the threshold angle and the threshold speed, which are the core of the modified prediction scheme. In brief, they are two values in which the node takes an action if its speed and direction changes beyond these values during any update period T by sending an immediate update without waiting for the update period T to finish. Details of the threshold speed and angle are available in section 4.1.

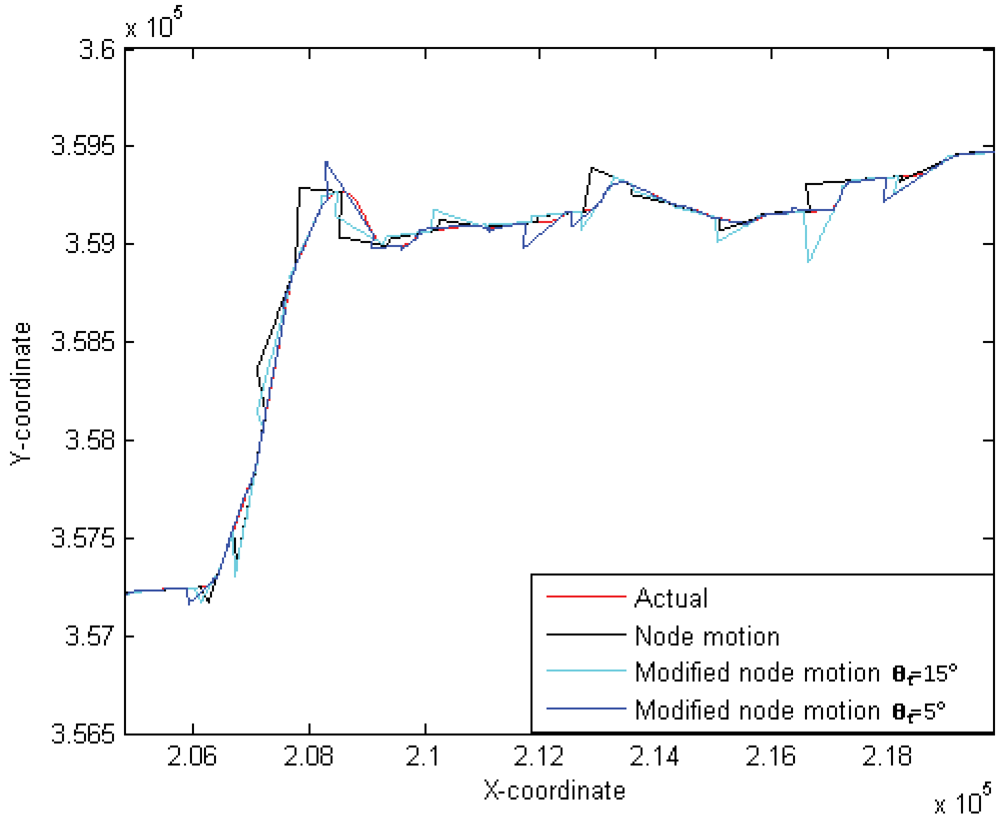

Figure 7.7: a comparison between the actual node motion and modified node motion prediction schemes using T=50 seconds. Two values of threshold angle in the modified node motion scheme are used.

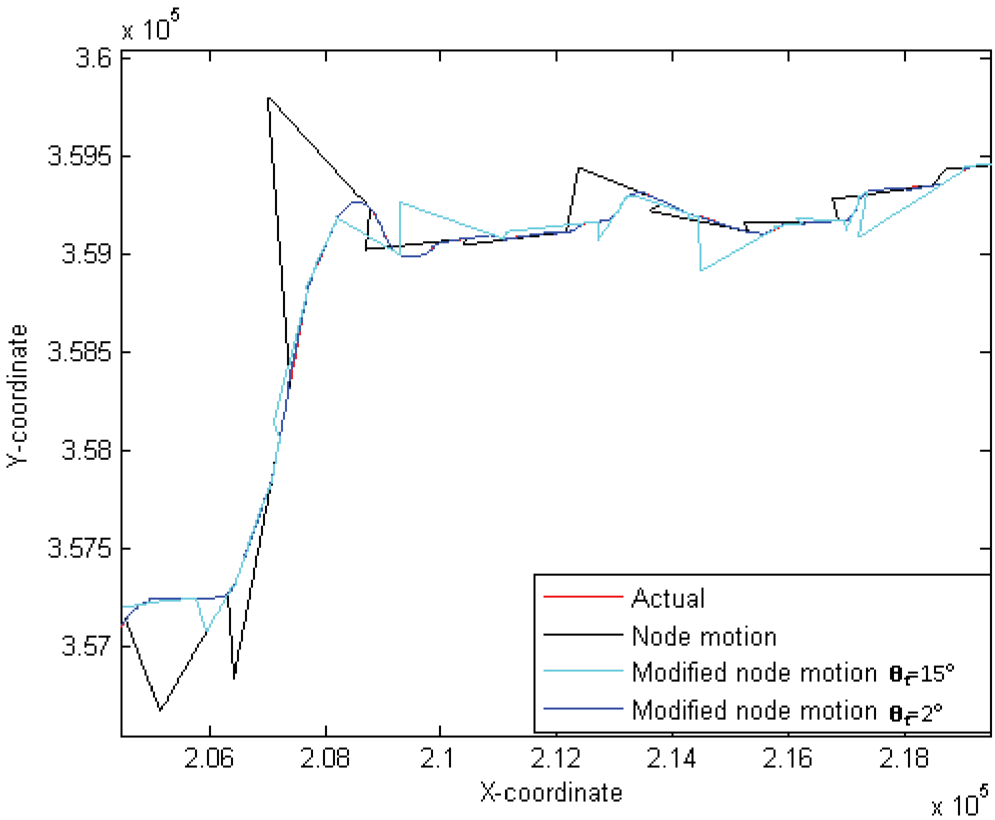

Figure 7.8: a comparison between the actual node motion and modified node motion prediction schemes using T=100 seconds. Two values of threshold angle in the modified node motion scheme are used.

To highlight the effect of the threshold speed and threshold angle, consider Figures 7.7 and 7.8 which show comparisons between the actual and the predicted trajectories of a node using node motion scheme and modified node motion scheme, using two different values of the threshold angle in the modified node motion scheme. In Figure 7.7, the cyan curve represents the predicted trajectory of the node using modified node motion with a threshold angle θ_t=15^0, and the blue curve represents the predicted trajectory of the node using the same prediction scheme with a threshold angle $\boldsymbol{\theta}_t$=5^0. From this Figure, it is clear how a smaller threshold angle achieves better prediction even with a higher value of T, which is set to 50 seconds in this experiment. The threshold angle and speed are determining factors in the modified node motion scheme. They affect the accuracy of the prediction more than the value of T does; for example, in Figure 7.8, the predicted trajectory in the modified node motion scheme where the value of T was set to 100 second and the threshold angle was set to 2^0 is closer to the actual trajectory than the predicted trajectory in Figure 7.7 where T was set to 50 seconds and the threshold angle is 15 degrees.

The polynomial regression prediction scheme is less affected by varying T than node motion schemes. As with the assumption in section 5.1, the longer value of T means that the magnitude of the speed is constant in this period, and this assumption affects the accuracy of the prediction; however, the greater effect of T in this prediction scheme reflects directly on the polynomial that represents the trajectory of the node. Choosing a high value for T means there are more points along the path to fit on the perspective polynomial, as mentioned in section 5.2 which require more regression computation to find a polynomial that fits most of these points. Thus, T still affects the accuracy of the predicted trajectory but not as much as it affects the complexity of the polynomial regression.

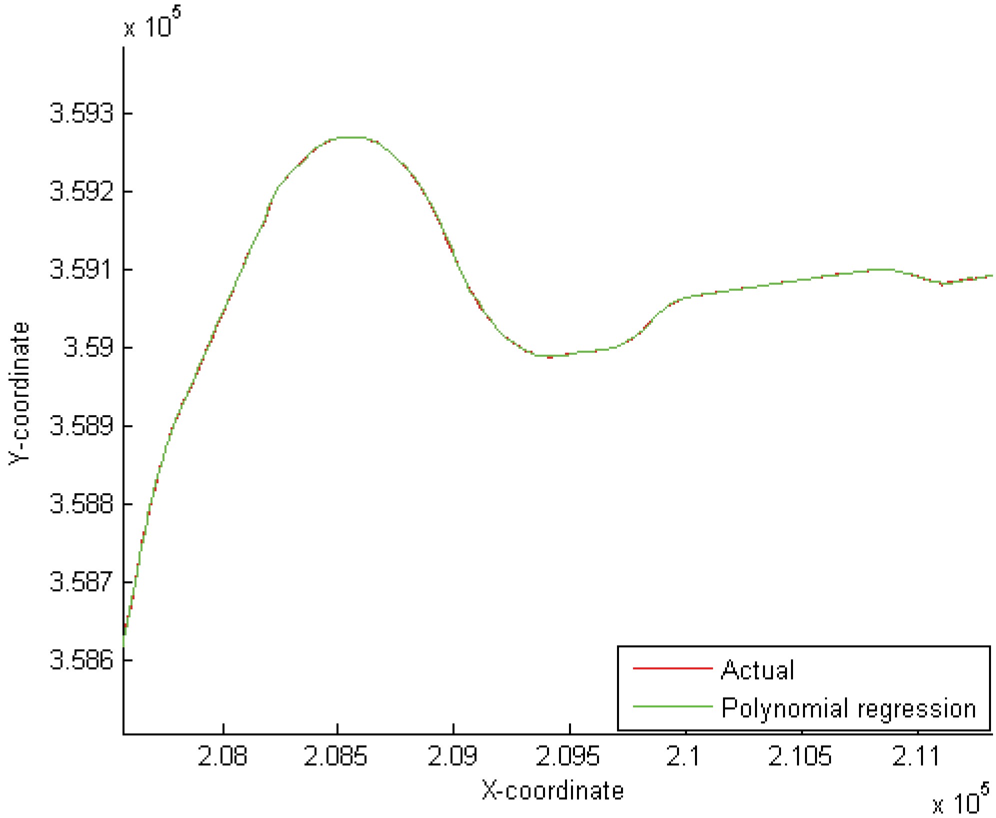

Figure 7.9: a comparison between actual and predicted trajectories for a node using polynomial regression prediction scheme with the value of the prediction period T=10 seconds.

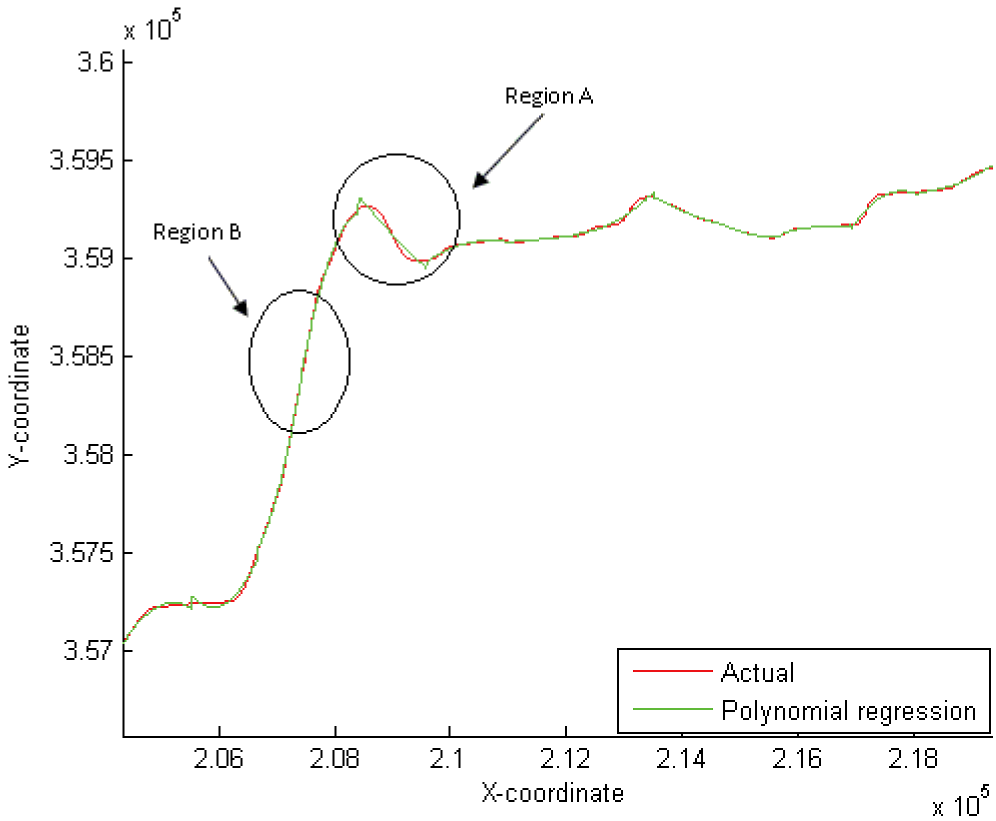

Figure 7.10: a comparison between actual and predicted trajectories for a node using polynomial regression prediction scheme with the value of the prediction period T=75 seconds.

In Figure 7.9 the predicted and actual trajectories are almost identical because the speed is updated more frequently, and the small number of points makes it easer to find the perfect polynomial that fits all these points. However in Figure 7.10, where the value of T is 75 seconds, the speed is assumed to be constant during this period, and there are many more points along the path which makes it more difficult to find a polynomial that fits, especially when there are more curvatures along the path.

The predicted trajectory is closer to the actual one when the path has less curvature, as in region B on Figure 7.10, and is not close to the actual when more curvature exists on the path, as in region A on the same Figure.

Pattern matching is an accurate prediction scheme especially when large quantities of past data are available. It is less affected by the length of the prediction period than all other prediction techniques; however it requires more computational power in the node to search for matching, and a dedicated memory in which to store the past location data.

The length of the predicted future location (PFL) – which is equivalent to the prediction period T in other prediction scheme – affects the accuracy of the prediction. When a match between the current query and the data is found, it is not necessary for the section of the data that follows the matched section to perfectly fit the real trajectory. Matching means that part of the actual trajectory which is the current query and a section of the data are similar; however, this does not guarantee similarity for the following section because the actual measurement is on the matched section and the following section is just a prediction. Therefore, this is a degree of uncertainty in the prediction which gets higher as the required length of PFL gets bigger. Figures 7.11 and 7.12 show how the length of PFL affects the accuracy of the prediction. The actual and predicted trajectory of the node in Figure 7.11 where the length of PFL is 15 converge more than the actual and predicted trajectory in Figure 7.12, which represents the trajectory of the same node setting the length of the prediction PFL to 75.

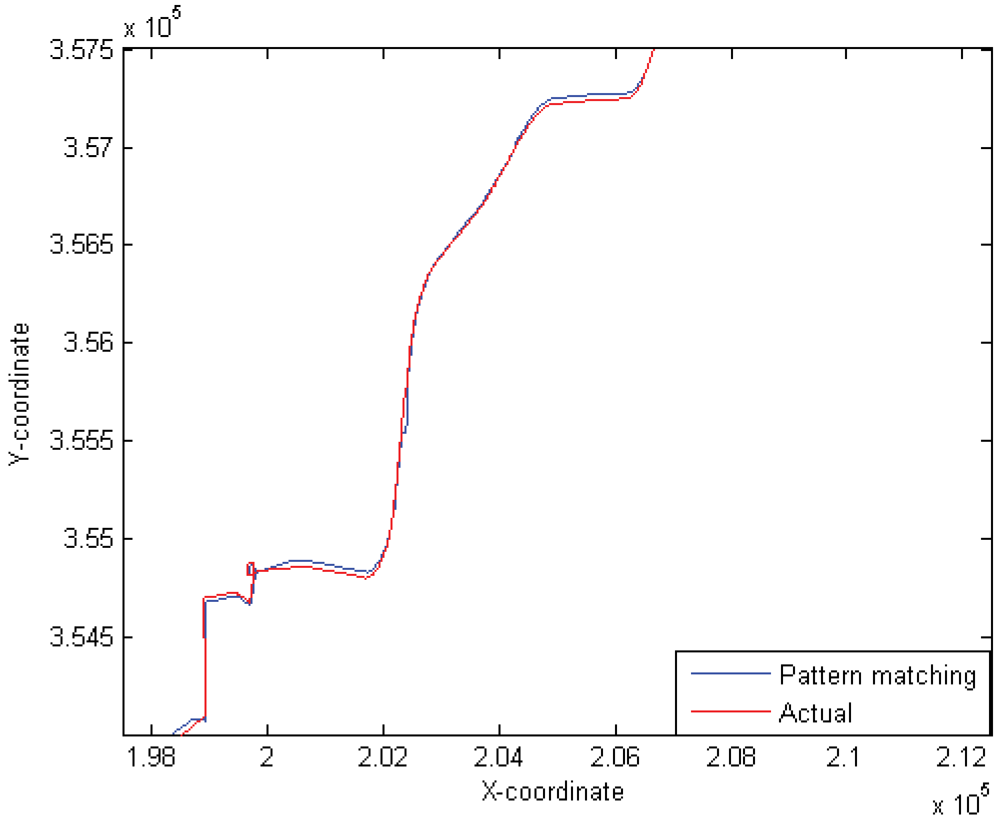

Figure 7.11: a comparison between actual and predicted trajectory for a mobile node using pattern matching scheme with a length of PFL=15.

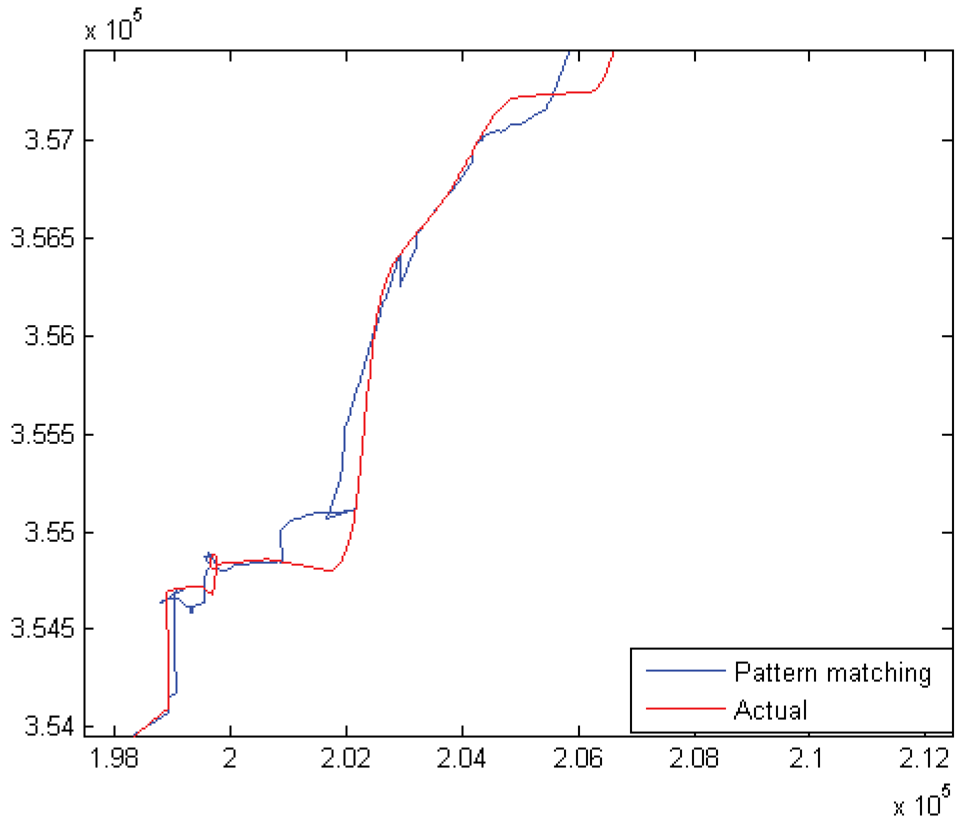

Figure 7.12: a comparison between actual and predicted trajectory for a mobile node using pattern matching scheme with a length of PFL=75.

The availability of more data sets may reduce the effect of varying the value of PFL on the prediction because it gives a higher probability of finding a similarity compared to smaller data sets, and increases the search space for better matching. The comparison between Figure 7.13, where two data sets are used for the search, and Figure 7.14, where ten sets are used, clearly shows that the predicted trajectory in Figure 7.14 is closer to the actual trajectory than that in Figure 7.13 when the same value of PFL length is used in both cases. However, this improvement is at a cost of extra search and more memory size within the node.

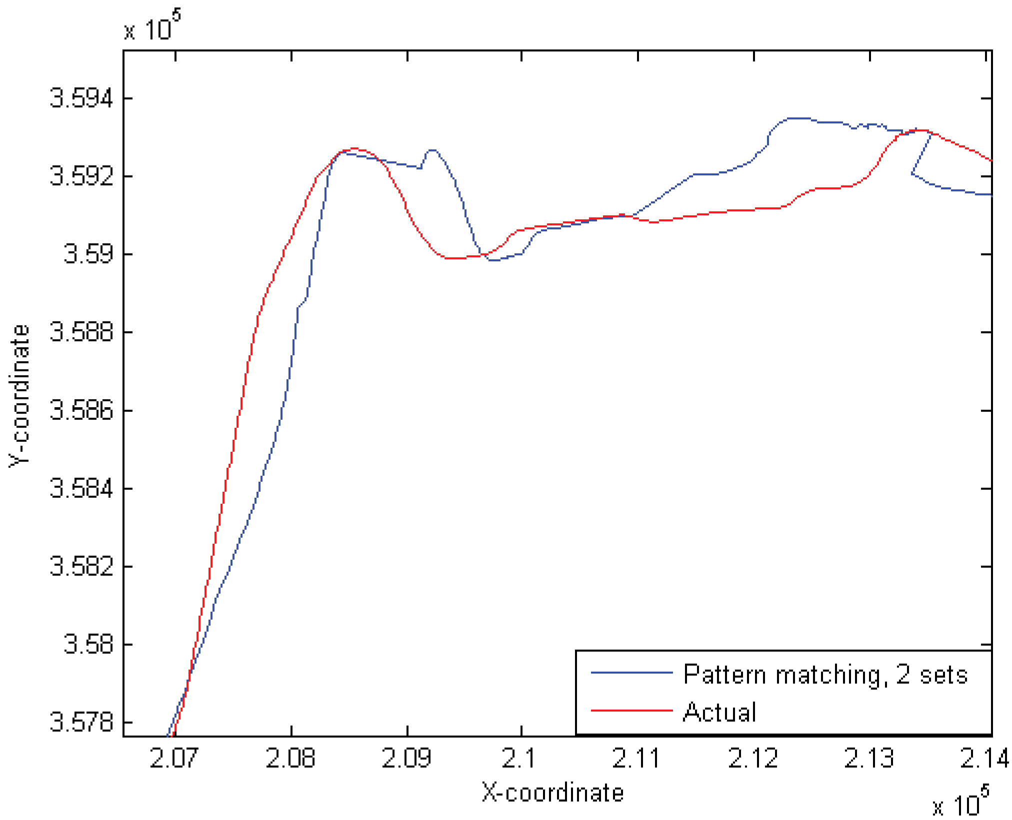

Figure 7.13: a comparison between actual and predicted trajectory for a mobile node using pattern matching scheme with a length of PFL=15 and 2 data sets used for the search.

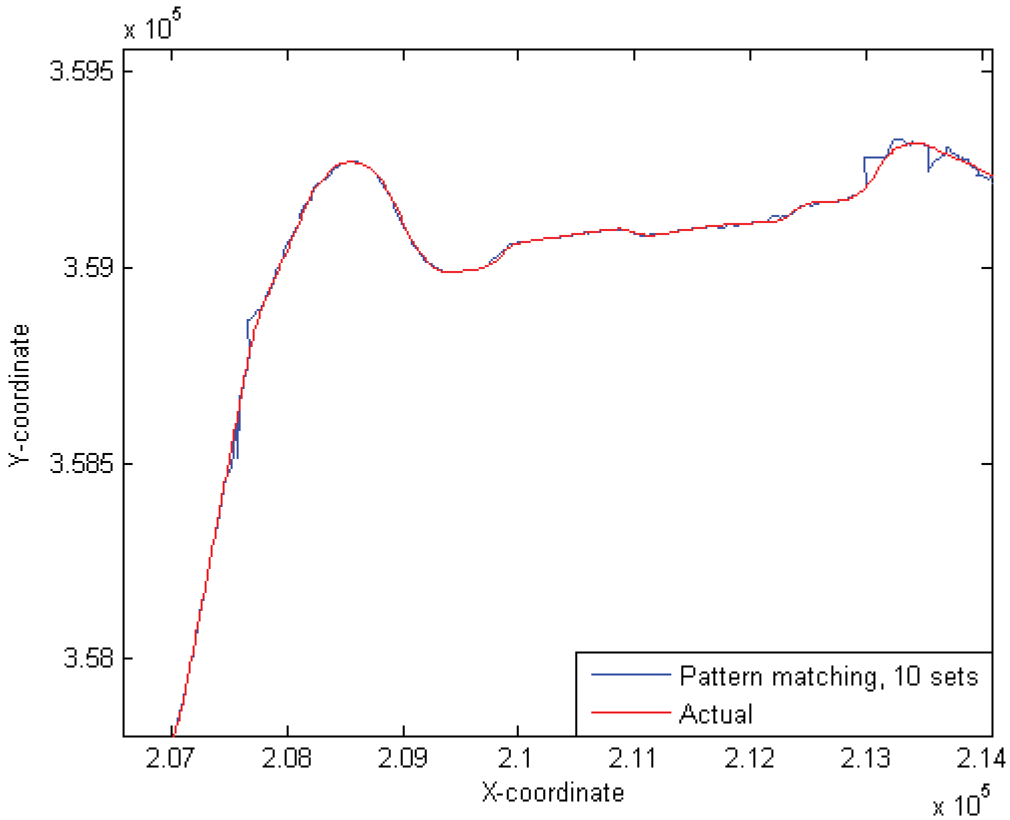

Figure 7.14: a comparison between actual and predicted trajectory for a mobile node using pattern matching scheme with a length of PFL=15 and 10 data sets used for the search.

7.3 Conclusion

In this chapter, a comparison was made between the four prediction schemes presented in previous chapters. The comparison tests the ability of the different schemes to predict the future mobility for a node using the same actual data under the same mobility conditions.

The results show that the accuracy in the node motion prediction scheme is most affected by the length of the prediction period; however, it requires minimal computational power within the node and does not requires any memory. This method can be used in an environment with stable mobility change, such as the mobility of vehicles on a highway.

The modified node motion which is less affected by the length of the prediction period is still simple and does not need any memory. The extra update makes this technique suitable in an environment with more available bandwidth.

Pattern matching and cross correlation are accurate and efficient prediction techniques, although they require high computational power. Pattern matching also requires the existence of memory to store past movements.

Since they are less affected by the length of the prediction period and by changing directions compared to node motion models, these methods are suitable for environments with higher mobility and frequent directional changes.

Chapter 8

8 Conclusions and future work

8.1 Conclusion

In this thesis, we have addressed the effects of the mobility of nodes on the topology of the network, and explained its downsides, including the delay of data delivery associated with route searches and route updates, and the consumption of bandwidth due to these updates. We have presented four novel prediction schemes for mobile ad hoc networks to predict future changes in the mobility of the nodes and to determine their future locations as a time index. The predicted locations were then used to predict the future topologies of the network. The presented schemes are:

- Topology / mobility prediction using equations of motion. This scheme is simple, efficient and does not required extensive computation
- Modified topology / mobility prediction using equations of motion. This scheme is simple, efficient and dealing with the sudden mobility change of the node.
- Topology / mobility prediction using polynomial regression. This method is efficient, and takes into consideration sudden changes in the status of the node's mobility.
- Topology / mobility prediction using pattern matching. This scheme is accurate, efficient and gives a longer prediction period.

The significant features of the proposed schemes are:

- Efficiency: analytical and test results have shown that the proposed schemes are more efficient than traditional algorithms in terms of data delivery and the delay associated with route search.
- Applicability: the proposed schemes are applicable to various routing algorithms including topology-based routing algorithms and geographical

routing algorithms. Moreover, all methods presented in this thesis reside in the upper layer, which gives them the advantage of platform independence.

- Viability: The test results show that the proposed methods are viable and the predicted future trajectories are very similar to the actual trajectories of nodes.

We have argued that the frequent changes in the topology impose a need to find a method to predict changes in order to reduce their impact on the efficiency of the network. The impact includes the delay associated with route updates and the dissipation of the available bandwidth through these updates. We have surveyed many wireless networks and ad hoc networks protocols and find there is a failure to consider topology prediction methods, especially for mobile networks in which the changes in topology are frequent and inevitable.

8.2 Achievements

In Chapter 3 we were inspired by the model of motions for a particle in two dimensional-spaces to develop a prediction scheme that predicts the future mobility of the mobile nodes. This model is computationally simple in that it uses mobility information received from the nodes at each update cycle to produces reliable predicted topologies for the future of mobile networks. Nodes can use this information to route data efficiently, reduce the delay associated with route updates and minimize the dissipation of the bandwidth. Although this model is computationally simple, it does not detect a sudden change in the status of the mobility for a node during the period between two consecutive updates. Therefore, it produces some prediction error that is accumulated until the node receives the next update. The solution for this problem was introduced in Chapter 4 where the algorithm in Chapter 3 was modified to respond to the sudden changes in the mobility of the node. These changes include a sudden change in the direction of movement and/or speed. The node which exhibits any sudden changes in its mobility status broadcasts an immediate update; therefore, the error will not be accumulated. Compared to the solution in Chapter 3, the modified algorithm performs better in terms of mobility and topology prediction with less error and higher performance, and with an extra, immediate update when required. Nevertheless, the algorithm is still considered to be computationally simple and more efficient.

In Chapter 5, we introduced the polynomial regression trajectory prediction scheme to use in the prediction of the future mobility of a dynamic network. We developed a mathematical model based on polynomial regression to find the relation between X and Y coordinates along the trajectory of the mobile node. We derived the relation between both X and Y coordinates and the respective times for the periods of prediction.

Unlike the solutions presented in Chapter 3 and Chapter 4 in which nodes predict the mobility of other nodes, this solution allows the node to predict its own mobility, which is more accurate and does not require any information from the other nodes.

The resultant predictions in both X and Y are functions of time where the future topology can be obtained at any time during the prediction period. The algorithm considers the sudden change in the direction of movement of the mobile node, hence the concept of immediate update does not exist in this prediction algorithm

We have derived several equations that have been used in the design of the algorithm and the choice of the prediction parameters.

The resultant prediction error is small and it does not accumulate along the prediction cycle, as the predicted value at a certain point does not depend on the previous value.

In Chapter 6 we proposed and tested a prediction scheme using pattern matching to predict the future mobility and topology of mobile networks. We assumed that the node exhibits a periodicity in its behaviour in term of mobility. Real location data was used to test the validity of the proposed algorithm which gave promising results. This solution can produce a longer prediction period than all other methods but it requires the use of memory to store the past locations and needs more computations.

Similar to polynomial regression prediction, this solution allows the node to predict its own mobility. The resulting error is very small compared to other solutions and does not accumulate because the predicted value at time t_{i+1} does not depend directly on the predicted value at time t_i. It depends only on the length of the predicted future locations

8.3 Future research

In this thesis, we have presented and individually tested the validity of three predictions schemes. In the future work, we intend to merge them into one prediction schemes to enable the node to:

- Compare the resultant predictions for optimization purposes and to choose the best predictions among them.
- Use the three schemes as backup to each other in case of the failure of one scheme under any circumstance.

References

[1] C. R. Dow, P. J. Lin, S. C. Chen, J. H. Lin, and S. F. Hwang, "A study of recent research trends and experimental guidelines in mobile ad-hoc network," in *Proceedings of 19th International Conference on Advanced Information Networking and Applications*, 2005, pp. 72-77.

[2] C. Liu and J. Kaiser, "A survey of mobile ad hoc network routing protocols," Carleton University, Systems and Computer Engineering, Technical Report SCE-04-14, 2003.

[3] S. Sesay, Z. Yang, and J. He, "A survey on mobile ad hoc wireless network," *Information Technology Journal,* vol. 3, pp. 168-175, 2004.

[4] L. Qin and T. Kunz, "Survey on mobile ad hoc network routing protocols and cross-layer design," Carleton University, Systems and Computer Engineering, Technical Report SCE-04-14, 2004.

[5] M. Mohsin and R. Prakash, "IP address assignment in a mobile ad hoc network," in *Proceedings of Military Communications Conference*, 2002, pp. 856-861.

[6] R. Chellappa-Doss, A. Jennings, and N. Shenoy, "A review of current mobility prediction techniques for ad hoc networks," in *Proceedings of IASTED International Multi-Conference, Wireless and Optical Communications*, 2004, pp. 536-542.

[7] D. Gavalas, C. Konstantopoulos, B. Mamalis, and G. Pantziou, "Mobility prediction in mobile ad-hoc networks " in *Encyclopedia of Next Generation Mobile Networks and Ubiquitous Computing*, S. Pierre, Ed.: Ecole Polytechnique ed Montreal: Montreal, Canada, 2010, pp. 226-240.

[8] A. Nasipuri and S. R. Das, "On-demand multipath routing for mobile ad hoc networks," in *Proceedings of International Conference on Computer Communications and Networks*, 1999, pp. 64-70.

[9] M. Al-Hattab and J. I. Agbinya, "Planning of wireless mesh networks," in *Planning and Optimisation of 3G and 4G Wireless Networks*. vol. 8, J. I. Agbinya, Ed.: River, Aalborg, 2010, pp. 201-222.

[10] E. M. Royer and C.-K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Personal Communications,* vol. 6, pp. 46-55, 1999.

[11] M. Talati, S. Valiveti, and K. Kotecha, "Trust based routing in ad hoc network," in *Communication and Networking*. vol. 120, T.-h. Kim, T. Vasilakos, K. Sakurai, Y. Xiao, G. Zhao, and D. Ślęzak, Eds.: Springer: Berlin Heidelberg, 2010, pp. 381-392.

[12] K. Farkas, T. Hossmann, L. Ruf, and B. Plattner, "Pattern matching based link quality prediction in wireless mobile ad hoc networks," in *Proceedings of the 9th ACM International Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems*, 2006, pp. 239-246.

[13] W. Su, S. J. Lee, and M. Gerla, "Mobility prediction in wireless networks," in *Proceedings of 21st Century Military Communications Conference*, 2000, pp. 491-495.

[14] 2006, "Real trajectory data," Accessed on 15/09/2010. Available: http://www.cs.uic.edu/~boxu/mp2p/gps_data.html.

[15] M. Al-Hattab and J. I. Agbinya, "Topology prediction and convergence for networks on mobile vehicles," in *Proceedings of International Conference on Computer and Communication Engineering*, 2008, pp. 266-269.

[16] B. H. Sathyaraj and R. C. Doss, "Route maintenance using mobility prediction for mobile ad hoc networks," in *Proceedings of IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, 2005, pp. 96-101.

[17] P. S. Prasad and P. Agrawal, "Mobility prediction for wireless network resource management," in *Proceedings of 41st Southeastern Symposium on System Theory*, 2009, pp. 98-102.

[18] I. Chlamtac, M. Conti, and J. J.-N. Liu, "Mobile ad hoc networking: imperatives and challenges," *Ad Hoc Networks,* vol. 1, pp. 13-64, 2003.

[19] G. Jayakumar and G. Gopinath, "Ad Hoc Mobile Wireless Networks Routing Protocols – A Review," *Journal of Computer Science,* vol. 3, pp. 574-582, 2007.

[20] C. Cordeiro, H. Gossain, and D. P. Agrawal, "Multicast over wireless mobile ad hoc networks: present and future directions," *IEEE Network,* vol. 17, pp. 52-59, 2003.

[21] P. Kuosmanen. *Classification of Ad Hoc Routing Protocols*. Available: http://www.netlab.tkk.fi/opetus/s38030/k02/Papers/12-Petteri.pdf

[22] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks," *Ad Hoc Networks,* vol. 2, pp. 1-22, 2004.

[23] R. C. Shah and J. M. Rabaey, "Energy aware routing for low energy ad hoc sensor networks," in *Proceedings of Wireless Communications and Networking Conference*, 2002, pp. 350-355.

[24] M. Mauve and J. Widmer, "A survey on position-based routing in mobile ad-hoc networks," *IEEE Network,* vol. 15, pp. 30-39, 2001.

[25] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks " in *Mobile Computing*. vol. 353, T. Imielinski and H. Korth, Eds.: Kluwer Academic Publishers, 1996, pp. 153-181.

[26] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," in *proceedings of Second IEEE Workshop on Mobile Computing Systems and Applications*, 1999, pp. 90-100.

[27] S. R. Das, R. Castaneda, and J. Yan, "Simulation-based performance evaluation of routing protocols for mobile ad hoc networks," *Mobile Networks and Applications,* vol. 5, pp. 179-189, 2000.

[28] C. E. Perkins and P. Bhagwat, "Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers," in *Proceedings of the*

conference on communications architectures, protocols and applications, 1994, pp. 234-244.

[29] P. Jacquet and T. Clausen, "RFC-3626 Optimized Link State Routing Protocol (OLSR)," IETF Internet Draft, Available: http://www.ietf.org/rfc/rfc3626.txt, 2003

[30] R. Ogier, F. Templin, and M. Lewis, "RFC-3684 Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)," IETF Internet Draft, Available: http://www.rfc-editor.org/rfc/rfc3684.txt, 2004.

[31] L. M. Feeney, "A taxonomy for routing protocols in mobile ad hoc networks," Swedish Institute of Computer Science, Technical Report T99/07, 1999.

[32] Z. J. Haas and M. R. Pearlman, "The performance of query control schemes for the zone routing protocol," *IEEE/ACM Transactions on Networking,* vol. 9, pp. 427-438, 2001.

[33] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, 2000, pp. 243 - 254

[34] Y.-B. Ko and N. H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," *Wireless Networks,* vol. 6, pp. 307-321, 2000.

[35] W. Liu, C. Chiang, H. Wu, and C. Gerla, "Routing in clustered multihop mobile wireless networks with fading channel," in *Proceedings of The IEEE Singapore International Conference on Networks*, 1997, pp. 197-211.

[36] R. A. Santos, R. M. Edwards, and A. Edwards, "Cluster-based location routing algorithm for inter-vehicle communication," in *Proceedings of IEEE 60th Vehicular Technology Conference*, 2004, pp. 914-918.

[37] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized Link State Routing Protocol (OLSR) for ad hoc networks," in

Proceedings of 2001 IEEE International Multi Topic Conference, Technology for the 21st Century, 2001, pp. 62 - 68

[38] G. Dommety and R. Jain, "Potential networking applications of global positioning systems (GPS)," CS Department, The Ohio State University, Technical Report TR-24, 1996.

[39] E. D. Kaplan, *Understanding GPS: principles and applications*, second ed. Boston: Artech House, 2006.

[40] S. Capkun, M. Hamdi, and J. P. Hubaux, "GPS-free positioning in mobile ad-hoc networks," *Cluster Computing,* vol. 5, pp. 157-167, 2002.

[41] H. Akcan, V. Kriakov, H. Bronnimann, and A. Delis, "GPS-Free node localization in mobile wireless sensor networks," in *Proceedings of the 5th ACM International workshop on Data Engineering for Wireless and Mobile Access*, 2006, pp. 35-42.

[42] S. Basagni, I. Chlamtac, V. Syrotiuk, and B. Woodward, "A distance routing effect algorithm for mobility (DREAM) " in *Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking* 1998, pp. 76-84.

[43] S.-C. Woo and S. Singh, "Scalable routing protocol for ad hoc networks," *Wireless Networks,* vol. 7, pp. 513-529, 2001.

[44] S. Basagni, "Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks," in *Proceedings of 1999 - Fall. IEEE 50th Vehicular Technology Conference* 1999, pp. 889-893.

[45] Y. Ko and N. Vaidya, "Using location information to improve routing in ad hoc networks," Texas A & M University, Technical Report 97-013, 1997.

[46] V. Kumar and S. R. Das, "Performance of dead reckoning-based location service for mobile ad hoc networks: Research Articles," *Wireless Communications & Mobile Computing,* vol. 4, pp. 189-202, 2004.

[47] S. Marwaha, T. Chen Khong, and D. Srinivasan, "Mobile agents based routing protocol for mobile ad hoc networks," in *Proceedings of IEEE Global Telecommunications Conference*, 2002, pp. 163-167.

[48] R. R. Choudhury, S. Bandyopadhyay, and K. Paul, "Topology discovery in ad hoc wireless networks using mobile agents," in *Proceedings of the Second International Workshop on Mobile Agents for Telecommunication Applications*, 2000, pp. 1-16.

[49] S. Marwaha, T. Chen Khong, and D. Srinivasan, "A novel routing protocol using mobile agents and reactive route discovery for ad hoc wireless networks," in *Proceedings of 10th IEEE International Conference on Networks*, 2002, pp. 311-316.

[50] N. Migas, W. J. Buchanan, and K. A. Mcartney, "Mobile agents for routing, topology discovery, and automatic network reconfiguration in ad-hoc networks," in *Proceedings of 10th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems*, 2003, pp. 200-206.

[51] R. Chandra, C. Fetzer, and K. Högstedt, "Adaptive topology discovery in hybrid wireless networks," Available: http://research.microsoft.com/pubs/73487/topology.pdf, 2002.

[52] G. Liu and G. Maguire, "A class of mobile motion prediction algorithms for wireless mobile computing and communications," *Mobile Networks and Applications,* vol. 1, pp. 113-121, 1996.

[53] F. Erbas, J. Steuer, D. Eggesieker, K. Kyamakya, and K. Jobinann, "A regular path recognition method and prediction of user movements in wireless networks," in *Proceedings of 2001- Fall IEEE 54th Vehicular Technology Conference*, 2001, pp. 2672-2676.

[54] J. Chan, S. Zhou, and A. Seneviratne, "A QoS adaptive mobility prediction scheme for wireless networks," in *Proceedings of IEEE Global Telecommunications Conference*, 1998, pp. 1414-1419.

[55] R. Chellappa-Doss, A. Jennings, and N. Shenoy, "A review on current work in mobility prediction for wireless networks " in *Proceedings of 3rd Asian International Mobile Computing Conference* 2004, pp. 1-7.

[56] R. Chellapa, A. Jennings, and N. Shenoy, "A comparative study of mobility prediction in fixed wireless networks and mobile ad hoc networks," in *Proceeding of IEEE International Conference on Communications*, 2003, pp. 891-895.

[57] P. Basu, N. Khan, and T. D. C. Little, "A mobility based metric for clustering in mobile ad hoc networks," in *Proceedings of International Conference on Distributed Computing Systems Workshop*, 2001, pp. 413-418.

[58] Q. Huang, S. Chan, and M. Zukerman, "Improving handoff QoS with or without mobility prediction," *Electronics Letters,* vol. 43, pp. 534-535, 2007.

[59] N. Meghanathan, "Location prediction based routing protocol for mobile ad hoc networks," in *Proceedings of IEEE Global Telecommunications Conference*, 2008, pp. 569-573.

[60] Z. R. Zaidi and B. L. Mark, "Mobility estimation for wireless networks based on an autoregressive model," in *Proceedings of IEEE Global Telecommunications Conference*, 2004, pp. 3405-3409.

[61] Z. R. Zaidi and B. L. Mark, "Real-time mobility tracking algorithms for cellular networks based on Kalman filtering," *IEEE Transactions on Mobile Computing,* vol. 4, pp. 195-208, 2005.

[62] G. Eshel, "The Yule Walker Equations for the AR Coefficients," University of South Carolina, Accessed on 1/10/2010. Available: http://www.stat.sc.edu/~vesselin/STAT520_YW.pdf.

[63] G. Welch and G. Bishop, "An introduction to the Kalman filter," University of North Carolina at Chapel Hill, Technical Report TR 95-041, 2006.

[64] T. Liu, P. Bahl, and I. Chlamtac, "Mobility modeling, location tracking, and trajectory prediction in wireless ATM networks," *IEEE Journal on Selected Areas in Communications,* vol. 16, pp. 922-936, 1998.

[65] K. Farkas, T. Hossmann, F. Legendre, B. Plattner, and S. K. Das, "Link quality prediction in mesh networks," *Computer Communications,* vol. 31, pp. 1497-1512, 2008.

[66] P. Santi, *Topology control in Wireless Ad Hoc and Sensor Networks*: John Wiley & sons, 2005.

[67] P. Santi, "Topology control in wireless ad hoc and sensor networks," *ACM Computing Surveys (CSUR),* vol. 37, pp. 164-194, 2005.

[68] R. Rajaraman, "Topology control and routing in ad hoc networks: a survey," *SIGACT News,* vol. 33, pp. 60-73, 2002.

[69] T. S. Rappaport, *Wireless Communications: Principles and Practice* second ed.: Prentice Hall PTR, 2002.

[70] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," in *Proceedings of Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, 2000, pp. 404-413.

[71] Z. Mir, D. Shrestha, G.-H. Cho, and Y.-B. Ko, "Mobility-aware distributed topology control for mobile multi-hop wireless networks," in *Information Networking. Advances in Data Communications and Wireless Networks*. vol. 3961, I. Chong and K. Kawahara, Eds.: Springer Berlin / Heidelberg, 2006, pp. 257-266.

[72] H. Hurskainen, J. Raasakka, and J. Nurmi, "Specification of GNSS application for multiprocessor platform," presented at the International Symposium on System-on-Chip, 2008, Tampere, Finland, 2008.

[73] J. Fulton, "Global Positioning System (GPS) Technology An Explanation of Global Navigation Satellite System (GNSS)," Alabama Cooperative Extension System, Technical Report ANR-1352, 2009.

[74] E. Kaplan, *Understanding GPS: principles and applications*, Second ed. Boston: Artech House, 2006.

[75] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE, Personal Communications,* vol. 7, pp. 28-34, 2000.

[76] S. M. Das, H. Pucha, and Y. C. Hu, "Performance comparison of scalable location services for geographic ad hoc routing," in *Proceedings of IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.* , 2005, pp. 1228-1239.

[77] P. Hsiao, "Geographical region summary service for geographical routing," in *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, 2001, pp. 263-266.

[78] A. Agarwal and S. R. Das, "Dead reckoning in mobile ad hoc networks," in *Proceedings of IEEE Wireless Communications and Networking*, 2003, pp. 1838-1843.

[79] Z. J. Haas and B. Liang, "Ad Hoc mobility management with uniform quorum systems," *IEEE/ACM Transactions on Networking,* vol. 7, pp. 228-240, 1999.

[80] J. Li, J. Jannotti, D. S. J. D. Couto, D. R. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, 2000, pp. 120-130.

[81] I. Stojmenovic and B. Vukojevic, "A routing strategy and quorum based location update scheme for ad hoc wireless networks," SITE, University of Ottawa, Technical Report TR-99-09, 1999.

[82] S. Corson and J. Macker, "RFC 2501-Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," IETF Internet Draft. Accessed on 5/10/2006. Available: http://www.ietf.org/rfc/rfc2501.txt.

[83] M. Gerla, L.-J. Chen, Y.-Z. Lee, B. Zhou, J. Chen, G. Yang, and S. Das, "Dealing with node mobility in ad hoc wireless network," *Lecture Notes in Computer Science,* vol. 3465, pp. 69-106, 2005.

[84] B.-C. Seet, G. Liu, B.-S. Lee, C.-H. Foh, K.-J. Wong, and K.-K. Lee, "A-STAR: A mobile ad hoc routing strategy for metropolis vehicular communications," in *NETWORKING 2004, Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications*. vol. 3042, N. Mitrou, K. Kontovasilis, G. N. Rouskas, I. Iliadis, and L. Merakos, Eds.: Springer Berlin / Heidelberg, 2004, pp. 989-999.

[85] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing,* vol. 2, pp. 483-502, 2002.

[86] C. Bettstetter, "Mobility Modeling in Wireless Networks: Categorization, Smooth Movement, and Border Effects " *Mobile Computing and Communications Review,* vol. 5, pp. 55-67, 2001.

[87] E. M. Royer, P. Michael Melliar-smith, and L. Moser, "An analysis of the optimum node density for ad hoc mobile networks," in *Proceedings of the IEEE International Conference on Communications*, 2001, pp. 857-861.

[88] RTA Traffic and Transport Directorate, "Traffic Volume Data for Sydney Region 2002,". Available: http://www.rta.nsw.gov.au/trafficinformation/downloads/aadt_data_files/aadtsydney2002_i.pdf.

[89] C. H. Edwards and D. E. Penney, *Calculus and Analytic Geometry*, Third ed.: Prentice-Hall International, 1990.

[90] A. Baggio and K. Langendoen, "Monte-carlo localization for mobile wireless sensor networks," *Ad-hoc Networks,* vol. 6, pp. 718-733, 2008.

[91] R. S. Wolff and G. Dawra, "A Terrain-based Routing Protocol for Sparse Ad-hoc Intermittent Networking (TRAIN)," presented at the International Conference on Wireless and Optical Communication Systems and Networks, Banff, Alberta, Canada, 2006.

[92] A. Spyropoulos and C. S. Raghavendra, "Energy efficient communications in ad hoc networks using directional antennas," in *Proceedings of Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, 2002, pp. 220-228.

[93] 2009, "http://www.skyhookwireless.com," [online]. Accessed on: 10/01/2009, Available: http://www.skyhookwireless.com.

[94] S. Keshav, *An Engineering Approach to Computer Networking: ATM Networks, the Internet, and other Telephone Network*: Addison Wisley, 1997.

[95] J. F. Kurose and K. W. Ross, *Computer Networking : A Top-down Approach Featuring the Internet*: Addison Wesley, 2003.

[96] M. Maroti, P. Volgyesi, S. Dora, B. Kusy, A. Nadas, A. Ledeczi, G. Balogh, and K. Molnar, "Radio interferometric geolocation," presented at the The 3rd international conference on Embedded networked sensor systems, San Diego, California, USA, 2005.

[97] S. Woo and S. Singh, "Scalable routing protocol for ad hoc networks," in *Wireless Networks*. vol. 7: Kluwer Academic Publishers, 2001, pp. 513-529.

[98] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, 2000, pp. 243 - 254.

[99] N. Yaakob, F. Anwar, S. Zeldi, and A. H. Abdalla, "Investigating mobile motion prediction in supporting seamless handover for high speed mobile

node," in *Proceedings of International Conference on Computer and Communication Engineering*, 2008, pp. 1260-1263.

[100] C. L. Wu, K. W. Chau, and C. Fan, "Prediction of rainfall time series using modular artificial neural networks coupled with data-preprocessing techniques," *Journal of Hydrology,* vol. 389, pp. 146-167, 2010.

[101] H. Roosma, "Prediction of stock market turning points," *International Advances in Economic Research,* vol. 1, pp. 272-277, 1995.

[102] A. S. Kulkarni, "Application of neural network to stock market prediction," 1996. Available: http://machine-learning.martinsewell.com/ann/Kulk96.pdf.

[103] P. Bourke, "Cross Correlation," 1996. Accessed on 15/07/2010. Available: http://local.wasp.uwa.edu.au/~pbourke/miscellaneous/correlate/