

REALIZING QUALITY-OF-SERVICE WITH LINK LIFETIME PREDICTION IN
MOBILE AD HOC NETWORKS

A Dissertation

Presented to the Faculty of the Graduate School
of Cornell University

In Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

by

Edward Yi Hua

August 2009

© 2009 Edward Yi Hua

REALIZING QUALITY-OF-SERVICE WITH LINK LIFETIME PREDICTION IN MOBILE AD HOC NETWORKS

Edward Yi Hua, Ph. D.

Cornell University 2009

Recent research in mobile ad hoc networks (MANETs) has been studying the feasibility of provisioning the Quality-of-Service in such a network. One major factor that makes this a difficult task is the node mobility, which induces a dynamic network topology and makes a multi-hop path susceptible to abrupt breakage when any of its constituent links breaks. It is therefore desirable to acquire predictive knowledge of the path lifetime, which reflects its reliability and is closely associated with the lifetime of each constituent link.

We first study how node mobility impacts the lifetime of a path in the MANET, and propose three algorithms that employ link age as the decision parameter to select the best path from all available paths between the sender (source) and recipient (destination) of data for transmissions. Performance evaluation results show that these path-selection algorithms achieve a robust performance in choosing a path that meets a specified path-lifetime requirement.

We next propose the Mobile-projected Trajectory (MPT) algorithm that employs four periodically measured distances between two nodes of a link to compute the relative movement trajectory and the remaining lifetime of a link. This algorithm is based on linear curve fitting, and does not require any knowledge of node position, speed, and direction. To account for the frequent occurrences of velocity change during the link lifetime, we propose a simple velocity-change detection test that is augmented to the MPT. The new algorithm, Mobile-Projected Trajectory with

Velocity Change Detection, improves the performance of the algorithm by re-computing the remaining link lifetime more accurately after a velocity change is detected.

We also propose a novel link lifetime prediction algorithm based on the Unscented Kalman Filter, which casts the link lifetime as a non-linear dynamic system model, and recursively estimates its states to compute the remaining link lifetime. Performance evaluation of the proposed algorithm demonstrates robust performance to estimate the link lifetime while nodes move along various trajectories induced by velocity changes.

BIOGRAPHICAL SKETCH

Edward Yi Hua was born in Shanghai, and spent his childhood in both Shanghai and Los Angeles. He graduated with honors from University of California, Irvine, in 1998, and subsequently decided to pursue a Master's degree at Princeton University despite the hi-tech industry boom at the time. After earning the Master's in 1999, Ed decided to stay in the East Coast and worked at Lucent Technologies in Whippany, New Jersey, where he was a Member of Technical Staff-I in the Wireless Networks Group. He was involved in the testing and deployment of both base-station hardware and software for the IS136 North American TDMA Standard, and wrote a technical manual on the use of AutoPACE, a Lucent-developed software data-analysis tool that collected live data from the field and turned them into a number of metrics to evaluate the performance of the base station. In 2001, with the downturn in economy, Ed chose to return to academia and enrolled in the Ph.D. program at Cornell, under the supervision of Dr. Zygmunt J. Haas.

At his leisure time, Ed is a classically trained pianist, and has since 2002 released two piano solo CD albums in limited edition (i.e., among family and friends). He also enjoys giving concerts through Cornell Piano Society. He sporadically maintains an on-line movie review column and enjoys cooking, among other interests and hobbies.

ACKNOWLEDGMENTS

First of all, I would like to express my gratitude to Professor Zygmunt J. Haas, my research advisor who has contributed so much during my Ph.D. program. Professor Haas has always provided me with candid and honest advice to guide me through various stages of my research progress. At times when I seemed to be at the wits' end, he would always point out a new direction to get the progress moving forward again. Moreover, he has always looked after my interests and acted on my behalf when there are administrative commitments to be fulfilled with the Graduate School. I am indebted to his guidance, without which I would not have completed this dissertation and brought my Ph.D. study to a closure.

I would also like to thank Professor Stephen B. Wicker and Professor David Shmoys for serving on my committee. They have given me helpful suggestions to improve my research work both within and outside of this dissertation. During my A and B exams, they asked very insightful questions and provided comments that helped set the next stages of the research and revise the dissertation.

My thanks also goes to Professor Kevin A. Tang, who graciously agreed to serve as proxy to Professor Wicker on my B exam committee. His participation on my committee prevented a further delay in scheduling my dissertation defense, and for that I am truly grateful.

A special Thank You goes to Professor Terrence L. Fine, who offered me Teaching Assistantships during the later year of my Ph.D. program that provided me with much needed financial resources to continue with my studies. Over the last year and a half, I have come to see Professor Fine as more than just a distinguished

academician, but also as a friend.

My hats off also to all the administrative staff at the ECE department and the Graduate School who made my life so much easier by helping me navigate through the myriad administrative necessities: Sue Bulkley, Scott Coldren, Kimberly Cotton, Karen Crane, Jamie Del Cero, Barbara Edinger, Sutapa Gosh, Susan Ferrara, Lisa Gould, and Pattie Place.

Above all, I am most indebted to my parents, Richard and Catherine Hua, for their love and unwavering support. They are always there for me in times of need, and have never lost their faith in me, even when I seemed to have lost the sight of the end goal. Their encouragement has been a main source of motivation that has kept me on course. They are truly the best parents in the world that anyone can ever hope for. Furthermore, I want to express my gratitude to all members of my family and friends, who have been caring, encouraging and supportive over the years.

Finally, I would like to acknowledge that my Ph.D. research work was supported in part by the National Science Foundation under the grants ANI-0329905 and CNS-0626751 and by the Air Force Office of Scientific Research under contract FA9550-09-1-0121/Z806001.

TABLE OF CONTENTS

| | |
|--|---------------|
| BIOGRAPHICAL SKETCH..... | III |
| ACKNOWLEDGMENTS..... | IV |
| LIST OF FIGURES | VIII |
| LIST OF TABLES | X |
| CHAPTER 1 INTRODUCTION..... | - 1 - |
| 1.1 OVERVIEW OF EXISTING WIRELESS NETWORK PARADIGMS | - 1 - |
| 1.2 INTRODUCTION TO MOBILE AD HOC NETWORKS | - 5 - |
| 1.3 NODE MOBILITY AND PATH LIFETIME IN MANET..... | - 8 - |
| CHAPTER 2 MOBILITY AND PATH LIFETIME IN MANET..... | - 13 - |
| 2.1 INTRODUCTION TO MOBILITY AND PATH LIFETIME IN MANET..... | - 13 - |
| 2.2 MEAN RESIDUAL PATH LIFETIME UNDER DIFFERENT MOBILITY MODELS - | 16 - |
| 2.2.1 <i>Overview of the Mobility Models.....</i> | - 17 - |
| 2.2.2 <i>Analysis of Mean Residual Path Lifetime</i> | - 20 - |
| 2.3 EFFECTS OF NETWORK PARAMETERS ON THE RPL..... | - 25 - |
| 2.4 SUMMARY..... | - 29 - |
| CHAPTER 3 LINK AGE-BASED PATH-SELECTION ALGORITHMS. - | 30 - |
| 3.1 INTRODUCTION..... | - 30 - |
| 3.2 MODEL AND ASSUMPTIONS..... | - 33 - |
| 3.2.1 <i>A Node Model with a Statistical FLL-Collecting Mechanism.....</i> | - 34 - |
| 3.2.2 <i>Underlying Route-Searching Protocol.....</i> | - 35 - |
| 3.3 PATH SELECTION ALGORITHMS BASED ON LINK AGE..... | - 36 - |
| 3.3.1 <i>Path-Selection Algorithm 1 (PSA1).....</i> | - 36 - |
| 3.3.2 <i>Path-Selection Algorithm 2 (PSA2).....</i> | - 38 - |
| 3.3.3 <i>Path-Selection Algorithm 3 (PSA3).....</i> | - 39 - |
| 3.4 PERFORMANCE EVALUATION | - 41 - |
| 3.4.1 <i>Performance Metrics.....</i> | - 44 - |
| 3.4.2 <i>Simulation Results.....</i> | - 47 - |
| 3.5 SUMMARY..... | - 53 - |
| CHAPTER 4 MOBILE-PROJECTED TRAJECTORY RLL PREDICTION .. | 55 - |
| 4.1 INTRODUCTION..... | - 55 - |
| 4.2 EXPONENTIAL BEHAVIOR OF LINK LIFETIME | - 59 - |
| 4.3 PREMISES OF DISTANCE MEASUREMENT-BASED RLL PREDICTION | - 62 - |
| 4.3.1 <i>Two-Node Link Model.....</i> | - 63 - |
| 4.3.2 <i>Minimal Number of Distance Measurements</i> | - 64 - |
| 4.3.3 <i>Effects of Measurement Errors.....</i> | - 68 - |
| 4.4 MOBILE-PROJECTED TRAJECTORY ALGORITHM..... | - 73 - |
| 4.4.1 <i>Operations of MPT</i> | - 73 - |

| | | |
|---|---|----------------|
| 4.4.2 | <i>MPT with More Distance Measurements</i> | - 78 - |
| 4.4.3 | <i>Theoretical Upper Bound on RLL Prediction Inaccuracy</i> | - 78 - |
| 4.5 | MPT WITH VELOCITY-CHANGE DETECTION | - 83 - |
| 4.5.1 | <i>Velocity-Change Detection (VCD) Test</i> | - 85 - |
| 4.5.2 | <i>MPT-VCD Algorithm</i> | - 86 - |
| 4.6 | PERFORMANCE EVALUATION | - 88 - |
| 4.6.1 | <i>MPT with Additional Distance Measurements</i> | - 88 - |
| 4.6.2 | <i>Acquisition Time-to-FLL Ratio</i> | - 91 - |
| 4.6.3 | <i>Performance of the VCD Test</i> | - 93 - |
| 4.6.4 | <i>Performance of MPT-VCD</i> | - 97 - |
| 4.7 | SUMMARY | - 101 - |
| CHAPTER 5 KALMAN FILTER-BASED RLL PREDICTION | | - 104 - |
| 5.1 | INTRODUCTION | - 104 - |
| 5.2 | UNSCENTED KALMAN FILTER-BASED RLL PREDICTION | - 106 - |
| 5.2.1 | <i>Formulation of Unscented Kalman Filter</i> | - 106 - |
| 5.2.2 | <i>Initialization and RLL Computation of the UKF</i> | - 111 - |
| 5.3 | PERFORMANCE EVALUATION | - 114 - |
| 5.3.1 | <i>Performance of the Algorithm without Velocity Change</i> | - 114 - |
| 5.3.2 | <i>Performance of the Algorithm with Velocity Change</i> | - 116 - |
| 5.4 | SUMMARY | - 119 - |
| CHAPTER 6 CONCLUSIONS AND DISCUSSIONS | | - 121 - |
| 6.1 | WHAT IS DONE | - 121 - |
| 6.2 | SOME THOUGHTS AND FUTURE WORK | - 123 - |
| APPENDIX A PROOF OF THEOREM ON MINIMAL NUMBER OF DISTANCE MEASUREMENTS | | - 127 - |
| APPENDIX B DERIVATIONS OF MPT WITH N DISTANCE MEASUREMENTS | | - 129 - |
| BIBLIOGRAPHY | | - 138 - |

LIST OF FIGURES

| | |
|--|-------|
| Figure 1-1: A wireless cellular network..... | 2 - |
| Figure 1-2: A wireless local area Network..... | 4 - |
| Figure 1-3: A wireless sensor network..... | 5 - |
| Figure 1-4: A mobile ad hoc network. | 7 - |
| Figure 2-1: Movements of constituent nodes along a four-hop path in MANET. ... | 15 - |
| Figure 2-2: Mean RPL vs. oldest link age for various path lengths (RM)..... | 23 - |
| Figure 2-3: Mean RPL vs. oldest link age for various path lengths (RWP). | 23 - |
| Figure 2-4: Mean RPL vs. oldest link age for various path lengths (G-M). | 24 - |
| Figure 2-5: Node movements on a multi-hop path in MANET. | 24 - |
| Figure 2-6: Effects of NVUI on the mean RPL in one-hop paths. | 27 - |
| Figure 2-7: Effects of pause time on the mean RPL (RWP). | 28 - |
| Figure 3-1: PM1 for Path-Selection Algorithm 1. | 49 - |
| Figure 3-2: PM1 for Path Selection Algorithm 2. | 49 - |
| Figure 3-3: PM1 for Path Selection Algorithm 3. | 50 - |
| Figure 3-4: PM2 for Path Selection Algorithm 1. | 50 - |
| Figure 3-5: PM2 for Path Selection Algorithm 2. | 51 - |
| Figure 3-6: PM2 for Path Selection Algorithm 3. | 51 - |
| Figure 3-7: Difference in PM1 using PSA1 with different average node speeds.... | 53 - |
| Figure 4-1: Statistical PDF of FLL under the RWP, RM, and G-M models..... | 61 - |
| Figure 4-2: Analytical and statistical PDFs of FLL under the RWP. | 61 - |
| Figure 4-3: Analytical and statistical CDF of FLL under the RWP. | 62 - |
| Figure 4-4: Approaching state at d_2 | 66 - |
| Figure 4-5: Receding state at d_2 | 66 - |
| Figure 4-6: S1 and the range of values each d_i can take. | 67 - |
| Figure 4-7: S2 and the range of values each d_i can take. | 67 - |
| Figure 4-8: S3 and the range of values each d_i can take. | 68 - |
| Figure 4-9: Average RLL prediction inaccuracy vs. systematic error. | 72 - |
| Figure 4-10: The MPT Cartesian coordinate system. | 75 - |
| Figure 4-11: Deriving the upper bound of RLL prediction inaccuracy. | 80 - |
| Figure 4-12: CDF of η for various numbers of distance measurements. | 90 - |
| Figure 4-13: Ratio of prediction misses to RLL predictions by MPT variants. | 91 - |
| Figure 4-14: RLL prediction inaccuracy w.r.t. T_{acq}/FLL ratio under Fluid model. . | 92 - |
| Figure 4-15: Statistical CDFs of Z_M ($\varepsilon_d=0.3\%R$)..... | 96 - |
| Figure 4-16: Statistical CDFs of Z_D ($\varepsilon_d=0.3\%R$)..... | 96 - |
| Figure 4-17: CDFs of RLL prediction inaccuracy ($\delta_{th}=0.5\varepsilon_d$). | 98 - |
| Figure 4-18: CDFs of RLL prediction inaccuracy w.r.t. $\Delta\tau_{req}$ | 99 - |
| Figure 4-19: Performance of MPT-VCD with multiple velocity changes ($\Delta\tau_{req}=1.0[\text{sec}]$). | 100 - |
| Figure 4-20: Performance of MPT-VCD with multiple velocity changes ($\Delta\tau_{req}=2.0[\text{sec}]$). | 100 - |
| Figure 5-1: Illustration for the UKF-based RLL-prediction algorithm..... | 108 - |
| Figure 5-2: Time diagram of tracking the state variables and RLL..... | 113 - |

Figure 5-3: Statistical CDFs of η (constant velocity). - 115 -
Figure 5-4: CDF of η under the G-M mobility model. - 118 -
Figure 5-5: CDF of η under the G-T mobility model. - 118 -

LIST OF TABLES

| | |
|--|---------|
| TABLE 2-1: SIMULATION PARAMETERS..... | - 21 - |
| TABLE 3-1: SIMULATION PARAMETERS FOR PATH-SELECTION ALGORITHMS. | - 43 - |
| TABLE 4-1: MEASUREMENT ERRORS AND PREDICTION INACCURACY- | 72 - |
| TABLE 4-2: d'_0 , d'_2 , AND d'_3 FOR MINIMAL-SLOPE TRAJECTORY..... | - 82 - |
| TABLE 4-3: d'_0 , d'_2 , AND d'_3 FOR MAXIMAL-SLOPE TRAJECTORY. | - 83 - |
| TABLE 4-4: PERFORMANCE OF THE VCD TEST..... | - 94 - |
| TABLE 4-5: PERCENTAGE OF PREDICTION MISSES IN MPT-VCD | - 101 - |

Chapter 1 Introduction

1.1 Overview of Existing Wireless Network Paradigms

Over the last three decades, the field of wireless communications has enjoyed an explosive growth in both technological research and commercial deployment. Advances made in this field have profoundly changed people's daily lives around the globe. Along with breakthroughs in microelectronics and the proliferation of personal mobile computing devices, a variety of wireless networks have evolved from fixed landline networks to provide a diverse array of communications services encompassing textual, audio, and video transmissions.

Most of today's commercially deployed wireless networks can be categorized into three popular paradigms, Wireless Cellular Networks (WCNs), Wireless Local Area Networks (WLANs), and Wireless Sensor Networks (WSNs). The WCN, as illustrated in Figure 1-1, is the most widely deployed wireless network today. In this network, communications between two mobile nodes are established by an infrastructure of base stations that are equipped with sophisticated hardware and software to process tasks such as channel assignment, data packet coding/decoding, and routing/forwarding operations. Communications in the WCN can be characterized as *single-hop*, since a mobile node needs only to be connected a base station for the communication with a remote node to be established.

When the first-generation WCN, known as the Advanced Mobile Phone System (AMPS), was deployed in 1983, it was based on Frequency-Division Multiple Access (FDMA) and exclusively supported voice traffic as an alternative means of communications to the fixed-line telephone networks. In the ensuing years, more sophisticated cellular networks based on technologies such as Time-Division Multiple Access (TDMA) and Code-Division Multiple-Access (CDMA) have been developed that are increasingly capable of supporting multimedia-intensive and data-centric services. Examples of wireless cellular networks in use today include Personal Communications Services (PCS) and Universal Mobile Telephone Services (UMTS) [67]. These networks provide long-range communications between users on a global scale.

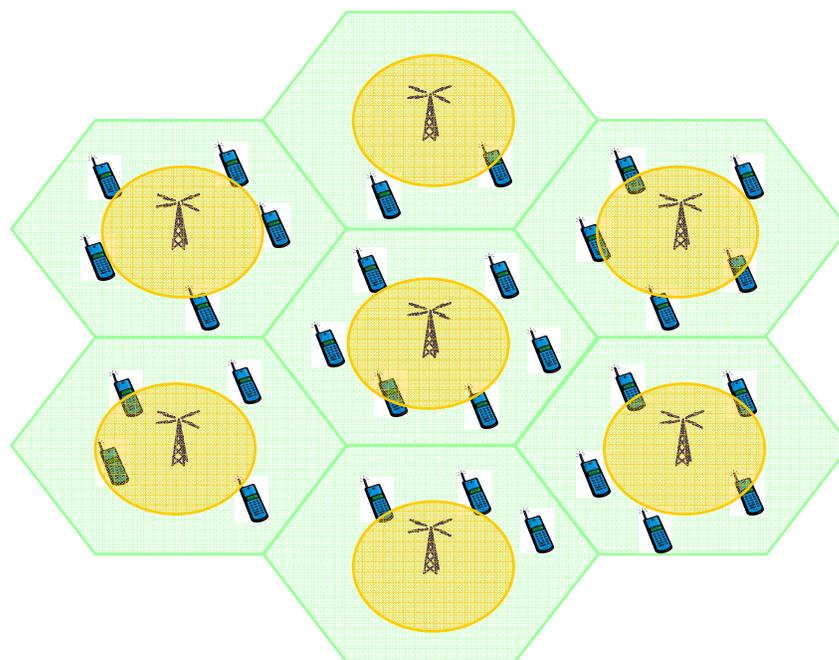


Figure 1-1: A wireless cellular network.

WLANs, on the other hand, represent a network paradigm that is similar to the

cellular technology, but support communications services in a much more localized area, e.g., an office building. As shown in Figure 1-2, the WLAN infrastructure consists of wireless Access Points (APs) that are positioned strategically across a local area to maintain network connectivity. One or several APs also serve as the gateways to the Internet. Like the WCN, the WLAN can also be characterized as a single-hop communications network. A mobile node establishes connection with one of the APs when it moves within the AP's coverage, and the data packets are transmitted via the intermediate APs to reach the gateway AP, connecting the user with the Internet. The growth of WLAN has spurred the development and commercialization of various mobile devices—cellular phones, laptops, personal digital assistants (PDAs), etc.—that support myriad communications services since the late 1990s. Because of its localized nature, the WLAN is capable of providing multimedia-intensive, real-time services to its users in the service area [67].

The WSN has been developed as a data-gathering network paradigm to be widely deployed in applications such as battlefield surveillance, environment monitoring, and traffic control. As illustrated in Figure 1-3, the sensors are scattered in a geographical area and must cooperate with each other to establish overall connectivity, forming a random network topology. The objective of the sensor network is to collect relevant information in the covered area (e.g., the degree of vibration in a bridge, the ground temperature on a prairie, etc.) and transmit this information to a data-collecting station, known as the sink, which is located either inside or outside of the sensor network area. Data transmissions in a WSN can be characterized as *multi-hop* transmissions, because the collected data are usually routed through several sensors before reaching the sink. Although the sensors remain stationary, the network connectivity can be dynamic over time as it is constantly influenced by factors such as battery power depletion and weather conditions. Consequently, the WSN induces a dynamic network topology in

which connections between neighboring sensors are established and torn down frequently, impacting the flow of data from the sensors to the sink.

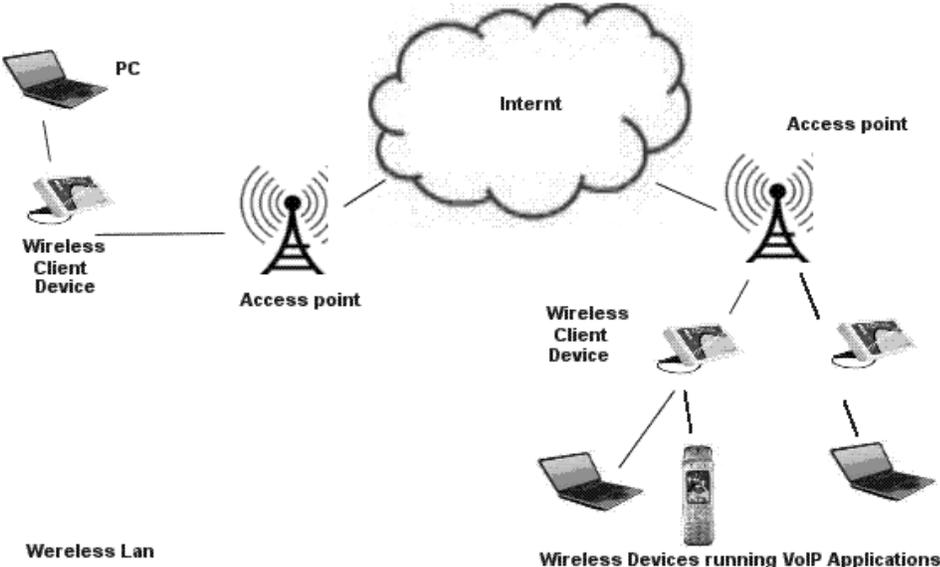


Figure 1-2: A wireless local area Network.

These three network paradigms each have their unique characteristics. In the WCN, although mobile nodes often move at high mobility, there exists a permanent, fixed backbone of powerful base stations to provide a stable network topology. As a result, an established traffic session between two mobile nodes has a low probability of experiencing disruptions. However, mobile nodes that are not in the coverage area of any base station are isolated and cannot establish connections with any node. In a WLAN, the infrastructure as formed by the APs also maintains a stable network topology in an environment where user mobility is expected to be low. In a WSN, all the sensors are stationary and clustered together in a geographical area. Data in the network are routed in a peer-to-peer fashion and are congregated towards the sink, and

each sensor can be either a source of the data or a relay to forward data for other sensors.

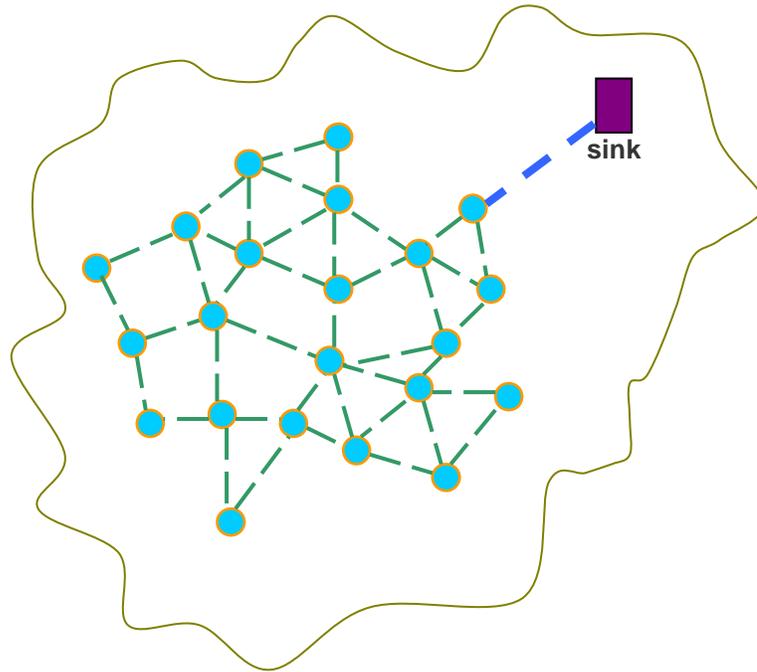


Figure 1-3: A wireless sensor network.

1.2 Introduction to Mobile Ad Hoc Networks

In real life, we often encounter situations where a mobile node attempts to reach another remote mobile node, only to find that there does not exist a communications infrastructure that would allow the connection to be established. For example, suppose a group of All-Terrain Vehicle (ATV) enthusiasts is organizing an ATV tournament in

the middle of the desert. Because of the remoteness and isolation of the location, it does not bode well economically for wireless service providers to establish a permanent network infrastructure there. The organizers, contestants, and spectators of the tournament who seek connectivity will find themselves unable to communicate with each other in such an environment. In this scenario, we must find another way to establish and maintain connectivity among all mobile nodes in the area. The answer to this challenge is the mobile ad hoc network (MANET).

A MANET consists entirely of mobile, autonomous nodes that do not constitute a permanent, fixed infrastructure to support communications services. As illustrated in Figure 1-4, since any of the nodes can be either a source, a destination, or a relay of data, all the nodes in the network must self-organize and cooperate with each other in order to establish connection for a traffic session between two remote nodes. Since all the nodes can freely move about in the network area, the network topology is dynamic over time. Furthermore, the network has to contend with the scarcity of resources such as bandwidth and battery power in the nodes, thus limiting the node lifetime and functionality. A MANET is most suitable for applications that require ad hoc, rapid deployment of a communications network in an environment lacking permanent infrastructure. Some of its applications include battlefield operations, disaster relief, vehicular communications, trade shows, and sporting tournaments.

In order to carry out the task of data routing between the data sender and recipient, the MANET is faced with unique challenges that are not encountered in the other wireless networks. Data routed from the source to the destination typically traverse through a communications path that consists of multiple intermediate relay nodes to the source and the destination. This path is termed a *multi-hop path*. Since the network topology is inherently dynamic, the path is prone to frequent and abrupt breakage while data are being routed. Any of a number of factors, such as mobility, physical

obstruction, thermal noise, multi-path fading, signal interference, weather conditions, etc., could lead to the path breakage, resulting in the disruption of data flow. Therefore, guarding data transmissions against path breakage is an important research topic in the field of MANET routing.

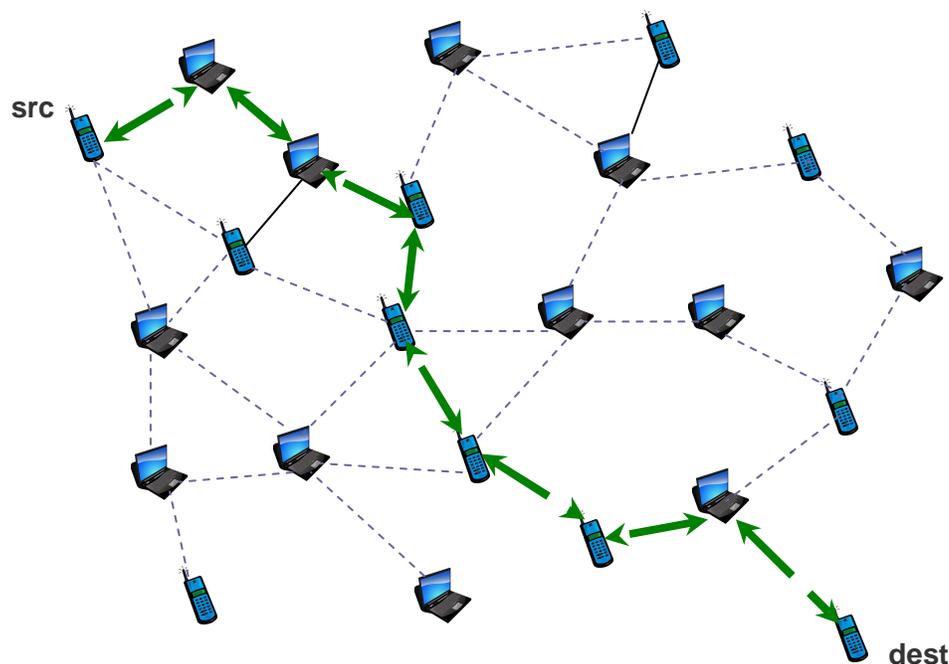


Figure 1-4: A mobile ad hoc network.

From a data-centric point of view, routing in a mobile ad hoc network can be generally categorized into two classes, *best-effort* and *Quality-of-Service (QoS)*. In a best effort-based routing protocol, the source and the intermediate nodes along the path do their best to deliver the data to the intended destination, without making any guarantee of the quality of delivery. Consequently, data could arrive at the destination with missing packets and/or with a much longer latency than an acceptable delivery time frame. Applications for which a best-effort scheme is suitable may include simple, textual data such as e-mail and Short Message Service (SMS). A QoS routing

scheme, on the other hand, must first guarantee that it can meet an explicit set of requirements demanded by the application before data transmissions can commence. A few well-established QoS requirements include the end-to-end delivery latency bound, data loss, degree of fault tolerance, and available bandwidth along every constituent link of the path. Obviously, QoS routing places a higher premium on the resources involved in data transmission and the knowledge on path quality. Applications that benefit from QoS routing include those that deliver real-time, multimedia-intensive data such as video and audio traffic sessions.

Both best-effort and QoS routing classes have been extensively studied, developed, and implemented in WCNs and WLANs. What makes them, and especially QoS routing, a daunting task in the MANET is that it must confront a number of challenging issues, including the lack of fixed infrastructure with stationary and powerful base stations, node mobility, limited resources, and the multi-hop characteristics of a path. Within such a topologically dynamic network, the task of robust data routing makes it imperative to find a reliable, long-living path along which transmitted packets are least disrupted.

1.3 Node Mobility and Path Lifetime in MANET

Node mobility is the dominating factor that impacts the reliability of a multi-hop communications path in the MANET. In our study, we consider the lifetime of a path to be directly related to path reliability, which in turn is associated with the lifetime of

each constituent link along the path. The establishment and breakage of a link between two mobile nodes is said to be *mobility-induced*. That is, a link is alive, or *up*, when the Euclidean distance between two mobile nodes is less than or equal to the transmission range of each other¹. Similarly, when the distance between the nodes is greater than the transmission range, the link is said to have *broken*. Subsequently, we define the *full link lifetime* (FLL) as the continuous duration from the first moment the link is up to the instant it breaks. We also define the *residual link lifetime* (RLL) at time t ($0 \leq t \leq FLL$), denoted as $RLL(t)$, as the duration from time t until the time that the link breaks, i.e., $RLL(t) + t = FLL$. These terminologies shall be used recurrently throughout this dissertation.

To ensure that data packets can be routed along a reliable path, it is desirable to gain some predictive knowledge of the path lifetime before data transmissions commence. The establishment and breakage of a multi-hop path can be defined based on lifetimes of its constituent links. A path is up when there exists a chain of links between the source and the destination that enable the connection between the two nodes. The residual *path lifetime* (RPL), denoted as $RPL(t)$, is defined as the duration measured from time t while the path has been up to the time when the first constituent link on the path breaks. Thus, the RPL of a path is the minimum of the RLLs of its constituent links. It stands to reason that in order to predict when a path breaks, we need learn of when any constituent link breaks first.

The residual path lifetime is defined in the context of a multi-hop path after its discovery in the MANET. A path may have been in existence since some time t_0 , long before it is discovered by some route-searching protocol, and it becomes useful only when it is needed by a node that either uses it immediately to send data packets, or

¹ In this dissertation, we assume that nodes in a MANET all have the same transmission range.

caches it as a back-up path to the current path. Therefore, the adjective “residual” refers to the remaining path duration from some time $t \geq t_0$ to the time it breaks. In this dissertation, we shall use the two terms “residual path lifetime” and “path lifetime” interchangeably, when the context is clear.

We also study the problem of the prediction of residual link lifetime in the MANET. To address this problem, we assume that there is relatively little information available to the RLL-prediction algorithm with which to compute the residual link lifetime. This is because we cannot expect each node in a large-scale network to be equipped with sophisticated hardware and software, which would incur a substantial cost of deployment. Rather, the functionality of the nodes can be rudimentary in order to achieve the economy of scale for mass-manufacturing. Possessing the predictive knowledge of residual link lifetime can greatly aid the mobile nodes in searching for a long-living path along which to transmit the data. In turn, having the predictive knowledge of when the path will break would allow the sender of the data to take appropriate actions to protect the data in transit before it breaks.

The remainder of this dissertation is organized as follows. Chapter 2 aims to gain an understanding of the behavior of mean residual path lifetime under several popularly used mobility models. The reason the mean RPL is chosen for study is because in the literature, some works have proposed using it as a parameter to gain the predictive knowledge of a current path lifetime. We study how parameters in these mobility models impact the mean lifetime of a path and verify the validity of choosing mean RPL as a prediction parameter.

In Chapter 3, we utilize the knowledge of *link age*-based link lifetime prediction and study its application of the best-path selection in the MANET. Link age is defined as the duration from the moment the link first comes into being up to the current time instant, while it is still up. Because of richness in connectivity in the MANET,

multiple paths could exist between a source and a destination. Since data transmissions require uninterrupted connectivity from the source to the destination, the ability to select the longest-living path from all the available ones is very desirable. We incorporate the link-age-based RLL-prediction algorithm as a subroutine in several path-selection algorithms and evaluate their performance in the network based on two selection criteria.

In Chapter 4, we take a different approach of RLL prediction based on distance measurements between two mobile nodes. We prove a theorem that states the necessity of measuring at least four distances to compute a unique RLL solution when the knowledge of node velocity and position is not available. Furthermore, we study the effects of systematic and random measurement errors on RLL prediction. These serve as the basis for the proposed Mobile-Projected Trajectory (MPT) RLL-prediction algorithm, which employs four periodically measured distances between two nodes of the link to estimate the relative trajectory, from which the RLL can be computed. We then derive the theoretical upper bound of the prediction inaccuracy achievable by any distance measurement-based RLL prediction algorithm when the measurement error has an unknown but bounded distribution. To account for the occurrences of velocity change during a link lifetime, we propose a simple velocity-change detection (VCD) test and augment it to the basic MPT to improve the accuracy of RLL prediction.

In Chapter 5, we propose an alternative distance measurement-based algorithm that utilizes the Unscented Kalman Filter (UKF). We formulate the link lifetime as a non-linear dynamic system model, and apply the UKF to recursively estimate its states, from which predicted RLL can be computed. UKF-based RLL prediction algorithm does not waste any distance measurement in estimating the states. Performance evaluation is also provided to demonstrate its effectiveness in various

velocity-change scenarios.

Chapter 6 summarizes the work from the preceding chapters, and discusses open problems and future research topics.

Chapter 2 Mobility and Path Lifetime in MANET

2.1 Introduction to Mobility and Path Lifetime in MANET

Node mobility is a major contributor to the dynamics of network topology in a mobile ad hoc network, where communications links between mobile nodes are established and broken frequently. Under certain circumstances, it can play a constructive role in improving the performance of the network. Several works have explored the benefits of taking advantage of mobility to improve network performance [25][34][9]. However, in order to harvest these benefits, it requires a deterministic mobility pattern and some coordination among the nodes to move closer to each other. When routing data in the MANET, nodes often move independently. Transmissions of data packets along a multi-hop path can be easily disrupted if any two nodes of a constituent link move away from each other beyond the transmission range. To protect the traffic session when this happens, the intermediate nodes have to buffer the undelivered packets, while control overhead packets such as Route Error (RERR) are generated to notify the source of the breakage so that it can launch a new Route Request (RREQ) to find an alternative path. Failure to find a new path in a timely manner would result in the permanent loss of these queued packets and premature termination of the traffic session. If re-transmissions of old data are required at a later

time, this could result in a waste of bandwidth and directly affecting the *goodput* of the network (i.e., the ratio of the number of data packets correctly received by the intended recipient to the number of transmitted packets).

Analytically modeling the residual lifetime of a multi-hop path due to node mobility is a difficult task. Let $Y_L(t)$ be the random process that denotes the RPL at time t of an L -hop path, and $X_i(t)$ be the random process that denotes the RLL of the i -th constituent link at time t , where $i = 1, \dots, L$. The relationship between the RPL and the constituent RLLs can be described by the following equation:

$$Y_L(t) = \min \{ X_i(t) : \forall i = 1, \dots, L \} \quad (2-1)$$

This simple equation belies the complexity of analytically modeling the RPL. Each $X_i(t)$ is not an independent random process. Rather, it is correlated with the links both before and after it. This can be visualized in Figure 2-1, where the arrow above each node of the four-hop path indicates the direction of node movement. When Node C moves towards the right, it both lessens $X_2(t)$ and prolongs $X_3(t)$ at the same time. The challenge therefore is to quantify the correlation between each pair of successive link lifetimes along each path. This is difficult, since we assume that each node independently and randomly chooses its own velocity². Because of this difficulty, researchers have chosen either to make the simplifying assumption of treating each $X_i(t)$ as an independent random process in order to make the mathematics of modeling the RPL more tractable, or to conduct extensive simulations to obtain numerical results.

² As in physics, we use the word “velocity” to refer to both speed and direction.

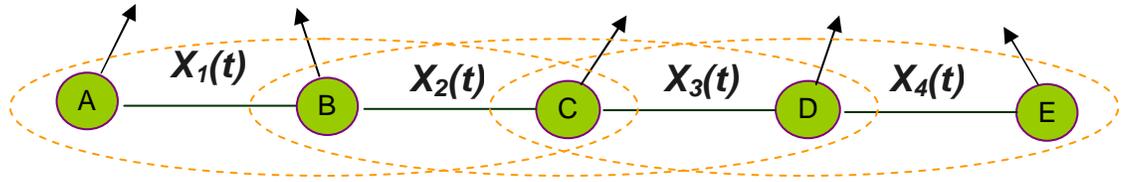


Figure 2-1: Movements of constituent nodes along a four-hop path in MANET.

A number of papers published in the literature have explored the statistical distribution and the expectation of residual path lifetime in the MANET. Bai, Sadagopan, Krishnamachari, and Helmy [3] studied the path duration in simulation under four mobility models, Random Waypoint, Freeway Mobility, Manhattan Mobility, and Reference Point Group Mobility (RPGM), and related the expected path duration to the performance of reactive routing protocols. They showed that the distribution of the RPL could be modeled as exponential, provided that the path was at least two hops in length, and that the average relative speed of all nodes in the network was medium to high. Han, La, and Makowski [32] treated the constituent link lifetimes as independent random processes, and analytically derived the distribution of a multi-hop path lifetime. A similar analytical model for computing the path availability was also proposed by Yu, Li, and Gruber [90]. Turgut, Das, and Chatterjee [83] studied the expected path lifetime to investigate the predictability of path lifetime under four mobility models in simulation, deterministic, partially deterministic, Brownian motion, and Brownian motion with drift. They concluded that the path lifetime could be exactly determined if the mobility pattern of the mobility model was absolutely deterministic, and a more chaotic mobility pattern would bring more uncertainty to predicting the path lifetime. The authors also envisioned the use of the predictive path lifetime knowledge in a route-discovery protocol.

In this chapter, we study the characteristics of the mean residual path lifetime.

Some published works (e.g., [3][83]) have attempted to use this quantity as a means of predicting the residual lifetime of a current path. We conduct extensive simulations to examine whether such an approach can be validated, i.e., the mean RPL as a reliable prediction parameter for the path lifetime in various mobility scenarios.

2.2 Mean Residual Path Lifetime under Different Mobility Models

Much of the study on the path lifetime in MANET employs extensive simulations to empirically investigate its characteristics, where node movements are governed by *mobility models*. A mobility model is a synthetic model defined by a number of mobility attributes, including, but not limited to, distribution of speed, distribution of direction, distribution of acceleration, distribution of pause time, correlation between consecutive velocities, dependency among different nodes (individual vs. group movements), etc. Depending on the mobility scenarios, these attributes are selectively applied to create as realistic an environment as possible.

In this section, we study the residual path lifetime of a multi-hop path under three popular mobility models: the *Random Waypoint Model* (RWP), *Random Mobility Model* (RM), and *Gauss-Markov Mobility Model* (G-M)}. We present a brief introduction to these mobility models, and discuss the details that are specifically adapted in their implementations in setting up our simulations.

2.2.1 Overview of the Mobility Models

The Random Waypoint mobility model was first proposed by Johnson and Maltz [41] and was adopted in simulation for the performance evaluation of Dynamic Source Routing (DSR). The network has a *closed network area*, where each node bounces back with a reflected angle once it reaches the network boundary. From its current position in the network area, each node independently chooses a speed that is uniformly distributed between V_{min} and V_{max} , where $V_{min}=0$ and V_{max} are the minimal and maximal node speeds, respectively. It also independently selects a uniformly distributed destination in the network area. The node then moves towards the destination with the chosen speed. Upon reaching the destination, it pauses for a constant period before independently selecting a new speed and destination again, repeating the above procedure.

Yoon, Liu, and Noble [89] reported that with the original RWP model, the nodes would gradually slow down as the simulation time progresses, leading to a declining average node speed. Eventually, the average speed approaches zero, with all the mobile nodes clustered at the center of the network area. This is known as the *clustering effect of the RWP*. To mitigate this shortcoming, the authors proposed a solution by employing a non-zero V_{min} . We also adopt this solution in our implementation of the RWP. Furthermore, we allow the nodes to move in a *torus*. A torus can be visualized as having a donut-like shape; it is an *open network area* in which a node that reaches the boundary of a rectangular network area passes through it and re-appears from the opposite boundary, maintaining its current velocity. Since there can be more than one direction for a node to reach the next destination in a torus, the node always chooses the one that yields the shortest trajectory to reach the

destination. This implies that the node could pass through one of its boundaries. The use of the torus therefore eliminates the notion of the center of the network area, and consequently mitigates the clustering effect.

The Random Mobility model, also known as the Brownian Motion mobility model, is based on the 2-D Random Walk process. In the literature, researchers have adopted several variants, depending on the specific applications (see, e.g., [70][81]). In this model, each node randomly and independently chooses a speed uniformly distributed between 0 and V_{max} , and a direction uniformly distributed between 0 and 2π . It then moves at the chosen velocity for a certain epoch. At the end of this epoch, the node randomly and independently chooses a new speed and direction and continues moving again without any pause. Boleng and Davies [7] have shown that because the current speed and direction of the mobile node are independent of its past speed and direction, the mobility model can result in movements with sudden stops and sharp directional changes. Our implementation of RM adopts this original model.

The Gauss-Markov (G-M) model [50] is a mobility model initially proposed for simulating node movements in a PCS network. It introduces correlation between successive velocity changes to allow more smooth and less drastic transition from one velocity to the next, thus eliminating abrupt changes in speed and direction, and mimicking more realistic movements. In its original form, a node updates its velocity from the previous one by breaking it down to its x and y components in a Cartesian system (for a 2-D velocity), where each component is Gaussian-distributed and independent of the other. The update is described by the following equations:

$$\begin{aligned}
v_n^x &= \alpha_x v_{n-1}^x + (1 - \alpha_x) \bar{v}_x + \sqrt{1 - (\alpha_x)^2} w_n^x \\
v_n^y &= \alpha_y v_{n-1}^y + (1 - \alpha_y) \bar{v}_y + \sqrt{1 - (\alpha_y)^2} w_n^y
\end{aligned} \tag{2-2}$$

In the above equations, v_n^x and v_n^y respectively denote the x and y components of the node velocity at time step n ; \bar{v}_x and \bar{v}_y denote their respective average speeds as n approaches infinity; w_n^x and w_n^y are uncorrelated Gaussian processes with zero mean and unit variance, and are independent of v_n^x and v_n^y ; α_x and α_y denote the correlation factors such that $\alpha_x, \alpha_y \in [0, 1]$. The authors of [50] further assumed isotropic velocity updates in both x and y components, i.e., $\alpha_x = \alpha_y = \alpha$. The node moves at the current velocity for a constant epoch, at the end of which a new velocity is updated from the current one, and the node continues moving at the new velocity without pause. The correlation factor determines the degree of randomness in the mobility model. When $\alpha = 0$, the current speed and direction are independent of the previous speed and direction, and the mobility model behaves like a RM model. When $\alpha = 1$, the speed and direction remain constant throughout the simulation, and the mobility model behaves like a constant-velocity fluid mobility model [88].

Our implementation of the G-M model follows that provided by Camp, Boleng and Davies [7] and adopted for the MANET. It defines the speed and direction in a polar-coordinate system as follows:

$$\begin{aligned}
s_n &= \alpha s_{n-1} + (1 - \alpha) \bar{s} + \sqrt{1 - \alpha^2} s_{x_{n-1}} \\
d_n &= \alpha d_{n-1} + (1 - \alpha) \bar{d} + \sqrt{1 - \alpha^2} d_{x_{n-1}}
\end{aligned} \tag{2-3}$$

where s_n and d_n denote the node speed and direction at time step n , \bar{s} and \bar{d} represent the mean values of speed and direction as $n \rightarrow \infty$, and $s_{x_{n-1}}$ and $d_{x_{n-1}}$ denote the

random variables from a Gaussian distribution.

We choose these mobility models because they represent various degrees of randomness in their respective node movements. The RM model has the highest degree of randomness, followed by RWP and G-M, which can be tuned to demonstrate a more deterministic mobility pattern. Comparisons among these mobility models thus allow us to observe the effects of randomness in mobility models that affect the path lifetime.

2.2.2 Analysis of Mean Residual Path Lifetime

We simulate a network of mobile nodes, in which node movements are governed by the three mobility models introduced in Section 2.2.1. A list of key simulation parameters are presented in TABLE 2-1. These parameters define an ad hoc network with a very dynamic topology. The network has a squared area, in which all the nodes move under the same mobility model. When node movements are governed by the RWP model, the network area is a torus; for RM and G-M models, the network area is open that nodes reaching the boundary will bounce back. The simulator randomly chooses a node as either the source or the destination. Once the source and the destination are chosen, we employ the *Dijkstra shortest-path algorithm* [4] to discover a path between them. 800,000 RPL statistics are collected for each of the one-, two-, three-, and four-hop paths under each mobility model.

Guided by an intuitive conjecture that an older link is more likely to break than a younger one, we first investigate the effects of the oldest link age of each path on the mean RPL. Figure 2-2, Figure 2-3, and Figure 2-4 plot the statistical mean RPL with

respect to the oldest link age for the three mobility models. These figures demonstrate that, contrary to the intuitive conjecture, the mean RPL in fact remains nearly constant with an increasing oldest link age for multi-hop paths under all three mobility models.

TABLE 2-1: SIMULATION PARAMETERS.

| Parameters | Values |
|-------------------------------------|-----------------|
| Network Size [m ²] | 700×700 |
| Num. of Nodes | 40 |
| TX Range [m] | 150 |
| Min. Speed [m/s] | 5 (RM and RWP) |
| Max. Speed [m/s] | 20 (RM and RWP) |
| Average Speed [m/s] | 12.5 (G-M) |
| Std. Dev. of Speed [m/s] | 3 (G-M) |
| Std. Dev. of Direction [degrees] | 30 (G-M) |
| Pause Time [sec] | 5 (RWP) |
| Node Velocity Update Interval [sec] | 10 (RM and G-M) |

This phenomenon can be explained as follows. The intuitive conjecture holds if all the constituent nodes along a path continuously move away from each other until one of the constituent link breaks. This can be seen in Figure 2-5(a), where the arrow on each node of the path indicates the direction of node movement. However, this is generally not the case when each node's velocity is chosen independent of the others, as demonstrated in Figure 2-5(b). When Nodes *A* and *B* move towards each other at

their independently chosen velocities, the distance between them decreases, leading to a longer residual link lifetime for the link $l_{(A,B)}$. This, however, comes at the expense of a shortened residual lifetime of the link between B and C , as they move further apart. We call these two links, $l_{(A,B)}$ and $l_{(B,C)}$, the *adjacent links* of a path, and define this effect of one link with a longer RLL and the other with a shorter RLL it as the *adjacent link correlation (ALC)*.

As illustrated in Figure 2-2 through Figure 2-4, when the oldest link age on a path is still very young (e.g., less than $10[sec]$ in the simulations), the mobile nodes behave more like those illustrated in Figure 2-5(a), and the link with the oldest age tends to be the first to break on the path. This is confirmed by the rapid decline in mean RPL in the figures when the oldest link age is less than $10[sec]$. As this link grows older, the nodes on the path have entered into a more stable phase, and all the constituent links of the path become equally likely to break first, diminishing the effects of the oldest link age on the overall path lifetime. The mean RLL therefore becomes uncorrelated with respect to the oldest link age on the path.

In fact, this behavior of non-correlation between the mean RPL and the oldest link age can be observed also with respect to any other constituent link of a multi-hop path. Furthermore, as the path length increases, the mean RPL decreases at a gradually reduced rate. Using Figure 2-2 as an example, it can be seen that for single-hop paths, the mean RPL achieves a range from $12[sec]$ to $15[sec]$. With two-hop paths, the achieved mean RPL is substantially reduced, ranging from $5[sec]$ to $8[sec]$. With three-hop paths, it further flattens and decreases to about $3[sec]$. With four-hop paths, the mean RPL has only slightly decreased to about $2.5[sec]$. It can be deduced that this decreasing trend shall continue for longer paths. The relative decrease in mean RPL will be smaller and smaller with respect to increasing path length, until it eventually approaches $0[sec]$ when the path length approaches infinity.

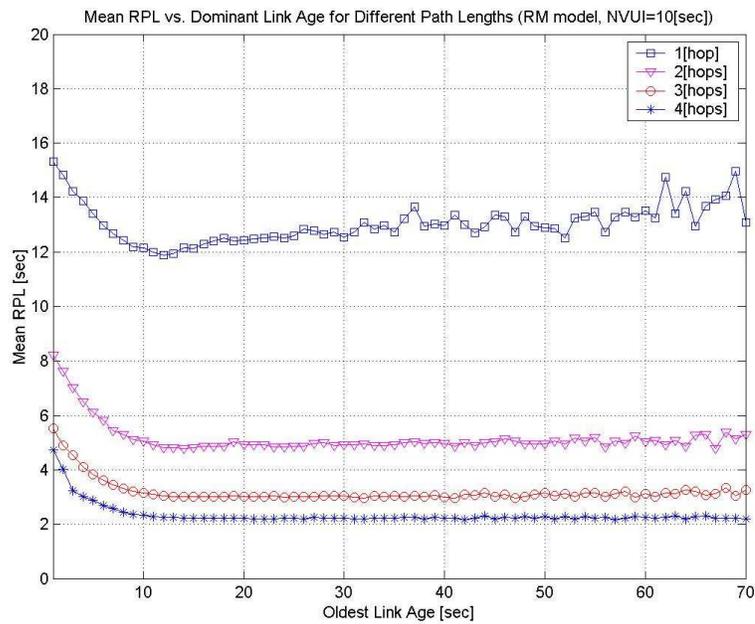


Figure 2-2: Mean RPL vs. oldest link age for various path lengths (RM).

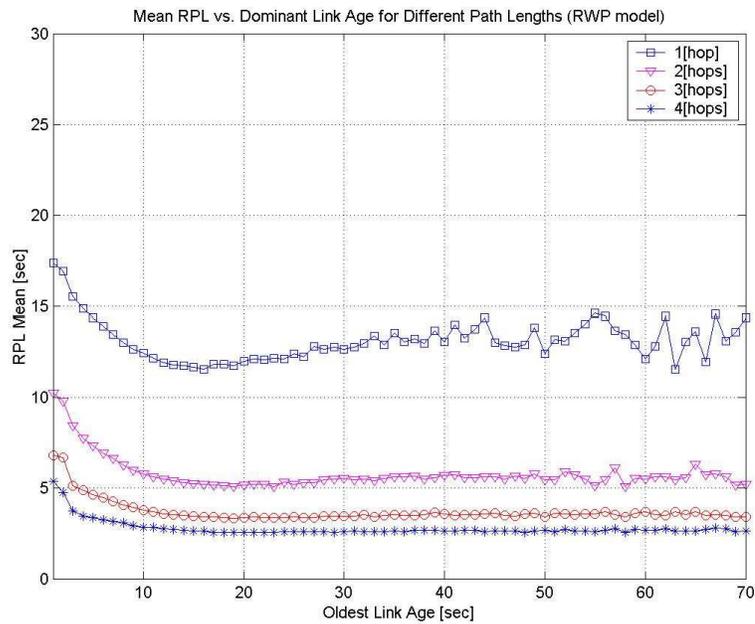


Figure 2-3: Mean RPL vs. oldest link age for various path lengths (RWP).

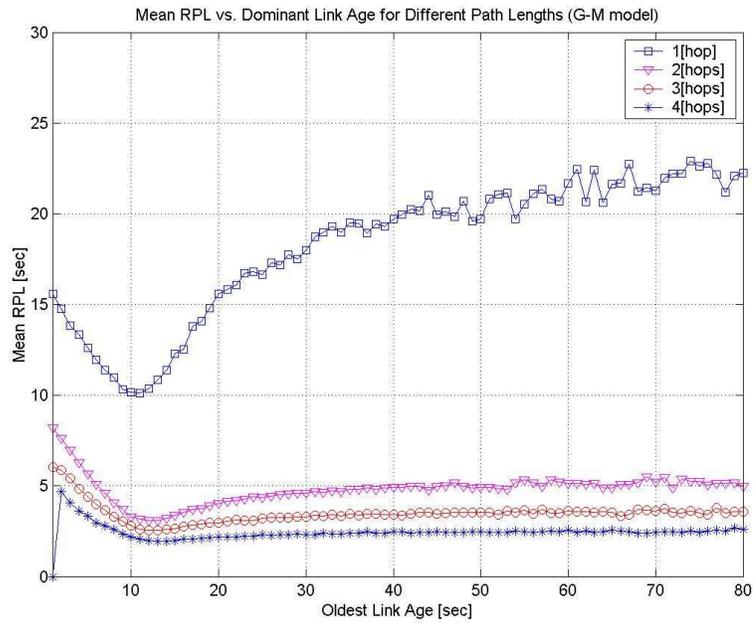


Figure 2-4: Mean RPL vs. oldest link age for various path lengths (G-M).

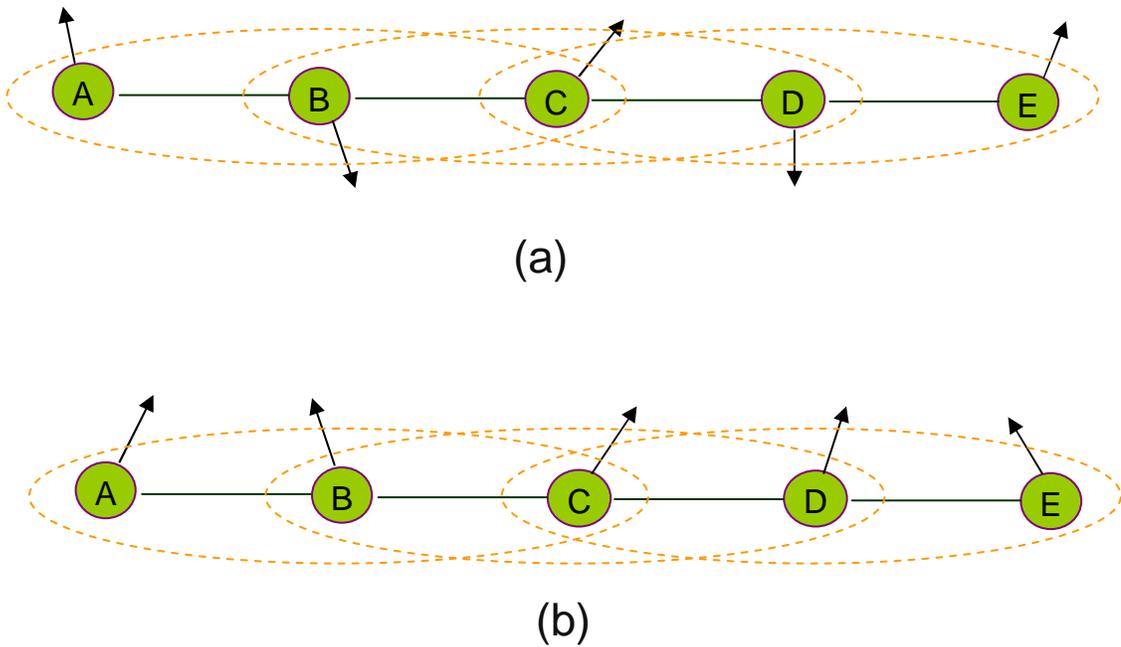


Figure 2-5: Node movements on a multi-hop path in MANET.

2.3 Effects of Network Parameters on the RPL

We now examine the effects of network parameters on the mean RPL in the mobile ad hoc network, with the same simulation parameters as given in TABLE 2-1. Specifically, we study the effects of the following parameters: *node density*, *node velocity update interval* (NVUI), and *pause time*. Node density is defined as the average number of nodes in a node's neighborhood³, and its value is related to three other network parameters: the number of nodes in the network area, the size of network area, and transmission range. Therefore, studying node density on the mean RPL is equivalent to studying the effects of these three parameters for their respective effects. The NVUI parameter defines the time duration in which a node moves at constant velocity between two consecutive velocity changes. It is closely related to the movement predictability in both Random Mobility and Gauss-Markov models. Note that NVUI does not apply to the RWP mobility model. The pause time defines the time duration in which a node remains stationary before moving again. It plays an important role in affecting the degree of dynamics in the network topology: the longer the pause time, the less dynamic the network topology. In this study, it is only defined in the RWP model.

With respect to node density, our investigations have shown that it generates an insignificant effect on the mean RPL for a path with a length greater than two hops under all three mobility models. Using the RM model as an example, it can be seen

³ A node neighborhood is defined as a circular region with the transmission range being its radius.

that for two-hop paths in a $900 \times 900 [m^2]$ network, i.e., a node density of $3.49 [nbrs/node]$, we have observed only a very small decrease in mean RPL (approximately $0.5 [sec]$) from a $700 \times 700 [m^2]$ network (i.e., a node density of $5.77 [nbrs/node]$). With longer paths, the difference between the two different node densities becomes indistinguishable. This can be explained by the fact that each path is discovered using the Dijkstra shortest-path algorithm, which connects the source and the destination through intermediate nodes that are as far apart as possible. Having more intermediate nodes in between does not affect the search for the shortest path, and therefore exerts no influence over the resultant mean residual path lifetime.

To observe the effects of NVUI on the mean RPL, we select two NVUI values, $10 [sec]$ and $40 [sec]$, in RM and G-M mobility models. Simulation results show that for both mobility models, discernable changes in mean RPL are observed only in one-hop paths (i.e., the links), whereas the effects become negligible in paths of two hops and longer. Figure 2-6 presents the case of links at the two NVUI values. In each mobility model, increasing the NVUI slightly increases the mean lifetime of the link when the link age is older than $20 [sec]$. Otherwise, the two curves for each mobility model overlap closely. This demonstrates that a link that has already persisted for a long time is more likely to have a longer residual lifetime, which contributes to a slightly longer mean residual link lifetime.

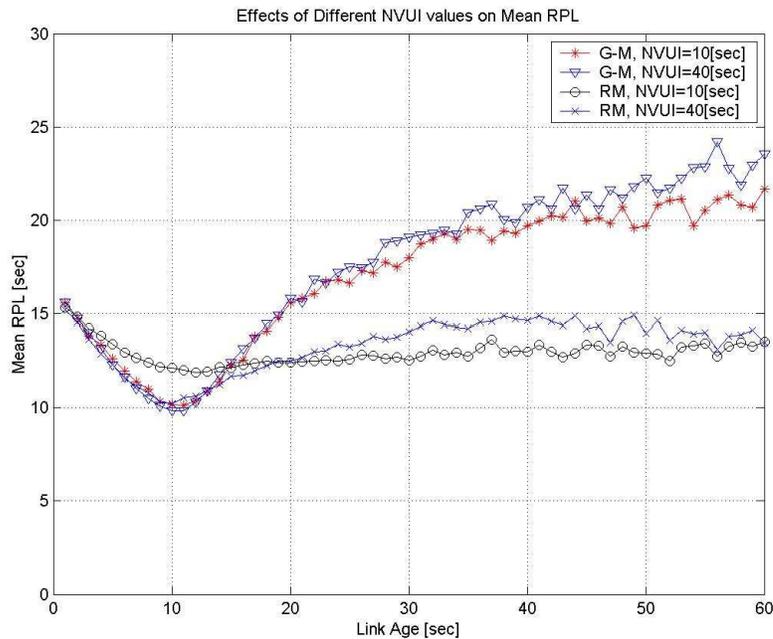


Figure 2-6: Effects of NVUI on the mean RPL in one-hop paths.

The effects of the pause time on the mean RPL in the RWP model are illustrated in Figure 2-7, which plots the mean RPL with respect to the oldest link age for a pause time of 50[sec]⁴. Comparing this figure with Figure 2-3, which has a pause time of 5[sec], it can be seen that for shorter paths (i.e., one- and two-hop paths), the pause time indeed leads to a substantial impact on the behavior of the mean RPL. When the pause time is 5[sec], each node does not remain stationary for long, and the network topology is very dynamic. The effects of the short pause time on each constituent link lifetime of the shorter paths are thus masked by the dynamic mobility of other nodes moving around the stationary node, and the mean RPL appears nearly constant regardless of the link age. On the other hand, when the pause time is 50[sec],

⁴ Since we have established in Section 2.2.2 that the mean RPL is uncorrelated with the age of any constituent link on a multi-hop path, the results presented for the link with the oldest link age equally apply to those other links.

neighboring nodes are more likely to remain longer in each other's transmission range, making the overall network less dynamic, leading to a longer mean RPL. Furthermore, a longer pause time makes it more likely to find a path along which all the constituent nodes are stationary, resulting in a longer RPL. In other words, it is more probable with a longer pause time to create alternating moments of dynamic and stable network topologies, making the mean RPL more fluctuating at different times. Finally, as can be observed in both figures, regardless the length of pause time, an increasing path length would still have a dominating impact in shortening the mean RPL, thus mitigating the effects of a longer pause time.

Therefore, we conclude from the results that the mean RPL is a quantity that is not significantly impacted by the key network and mobility parameters that we have chosen to study. Consequently, it will not be able to serve as a reliable parameter for predicting the current residual path lifetime.

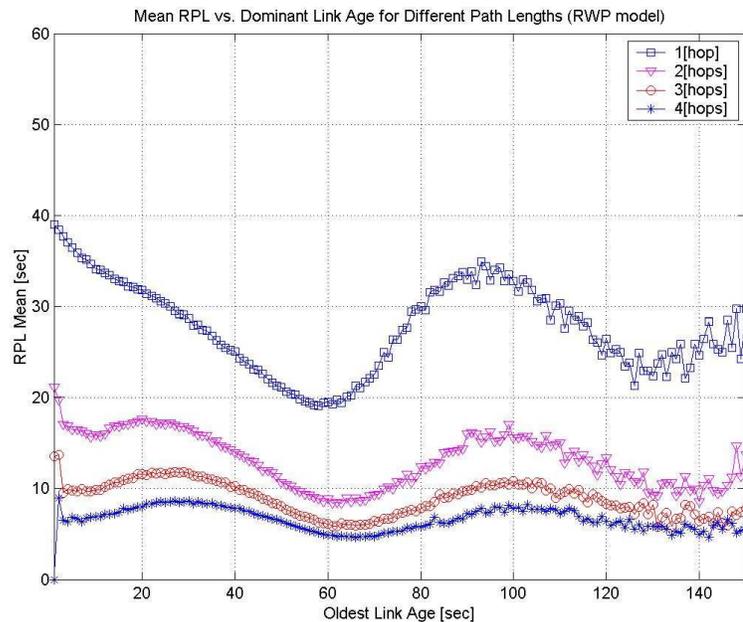


Figure 2-7: Effects of pause time on the mean RPL (RWP).

2.4 Summary

In this chapter, we investigated the effects of mobility on the residual lifetime of a multi-hop path in the MANET via simulations. We chose the mean residual path lifetime as the object of our study, since it has been proposed in the literature as a parameter of path lifetime prediction. Insight gained in this chapter may help us develop a path-selection algorithm for mobile ad hoc networks.

Three mobility models, Random Mobility, Random Waypoint, and Gauss-Markov, were implemented to govern node movements because they represented various degrees of mobility randomness that would impact the path lifetime. Through extensive simulations, we have observed that under all the mobility models studied, the mean RPL of a multi-hop path is uncorrelated with any constituent link age. In addition, we examined the effects of three network parameters, namely the node density, node velocity update interval, and pause time. We have found that these parameters do not contribute to substantial impact on the mean RPL of multi-hop paths, neither. We thus conclude from these results that the mean RPL in the MANET would not serve as a reliable prediction parameter for path lifetime prediction.

Since a MANET is usually characterized as a network with a high node density, there often exist multiple paths that connect the sender and receiver of the data. It is unlikely that these paths would all break simultaneously due to the dynamic network topology. Therefore, some paths will have longer lifetimes than the others. It would be desirable to be choose a path from all the available ones that is the most suitable to route data packets to the destination. In Chapter 3, we take advantage of this richness in connectivity and propose three link age-based path-selection algorithms.

Chapter 3 Link Age-based Path-Selection Algorithms

3.1 Introduction

Recent research has generated a growing interest in real-time, multimedia-intensive applications in the MANET. Supporting such applications demands that certain Quality-of-Service requirements be met before commencing data transmissions. Provisioning of the QoS has been extensively studied and implemented in the wired networks such as the Internet, and researchers have defined a number of QoS metrics such as data-loss rate, packet-delivery latency, delay jitter, etc. for that need to be satisfied in routing data in the networks.

The provisioning of QoS in the MANET presents both challenges and opportunities. On one hand, the network's inherent nature of dynamic topology causes frequent link breakage that can disrupt the flow of data packets along a multi-hop path. When this happens, the routing protocol needs to execute relevant path-management operations such as failure notifications and path rediscovery. Failure to discover an alternative path for the ongoing traffic session in a timely manner would result in permanent data loss, waste of network bandwidth, and premature termination of the traffic session. On the other hand, the typically high node density induces a rich and redundant connectivity in the MANET, which motivates the development of a routing protocol to seek out the simultaneous existence of multiple paths between the source and the destination.

Generally, multiple paths that exist between the source and the destination have different residual path lifetimes. Therefore, in the interest of the successful completion of data delivery, it is always desirable for the source node to intelligently select a path that is most reliable among all available ones to route data packets. In our study, we associate the reliability of a path with its lifetime induced by the node mobility. Possessing some predictive knowledge of when the path might break would allow the routing protocol to select the most reliable path. This in turn would reduce the overhead as a result of fewer path failure notifications and path re-discoveries. Consequently, this would lead to meeting QoS requirements such as lower data loss rate, less control overhead, and reduced packet delivery latency. A *path-selection algorithm* (PSA) is thus charged with making an intelligent decision of choosing the best path for data transmissions.

The first step towards developing the PSA is to gain some insight of the characteristics of the path lifetime. However, as we have shown in Chapter 2, accurately modeling the lifetime of a multi-hop path to extract its characteristics is a difficult task. In light of this difficulty, we take the approach of first identifying the reliability of residual lifetime for each of the path's constituent links, then extending the results to the overall path lifetime [3][32]. One popular parameter used in assessing the reliability of a link is the link age.

In this chapter, we propose three path-selection algorithms, whose decision making is based on simple reliability parameters computed from link age that can be readily sampled from the network. They are proposed with the following three design guidelines in mind. Firstly, each path-selection algorithm is transparent to the underlying routing protocol. This means that the routing protocol will be responsible for discovering a set of available paths between the source and the destination, and the PSA will only utilize this discovered path set for its computations without knowing

how it was discovered. This transparency allows the PSA to be implemented more flexibly across a wide array of multi-path routing protocols.

Secondly, the decision-making computations of a PSA are independent of any underlying node mobility attributes such as the distribution of speed, the distribution of direction, and the distribution of node pause time. A PSA that is conditioned on a particular attribute is limited in its scope of applications. For example, assume that nodes move according to some mobility model that incorporates the pause time, which has a significant impact on the dynamics of the network topology. Since in practice, a mobile node is usually unable to learn of the pause time, a path-selection algorithm cannot rely on the knowledge of this parameter to make selection decisions. Similarly, “hardwiring” a mobility attribute value into the PSA would cause the algorithm to perform unsatisfactorily once the mobile environment changes. Therefore, independence from mobility attributes makes a path-selection algorithm more adaptive in a wide variety of mobility scenarios.

Thirdly, a mobile node is not equipped with sophisticated hardware and software. For example, if a node is supported by the GPS, it could provide the node with accurate velocity and location information that may make the computations of the residual link lifetime much easier in many scenarios. However, too much sophisticated HW/SW equipment built into each mobile node would make it harder to achieve an economy of scale. Therefore, a challenge is how to design a path-selection algorithm that makes intelligent decisions based on very limited information that can be readily sampled from the network.

A number of routing protocols have been proposed in literature with the objective of discovering multiple paths between a source-destination pair. For example, the Split Multi-path Routing (SMR) protocol [48] discovers two paths and chooses the shorter one to transmit data packets, where the length of the path is determined by the time

when the source received a RREP packet from the destination. Nasipuri and Das [57] proposed a multi-path extension to the DSR protocol by allowing the RREQ packets to always reach the destination before a path is declared discovered. Each RREP packet contains the discovered path as it traverses back to the source. After receiving all the RREP packets, the source only chooses those that are mutually node-disjoint, i.e., no intermediate node shared by more than one path, and stores them in a cache. When the primary path breaks, the shortest secondary path takes over the task of data routing. Das et al. [16] proposed searching for multiple paths in both temporal and spatial domains. A path's reliability is determined by the minimum of the signal strengths of its constituent links, and the signal strength is assumed to be inversely proportional to the distance between the two nodes. Before the predicted path lifetime expires, path re-discovery operations are preemptively executed to search for a new path to prevent disruption in the current data routing.

This chapter is organized as follows. Section 3.2 provides the model and assumptions for the proposed path-selection algorithms. Section 3.3 proposes three path-selection algorithms based on link age. We then develop two performance metrics in Section 3.4 and employ them for the performance evaluation of the proposed PSAs. Section 3.5 concludes the chapter.

3.2 Model and Assumptions

We propose three path-selection algorithms, which aim to select the best path from all available paths between the source and the destination. To define what it means to

be the best path, we propose the following two objectives that the algorithms strive to achieve while making path-selection decisions:

- 1) the selected path is most likely to meet a specified target RPL requirement among all available paths set by the application, and
- 2) the selected path has the longest RPL among all available paths.

3.2.1 A Node Model with a Statistical FLL-Collecting Mechanism

We now present the assumptions on the node model and the underlying routing protocol on which the path-selection algorithms are developed. We first employ a simple statistic-collecting mechanism in each node that will be used in two of the three proposed algorithms. This mechanism collects full link lifetime statistics empirically sampled from the network, for all the nodes in its neighborhood⁵, and works as follows. Each node is equipped with a beacon, which periodically broadcasts its ID signal to announce its presence to all other nodes in its neighborhood. When Node B enters Node A's neighborhood, Node A receives Node B's ID signal for the first time and assumes there now exists a link from B to A. Node A then initiates a timer that keeps track of B's presence by listening to its ID signal as long as it moves in A's neighborhood, until a time when it no longer hears the signal. This entire duration—from the first time A hears of B's ID signal to the first time it stops hearing it—is

⁵ A similar scheme was also used by Gerharz, de Waal, Martini and James in their work [22].

recorded as the full link lifetime between the two nodes and is stored in Node A's memory cache. Likewise, Node B that has been keeping track of Node A's ID signal will have also generated and stored the FLL statistic. Each node continuously collects FLL statistics in this manner with nodes entering and exiting its neighborhood, until a sufficient number of statistics has been collected that allows the node to generate a stable histogram of FLL. In the subsequent sections, we assume that links in the network are symmetric. That is, if a link exists from Node A to Node B, another link exists from B to A.

3.2.2 Underlying Route-Searching Protocol

For our path-selection algorithms, we assume the availability of a generic route-searching protocol has already discovered multiple paths that connect the source and the destination, and it is transparent to the path-selection algorithm. This allows great flexibility for the proposed algorithms to be implemented on top of a number of route-searching protocols proposed in the literature.

The generic route-searching algorithm is modeled after the Dynamic Source Routing protocol and works as follows. When a source node has data packets to send to a destination, it initiates the route-searching operations by sending out RREQ packets to all its neighbors. As each RREQ packet is forwarded from one node to the next in the network, it records these nodes so that by the time it reaches the destination, it will have learned the entire path from the source to the destination. After discovering a set of multiple paths, the destination copies the path information collected in the RREQ to a RREP packet, and sends it in reverse direction of each

discovered path towards the source. Each node that receives the RREP invokes the proposed PSA to compute the reliability information for each link between itself and the downstream node⁶, and appends this information in the RREP before forwarding it to its upstream node. Upon receiving all the RREP packets, the source will have obtained the knowledge of each path to reach the destination, as well as the reliability of each path. The source will then be able to make an intelligent path-selection decision based on the information it has gathered to choose the best path for data transmissions to commence.

3.3 Path Selection Algorithms Based on Link Age

3.3.1 Path-Selection Algorithm 1 (PSA1)

The first path-selection algorithm, PSA1, works as follows. When a mobile node is first deployed in the network, it continuously collects FLL statistics to construct the empirical FLL histogram. The network is ready to execute the path-selection algorithm once a stable FLL histogram has emerged for every node in the network.

After a set ψ of paths has been reported to the destination by the underlying route-

⁶ If a link is a constituent link along a multi-hop path, its *upstream node* is the one that forwards the data packets from the source to the other node of the link, known as the *downstream node*. The words “upstream” and “downstream” thus indicate a direction of data flow from the source to the destination.

searching protocol, it returns a RREP packet for each discovered path i of length L_i [hops], where $i \in \{1, \dots, |\Psi|\}$, along its reverse direction towards the source. Upon receiving the RREP packet, each upstream node j on Path i , where $j \in \{1, \dots, L_i\}$ ⁷, computes the probability that the link between Node j and its downstream neighbor Node k , denoted as $l_{(j,k)}^i$, has a RLL of at least τ [sec] given its current age a_{jk}^i [sec]. τ is a system parameter whose value is specified by the upper-layer application.

To compute this probability, Node j first computes a_{jk}^i by counting the number of times it has continuously received Node k 's ID signal. Denote \tilde{T}_{jk}^i as the residual lifetime of the constituent link $l_{(j,k)}^i$, and T_{jk}^i as the corresponding FLL of the link, i.e., $T_{jk}^i = \tilde{T}_{jk}^i + a_{jk}^i$. The desired probability can thus be computed as follows:

$$\begin{aligned}
 p_{jk}^i(\tau) &= P\{\tilde{T}_{jk}^i \geq \tau \mid a_{jk}^i\} = P\{\tilde{T}_{jk}^i \geq \tau + a_{jk}^i \mid a_{jk}^i\} \\
 &= \frac{P\{T_{jk}^i \geq \tau + a_{jk}^i\}}{P\{T_{jk}^i \geq a_{jk}^i\}}
 \end{aligned} \tag{3-1}$$

Note that the numerator and the denominator in the second equality of Equation (3-1) can be numerically computed from Node j 's FLL histogram. Node j then appends $p_{jk}^i(\tau)$ to the RREP packet and forwards it to its upstream neighbor on the path. Upon receiving the RREP, the source computes *the probability that Path i 's residual lifetime is at least τ [sec]*, denoted as $p_i(\tau)$, as follows:

⁷ The nodes on Path i are numbered 1 through L_i+1 , where L_i is the path length in hops.

$$p_i(\tau) = \prod_{j=1}^{L_i} p_{jk}^i(\tau), \forall i \in \{1, \dots, |\Psi|\}. \quad (3-2)$$

Note that we assume that the residual link lifetimes along the path are treated as independent random variables, an assumption commonly adopted by a number of works in the literature (e.g., [16]).

The decision of selecting the path with the highest probability of meeting the target RPL of τ [sec] is given by:

$$i^* = \arg \max_i \{p_i(\tau) : i = 1, \dots, |\Psi|\}. \quad (3-3)$$

If more than one path in Ψ has the same maximal $p_i(\tau)$, the shortest path (in hops) with $p_i(\tau)$ is chosen as i^* .

3.3.2 Path-Selection Algorithm 2 (PSA2)

The second path-selection algorithm, PSA2, is proposed as follows. When the destination has received the RREQ packets and learned of a set Ψ of paths discovered by the route-searching protocol, it sends back a RREP packet to the source in reverse direction of each discovered path i . Each intermediate node j on Path i that receives the RREP computes the current age a_{jk}^i of the link between itself and its downstream neighbor k , and appends it to the RREP before continuing to forward it to its upstream neighbor. After the source has received the RREP packet from each path, it will have

learned all the paths' constituent link ages. We define the *path age* of a Path i to be the age of the youngest link of the path:

$$a_i = \min \{ a_{jk}^i : j = 1, \dots, L_i - 1; k = j + 1 \}. \quad (3-4)$$

The decision made by the path-selection algorithm is to choose the path i^* that has the minimum path age:

$$i^* = \arg \min_i \{ a_i : i = 1, \dots, |\Psi| \}. \quad (3-5)$$

If more than one path in Ψ has the same path age, the shortest one is chosen as i^* .

3.3.3 Path-Selection Algorithm 3 (PSA3)

The operations of the third path-selection algorithm, PSA3, are based on the assertion that a reliable multi-hop path should be composed of links that are neither too old nor too young. Given the current age a_{jk}^i for each link $l_{(j,k)}^i$ on Path i , each node j computes the expected full lifetime of Link $l_{(j,k)}^i$ as follows:

$$t_{jk}^i = \frac{\int_{a_{jk}^i}^{\infty} t f_T(t) dt}{\int_{a_{jk}^i}^{\infty} f_T(t) dt}, \quad (3-6)$$

where $f_T(t)$ is the *probability density function* (PDF) of the FLL as observed by Node j , which can be easily obtained from j 's statistical FLL histogram.

With t_{jk}^i , node j assigns to each link $l_{(j,k)}^i$ a *link grade* G_{jk}^i . The link grades are assigned with respect to each FLL interval that t_{jk}^i falls into, and are defined as follows:

- 1) If $t_{jk}^i \in (0, \tilde{\mu}_{jk}^i - \tilde{\sigma}_{jk}^i / 2)$, $G_{jk}^i = \text{SOSO}$;
- 2) If $t_{jk}^i \in (\tilde{\mu}_{jk}^i - \tilde{\sigma}_{jk}^i / 2, \tilde{\mu}_{jk}^i + \tilde{\sigma}_{jk}^i / 4)$, $G_{jk}^i = \text{EXCELLENT}$;
- 3) If $t_{jk}^i \in (\tilde{\mu}_{jk}^i + \tilde{\sigma}_{jk}^i / 4, \tilde{\mu}_{jk}^i + \tilde{\sigma}_{jk}^i)$, $G_{jk}^i = \text{GOOD}$;
- 4) If $t_{jk}^i \geq \tilde{\mu}_{jk}^i + \tilde{\sigma}_{jk}^i$, $G_{jk}^i = \text{POOR}$.

In the link grades, $\tilde{\mu}_{jk}^i$ and $\tilde{\sigma}_{jk}^i$ denote the sample mean and standard deviation of FLL on $l_{(j,k)}^i$, respectively, which are computed from Node j 's statistical FLL histogram. The ranking of these link grades in descending order is $\text{EXCELLENT} > \text{GOOD} > \text{SOSO} > \text{POOR}$. The values that define the boundaries of the FLL intervals are chosen by observing the distribution of the FLL in simulation scenarios.

After receiving the RREQ packets and learning of a set Ψ of paths from the source to the destination, the destination disseminates the RREP packets in reverse directions of the discovered paths. Upon receiving the RREP, each intermediate node j on Path i computes t_{jk}^i for the link with its downstream neighbor k , and assigns a link grade G_{jk}^i . Node j then appends the link grade to the RREP packet before forwarding it to its upstream neighbor on the path.

When the source receives the RREP and performs its own link-grade computations, it assigns a *path grade* G_i to Path i . The assignment of G_i can be a choice of the path-selection algorithm. We propose the following two path-grade assignments. In the first, conservative assignment, the worst of the link grades on the path is selected as the path grade. In the second, aggressive assignment, the best of the link grades is selected as the path grade. Once every path has been assigned a path grade, the source chooses the best path i^* that has the highest path grade. If more than one path in Ψ has the same best path grade, the one with the fewest hops is selected as i^* .

3.4 Performance Evaluation

We evaluate the performance of the three proposed path-selection algorithms in simulation. TABLE 3-1 tabulates the relevant network and mobility parameters used to construct simulation scenarios. Initially, the mobile nodes are uniformly distributed in the squared network area. We have chosen values for network dimensions, number of nodes, and transmission range to ensure that the node density is sufficiently high such that multiple paths can be discovered between the source and the destination. The simulator randomly chooses a source and a destination among the nodes. The underlying mobility model that governs node movements is the Random Mobility model, as explained in Section 2.2.1. The simulator discovers a set Ψ of paths that are *node-disjoint*, i.e., no two paths in Ψ share the same intermediate node. Such paths

have the property that the breakage of one path is independent of all others⁸, and permit us to better evaluate the performance of the proposed path-selection algorithms by treating each path lifetime independent of the others. In practice, this restriction can be easily lifted to accommodate paths that share common intermediate nodes.

The simulator simulates the discovery of the path set \mathcal{P} by employing the *multi-path Dijkstra's algorithm*, an extension to the traditional Dijkstra's shortest-path algorithm [4]. The initial network is modeled as a graph G_0 , in which each node is a vertex and each link an undirected edge. In G_0 , the Dijkstra's algorithm finds the shortest path that exists between the source and destination, denoted as P_1 . It then removes all of P_1 's intermediate nodes and all the edges incident to them to produce the residual graph G_1 . The Dijkstra's algorithm is again applied to G_1 in which to search for the shortest path P_2 , and removes all of its intermediate nodes and incidental edges to generate G_2 . With each such iteration, a new path is discovered in the new residual graph, until a specified number of paths have been discovered, at which time the Dijkstra's algorithm stops and returns the discovered set of node-disjoint paths to the path-selection algorithm.

Furthermore, shorter paths tend to have a longer residual path lifetime than a longer path. In a discovered path set, the shortest path often (but not always) is the best one. However, choosing the shortest path fails to handle situations in which some of the discovered paths are of equal path length. Therefore, to objectively evaluate their performance, we allow the proposed path-selection algorithms to choose the best path from a set of paths with equal path length.

⁸ Of course, the source and destination nodes are the same for all the paths in \mathcal{P} .

TABLE 3-1: SIMULATION PARAMETERS FOR PATH-SELECTION ALGORITHMS.

| Parameters | Values |
|--|-----------|
| Network Size [m^2] | 800×800 |
| Num. of Nodes | 100 |
| TX Range [m] | 150 |
| Min. Speed [m/s] | 42.5 (5) |
| Max. Speed [m/s] | 57.5 (20) |
| Path-set Size [<i>paths</i>] | 2,3,4,5 |
| Node Velocity Update Interval [<i>sec</i>] | 10 |
| Target Residual Lifetime [<i>sec</i>] | 1.25 (5) |

In the simulation, we assume a network environment, where all node movements are governed by the same mobility model with the same set of mobility attribute values. Thus, each node will have recorded FLL statistics of identical distribution. This allows us to pool the FLL statistics from all the nodes to generate the FLL histogram, avoiding the need to wait for a longer simulation time before a stable FLL histogram is reached.

3.4.1 Performance Metrics

We develop two performance metrics to evaluate the effectiveness of the proposed path-selection algorithms in achieving the two objectives stated in Section 3.2 that define the best path. Each performance metric is defined by comparing the proposed path-selection algorithm with a baseline *random-selection* algorithm to evaluate the ability of the algorithm to select a path that is most likely to meet the *target RPL requirement* τ^9 . The baseline algorithm does not possess any information regarding the reliability of a path; rather, it arbitrarily selects a path from Ψ .

The first performance metric, PM1, is based on two reward schemes, one for the proposed algorithm, and the other for the baseline algorithm. Denote the path selected by the proposed algorithm as i^* . For the k -th path-selection decision made in simulation, the proposed algorithm receives a decision reward D_k as follows:

$$D_k = \begin{cases} 1, & T_{i^*} \geq \tau \\ 0, & o.w. \end{cases}, \quad (3-7)$$

where T_{i^*} denotes the actual residual lifetime of Path i^* . The baseline algorithm makes its selection decision and receives a decision reward E_k as follows:

⁹ τ is a system parameter value set by the application that must be met in order for the data transmissions to commence.

$$E_k = \begin{cases} 0, & T_i^* < \tau, \forall i \in \Psi \\ \frac{1}{|\Psi|}, & o.w. \end{cases}, \quad (3-8)$$

where $|\Psi|$ denotes the cardinality of the path set. Equation (3-8) states that the baseline algorithm receives a non-zero reward as long as there exists at least one path in Ψ that meets the target RPL requirement, regardless of whether its selected path meets that requirement.

Denote N_Ψ as the total number of path-selection decisions made in simulation for a path set of size $|\Psi|$. PM1, denoted as γ_1 , is therefore defined as follows:

$$\gamma_1 = \frac{\sum_{k=1}^{N_\Psi} D_k}{\sum_{k=1}^{N_\Psi} E_k}. \quad (3-9)$$

Note that the denominator of Equation (3-9) is equivalent to the average number of path-selection decisions in which the baseline algorithm finds a path meeting the target RPL requirement in the long run. By this definition, $100(\gamma_1 - 1)\%$ can be viewed as the *performance gain* of the proposed algorithm over the baseline algorithm. The range of γ_1 is $0 \leq \gamma_1 \leq |\Psi|$. The greater the value of γ_1 , the better the performance of the proposed algorithm, and the greater its performance gain over the baseline algorithm.

Note that evaluating the ability of a PSA to select a path that meets the target RPL requirement is not as straightforward as it appears. If τ is set too large, none of the

paths would likely meet this requirement, resulting in $E_k=0$ all the time. Similarly, if it is set too small, all the paths would likely meet this requirement, resulting in $E_k=1/|\Psi|$ all the time. Therefore, some adaptive normalization must be built into the performance metrics in order to compensate for the arbitrary choice of the target RPL requirement and make the evaluation meaningful.

The second performance metric, PM2, evaluates the ability of each proposed path-selection algorithm to choose the path with the longest residual lifetime from Ψ . As in PM1, each proposed algorithm is compared with the baseline algorithm, and their performance is measured as follows. When a path set Ψ is discovered, the proposed and baseline algorithms each select a path that they believe to have the longest RPL, while the simulator continues tracking the lifetimes of all paths in Ψ until the last one breaks. For either the proposed or baseline algorithm, if its chosen path is the last to break, the decision is called a *success*. After conducting path-selection decisions a number N_Ψ of times for a given path-set size $|\Psi|$, we denote $N_{\Psi,s}$ as the number of successes for the proposed algorithm. Therefore, the rate of success by the proposed algorithm is $N_{\Psi,s}/N_\Psi$. For the baseline algorithm, its rate of success is equivalent to the probability of randomly selecting a longest-living path from Ψ , and is given by $1/|\Psi|$. Thus, PM2, denoted as γ_2 , is defined as follows:

$$\gamma_2 = \frac{\frac{N_{\Psi,s}}{N_\Psi}}{\frac{1}{|\Psi|}} = \frac{|\Psi| N_{\Psi,s}}{N_\Psi}. \quad (3-10)$$

The performance gain of the proposed algorithm over the baseline algorithm is given by $100(\gamma_2 - 1)\%$. The range of values for γ_2 is $0 \leq \gamma_2 \leq |\Psi|$, where a larger value

indicates a greater performance gain of the proposed over the baseline algorithm.

3.4.2 Simulation Results

We generate a number of simulation scenarios and apply the two performance metrics to each of the three proposed path-selection algorithms. For each scenario, *10,000* selection decision statistics are collected to compute the performance metrics. In evaluating the performance of PSA3, the conservative grade assignment—explained in Section 3.3.3—is adopted to assign path grades.

Figure 3-1, Figure 3-2, and Figure 3-3 plot the PM1 values of PSA1, PSA2, and PSA3, respectively, as a function of the path-set size $|\Psi|$, for path lengths of 2, 3, and 4[hops]. The node speed is uniformly distributed between $42.5[m/s]$ and $57.5[m/s]$ (i.e., a high-mobility environment)¹⁰, and a target RPL requirement of $1.25[sec]$. It can be seen that with a fixed path length, each proposed PSA is able to achieve a more substantial performance gain over the baseline algorithm when $|\Psi|$ increases. For example, with a path length of three hops, PSA1 obtains a γ_1 value of approximately *1.4* when $|\Psi|=2$ (i.e., a 40% performance gain over the baseline algorithm). As $|\Psi|$ increases to five, γ_1 reaches a value of *2.3* (i.e., a 130% performance gain). This is because as $|\Psi|$ increases, it makes both the proposed and baseline algorithms more difficult to select a path that most likely meets the target RPL requirement. However, the rate of performance degradation by the baseline algorithm is much greater than

¹⁰ Although large for a practical scenario, these values are chosen in order to demonstrate the comparison in path-selection performance between high and low mobility, as will be explained later.

that of the proposed PSAs. These figures also demonstrate that as the path length increases, the performance of each proposed PSA approaches that of the baseline algorithm. This illustrates the fact that in a dynamic network topology, the length of a multi-hop path is a significant factor in determining its lifetime path. Furthermore, the figures demonstrate that of the three PSAs, PSA1 outperforms the other two in meeting the target RPL requirement as $|\Psi|$ increases, and PSA2 slightly performs better than PSA3 for paths of three hops and longer.

Figure 3-4, Figure 3-5, and Figure 3-6 demonstrate the behavior of PM2 in the three PSAs with an increasing path-set size. For PSA1 and PSA2, there exists a discernable trend that the performance of both path-selection algorithms increases while $|\Psi|$ increases, regardless of the path length. This trend can again be attributed to the fact that as the path-set size increases, the rate of degradation for the baseline algorithm is much greater than that for the proposed PSAs. Furthermore, PSA1 outperforms PSA2 and PSA3 in achieving the greatest gain over the baseline algorithm. For example, with $|\Psi|=5$ and a path length of three hops, PSA1 achieves a γ_2 value of approximately 1.55, i.e., a 55% performance gain over the baseline algorithm, whereas PSA2 achieves a γ_2 value of approximately 1.38 (i.e., a 38% performance gain), and the γ_2 value for PSA3 is approximately 1.12 (i.e., a 12% performance gain).

Comparing the performance results for PM1 and PM2, it is clear that generally, all three path-selection algorithms achieve higher γ_1 than γ_2 in absolute values, which represent a greater performance gain over the baseline algorithm in γ_1 than in γ_2 . This observation leads us to conclude that in selecting the best path, it is usually simpler to identify a path that satisfies a path lifetime requirement specified by the application, rather than to find one with the absolute longest path lifetime.

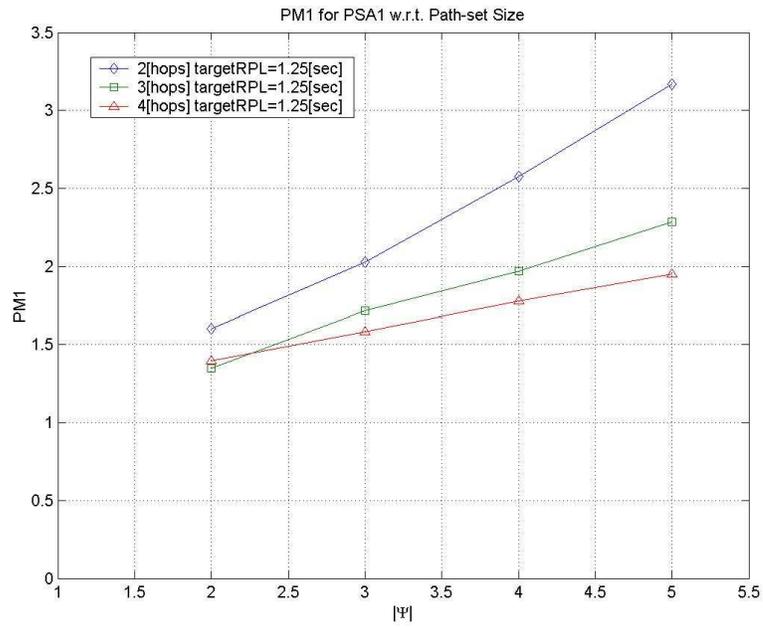


Figure 3-1: PM1 for Path-Selection Algorithm 1.

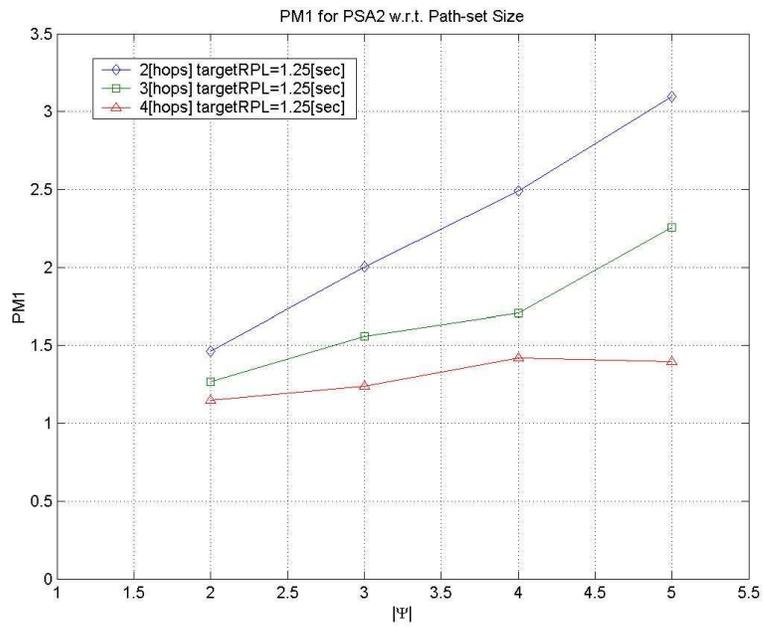


Figure 3-2: PM1 for Path Selection Algorithm 2.

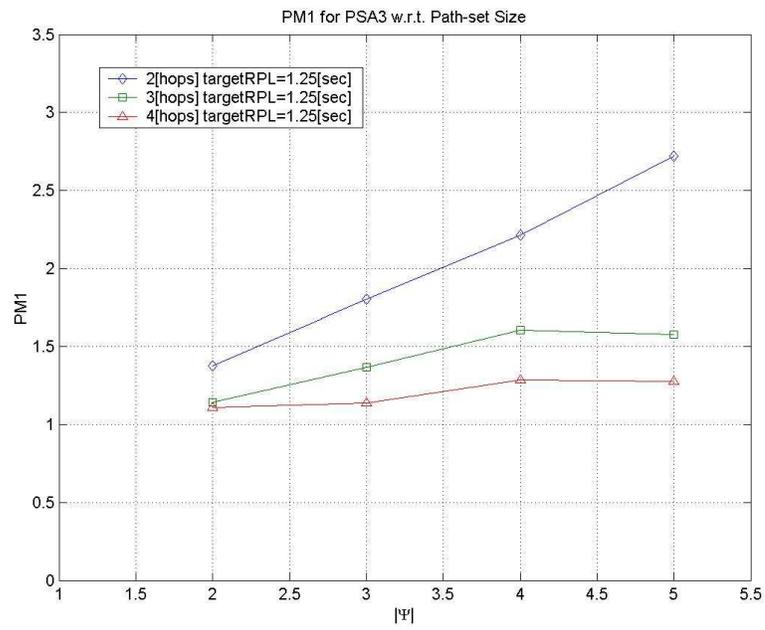


Figure 3-3: PM1 for Path Selection Algorithm 3.

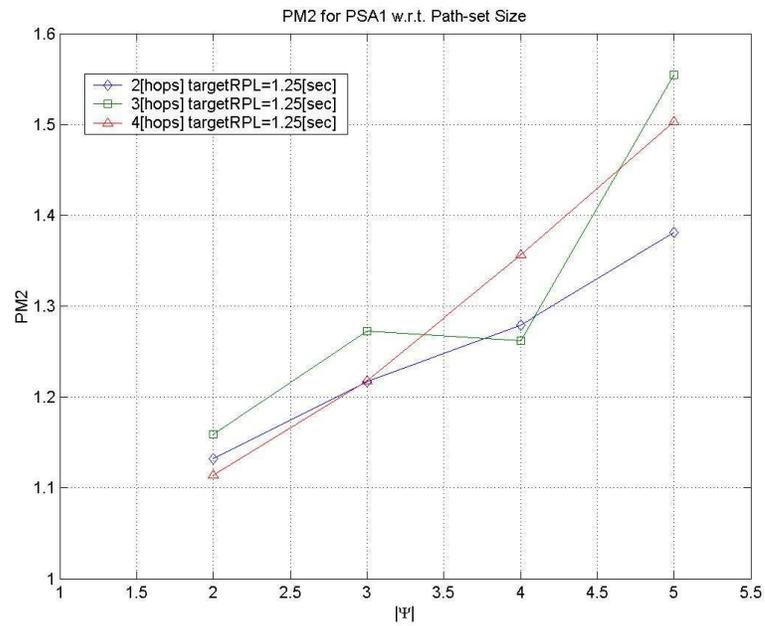


Figure 3-4: PM2 for Path Selection Algorithm 1.

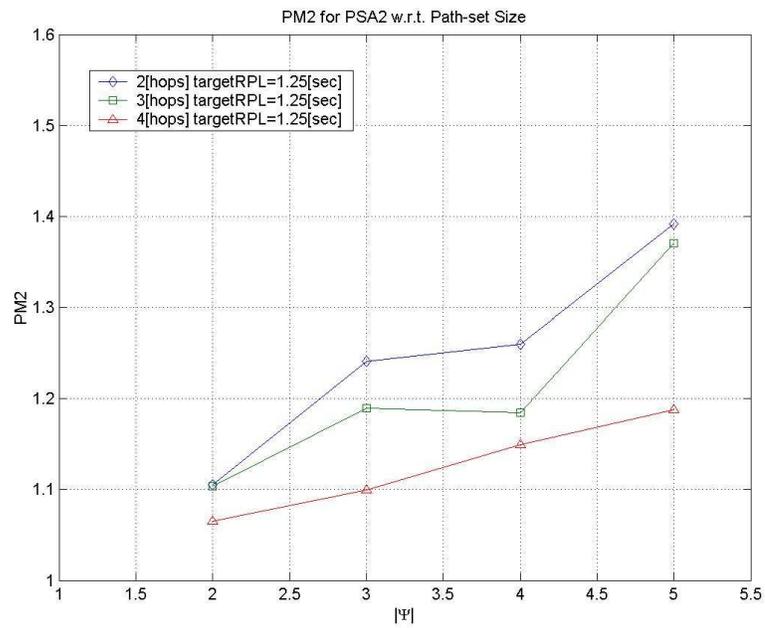


Figure 3-5: PM2 for Path Selection Algorithm 2.

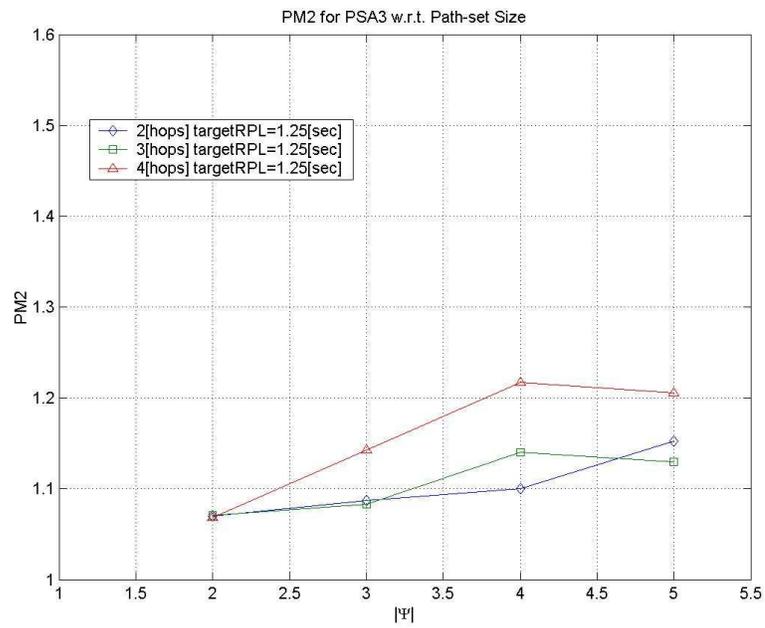


Figure 3-6: PM2 for Path Selection Algorithm 3.

Figure 3-7 illustrates the difference in PM1 when PSA1 is employed in two different mobility scenarios where nodes move according to the RM model. In the high-mobility environment, the average node speed is $50[m/s]$ (illustrated by solid lines in the figure) and target RPL requirement $1.25[sec]$. In the low-mobility environment, the average node speed is $12.5[m/s]$ ¹¹ (illustrated by dash lines), and the target RPL requirement $5[sec]$. Note the four-fold increase in the target RPL requirement, because we must compensate for the 75% decrease in average node speed accordingly so as to make path-selection performances in the two mobility environments comparable. The figure shows that the algorithm's performance is consistently better in the high-mobility environment compared with the results in the low-mobility environment. This is because with a higher average node speed, the average link lifetime becomes shorter. The FLL histogram that PSA1 employs is therefore composed of FLL statistics with a smaller range between the shortest and the longest FLL. This improves the accuracy of the probabilistic computations for PSA1, resulting in a better path-selection performance.

An interesting observation is that PSA2, despite its simplicity, has comparable performance to PSA3 in both the path-selection criteria. Unlike PSA1 and PSA3, PSA2 does not rely on the FLL histogram for its computations. This implies that it does not have to go through a initialization period during which it must collect FLL statistics to obtain a stable FLL histogram. Therefore, PSA2 can be invoked as soon as the network is deployed, without undergoing the initial FLL-collecting phase.

¹¹ The relevant simulation parameters for the average speed in the low-mobility environment are parenthesized in TABLE 3-1.

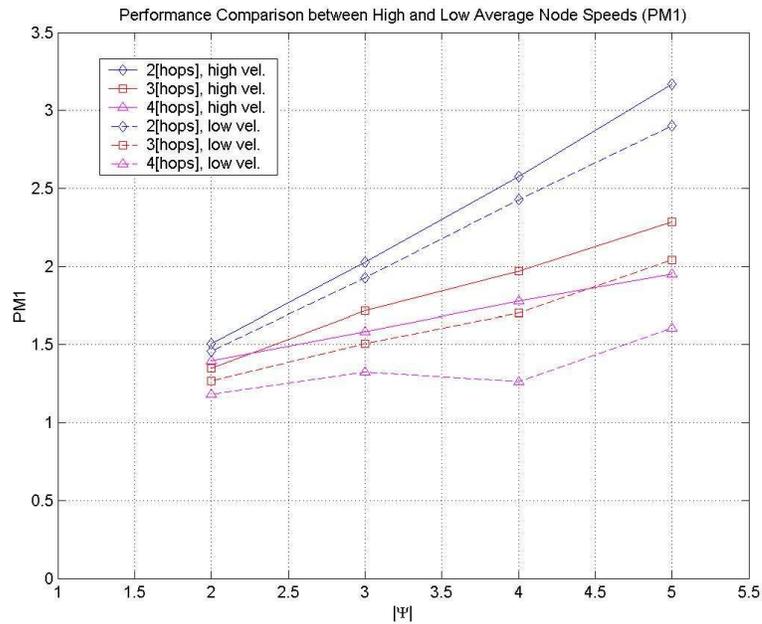


Figure 3-7: Difference in PM1 using PSA1 with different average node speeds.

3.5 Summary

In this chapter, we proposed three link age-based path-selection algorithms that make intelligent path-selection decisions to choose the best path from all available paths between the source and the destination. The quality of the path-selection decision is measured by two objectives: 1) the ability of the path-selection algorithm to choose a path that is most likely to meet a target RPL requirement specified by the upper-layer application, and 2) the ability of the algorithm to choose the path with the longest residual path lifetime among all paths.

The performance of the path-selection algorithms was evaluated via extensive

simulations. We defined two performance metrics to measure the algorithms' path-selection objectives. They allow the proposed algorithms to be compared with each other as well as with a baseline random-selection algorithm that arbitrarily choose a path from all available paths with no regard to path reliability. The simulations demonstrated that the performance of all three PSAs over the baseline algorithm improved as the size of the path set increases. Furthermore, these algorithms performed better in a high-mobility environment than in a low-mobility one. Our simulations showed that while PSA2 and PSA3 had comparable performance among themselves, while PSA1 achieved the best overall performance among the three algorithms.

An immediate application of the best-path selection is the development of multi-path routing schemes. Multi-path routing is being increasingly viewed as a candidate approach to realize QoS in MANET (see, for example, [16], [48], [57], [58], [85], and [86]). Each path in an available path set can be associated with a parameter to indicate the path lifetime, i.e., its reliability. We can exploit this information that would allow us to select multiple paths to maximize the performance of one or more routing metrics. The paths can be ranked in order of their reliability, prompting the multi-path routing algorithm to assign more data packets to the more reliable paths, and fewer data packets to the less reliable ones for transmissions. This would allow the sender of data to spread all its packets across multiple paths, and the potential benefits include a more expedient data delivery time, reduced data loss, and more robust network balancing.

Chapter 4 Mobile-Projected Trajectory RLL Prediction

4.1 Introduction

The PSAs proposed in Chapter 3 employed the link age as the decision metric in selecting the best path from a path set in the MANET. Similar schemes using link age have been proposed in literature (e.g., [19][36][45]). However, link age is not a very reliable metric in choosing the best path. As we have found in Chapter 2, the path lifetime is often uncorrelated with link age, when the nodes in the network move under several popular mobility models such as Random Mobility, Random Waypoint, and Gauss-Markov. Consequently, using link age as a prediction parameter poses a limitation on the performance of the path-selection algorithms.

As will be explained in this chapter, this performance limitation is in fact related to the probability density function of the FLL induced by these mobility models. Namely, the distribution of the FLL exhibits a behavior that can be modeled by the *general exponential distribution*. Since an exponential distribution has the memoryless property, this implies that the knowledge of link age does not give us any new information about how much longer the link will persist before its breakage. Consequently, this property invalidates link age as a reliable measure to accurately predict the residual link lifetime.

In this chapter, we focus on studying the problem of link lifetime prediction in MANET. Link lifetime is a quantity impacted by a number of factors such as node

mobility, physical obstruction, noise, and weather conditions. Therefore it is very difficult to mathematically model the link lifetime that takes into account all such factors. Of these factors, node mobility is a major contributor to the dynamics of network topology and wields considerable influence over the behavior of link lifetime. Since we are interested in the effects of mobility, we shall therefore model link lifetime induced only by node mobility. That is, a communications link is up when the Euclidean distance between two mobile nodes is less than or equal to the transmission range. Similarly, when the Euclidean distance between the two nodes is greater than the transmission range, the link breaks.

Two approaches of studying link lifetime prediction are adopted by the research community: analytical modeling and simulations. As an example of analytical modeling, McDonald and Znati [53] proposed an Ad Hoc Mobility Model that assumes an exponentially-distributed epoch, during which a node moves at constant velocity. The authors then analytically derived the conditional probability that a link would be up at a time t_0+t given that it is up at time t_0 . Jiang, He, and Rao [40] furthered this work by proposing a link-availability prediction algorithm to compute the conditional probability that a link remains up until t_0+t given it is up at t_0 . They also developed a routing metric for a multi-hop path where the path availability is determined by constituent link availability derived from precise range measurements provided by the GPS. Tseng, Li, and Chang [84] proposed an analytical model for computing the multi-hop path lifetime based on a discrete-time, random walk mobility model. The path lifetime is determined from each constituent link lifetime, which is modeled as a state and transitions to a different state at every discrete time instant with some probability.

Among the published works that employ simulations, Korsnes et al. [45] modeled the link lifetime as a heavy-tailed distribution from the observation that some link

lifetimes can be extremely long. They also proposed a prediction criterion whereby a link with an older link age is shown to have a longer expected remaining lifetime than a younger age. Gerharz, de Waal, Frank, and Martini [21] employed a histogram of FLL simulation statistics to probabilistically predict link lifetime. Subsequently they proposed several methods of finding stable paths based on some link-age-based criteria [22]. Toh [82] proposed the Associativity-Based Routing (ABR) protocol that considers a link stable if it has been up longer than a threshold period in a network where node movements are governed by an in-door mobility model. Dube, Rais, Wang, and Tripathi [18] proposed a routing protocol in which the link stability is classified as strongly-connected if the received signal strength is strong and persists for a threshold time period, and as weakly-connected otherwise. Bai, Sadagopan, Krishnamachari, and Helmy [3] collected link- and path-lifetime statistics via extensive simulations to derive their respective statistical PDFs under four mobility models, and concluded that for a path length of two hops or more and with medium to high node speeds, the path lifetime can be modeled as an exponential distribution.

In this chapter, we propose a novel, distance measurement-based algorithm, called *Mobile-Projected Trajectory* (MPT), which computes the residual link lifetime by periodically measuring distances between the nodes to estimate the movement trajectory. MPT is based on linear curve fitting that aims to mitigate the negative effects of measurement errors. Furthermore, the algorithm does not require nodes to possess knowledge of node velocity and location before performing its computations. This is particularly attractive to mobile nodes that are not equipped with sophisticated hardware and software mechanisms. To account for the frequent occurrences of velocity change during the link lifetime, we augment the MPT with a velocity-change detection test to improve the accuracy of link lifetime prediction.

A number of proposals employing distance measurements for various objectives

have been published in the literature. Su, Lee, and Gerla [79] computed the link expiration time between two neighboring nodes, with distance measurement computed from GPS-provided velocity and location information. Savvides, Han, and Strivastava [74] employed distance measurements to address the problem of sensor localization in a stationary WSN in which a few beacon nodes, which possess precise position information provided by either pre-deployment manual configuration or GPS.

Two major differences distinguish our work from the others. Firstly, many proposals that employ distance measurements assume a network with a very stable topology (e.g., [64][74]). Our proposed MPT algorithm, on the other hand, addresses the problem of predicting the RLL in a network with a very dynamic topology. Secondly, many proposals assume that some or all of the nodes in the network possess precise location and/or velocity information (e.g., [64][73]) via external infrastructure such as the GPS. Such information in practice may be expensive to acquire since nodes need be equipped sophisticated hardware and software. Our algorithm can be implemented without any external support to any node in the network, and therefore does not rely on such expensive information. For example, a mobile node might not possess the knowledge of its current speed and direction. Likewise, it may be aware of the presence of another node in its neighborhood, but does not know the neighbor's orientation (i.e., relative position) with respect to itself. The challenge of our problem is thus to achieve a robust RLL prediction performance with only limited available information.

The remainder of the chapter is organized as follows. Section 4.2 performs a statistical test to verify the modeling of the full link lifetime as an exponential distribution under the mobility models studied in Chapter 3. Section 4.3 proves the necessary condition to yield a unique RLL solution with periodical distance measurements, and discusses the effects of measurement errors on the accuracy of

predicted link lifetime. Section 4.4 presents the Mobile-Projected Trajectory algorithm. Section 4.5 takes into consideration the occurrences of velocity change during the link lifetime, and augment the MPT with a simple velocity-change detection test to propose an enhanced algorithm called MPT-VCD. Section 4.6 evaluates the performance of MPT and MPT-VCD through extensive simulations. Section 4.7 concludes the chapter.

4.2 Exponential Behavior of Link Lifetime

We first explain the performance limitation of the link-age-based algorithms by examining the PDFs of full link lifetime in the three mobility models from Chapter 2. For each mobility model, the statistical PDF of the FLL in each mobility model takes the shape as shown in Figure 4-1, which for the purpose of illustration is generated under the RWP model. In the figure, the probability density initially rises to a peak before decreasing as FLL further increases. We define the range of FLLs from zero to the *mode*, where the peak is located, as the *under-mode range*, and the range of FLLs greater than the mode as the *over-mode range*. The objective is to demonstrate that the FLL in the over-mode range can be modeled as a *general exponentially-distributed* random variable, whose memoryless property implies that reliable RLL predictions are not possible even given the knowledge of a link age that falls into this range.

The PDF and the cumulative distribution function (CDF) of the general exponential distribution are given as follows:

$$f_x(x) = \frac{1}{\lambda} e^{-\frac{x-\mu}{\lambda}}, F_x(x) = 1 - e^{-\frac{x-\mu}{\lambda}}, \lambda > 0, x > \lambda, \quad (4-1)$$

where μ and λ denote the *location* and *scale* parameters, respectively. To verify the validity of modeling the FLL in the over-mode range as a generic exponential distribution, we collect FLL statistics from a Monte Carlo simulation, curve-fit the empirical distribution of the FLL in the over-mode range to the analytical distribution, and evaluate the goodness of the fit by the *Kolmogorov-Smirnov (K-S) test* [17].

In the simulation, movements of the nodes are governed by the RWP model in a torus. The transmission range R for each node is 150[m], and the node speed $V \sim U(5, 20)[m/s]$. We collect 60,000 FLL statistics from a single node to generate the statistical PDF and CDF, as shown in Figure 4-2 and Figure 4-3, where the mode is located at approximately 13[sec]. The two parameters of the general exponential distribution are found to be $\lambda = 11.63[sec]$ and $\mu = 7.66[sec]$, with which the analytical PDF and CDF are also plotted in the figures. By applying the *K-S test*, we find the D-statistic to be 0.03387, confirming a very good fit between the analytical and simulated results for the over-mode range. Further simulations with the RM and G-M mobility models have also demonstrated a good fit of the FLL in the over-mode range as a generic exponential distribution.

Figure 4-3 shows the CDF value at the mode is 0.26. This means that, given the transmission range and speed distribution, 74% of all link-age-based RLL predictions could not be made reliably. In fact, the mode is approximately equal to the ratio of the transmission range to the average node speed. As the transmission range increases and/or node speed decreases, the mode will shift towards the right, decreasing the over-mode range, and the prediction performance would improve.

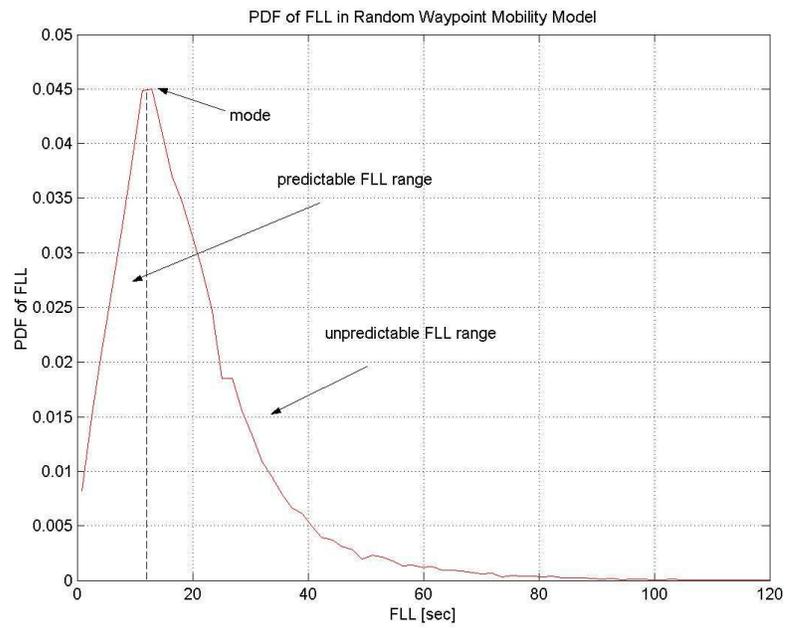


Figure 4-1: Statistical PDF of FLL under the RWP, RM, and G-M models.

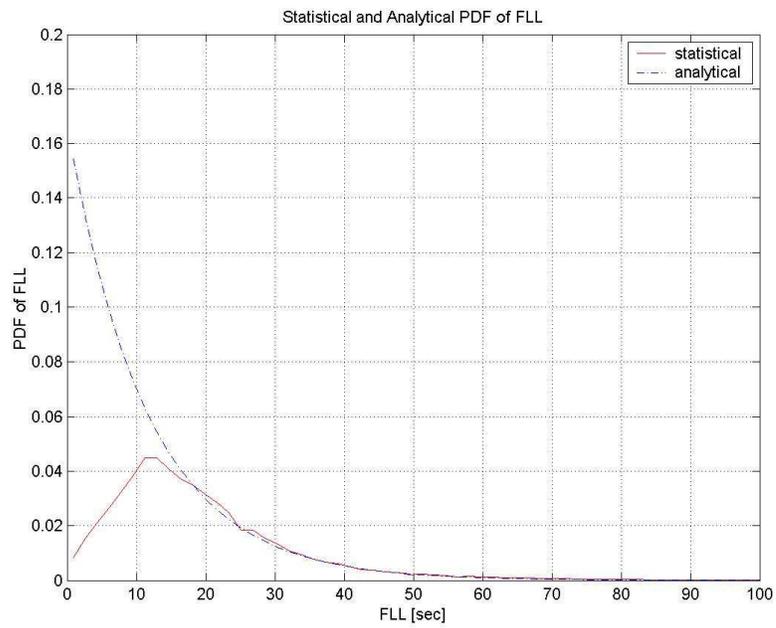


Figure 4-2: Analytical and statistical PDFs of FLL under the RWP.

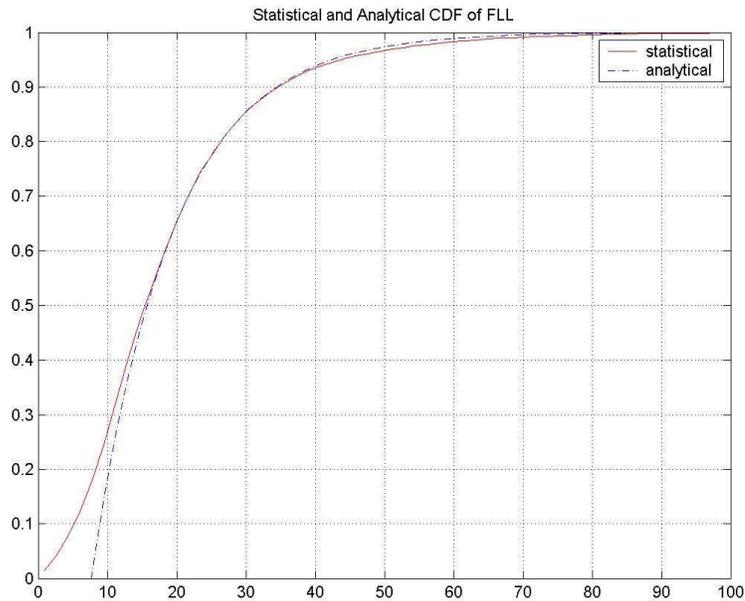


Figure 4-3: Analytical and statistical CDF of FLL under the RWP.

We have thus confirmed the validity of modeling the FLL in the over-mode range as exponentially-distributed to explain the limited effectiveness of the link-age-based RLL prediction algorithm. We now take an alternative approach of predicting the RLL based on measured distances. To begin with, we present the premises for such an approach.

4.3 Premises of Distance Measurement-Based RLL Prediction

In this section, we first develop a two-node link model upon which the proposed algorithm will be presented. We then present a theorem that states that, without any

knowledge of node velocity and position, it is necessary to measure at least four distances to obtain a unique RLL solution. Furthermore, we investigate the effects of measurement errors on RLL prediction. These serve as the basis of the MPT algorithm that will be proposed in Section 4.4.

4.3.1 Two-Node Link Model

We define a link model between two nodes, Node 1 and Node 2, as follows. Each node neighborhood is circular with the radius equal to the transmission range R . A link comes into being when the distance between the two nodes is less than or equal to R . This protocol model makes relevant mathematics more tractable, and has been employed in a number of works in the literature [29]. Without loss of generality, let Node 1 be the node to measure distances between itself and Node 2 as Node 2 moves in Node 1's neighborhood. Furthermore, neither node possesses the knowledge of node velocity, location, and its orientation (i.e., relative position) with respect to each other.

Each node is equipped with the following three mechanisms, two of which have already been employed by the link age-based prediction algorithms in Chapter 3. Firstly, a node is equipped with a beaconing mechanism that periodically emits an ID signal and broadcasts it to the other node in its neighborhood. Node 1 can thus hear this signal emitted by Node 2 if and only if the distance between the two nodes is less than or equal to R , and assumes there exists a link from Node 2 to Node 1. We assume a link is symmetric. Secondly, each node is equipped with a timer that keeps track of the lifetime of the link by listening to the periodic ID signal emitted from the other

node. The timer gives the node the current knowledge of link age and full link lifetime. Thirdly, each node is equipped with a ranging mechanism to compute distance measurements between the two nodes. A few well-known ranging techniques include Time-of-Arrival (ToA) [13], Angle-of-Arrival (AoA) [60], and Time-Difference-of-Arrival (TDoA) [74].

One type of technology suitable for ranging purposes is the *ultra-wideband* (UWB) communications, the feasibility of which has been extensively explored in the literature (e.g., [12][13][19]). The extremely short temporal UWB pulse allows it to mitigate the negative effects of multi-path fading in the wireless medium. Furthermore, UWB-based ranging can also be made very cost-effective [19]. In the link model, we employ the same UWB pulses for both ID signaling and ranging in each node. This combination imposes no additional communications overhead on ranging. However, the distance measurements contain measurement errors that must be taken into consideration. We shall investigate their effects on the prediction of RLL in Section 4.3.3.

4.3.2 Minimal Number of Distance Measurements

Consider our link model between Node 1 and Node 2, in which both nodes move at constant velocity during the entire link lifetime. For purposes of illustration, we tentatively assume the distance measurements are error-free. Node 1 measures the first distance, denoted as d_0 , at time t_0 when Node 2 enters the transmission range of Node 1. Subsequently at times $t_0 + \Delta t$, $t_0 + 2\Delta t$, and $t_0 + 3\Delta t$, where Δt denotes the *sampling*

period, three more distance measurements, d_1 , d_2 , and d_3 , are measured, as illustrated in Figure 4-4.

Let d_{min} denote the minimal distance between the nodes given the relative direction of Node 2 with respect to Node 1. It can be seen that there exist exactly three possible scenarios for having four periodical distance measurements during the link lifetime:

S1: d_0 and d_1 measured before d_{min} ; d_2 and d_3 after d_{min} ;

S2: d_0 , d_1 , and d_2 measured before d_{min} ; d_3 after d_{min} ; and

S3: d_0 , d_1 , d_2 , and d_3 all measured before d_{min} .

No other scenarios with four periodical distance measurements are possible. If only d_0 were measured before d_{min} , this would result in at most three distances (i.e., d_0 , d_1 , and d_2) measured before the link breaks. Similarly, it is not possible to measure all four distances after d_{min} is reached.

Define the state in which the two nodes are moving towards each other at the time d_2 is measured as the *approaching state* (Figure 4-4, as described by S2 and S3), and the state in which they are moving away from each other when d_2 is measured as the *receding state* (Figure 4-5, as described by S1). We present the following theorem for computing the RLL based on distance measurements (details of the proof are presented in Appendix A):

Theorem. With our two-node link model, it is necessary to employ four periodically measured distances to uniquely compute a solution for the residual link lifetime.

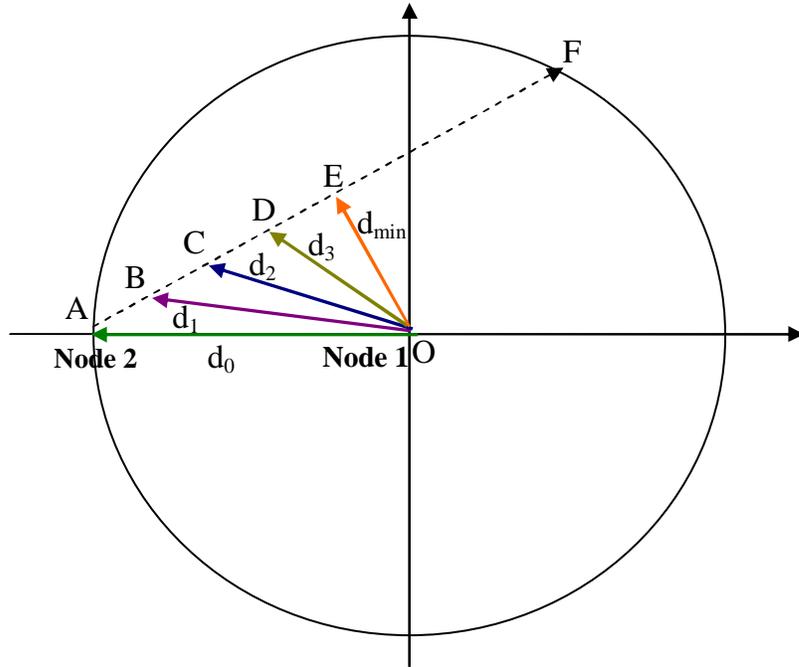


Figure 4-4: Approaching state at d_2 .

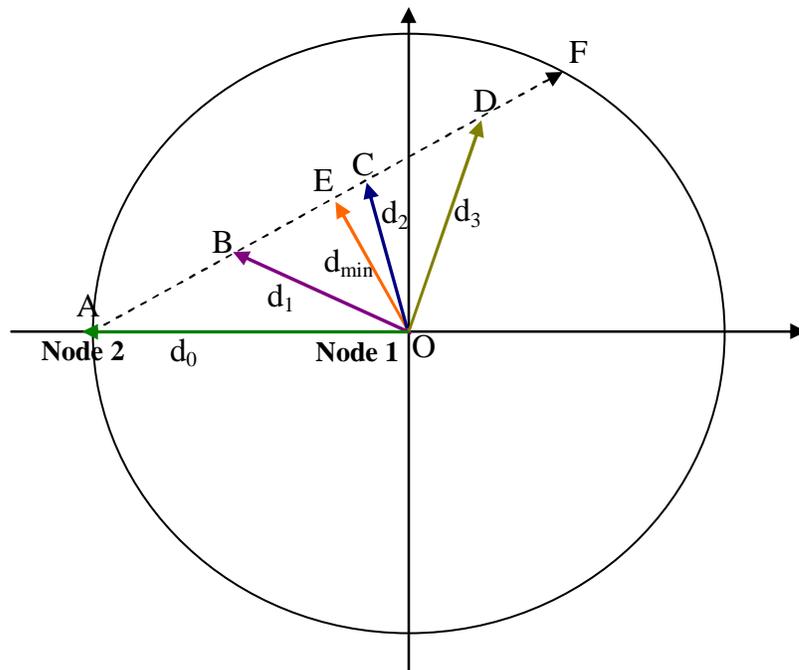


Figure 4-5: Receding state at d_2 .

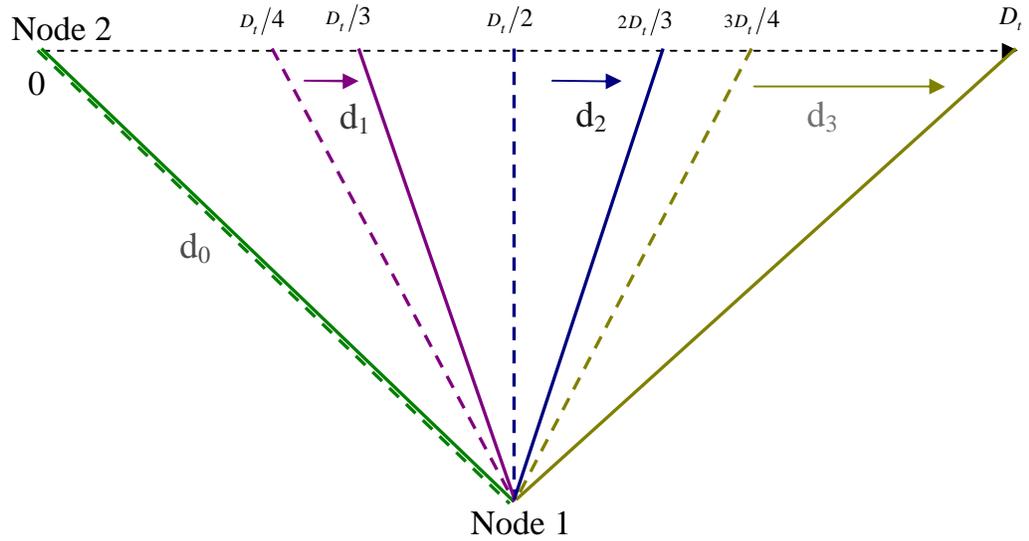


Figure 4-6: S1 and the range of values each d_i can take.

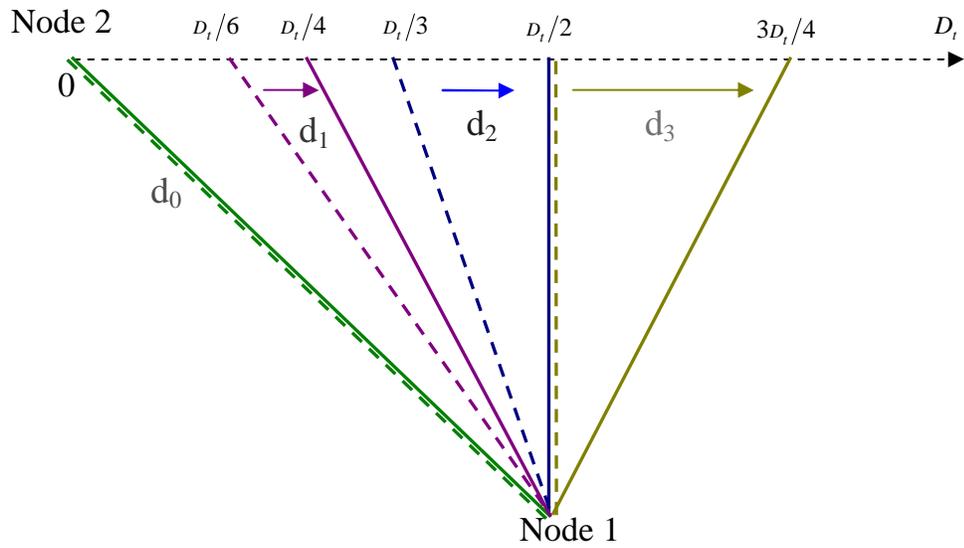


Figure 4-7: S2 and the range of values each d_i can take.

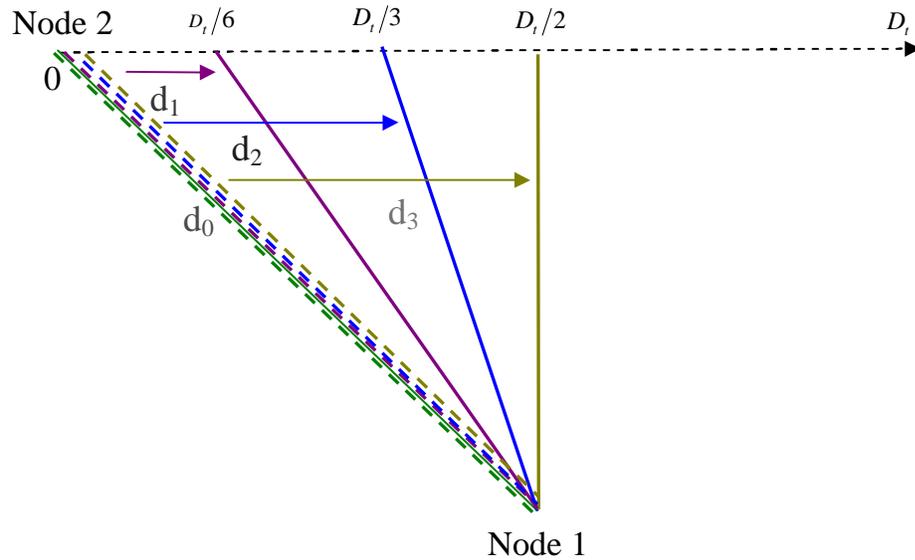


Figure 4-8: S3 and the range of values each d_i can take.

4.3.3 Effects of Measurement Errors

We investigate how distance measurement errors affect the accuracy of RLL prediction. We begin by deriving the formulas to compute the RLL with the four error-free d_i 's, which will then be replaced with the distance measurements that contain measurement errors.

To compute the RLL when two nodes are in approaching state at d_2 , let $b=AB=BC=CD$, and $a=CE$ in Figure 4-4, since both the node velocity and Δt are constant. A system of equations can be established as follows:

$$\begin{cases} (a+2b)^2 + d_{\min}^2 = d_0^2 \\ (a+b)^2 + d_{\min}^2 = d_1^2, \\ a^2 + d_{\min}^2 = d_2^2 \end{cases}, \quad (4-2)$$

where a and b are computed as follows:

$$\begin{cases} a = \frac{-d_0^2 + 4d_1^2 - 3d_2^2}{2\sqrt{2(d_0^2 - 2d_1^2 + d_2^2)}} \\ b = \sqrt{\frac{d_0^2 - 2d_1^2 + d_2^2}{2}} \end{cases}. \quad (4-3)$$

The residual link lifetime that is computed $\tau[sec]$ after d_3 is measured (i.e., at time $t_0 + 3\Delta t + \tau$) is:

$$RLL(t_0 + 3\Delta t + \tau) = \Delta t + 2\Delta t(a/b) - \tau. \quad (4-4)$$

Note that only d_0 , d_1 , and d_2 are involved in the actual computation of the RLL as shown in Equations (4-2) and (4-3).

Likewise, to compute the RLL when the nodes are in receding state at d_2 , a system of equations similar to Equation (4-2) is established from Figure 4-5 as follows:

$$\begin{cases} (2b-a)^2 + d_{\min}^2 = d_0^2 \\ (b-a)^2 + d_{\min}^2 = d_1^2, \\ a^2 + d_{\min}^2 = d_2^2 \end{cases}, \quad (4-5)$$

where:

$$\begin{cases} a = \frac{d_0^2 - 4d_1^2 + 3d_2^2}{2\sqrt{2(d_0^2 - 2d_1^2 + d_2^2)}} \\ b = \sqrt{\frac{(d_0^2 - 2d_1^2 + d_2^2)}{2}} \end{cases} . \quad (4-6)$$

The residual link lifetime that is computed $\tau[\text{sec}]$ after d_3 is measured is:

$$RLL(t_0 + 3\Delta t + \tau) = \Delta t - 2\Delta t(a/b) - \tau . \quad (4-7)$$

We now replace d_i by distance measurements with measurement errors, denoted as \hat{d}_i , in the above equations, and observe how they affect the *RLL prediction inaccuracy*, defined as follows:

$$\eta(t) = \frac{|RLL(t) - \hat{RLL}(t)|}{RLL(t)} \cdot 100\% , \quad (4-8)$$

where $RLL(t)$ and $\hat{RLL}(t)$ denote the true and predicted RLL at time t , respectively.

There are two types of measurement errors: *systematic* and *random*. A systematic error results from hardware mis-calibration of the ranging equipment, e.g., the imperfect synchronization between the transmitter and the receiver [13]. We model it as a constant offset Z , i.e., $\hat{d}_i = d_i + Z$, $\forall i = 0, \dots, 3$. The effects of systematic errors on $\eta(t)$ are demonstrated by simulation shown in Figure 4-9, which plots the average prediction inaccuracy $\bar{\eta}$ with respect to Z . Key simulation parameters include the transmission range $R=50[m]$, $\Delta t=0.5[\text{sec}]$, and $Z=\{-5, -4, \dots, 4, 5\}[m]$. The node speed is $V \sim U(5, 20)[m/s]$, and node direction $\theta \sim U(0, 2\pi)$. For each value of Z , 50,000 $\eta(t_0+3\Delta t)$ statistics are collected to compute $\bar{\eta}$.

It can be seen in Figure 4-9 that as the magnitude of Z increases, $\bar{\eta}$ increases. However, the rate of increase for $\bar{\eta}$ is smaller than that for Z . For example, at $|Z|=5[m]$, i.e., a $10\%R$ error that is well below the precision achievable in today's ranging equipment, $\bar{\eta}$ is only about 6% to 7%. With a more realistic, smaller choice of Z , the prediction inaccuracy is even smaller. The results in Figure 4-9 therefore demonstrate that the effects of systematic errors on the RLL prediction inaccuracy are insignificant.

Random errors arise from unpredictable phenomena such as channel fading and thermal noise. The effects of random measurement errors on prediction inaccuracy are demonstrated with the following computations that represent the extreme-case scenarios of random errors. With the given speed and direction distributions in the link model, we periodically measure four distances $\hat{d}_i = d_i \pm z, \forall i = 0, \dots, 3$. \hat{d}_i 's replace d_i 's in Equation (4-2) through Equation (4-6) to compute the RLL as if the measurements are error-free, and $\eta(t_0+3\Delta t)$ is computed with Equation (4-8).

Since only the first three distance measurements are involved in the actual computations of the predicted RLL, there are eight possible solutions for $\eta(t_0+3\Delta t)$. TABLE 4-1 lists these solutions for a relative speed $v=3[m/s]$ and relative direction¹² $\phi=0^\circ$. Let $z=0.3\%R$. The notations in the "Measurement Errors" column of the table denote the signed triples of z in $\hat{d}_i, \forall i=0,1,2$. For instance, "+-+" denotes $\hat{d}_0 = d_0 + z, \hat{d}_1 = d_1 - z, \text{ and } \hat{d}_2 = d_2 + z$. As can be seen from the table, six out of the eight triples of random measurement errors result in large prediction inaccuracy ranging from 46.05% to 282.56%, dominating the impact on the prediction accuracy of RLL.

¹² Note that this direction is chosen because our study has shown that a smaller relative direction produces lower prediction inaccuracy. In other words, $\phi=0^\circ$ computes a more favorable η .

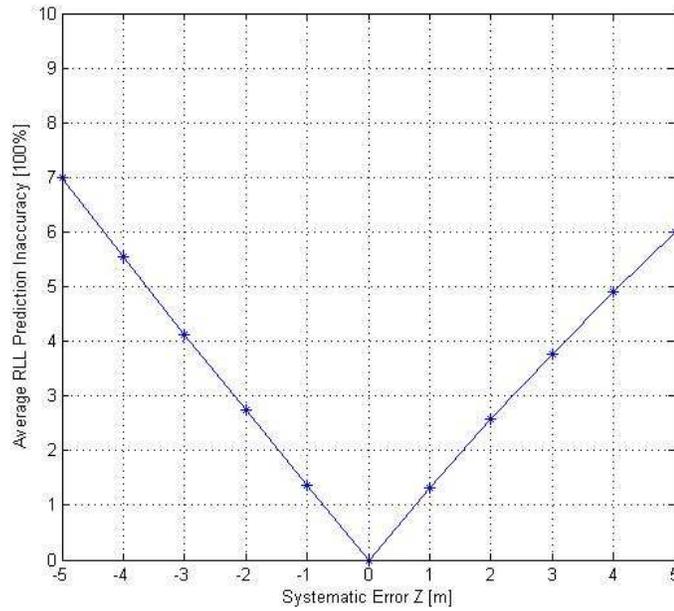


Figure 4-9: Average RLL prediction inaccuracy vs. systematic error.

TABLE 4-1: MEASUREMENT ERRORS AND PREDICTION INACCURACY

| Measurement Errors | Prediction Inaccuracy [%] |
|--------------------|---------------------------|
| +++ | 0.34 |
| ++- | 282.86 |
| + - + | 64.07 |
| + -- | 46.54 |
| - ++ | 63.78 |
| - +- | 283.56 |
| -- + | 46.05 |
| --- | 0.34 |

The preceding measurement error analysis shows that the effects of systematic errors on RLL prediction inaccuracy are negligible, whereas random errors have a

more significant impact, and must be taken into consideration when developing a distance measurement-based RLL prediction algorithm. With this reasoning, in presenting the proposed algorithm in the next section, we consider only random errors.

4.4 Mobile-Projected Trajectory Algorithm

4.4.1 Operations of MPT

The operations of MPT can best be illustrated in Figure 4-10 with the error-free d_i 's. As Node 2 enters Node 1's transmission range at time $t_0=0$, Node 1 measures d_0 , and establishes a Cartesian coordinate system, with Node 2 at its origin and Node 1 at the position $(d_0, 0)$. The coordinates of d_0 are defined as $(x_0, y_0)=(0,0)$. Subsequently, at times Δt , $2\Delta t$, and $3\Delta t$, Node 1 measures d_1 , d_2 , and d_3 respectively. We define the entire duration, at the beginning and end of which the first and the last distances are measured to perform RLL computations by the MPT, as the *acquisition time*.

We now show that with four periodical distance measurements, MPT is able to compute the (x_i, y_i) coordinates of d_i , $\forall i=1, 2, 3$, and the estimated relative trajectory between the two nodes, denoted as $y=\alpha x$. The (x_i, y_i) coordinates are equally spaced on the x - and y -axis due to node velocity and Δt being constant, i.e.:

$$\begin{cases} (x_0, y_0) = (0, 0) & (x_1, y_1) = (x_1, \alpha x_1) \\ (x_2, y_2) = (2x_1, 2\alpha x_1) & (x_3, y_3) = (3x_1, 3\alpha x_1) \end{cases} \quad (4-9)$$

Since $(d_0 - x_i)^2 + y_i^2 = d_i^2$, $\forall i=1, 2, 3$, with proper substitutions from Equation (4-9), we can establish the following system of three equations and two unknowns:

$$\begin{cases} (x_1 - d_0)^2 + \alpha^2 x_1^2 = d_1^2 \\ (2x_1 - d_0)^2 + 4\alpha^2 x_1^2 = d_2^2, \\ (3x_1 - d_0)^2 + 9\alpha^2 x_1^2 = d_3^2 \end{cases} \quad (4-10)$$

Where x_1 can be found as follows:

$$x_1 = (3d_0^2 - 4d_1^2 + d_2^2) / 4d_0 = (8d_0^2 - 9d_1^2 + d_3^2) / 12d_0. \quad (4-11)$$

After re-arranging Equation (4-11), it can be seen that in order for the coordinates of the four distance measurements to be equally spaced, they must satisfy the following equality:

$$d_0^2 - 3d_1^2 + 3d_2^2 - d_3^2 = 0. \quad (4-12)$$

Although Equation (4-12) is always satisfied with the error-free d_i 's, it is not so with \hat{d}_i 's, denoted as $\hat{d}_i = d_i + \varepsilon_i$, where ε_i denotes the i -th distance measurement error. A challenge for MPT is therefore to find the *estimated distance values*, denoted as \tilde{d}_i , such that \tilde{d}_i 's will satisfy Equation (4-12). Let $\tilde{d}_i = \hat{d}_i + e_i$, $\forall i=0, \dots, 3$, where e_i denotes the i -th *estimated measurement error*. e_i can be solved by formulating a minimization problem as follows:

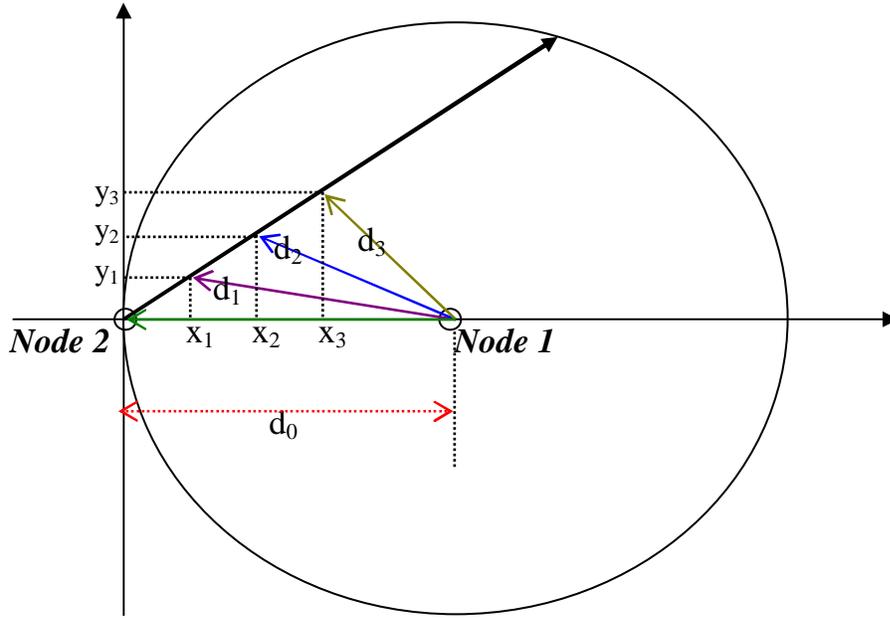


Figure 4-10: The MPT Cartesian coordinate system.

$$\text{Minimize: } \sum_{i=0}^3 e_i^2$$

$$\text{Subject to: } \tilde{d}_0^2 - 3\tilde{d}_1^2 + 3\tilde{d}_2^2 - \tilde{d}_3^2 = 0, \quad (4-13)$$

where the constraint function follows from Equation (4-12) with d_i replaced by \tilde{d}_i . It is solved by using the *Lagrange multiplier* λ as follows:

$$f(\underline{e}, \lambda) = \sum_{i=0}^3 e_i^2 + \lambda \left[(\hat{d}_0 + e_0)^2 - 3(\hat{d}_1 + e_1)^2 + 3(\hat{d}_2 + e_2)^2 - (\hat{d}_3 + e_3)^2 \right], \quad (4-14)$$

where \tilde{d}_i is replaced with $\hat{d}_i + e_i$. Setting the gradient of $f(\underline{e}, \lambda)$, ∇f , to zero allows us to compute e_i :

$$\nabla f = \begin{cases} \frac{\partial}{\partial e_0} f(\underline{e}, \lambda) = 2e_0 + \lambda [2(\hat{d}_0 + e_0)] = 0 \Rightarrow e_0 = -\frac{\lambda}{1+\lambda} \hat{d}_0 \\ \frac{\partial}{\partial e_1} f(\underline{e}, \lambda) = 2e_1 + \lambda [-6(\hat{d}_1 + e_1)] = 0 \Rightarrow e_1 = \frac{3\lambda}{1-3\lambda} \hat{d}_1 \\ \frac{\partial}{\partial e_2} f(\underline{e}, \lambda) = 2e_2 + \lambda [6(\hat{d}_2 + e_2)] = 0 \Rightarrow e_2 = -\frac{3\lambda}{1+3\lambda} \hat{d}_2 \\ \frac{\partial}{\partial e_3} f(\underline{e}, \lambda) = 2e_3 + \lambda [-2(\hat{d}_3 + e_3)] = 0 \Rightarrow e_3 = \frac{\lambda}{1-\lambda} \hat{d}_3 \\ \frac{\partial}{\partial \lambda} f(\underline{e}, \lambda) = (\hat{d}_0 + e_0)^2 - 3(\hat{d}_1 + e_1)^2 + 3(\hat{d}_2 + e_2)^2 - (\hat{d}_3 + e_3)^2 = 0 \end{cases} .$$

Next, we substitute e_i into $\partial f(\underline{e}, \lambda)/\partial \lambda$, leading to the following sixth-degree polynomial equation:

$$T_6 \lambda^6 + T_5 \lambda^5 + T_4 \lambda^4 + T_3 \lambda^3 + T_2 \lambda^2 + T_1 \lambda + T_0 = 0, \quad (4-15)$$

where the coefficients are given by the following expressions:

$$\begin{aligned} T_0 &= \hat{d}_0^2 - 3\hat{d}_1^2 + 3\hat{d}_2^2 - \hat{d}_3^2 \\ T_1 &= -2\hat{d}_0^2 - 18\hat{d}_1^2 - 18\hat{d}_2^2 - 2\hat{d}_3^2 \\ T_2 &= -17\hat{d}_0^2 - 21\hat{d}_1^2 + 21\hat{d}_2^2 + 17\hat{d}_3^2 \\ T_3 &= 36(\hat{d}_0^2 + \hat{d}_1^2 + \hat{d}_2^2 + \hat{d}_3^2) \\ T_4 &= 63\hat{d}_0^2 + 51\hat{d}_1^2 - 51\hat{d}_2^2 - 63\hat{d}_3^2 \\ T_5 &= -162\hat{d}_0^2 - 18\hat{d}_1^2 - 18\hat{d}_2^2 - 162\hat{d}_3^2 \\ T_6 &= 81\hat{d}_0^2 - 27\hat{d}_1^2 + 27\hat{d}_2^2 - 81\hat{d}_3^2 \end{aligned}$$

Of the six roots of λ , four are complex-valued. The feasible solution thus must be one of the two remaining real-valued roots. Since distance measurement errors are assumed to be small compared with the transmission range, the smaller of the two real roots is the desired solution. By substituting λ into ∇f , we solve for the e_i 's to yield

the \tilde{d}_i 's and \tilde{x}_1 from Equation (4-11). We then compute the *MPT-estimated trajectory slope* $\tilde{\alpha}$ by employing the first equation in Equation (4-10):

$$\tilde{\alpha} = \sqrt{\frac{\tilde{d}_1^2 - (\tilde{x}_1 - \tilde{d}_0)^2}{\tilde{x}_1^2}}. \quad (4-16)$$

With \tilde{x}_1 and $\tilde{\alpha}$ now known, all the $(\tilde{x}_i, \tilde{y}_i)$ coordinates of \tilde{d}_i can be easily derived via Equation (4-9). We have therefore obtained the MPT-estimated trajectory $\tilde{y} = \tilde{\alpha}\tilde{x}$.

The MPT-estimated trajectory $\tilde{y} = \tilde{\alpha}\tilde{x}$ is optimal in the sense that it minimizes the sum of the estimated measurement errors squared. On the other hand, there could exist other trajectories whose slopes are even closer to the true trajectory slope than $\tilde{\alpha}$. In Section 4.4.3, we shall derive the theoretical upper bound (i.e., the worst case) of the RLL prediction inaccuracy attainable by the best distance measurement-based RLL-prediction algorithm.

Following the above derivations, we can compute the RLL that is computed τ seconds after \hat{d}_3 is measured:

$$R\hat{L}L(t_0 + 3\Delta t + \tau) = \frac{2\tilde{d}_0\Delta t}{1 + \tilde{\alpha}^2} \sqrt{\frac{1 + \tilde{\alpha}^2}{\tilde{x}_1^2 + \tilde{y}_1^2}} - 3\Delta t - \tau. \quad (4-17)$$

4.4.2 MPT with More Distance Measurements

By formulating the problem as a minimization function shown in Equation (4-13), the MPT computes the relative trajectory by linear curve-fitting with four distance measurements. Therefore, if more distance measurements are available, the algorithm is able to estimate a relative trajectory whose slope is closer to the true trajectory slope. To verify this assertion, we have formulated several variants of MPT that employ N distance measurements, denoted as MPT- N , where $N=5, 6, 7$, and 8 . The formulation of the minimization problem for each of them and its solutions are presented in Appendix B. As we shall see in the performance evaluation in Section 4.6, more distance measurements lead to an improving performance of the algorithm. We will also discuss the trade-off between the performance level and the longer acquisition time it takes to measure more distances.

4.4.3 Theoretical Upper Bound on RLL Prediction Inaccuracy

While deriving the operations of the MPT, we did not impose any constraints on the distribution of the distance measurement error ε_i . We now assume that the distribution of ε_i is *unknown but bounded* by some finite ε_d . This allows us to derive a *theoretical upper bound on RLL prediction inaccuracy*, denoted as η_u , with four periodical distance measurements. This assumption on measurement errors is reasonable, since in many real-life scenarios, the ranging equipment can compute

range measurements that deviate within a small neighborhood of the true distance¹³.

The derivations of η_u are as follows. Each \hat{d}_i must be in the interval of $[d_i - \varepsilon_d, d_i + \varepsilon_d]$. Conversely, each d_i must be in the interval of $[\hat{d}_i - \varepsilon_d, \hat{d}_i + \varepsilon_d]$. Because a \hat{d}_i can take either of the two extreme values, $\hat{d}_i = d_i \mp \varepsilon_d$, this interval can be re-written as $[d_i - 2\varepsilon_d, d_i + 2\varepsilon_d]$. Since \tilde{d}_i is computed with linear curve fitting, this implies that this estimate of d_i will not depart from it by more than $2\varepsilon_d$, i.e., \tilde{d}_i too must be bounded by $[d_i - 2\varepsilon_d, d_i + 2\varepsilon_d]$.

There exist two possible *candidate trajectories* that would deviate the farthest from the true trajectory: 1) $y' = \alpha'x'$, where $\alpha' < \alpha$, and 2) $y'' = \alpha''x''$, where $\alpha'' > \alpha$. These are depicted in Figure 4-11 along with the true trajectory $y = \alpha x$. In the figure, three Cartesian systems, (x, y) , (x', y') , and (x'', y'') , are superimposed with the overlapping x , x' , and x'' axes. Node 1 is located at Point A, and Node 2 at Point O. Four d_i 's, $\forall i = 0, \dots, 3$, are measured along $y = \alpha x$, with intersection points O, D, C, and B. Let d_i' 's and d_i'' 's denote the two sets of rang measurements that are used to compute the two trajectories. Each semicircular area between two concentric semicircles with the radii $d_i - 2\varepsilon_d$ and $d_i + 2\varepsilon_d$ (centered at A) thus defines the region Ω_i of possible values that d_i' and d_i'' can take.

We need to derive the RLL prediction inaccuracy induced by both α' and α'' to obtain the theoretical upper bound. First, consider the slope α' . Let d_i' , $\forall i = 0, \dots, 3$, be the four periodical distance measurements on $y' = \alpha'x'$. $y' = \alpha'x'$ must satisfy the following two conditions:

¹³ One example of such a distribution used in the literature is the uniform distribution [65], i.e., $\varepsilon_i \sim U(-\varepsilon_d, \varepsilon_d)$, where $0 < \varepsilon_d < \infty$.

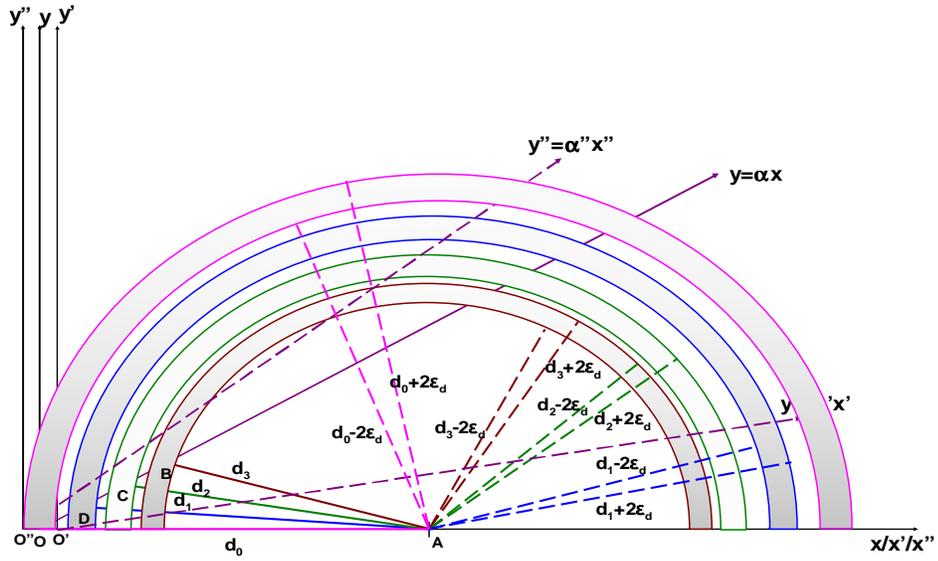


Figure 4-11: Deriving the upper bound of RLL prediction inaccuracy.

- the (x'_i, y'_i) coordinates for the successive d'_i 's must be equidistant from each other, and
- d'_0, d'_2 and d'_3 must all be on either boundary of Ω_1, Ω_2 , and Ω_3 , respectively.

There exist six possible (d'_0, d'_2, d'_3) triples, as listed in TABLE 4-2, which satisfy the above two conditions. For each triple, we write the following system of equations:

$$\begin{cases} (x'_3 - d'_0)^2 + (y'_3)^2 = (d'_3)^2 \\ (x'_2 - d'_0)^2 + (y'_2)^2 = (d'_2)^2 \\ x'_3 = \frac{3}{2}x'_2 \\ y'_i = \alpha' x'_i, \forall i = 2, 3 \end{cases}, \quad (4-18)$$

in which x_3' and α' are computed as follows:

$$\begin{cases} x_3' = \frac{15(d_0')^2 - 27(d_2')^2 + 12(d_3')^2}{12d_0'} \\ \alpha' = \frac{3\sqrt{(d_2')^2 - (d_0')^2 + \frac{4}{3}x_3'd_0' - \frac{4}{9}(x_3')^2}}{2x_3'} \end{cases} \quad (4-19)$$

The predicted RLL computed at d_3' , denoted as RLL' , is computed as:

$$RLL' = 3\Delta t \left(\frac{d_0' + \sqrt{(d_0')^2 - (1 + (\alpha')^2)((d_0')^2 - d_0^2)}}{x_3'(1 + (\alpha')^2)} - 1 \right). \quad (4-20)$$

For notational convenience, we denote the RLL' computed by each (d_0', d_2', d_3') triple as RLL'_k , $\forall k=1, \dots, 6$. Each respective prediction inaccuracy is computed as:

$$\eta_k' = \frac{|RLL'_k - RLL|}{RLL} \cdot 100\%, \forall k = 1, \dots, 6, \quad (4-21)$$

where RLL is the true residual link lifetime. The maximum of η_k' 's corresponds to the minimal trajectory slope that deviates the most from the true slope.

TABLE 4-2: d_0' , d_2' , AND d_3' FOR MINIMAL-SLOPE TRAJECTORY.

| | d_0' | d_2' | d_3' |
|---|------------------------|------------------------|------------------------|
| 1 | $d_0 + 2\varepsilon_d$ | $d_2 + 2\varepsilon_d$ | $d_3 + 2\varepsilon_d$ |
| 2 | $d_0 - 2\varepsilon_d$ | $d_2 + 2\varepsilon_d$ | $d_3 + 2\varepsilon_d$ |
| 3 | $d_0 + 2\varepsilon_d$ | $d_2 - 2\varepsilon_d$ | $d_3 - 2\varepsilon_d$ |
| 4 | $d_0 - 2\varepsilon_d$ | $d_2 - 2\varepsilon_d$ | $d_3 - 2\varepsilon_d$ |
| 5 | $d_0 + 2\varepsilon_d$ | $d_2 + 2\varepsilon_d$ | $d_3 - 2\varepsilon_d$ |
| 6 | $d_0 - 2\varepsilon_d$ | $d_2 + 2\varepsilon_d$ | $d_3 - 2\varepsilon_d$ |

For the slope α'' , there are two (d_0'', d_2'', d_3'') triples that satisfy the two conditions as those for the (d_0', d_2', d_3') triples, and they are listed in TABLE 4-3. Applying the same derivations as in the case of α' , the RLL computed at d_3'' is given as:

$$RLL'' = 3\Delta t \left(\frac{d_0'' + \sqrt{(d_0'')^2 - (1 + (\alpha'')^2)} \left((d_0'')^2 - d_0^2 \right)}{x_3'' (1 + (\alpha'')^2)} - 1 \right). \quad (4-22)$$

Denote the RLL computed by each of the (d_0'', d_2'', d_3'') triples as RLL_j'' , $\forall j = 1, 2$. Their respective prediction inaccuracies are given as follows:

$$\eta_j'' = \frac{|RLL_j'' - RLL|}{RLL} \cdot 100\%, \forall j = 1, 2. \quad (4-23)$$

The greater of the two η_j'' 's corresponds to the slope that deviates the farthest from the true slope.

TABLE 4-3: d_0' , d_2' , AND d_3' FOR MAXIMAL-SLOPE TRAJECTORY.

| | d_0' | d_2' | d_3' |
|---|------------------------|------------------------|------------------------|
| 1 | $d_0 + 2\varepsilon_d$ | $d_2 - 2\varepsilon_d$ | $d_3 + 2\varepsilon_d$ |
| 2 | $d_0 - 2\varepsilon_d$ | $d_2 - 2\varepsilon_d$ | $d_3 + 2\varepsilon_d$ |

Combining the formulas derived in Equations (4-21) and (4-23), η_u is defined as follows:

$$\eta_u = \max \left\{ \left\{ \eta_k' : k = 1, \dots, 6 \right\}, \left\{ \eta_j'' : j = 1, 2 \right\} \right\}. \quad (4-24)$$

Let α_{min} be the trajectory slope that induces the maximal η_k' , $\forall k = 1, \dots, 6$, and α_{max} the trajectory slope that induces the maximum of η_j'' , $\forall j = 1, 2$. The upper bound on RLL prediction inaccuracy represents the maximal prediction inaccuracy possible by the best distance measurement-based prediction algorithm.

4.5 MPT with Velocity-Change Detection

While presenting the operations of the MPT in Section 4.4, we allowed the nodes to move at constant velocity, thereby inducing a linear relative trajectory throughout the link lifetime. In reality, velocity change is a frequent occurrence that poses a

challenge to the algorithm achieving a robust RLL prediction accuracy. Furthermore, the performance of MPT depends on the sampling period Δt with which distances are measured. As will be shown in the performance evaluation, with a larger Δt , the distances are measured further apart, reducing the negative effects of measurement errors on the operations of curve fitting, and improving the prediction accuracy. Yet, it may be difficult to choose an appropriate value for Δt *a priori* that achieves the robust prediction performance.

To address these two issues, we propose an enhanced MPT that is augmented with a simple velocity-change detection (VCD) test. Instead of measuring only four distances, this enhanced algorithm, called MPT-VCD, periodically measures distances between the two nodes throughout the link lifetime. Concurrently, the VCD test is performed periodically to detect velocity-change occurrences, and allows the algorithm to re-compute the RLL when a velocity change is detected. Furthermore, the acquisition time increases with the periodic distance measuring and algorithmic invocation. We shall show how this eliminates the need to judiciously choose a Δt and still achieves a robust prediction performance.

To simulate velocity changes in the link model, we allow Node 2's movements to induce a *piecewise-linear trajectory*. That is, Node 2 moves at constant velocity for some time duration before randomly selecting a new velocity. Node 1 maintains constant velocity and periodically measures distances between itself and Node 2 at each time t_k , for $k=0, 1, 2...$

4.5.1 Velocity-Change Detection (VCD) Test

When Node 2 enters Node 1's transmission range at time t_0 , Node 1 periodically measures distances between the two nodes at each time step throughout the link lifetime. Each distance measurement is stored in a memory cache. Without loss of generality, let $t_0=0$. Every $3\Delta t$ [sec] (or equivalently, whenever k is a multiple of 3), denote $T_{acq}(k)=t_k-t_0=t_k$ as the current acquisition time at t_k . Node 1 then invokes the MPT by drawing four measurements, measured at t_0 , $T_{acq}(k)/3$, $2T_{acq}(k)/3$, and t_k , denoted as \hat{d}_0 , $\hat{d}_{k/3}$, $\hat{d}_{2k/3}$, and \hat{d}_k , respectively. It then computes $\tilde{d}_k = \sqrt{(R - \tilde{x}_k)^2 + \tilde{y}_k^2}$. The VCD test then compares \hat{d}_k and \tilde{d}_k as follows:

$$\begin{aligned} & \text{if } |\tilde{d}_k - \hat{d}_k| \leq \delta_{th} \\ & \quad \text{no velocity change at } t_k \\ & \text{else} \\ & \quad \text{velocity change at } t_k, \end{aligned}$$

where δ_{th} denotes the *detection threshold* that is set by the test.

We define the following terms to analyze the performance of the VCD test. Denote t_{vc} as the *velocity change time*, and t_{vcd} as the *velocity-change detection time*. A *miss* (M) occurs when the test did not detect any velocity change during the link lifetime, even though it did occur. A *false alarm* (FA) occurs when velocity change is detected without it actually occurring, i.e., $t_0 < t_{vcd} < t_{vc}$. A *detection* (D) occurs when velocity change is detected after it occurred, i.e., $t_{vcd} > t_{vc}$.

Assume an unknown but finitely bounded measurement error distribution, as we

did in the derivations for η_u in Section 4.4.3. As explained in that section, at each time step t_k , \hat{d}_k is bounded by $[d_k - \varepsilon_d, d_k + \varepsilon_d]$, and \tilde{d}_k is bounded by $[d_k - 2\varepsilon_d, d_k + 2\varepsilon_d]$. Consequently, the maximal possible difference between \tilde{d}_k and \hat{d}_k is $3\varepsilon_d$ if there is no velocity change. This implies that when we set $\delta_{th} = 3\varepsilon_d$, the probability that the VCD test raises a false alarm is zero. In fact, this is the minimal δ_{th} that achieves the zero probability of false alarms.

Furthermore, we define two VCD metrics related to misses and detections, Z_M and Z_D , respectively:

$$Z_M = \frac{FLL - t_{vc}}{FLL}, Z_D = \frac{t_{vcd} - t_{vc}}{FLL - t_{vc}}. \quad (4-25)$$

Z_M is computed with instances in which a miss occurs, and captures the relationship between a miss and how close t_{vc} is to the end of the link lifetime. Z_D is computed with instances in which a detection occurs, and reflects how much time has elapsed after a velocity change before it is detected, providing a measure of responsiveness by the VCD test to a velocity-change occurrence. Both metrics take values between 0 and 1. They are used to evaluate the effectiveness of the VCD test in Section 4.6.3.

4.5.2 MPT-VCD Algorithm

Once Node 2 enters its transmission range, Node 1 invokes the MPT-VCD to periodically measure distances (with periodicity Δt) and computes \tilde{d}_k and the residual link lifetime $R\hat{L}L_k$ (with periodicity $3\Delta t$). If a velocity change is detected at time t_{vcd}

(which is initialized to t_0), the MPT-VCD will employ the distance measurements that are measured after t_{vcd} to compute the RLL. When Node 1 receives a RLL-prediction request at a time t_{req} , the algorithm draws four periodical distance measurements from t_{vcd} to t_{req} to compute the RLL and reports it to Node 1.

In practice, such a request could arrive at a random time either before or after the velocity-change detection. Reporting the currently predicted RLL before velocity change occurs would likely result in erroneous RLL prediction. We first define a *separation time threshold* $\Delta\tau_{req}$, a minimal time duration between t_{vc} and t_{vcd} . When responding to a prediction request, the MPT-VCD needs to consider the following three cases with respect to the velocity change detection time versus the time of prediction request arrival:

- 1) If the prediction request arrives at Node 1 after a velocity change was detected at t_{vcd} , and the difference between t_{req} and t_{vcd} is at least $\Delta\tau_{req}$ (i.e., $t_{req} - t_{vcd} \geq \Delta\tau_{req} > 0$ and $t_{vcd} > t_0$), the algorithm computes the RLL at t_{req} and reports it to Node 1.
- 2) If the request arrives after a velocity change was detected at t_{vcd} , and the time difference between t_{req} and t_{vcd} is less than $\Delta\tau_{req}$ (i.e., $0 < t_{req} - t_{vcd} < \Delta\tau_{req}$ and $t_{vcd} > t_0$), the algorithm updates $t_{req} = t_{vcd} + \Delta\tau_{req}$, and continues measuring new distances until the new t_{req} , at which time it computes the RLL and reports it to Node 1.
- 3) If the request arrives before a velocity change is detected (i.e., $0 < t_{req} < t_{vcd}$), the algorithm computes the RLL at t_{req} and reports it to Node 1. It continues measuring new distances until it detects velocity-change at time t_{vcd} , at which it updates $t_{req} = t_{vcd} + \Delta\tau_{req}$, resumes measuring distances until the new t_{req} is reached, at which time it computes and reports the RLL to Node 1.

4.6 Performance Evaluation

In this section, we provide performance evaluation of the MPT and MPT-VCD in simulation, which also examines some of the theoretical results obtained in the previous sections.

4.6.1 MPT with Additional Distance Measurements

We first demonstrate how employing more distances at the input of the MPT improves the algorithm's RLL-prediction performance. Distance measurements can be measured in two approaches. In the first approach, N distances are measured with a constant period Δt , and the acquisition time is computed by $T_{acq}=(N-1)/\Delta t$. In the second approach, N distances are measured over a constant acquisition time T_{acq} , and $\Delta t=T_{acq}/N$. However, the second approach yields practically no improvement in performance. This is because of all the distance measurements used by MPT, the two having a dominating impact on estimating the relative trajectory are \hat{d}_0 and \hat{d}_{N-1} , measured at time t_0 and t_{N-1} , respectively. As long as the acquisition time (i.e., the duration between t_0 and t_{N-1}) remains constant and fixed at these two ends, how the other intermediate distances are measured does not have a significant impact on reducing the negative effects of the measurement errors, since it is \hat{d}_0 and \hat{d}_{N-1} that determine the amount of freedom with which the trajectory could tilt. In the first measuring approach, as N increases, MPT is invoked at a time ever closer to the end of

the link lifetime. This is equivalent to increasing the acquisition time between t_0 and t_{N-1} , and results in a more accurate estimated trajectory. Therefore, in the performance evaluation below, we will only present results from the first approach.

We perform simulations in the two-node link model and plot the statistical CDFs of η by MPT-N, where $N=4, 5, 6, 7, 8$. Nodes 1 and 2 are initially placed at $R=50[m]$ apart. The node speed is $V\sim U(1, 10)[m/s]$, the node direction $\theta\sim U(0, 2\pi)$, and $\Delta t=1[sec]$. The *i.i.d.* measurement errors $\varepsilon_i\sim U(-\varepsilon_d, \varepsilon_d)$, where $\varepsilon_d=0.3\%R$. For comparison consistency, all RLL predictions are computed at $\eta(t_0+7\Delta t)$.

Figure 4-12 plots the statistical CDF curves of these MPT variants. As a reference, we also plot the CDF for η_u as derived in Equation (4-24). As expected, measuring more distances with a constant sampling period greatly lowers the achievable RLL prediction inaccuracy. The CDF for η_u depicts the worst possible inaccuracies these predictions with the measured distances can achieve, and serves as the inviolable bound for MPT-4. Since MPT-5 through MPT-8 all achieve better prediction performance than MPT-4, the η_u upper-bounds their worst-case performance as well, as is clearly shown in the figure.

However, the improved prediction performance with more distance measurements comes at the expense of a longer acquisition time, as it denies a greater number of links the opportunity of RLL-prediction computations. A *prediction miss* is defined as an event in which the RLL prediction is not possible because the link breaks before all the distance measurements needed for RLL computations are measured. The amount of prediction misses incurred by each MPT variant is illustrated in Figure 4-13, which plots the number of prediction misses as a percentage of the sum of the numbers of predictions and prediction misses by each MPT variant.

It can be seen that in Figure 4-12, for $\eta\leq 10\%$, 40% of all predictions by MPT-4 achieve this prediction inaccuracy, whereas 80% of all predictions by MPT-8 (i.e., a

little more than twice as much as MPT-4) do so. For $\eta \leq 20\%$, 63% of MPT-4's predictions result in $\eta \leq 20\%$, whereas 91% of all predictions by MPT-8 (i.e., approximately 50% better than MPT-4) do so. On the other hand, as shown in Figure 4-13, MPT-4 leads to an 18% of prediction misses, compared with 41% of prediction misses by MPT-8. The two figures therefore demonstrate a trade-off between prediction inaccuracy and prediction misses. In implementing the algorithm, we are more concerned with allowing more links in the network to perform prediction computations rather than striving for higher accuracy for a smaller number of links. Therefore, in the subsequent performance evaluation, all RLL computations are performed with four distance measurements.

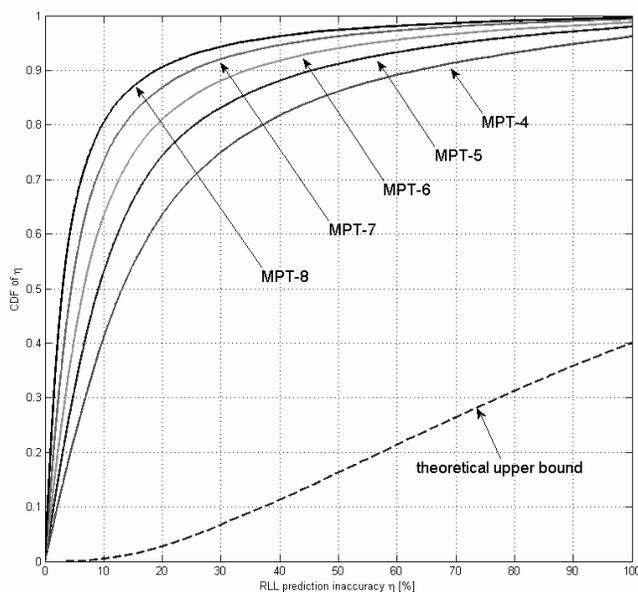


Figure 4-12: CDF of η for various numbers of distance measurements.

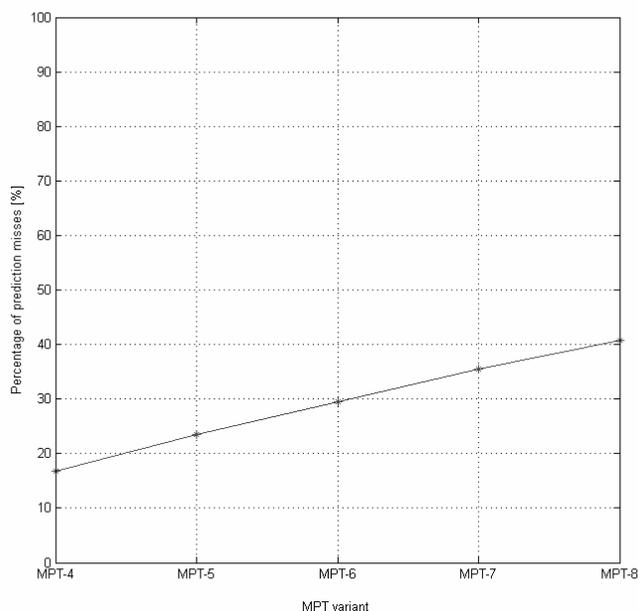


Figure 4-13: Ratio of prediction misses to RLL predictions by MPT variants.

4.6.2 Acquisition Time-to-FLL Ratio

We now evaluate the performance of MPT with respect to the acquisition time T_{acq} as a fraction of the FLL. The simulation scenario is set up in the two-node link model as follows. Before each RLL prediction, we compute the FLL given the relative speed and direction, and set the T_{acq}/FLL ratio, denoted as ρ_{acq} , at 5%, 10%, 15%, 20%, 25%, and 30%. The sampling period is thus equal to $\Delta t = T_{acq}/3$ for each ρ_{acq} value. The same parametric values and distributions for parameters R , ε_i , ε_d , node speed and direction as those in Section 4.6.1 are used in the simulation. Furthermore, both Node 1 and Node 2 maintain constant velocity throughout the link lifetime.

Figure 4-14 plots the statistical CDFs of RLL prediction inaccuracy with respect to

the T_{acq}/FLL ratio. As this ratio increases, the performance of MPT improves. This improvement is brought out because a larger ratio means that MPT is invoked at a time closer to the end of the link lifetime, and the first and last distance measurements are measured further apart. This reduces the effects of the measurement errors when the algorithm performs linear curve fitting to compute the estimated relative trajectory. Therefore, the length of the acquisition time casts a significant impact directly related to the prediction accuracy of the RLL. Conversely, it can also be seen that a larger Δt usually leads to a higher prediction performance, for the reasons given above. In Section 4.6.4, we see how this idea is naturally incorporated into the operations of the MPT-VCD.

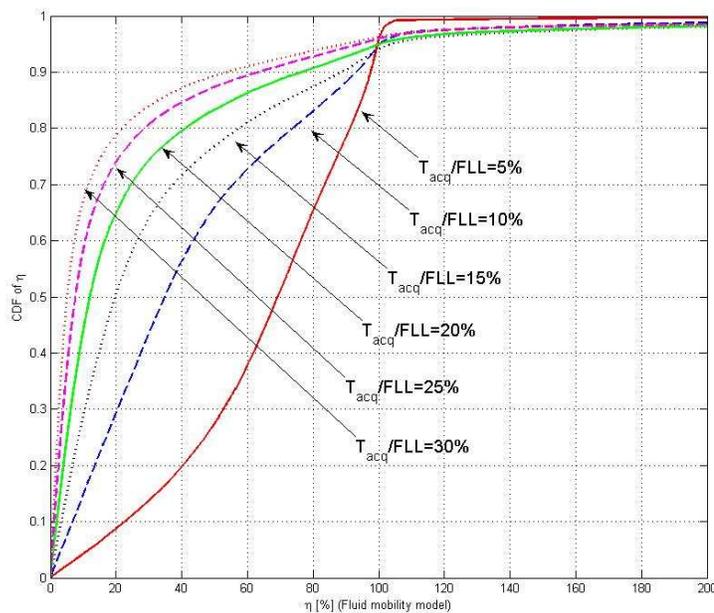


Figure 4-14: RLL prediction inaccuracy w.r.t. T_{acq}/FLL ratio under Fluid model.

4.6.3 Performance of the VCD Test

To evaluate the effectiveness of the VCD test, we let Node 2's movements induce a piecewise-linear trajectory. In this scenario, we allow the velocity change in Node 2 to occur only once at a random time t_{vc} during the link lifetime, while Node 1 maintains constant velocity. The velocity change time $t_{vc} \sim U(0, FLL_1)$, where FLL_1 denotes the FLL had Node 2 not changed its velocity. At t_{vc} , Node 2 chooses a new velocity and moves until the link breaks. Key simulation parameters remain the same as those in Section 4.6.1. Performance evaluation involving multiple velocity changes is presented in Section 4.6.4.

For each link lifetime, we collect 30,000 triples of the following statistics to generate the relevant performance metrics: t_{vc} , t_{vcd} , and FLL . We tabulate the probabilities of misses ($M[\%]$), false alarms ($FA[\%]$), and detections ($D[\%]$) in TABLE 4-4 with two values of δ_{th} , $3\varepsilon_d$ and $0.5\varepsilon_d$, and four values of ε_d , $0.1\%R$, $0.3\%R$, $0.5\%R$, and $0.7\%R$.

The results show that when $\delta_{th}=3\varepsilon_d$, the probability of false alarm is indeed zero regardless of the values of ε_d , confirming our assertion in Section 4.5.1. However, at this detection threshold, a larger ε_d results in the probabilities of miss and detection becoming more equally probable. On the other hand, as δ_{th} is smaller, the probability of false alarm becomes non-zero, the probability of miss decreases, and the probability of detection increases. This is because a smaller δ_{th} makes the VCD test more sensitive to velocity changes, reducing the number of potential misses and allowing for more detections, while becoming more prone to false alarms. The data in the table demonstrate that at a smaller δ_{th} , the proposed VCD test is able to achieve a higher detection probability with relatively low miss and false-alarm probabilities. Therefore,

to achieve a better performance in velocity change detections, we tolerate some false alarms by setting δ_{th} at a smaller value in the following evaluations.

TABLE 4-4: PERFORMANCE OF THE VCD TEST.

| ε_d | δ_{th} | M [%] | FA [%] | D [%] |
|-----------------|--------------------|---------|----------|---------|
| $0.1\%R$ | $0.5\varepsilon_d$ | 5.38 | 3.72 | 90.90 |
| $0.3\%R$ | $0.5\varepsilon_d$ | 9.13 | 4.35 | 86.52 |
| $0.5\%R$ | $0.5\varepsilon_d$ | 11.87 | 4.49 | 83.64 |
| $0.7\%R$ | $0.5\varepsilon_d$ | 14.37 | 4.86 | 80.77 |
| $0.1\%R$ | $3\varepsilon_d$ | 23.77 | 0 | 76.23 |
| $0.3\%R$ | $3\varepsilon_d$ | 41.59 | 0 | 58.41 |
| $0.5\%R$ | $3\varepsilon_d$ | 52.16 | 0 | 47.84 |
| $0.7\%R$ | $3\varepsilon_d$ | 60.09 | 0 | 39.91 |

Figure 4-15 and Figure 4-16 plot the statistical CDFs of Z_M and Z_D respectively, for $\delta_{th}=3\varepsilon_d$, $1.5\varepsilon_d$, and $0.5\varepsilon_d$, with $\varepsilon_d=0.3\%R$. It can be seen that at a smaller δ_{th} , a miss occurs less likely. That it does implies that t_{vc} is likely very close to the end of the FLL (i.e., corresponding to a small Z_M), leaving little time for the VCD test to detect the change before the link breaks. Indeed, Figure 4-15 shows that at $\delta_{th}=3\varepsilon_d$, approximately 33% of all misses occur when the velocity change occurs at a time when 90% of the FLL has gone by, i.e., $Z_M \leq 0.1$, whereas at $\delta_{th}=0.5\varepsilon_d$, nearly 75% of all misses occur for $Z_M \leq 0.1$. A smaller δ_{th} also allows the VCD test to detect a velocity change more quickly after its occurrence, as shown in Figure 4-16. Note that a smaller Z_D reflects a shorter time lapse between t_{vc} and t_{vcd} . Thus, the figure shows that at $\delta_{th}=3\varepsilon_d$, only 16% of all detections are made for $Z_D=0.1$, whereas at $\delta_{th}=0.5\varepsilon_d$, 63% of all detections are made for $Z_D=0.1$. This rapid response time, of course, comes at the expense of an increasing number of false alarms.

We also examine the effects of measurement errors on the VCD test with $\varepsilon_d=0.1\%R$, $0.3\%R$, $0.5\%R$, and $0.7\%R$. The data in TABLE 4-4 also show that with an increasing ε_d , the probability of miss increases, while the probability of detection

decreases. This is expected since a larger ε_d makes it easier for $|\tilde{d}_k - \hat{d}_k|$ to be bounded by δ_{th} , making a miss more likely to occur while reducing the responsiveness to velocity change. Furthermore, the data in the table show that the probability of false alarm increases only slightly despite the increasing ε_d compared with the probability of miss. This is because δ_{th} is always scaled by ε_d , and the ratio of the interval of values for $|\tilde{d}_k - \hat{d}_k|$ in which false alarms do *not* occur to $3\varepsilon_d$ (i.e., the maximal difference between \hat{d}_k and \tilde{d}_k without velocity change) remains unchanged. Consequently, the probability of false alarm does not register a significant increase.

These results allow us to make an appropriate choice of δ_{th} that trades off between misses and false alarms. The RLL can be either shorter or longer after a velocity change than if there were no velocity change. With a larger δ_{th} , a miss would occur if the RLL after t_{vc} is too short for the VCD test to react. Such a link may not be suitable for the purpose of RLL predictions, and the predicted RLL due to a miss could result in a larger prediction inaccuracy. On the other hand, a smaller δ_{th} leads to more false alarms and detections. Detections allow the MPT-VCD to perform RLL computations with distance measurements after the velocity change, leading to a lower prediction inaccuracy. It stands to reason that the cost of a miss is greater than the cost of a false alarm, and this justifies the choice of a smaller δ_{th} that tolerates more false alarms. In the subsequent evaluations, we set δ_{th} to be $0.5\varepsilon_d$.

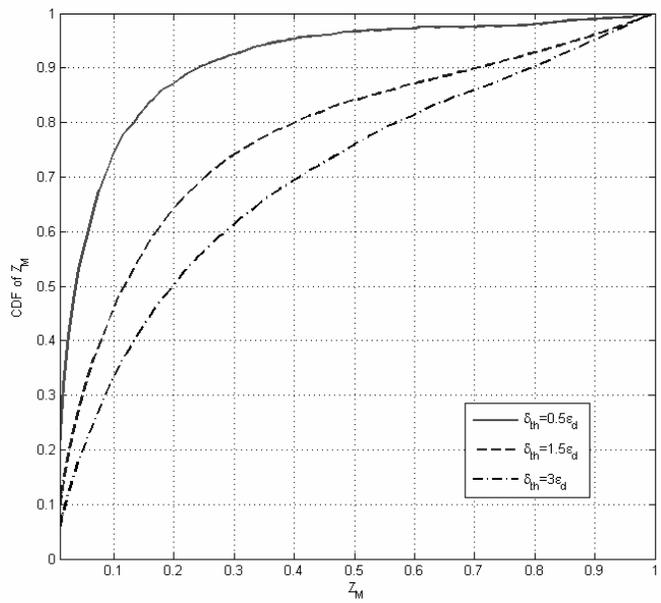


Figure 4-15: Statistical CDFs of Z_M ($\epsilon_d=0.3\%R$).

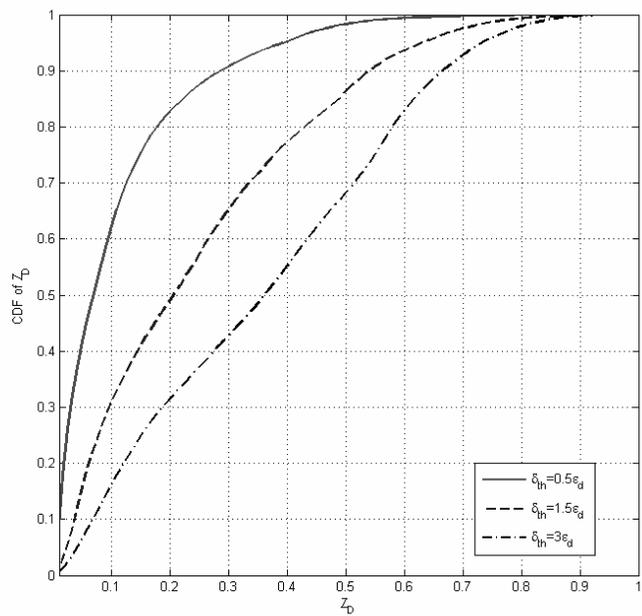


Figure 4-16: Statistical CDFs of Z_D ($\epsilon_d=0.3\%R$).

4.6.4 Performance of MPT-VCD

We begin the performance evaluation of MPT-VCD with the following scenario. During each link lifetime, Node 2 undergoes exactly one velocity change at a random time while Node 1 maintains constant velocity. The separation time threshold $\Delta\tau_{req}=1.5[sec]$ and the detection threshold $\delta_{th}=0.5\varepsilon_d$. Figure 4-17 plots the statistical CDFs of η by the MPT-VCD generated with this scenario at different values of ε_d . As expected, the performance of MPT-VCD decreases as ε_d increases.

The performance of MPT-VCD depends on the choice of $\Delta\tau_{req}$, as defined in Section 4.5.2. The algorithm inserts additional time $\Delta\tau_{req}$ between t_{req} and t_{vcd} so that the first and the last distance measurements used by the subsequent MPT-VCD invocations are sufficiently apart. Therefore, we expect a larger value of $\Delta\tau_{req}$ would result in a better performance of the algorithm, as demonstrated in Figure 4-18 that plots the CDFs of prediction inaccuracy for $\Delta\tau_{req}=0.5, 1.0, 1.5,$ and $2.0[sec]$. However, if $\Delta\tau_{req}$ is too large, the link could break before RLL is computed at t_{req} , resulting in a prediction misses. We have also calculated the percentage of prediction misses out of the sum of the numbers of predictions and prediction misses. For $\Delta\tau_{req}=0.5, 1.0, 1.5,$ and $2.0[sec]$, the percentages of prediction misses are 7.78%, 13.49%, 19.88%, and 25.46%, respectively. These results therefore show a trade-off between improved prediction performance and prediction misses.

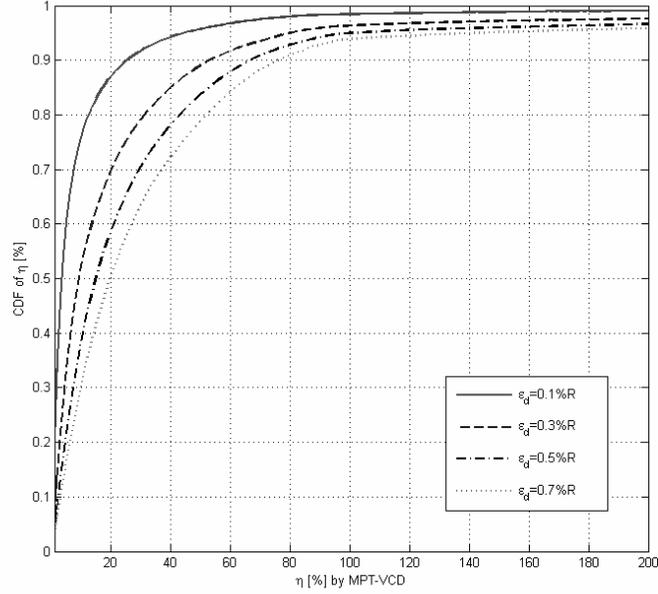


Figure 4-17: CDFs of RLL prediction inaccuracy ($\delta_{th}=0.5\epsilon_d$).

We next consider the effects of multiple velocity changes during the link lifetime on the performance of MPT-VCD. Let m be a simulation parameter that specifies the number of velocity changes Node 2 undergoes during a link lifetime. Denote $RLL_{vc,i}$ as the true RLL at the i -th velocity change at time $t_{vc,i}$. The next velocity-change time is computed as $t_{vc,i+1} \sim U(t_{vc,i}, t_{vc,i} + RLL_{vc,i})$. This allows Node 2's movements to induce a piecewise-linear trajectory with multiple velocity changes. At each $t_{vc,i}$, with 50% probability, the simulator sets t_{req} at the i -th segment of the trajectory until it is set for the first time, and $t_{req} \sim U(t_{vc,i}, t_{vc,i} + RLL_{vc,i})$. The three cases for the relationship between t_{req} and t_{vcd} in Section 4.5.2 still apply. Note that if a new velocity change is detected at time $t_{vcd,i+1}$ after the MPT-VCD reported the predicted RLL to Node 1, a new prediction request needs to be issued at a time $t_{req} = t_{vcd,i+1} + \Delta \tau_{req}$.

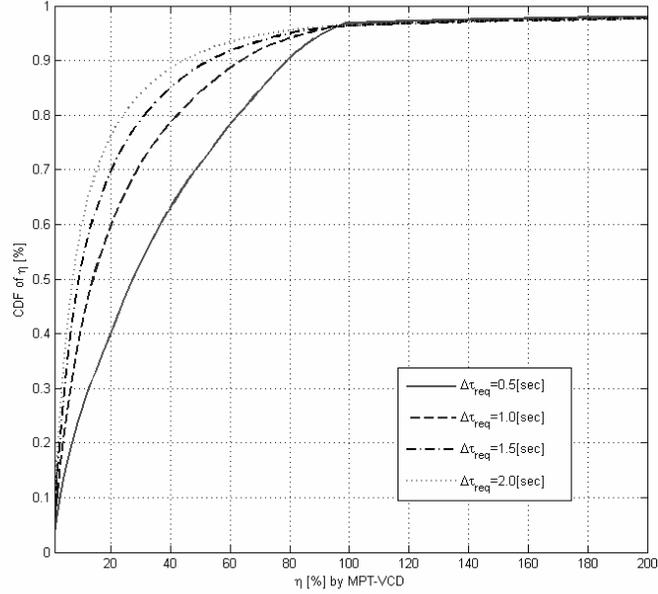


Figure 4-18: CDFs of RLL prediction inaccuracy w.r.t. $\Delta\tau_{req}$.

Figure 4-19 and Figure 4-20 plot the CDFs of the RLL prediction inaccuracy for multiple velocity changes during the link lifetime, with $\Delta\tau_{req}=1.0[sec]$ and $2.0[sec]$, respectively. The four CDF curves in each figure correspond to $m=1, 2, 3, 4$. Apart from showing that a larger $\Delta\tau_{req}$ leads to a better prediction performance as expected, the figures also demonstrate that more velocity changes lead to a slight degradation in the algorithm's prediction performance. This degradation comes from the fact that more velocity changes during the link lifetime increase the possibility of the algorithm making erroneous RLL predictions, especially when the node velocity changes abruptly and randomly. On the other hand, more velocity changes also imply that the last RLL computation likely occurs in the last segment of the trajectory because of the propagation of t_{req} , where the link is near the end of its lifetime. The RLL computations made at such a late stage of the link lifetime do not differ significantly regardless of the number of velocity changes during the link lifetime.

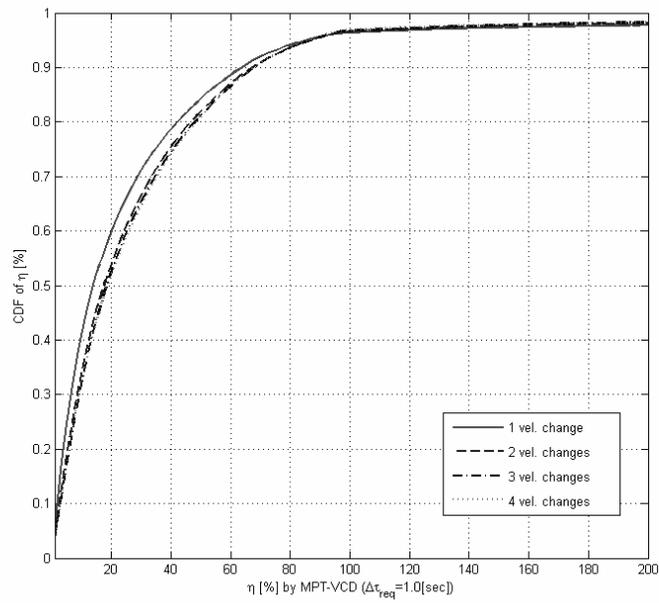


Figure 4-19: Performance of MPT-VCD with multiple velocity changes ($\Delta \tau_{req}=1.0[\text{sec}]$).

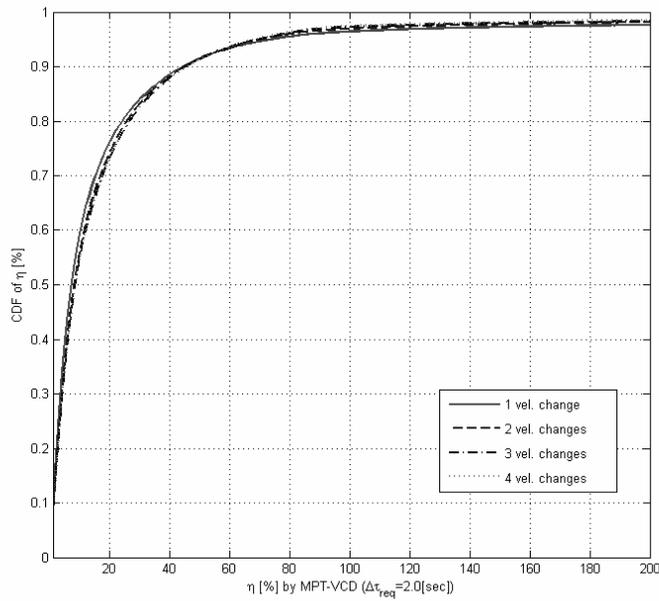


Figure 4-20: Performance of MPT-VCD with multiple velocity changes ($\Delta \tau_{req}=2.0[\text{sec}]$).

We also examine the occurrences of prediction misses due to the increasing number of velocity changes (VCs) and a larger $\Delta\tau_{req}$. TABLE 4-5 tabulates the percentage of prediction misses as a percentage of the total number of predictions and prediction misses for $\Delta\tau_{req}=1.0$ and $2.0[sec]$. It shows that as m increases, more prediction misses occur. This is because with more velocity changes, the last velocity-change detection time ($t_{vcd,m}$) is likely very close to the end of the link lifetime. When this happens, the link could break before the last updated t_{req} (equal to $t_{vcd,m}+\Delta\tau_{req}$), resulting in a prediction miss.

TABLE 4-5: PERCENTAGE OF PREDICTION MISSES IN MPT-VCD

| $\Delta\tau_{req}$ | 1 VC | 2 VCs | 3 VCs | 4 VCs |
|--------------------|--------|--------|--------|--------|
| 1.0 [sec] | 13.49% | 15.98% | 18.18% | 20.06% |
| 2.0 [sec] | 25.46% | 28.59% | 32.09% | 35.62% |

The preceding performance evaluation demonstrates that, from the perspective of RLL prediction accuracy, multiple velocity changes in a piecewise-linear trajectory do not significantly impact MPT-VCD's performance. However, they could also lead to an increase in prediction misses.

4.7 Summary

In order to find a reliable path in the topologically dynamic MANET along which to transmit data packets, it is beneficial to gain predictive knowledge about the path lifetime so that appropriate measures can be taken in advance to safeguard data transmissions. In this chapter, we studied the problem of link lifetime prediction,

which can be expanded to predicting the lifetime of the entire path. Although works have been proposed to employ link age as a prediction parameter of the residual link lifetime, we demonstrated that in several popular MANET mobility models, the full link lifetime beyond a certain threshold can be modeled as a general exponential distribution. This observation invalidates link age as a reliable prediction parameter of the residual link lifetime.

We thus considered an alternative approach for predicting the residual link lifetime, one based on distance measurements. We first proved that, with our two-node link model, it is necessary to measure at least four distances to compute a unique RLL solution. Furthermore, we showed that random measurement errors have a much more significant impact on inducing the inaccuracy of the RLL prediction than systematic errors.

We then proposed the Mobile-Projected Trajectory algorithm using four periodical distance measurements to estimate the relative trajectory and compute the residual link lifetime. The operations of the algorithm do not require the node to possess any knowledge of velocity and position. Since the algorithm is based on linear curve fitting, if more than four distance measurements are available, the prediction accuracy can be further improved. By assuming an unknown, but finitely-bounded distance measurement error distribution, we were able to derive an upper bound on RLL prediction inaccuracy. This is the best bound achievable by any distance measurement-based RLL prediction algorithm.

To account for the occurrences of velocity changes during the link lifetime, we augmented the MPT with a simple velocity-change detection test. With the same assumption about the distance measurement error, we were able to derive the minimal detection threshold that allowed the proposed VCD test to achieve a zero probability of false alarm.

In the performance evaluation, we have confirmed that employing more distance measurements indeed improves the prediction performance of MPT, but it is at the expense of more prediction misses. By analyzing the performance of MPT with respect to the acquisition time-to-FLL ratio, we showed that the more apart the first and last distance measurements are measured, the better prediction accuracy MPT is achieved. This idea was indeed incorporated into the MPT-VCD with its periodical distance measuring and algorithm invocations. We have demonstrated the effectiveness of the proposed VCD test that can achieve a very powerful detection probability if we tolerate some false alarms. Performance evaluation of the MPT-VCD showed a robust performance when the node movements induce a piece-linear trajectory. Under such a trajectory, we have shown that increasing the number of velocity changes does not significantly impact the algorithm's performance, although it leads to an increase in prediction misses.

Chapter 5 Kalman Filter-Based RLL Prediction

5.1 Introduction

Multi-hop paths in a MANET undergo frequent breakages due to the dynamic network topology, resulting in undesired outcomes such as excessive data delivery latency and data loss. It is in the interest of the data sender to find a path along which data can be routed reliably to reach the intended recipient.

In this chapter, we continue to study the problem of link lifetime prediction in the MANET. A number of published works have addressed this problem by employing link age as the prediction parameter [7][45][82]. In Chapter 4, we proposed the Mobile-Projected Trajectory algorithm that computes the predicted RLL by employing four periodical distance measurements. To account for the occurrences of velocity change during link lifetime, we augmented the MPT with a Velocity-Change Detection test.

Although the MPT-VCD continuously measures distances between two nodes of a link and is invoked periodically throughout its lifetime, at every invocation it utilizes only four distance measurements that are equally measured in time to compute the relative trajectory and RLL. This results in discarding a lot of distance measurements that would have been helpful in providing a more accurate RLL computation. Furthermore, the first time the MPT-VCD is invoked is $3\Delta t[sec]$ after the link came into being; subsequently, it is invoked every $3\Delta t[sec]$ in order to maintain the

periodicity of distance measurements utilized in RLL computations. Therefore, when a RLL-prediction request arrives at the node, the algorithm may not be able to immediately provide a predicted RLL to the prediction request.

We now propose a new algorithm that employs a Kalman Filter in computing the residual link lifetime. The original discrete-time linear Kalman filter was proposed by Rudolf Kalman in 1960 to recursively estimate the state of a linear system by sampling a measurement at each time step [76]. With certain assumptions, it can achieve the optimal solution by minimizing the mean square error of the estimated state. However, no practical systems can be modeled as entirely linear. Therefore, several Kalman filter variants, such as Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF), have been developed to account for system nonlinearity and approximate sub-optimal state estimations. Since its invention, Kalman filters have been implemented in a wide variety of applications, from aviation to the stock market.

The RLL-prediction algorithm proposed in this chapter is based on the Unscented Kalman Filter. While the EKF is more popular in many practical applications than the UKF due to its earlier introduction to the public, we adopt the UKF in our algorithm because of its ability of reaching approximations that are third-order to the optimal solution, whereas an EKF can only reach first-order approximations to the optimal solution [76]. Furthermore, it is much easier to initialize the UKF than the EKF. That is, the initial parameter values of an UKF can be chosen in a coarsely defined region without significantly impacting the performance of the filter, whereas a coarse initialization of an EKF could easily lead to the divergence in state estimations. Thus, it is much more likely for the state estimations made by the UKF to converge to the true states.

The development of UKF is based on the following two premises at each time step [42]. Firstly, it is easy to perform a nonlinear transformation on a single state point.

Secondly, it is not too difficult to find a set of individual state points in the state space whose sample PDF approximates the true PDF of the state. We will present how, in the context of our problem, the proposed RLL-prediction algorithm models the link lifetime as a non-linear dynamic system, in which the filter recursively updates the state estimations, and computes the RLL from the estimated states. As in the case of MPT and MPT-VCD, the proposed algorithm in this chapter does not require any knowledge of nodes' velocities and actual positions.

The remainder of the chapter is organized as follows. Section 5.2 introduces the operations of the UKF-based algorithm and discusses the initialization of the Unscented Kalman Filter. Section 5.3 evaluates the performance of the proposed algorithm through extensive simulations. Section 5.4 concludes the chapter with a summary.

5.2 Unscented Kalman Filter-based RLL Prediction

5.2.1 Formulation of Unscented Kalman Filter

We assume the same two-node link model developed for MPT and MPT-VCD in Chapter 4, in which Node 1 measures distances between itself and Node 2 with sampling period Δt . As illustrated in Figure 5-1, when two nodes enter each other's neighborhood at time t_0 (without loss of generality, let $t_0=0$), Node 1 measures the first

distance \hat{d}_0 and establishes a Cartesian system by placing Node 2 at its origin, and itself on the positive end of the x-axis at $(\hat{d}_0, 0)$. At each subsequent time step t_k ($t_k = \Delta t + t_{k-1}$), $k=1, 2, \dots$, Node 1 computes a new distance measurement \hat{d}_k and invokes the UKF to compute the RLL. When an RLL-prediction request arrives at some random time t_{req} , the most recently predicted RLL is reported to Node 1.

The link lifetime is modeled as a non-linear dynamic system for the UKF, and it consists of two state variables to be estimated at each t_k :

- 1) the slope of the relative movement trajectory, denoted as α_k , and
- 2) the x-coordinate of the currently measured distance, denoted as x_k .

Along with \hat{d}_k , the dynamic system model can be formulated as follows:

$$\left\{ \begin{array}{l} f \left(\begin{bmatrix} \alpha_{k-1} \\ x_{k-1} \end{bmatrix}, \begin{bmatrix} w_{\alpha, k-1} \\ w_{x, k-1} \end{bmatrix} \right) = \begin{bmatrix} \alpha_k \\ x_k \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 + \Delta t / t_{k-1} \end{bmatrix} \begin{bmatrix} \alpha_{k-1} \\ x_{k-1} \end{bmatrix} \\ h \left(\begin{bmatrix} \alpha_k \\ x_k \end{bmatrix}, v \right) = \hat{d}_k = \sqrt{(R - x_k)^2 + (\alpha_k x_k)^2} + \varepsilon_k \end{array} \right. \quad (5-1)$$

where ε_k denotes the distance measurement error with a distribution that has zero mean and variance $R_{\varepsilon, k}$. Note that since the states of the system are accurately described by $f(*)$, no state noise exists.

At each time step, the UKF recursively performs two updates: *time update* and *measurement update*. The time update is first performed as follows. At the current time step t_k , UKF computes the *state sigma points* for t_{k-1} , denoted as $\left[\hat{\alpha}_{k-1}^{(i)} \quad \hat{x}_{k-1}^{(i)} \right]^T$, as follows:

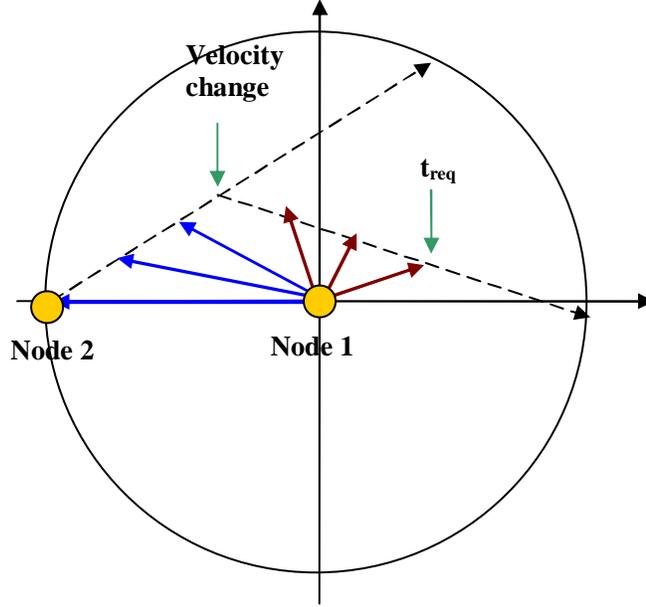


Figure 5-1: Illustration for the UKF-based RLL-prediction algorithm.

$$\begin{bmatrix} \hat{\alpha}_{k-1}^{(i)} \\ \hat{x}_{k-1}^{(i)} \end{bmatrix} = \begin{bmatrix} \hat{\alpha}_{k-1} + \tilde{\alpha}^{(i)} \\ \hat{x}_{k-1} + \tilde{x}^{(i)} \end{bmatrix}, \quad \forall k > 1, i = 1, \dots, 4. \quad (5-2)$$

In Equation (5-2), $\hat{\alpha}_{k-1}$ and \hat{x}_{k-1} denote the *a posteriori* state estimates at time t_{k-1} . They represent the states estimated with distance measurements up to and including \hat{d}_{k-1} . The two corrective terms $\tilde{\alpha}^{(i)}$ and $\tilde{x}^{(i)}$ are computed as follows:

$$\begin{bmatrix} \tilde{\alpha}^{(i)} \\ \tilde{x}^{(i)} \end{bmatrix} = \left(\sqrt{2P_{k-1}} \right)_i^T, \quad \begin{bmatrix} \tilde{\alpha}^{(2+i)} \\ \tilde{x}^{(2+i)} \end{bmatrix} = - \left(\sqrt{2P_{k-1}} \right)_i^T, \quad \forall i = 1, 2, \quad (5-3)$$

where P_{k-1} denotes the *a posteriori state covariance matrix* for the two states at t_{k-1} , and $\sqrt{2P_{k-1}}$ is an upper-triangular matrix that can be computed by the *Cholesky matrix*

square root algorithm [76]. Note that $(\sqrt{2P_{k-1}})_i^T$ is the transpose of the i -th row in $\sqrt{2P_{k-1}}$. The state sigma points are then promoted from time t_{k-1} to t_k , and are denoted as $[\hat{\alpha}_k^{(i)} \quad \hat{x}_k^{(i)}]^T$:

$$\begin{bmatrix} \hat{\alpha}_k^{(i)} \\ \hat{x}_k^{(i)} \end{bmatrix} = f \left(\begin{bmatrix} \hat{\alpha}_{k-1}^{(i)} \\ \hat{x}_{k-1}^{(i)} \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} 1 & 0 \\ 0 & 1 + \frac{\Delta t}{t_{k-1}} \end{bmatrix} \begin{bmatrix} \hat{\alpha}_{k-1}^{(i)} \\ \hat{x}_{k-1}^{(i)} \end{bmatrix}, \quad \forall i=1, \dots, 4, \quad (5-4)$$

We next employ these state sigma points from Equation (5-4) to compute the *a priori state estimates* and *a priori state covariance matrix* at time step t_k , denoted as $[\hat{\alpha}_k^- \quad \hat{x}_k^-]^T$ and P_k^- , respectively, which do not have the knowledge of \hat{d}_k at time step t_k . They are computed as follows:

$$\begin{bmatrix} \hat{\alpha}_k^- \\ \hat{x}_k^- \end{bmatrix} = \frac{1}{4} \sum_{i=1}^4 \begin{bmatrix} \hat{\alpha}_k^{(i)} \\ \hat{x}_k^{(i)} \end{bmatrix}, \quad P_k^- = \frac{1}{4} \sum_{i=1}^4 \begin{bmatrix} \hat{\alpha}_k^{(i)} - \hat{\alpha}_k^- \\ \hat{x}_k^{(i)} - \hat{x}_k^- \end{bmatrix} \begin{bmatrix} \hat{\alpha}_k^{(i)} - \hat{\alpha}_k^- \\ \hat{x}_k^{(i)} - \hat{x}_k^- \end{bmatrix}^T, \quad (5-5)$$

This concludes the time update.

The measurement update follows the time update, and it updates the results obtained from the time update by incorporating the currently measured distance \hat{d}_k in its state estimations. It begins by first refining the state sigma points with the newly acquired $[\hat{\alpha}_k^- \quad \hat{x}_k^-]^T$ and P_k^- from Equation (5-5):

$$\begin{bmatrix} \hat{\alpha}_k^{(i)} \\ \hat{x}_k^{(i)} \end{bmatrix} = \begin{bmatrix} \hat{\alpha}_k^- + \tilde{\alpha}^{(i)} \\ \hat{x}_k^- + \tilde{x}^{(i)} \end{bmatrix}, \quad \forall i=1, \dots, 4, \quad (5-6)$$

where:

$$\begin{bmatrix} \tilde{\alpha}^{(i)} \\ \tilde{x}^{(i)} \end{bmatrix} = \left(\sqrt{2P_k^-} \right)_i^T, \quad \begin{bmatrix} \tilde{\alpha}^{(2+i)} \\ \tilde{x}^{(2+i)} \end{bmatrix} = -\left(\sqrt{2P_k^-} \right)_i^T, \quad \forall i=1,2.$$

With the refined state sigma points from Equation (5-6), we can compute the *measurement sigma points* $\hat{d}_k^{(i)}$ as follows:

$$\hat{d}_k^{(i)} = h \left(\begin{bmatrix} \hat{\alpha}_k^{(i)} \\ \hat{x}_k^{(i)} \end{bmatrix}, 0 \right) = \sqrt{\left(R - \hat{x}_k^{(i)} \right)^2 + \left(\hat{\alpha}_k^{(i)} \hat{x}_k^{(i)} \right)^2}, \quad \forall i=1, \dots, 4, \quad (5-7)$$

The measurement sigma points can be seen as the four best estimated distance measurements around the true distance. The *predicted measurement*, denoted as \hat{d}_k , is computed by taking the average of the four $\hat{d}_k^{(i)}$ values:

$$\hat{d}_k = \frac{1}{4} \sum_{k=1}^4 \hat{d}_k^{(i)}. \quad (5-8)$$

The variance of \hat{d}_k , denoted as P_d , is computed as follows:

$$P_d = \frac{1}{4} \sum_{k=1}^4 \left(\hat{d}_k^{(i)} - \hat{d}_k \right)^2 + R_{\varepsilon,k}. \quad (5-9)$$

The cross-covariance matrix of $\begin{bmatrix} \hat{\alpha}_k^- & \hat{x}_k^- \end{bmatrix}^T$ and \hat{d}_k , denoted as $P_{y,d}$, is computed

using the results from Equations (5-5) through (5-8) as follows:

$$P_{\underline{y},d} = \frac{1}{4} \sum_{k=1}^4 \left(\begin{bmatrix} \hat{\alpha}_k^{(i)} - \hat{\alpha}_k^- \\ \hat{x}_k^{(i)} - \hat{x}_k^- \end{bmatrix} (\hat{d}_k^{(i)} - \hat{d}_k) \right). \quad (5-10)$$

Finally, we compute the *Kalman gain*, *a posteriori state estimates*, and a *posteriori state covariance matrix*, denoted as K_k , $[\hat{\alpha}_k \ \hat{x}_k]^T$, and P_k , respectively:

$$\begin{cases} K_k = P_{\underline{y},d} P_d^{-1} \\ \begin{bmatrix} \hat{\alpha}_k \\ \hat{x}_k \end{bmatrix} = \begin{bmatrix} \hat{\alpha}_k^- \\ \hat{x}_k^- \end{bmatrix} + K_k (\hat{d}_k - \hat{d}_k) \\ P_k = P_k^- - K_k P_d K_k^T \end{cases} \quad (5-11)$$

The results in Equation (5-11) will then be applied to the time update at the next time step t_{k+1} when UKF is invoked again, commencing a new iteration of time and measurement updates.

5.2.2 Initialization and RLL Computation of the UKF

We now discuss the initialization of the following UKF parameters at t_1 , namely, α_1 , x_1 , $R_{\epsilon,1}$, and P_1 . Although no specific guidelines exist for initializing filter, and the choices of these initial parametric values are often coarse, UKF achieves very robust

performance with them, demonstrating its considerable leeway in parametric tuning when relatively little information is available to initialize the filter.

Although we assume a node has no knowledge of its actual speed and direction, in practice, it is reasonable to assume that the distribution of the node speed from a minimal to maximal value is known. Therefore, the average speed V_{avg} of the minimal and maximal speeds is known. Since Node 2 can approach Node 1 with any relative angle between 0 and $\pi/2$, it is reasonable to initialize the relative angle to be $\pi/4$; therefore, $\alpha_l = \tan(\pi/4) = 1$. x_l is thus initialized to be $x_l = V_{avg} \cdot \Delta t \cos(\pi/4)$. As to P_l , if we were absolutely certain of our initial state estimates, P_l would be a two-by-two null matrix $\underline{0}$. However, since it is impossible to guarantee error-free initial state estimates, P_l cannot be a null matrix. As it turns out, when initializing the UKF, any particular choice of P_l is not critical [76], and any $P_l \neq \underline{0}$ would allow the filter to eventually converge. Furthermore, it is acceptable to assume zero correlation between the states' initial values. In our performance evaluation, we choose the following P_l :

$$P_l = \begin{bmatrix} 0.05 & 0 \\ 0 & 0.05 \end{bmatrix}. \quad (5-12)$$

$R_{\varepsilon,l}$ is the variance of the random measurement error that depends on the distance-measurement technology and equipment used and can be obtained by gathering the long-term statistics of the measurement error or through other means. Furthermore, we can reasonably assume that the measurement errors ε_k are *i.i.d.*, and that they are independent of the true distances. Therefore, $R_{\varepsilon,k} = R_{\varepsilon}$, for $k=0, 1, 2, \dots$

Without loss of generality, let $t_0=0$. At each t_k , the predicted RLL, denoted as $\hat{RLL}_{ukf,k}$, is computed as follows:

$$R\hat{L}_{ukf,k} = (k\Delta t) \left[\frac{2\hat{d}_0}{x_k(1+\alpha_k^2)} - 1 \right], k=1,2,\dots \quad (5-13)$$

Figure 5-2 demonstrates the effectiveness of the above initialization by tracking the state variables and the RLL in the case of constant velocity. We randomly choose a link lifetime in the simulation and apply the UKF-based RLL prediction algorithm to compute its predicted RLL at every time step. For the key simulation parameters, the transmission range is $R=50[m]$, the relative speed between the nodes is $6.45[m/s]$, the relative angle is 24.58° , and the sampling period is $\Delta t=0.1[sec]$. As expected, after an initial period of fluctuation, the estimated states eventually converge to the true states, as does the estimated RLL to the true RLL. Repeated experiments with other values for the node speed, direction, and Δt have yielded the similar results as depicted in the figure.

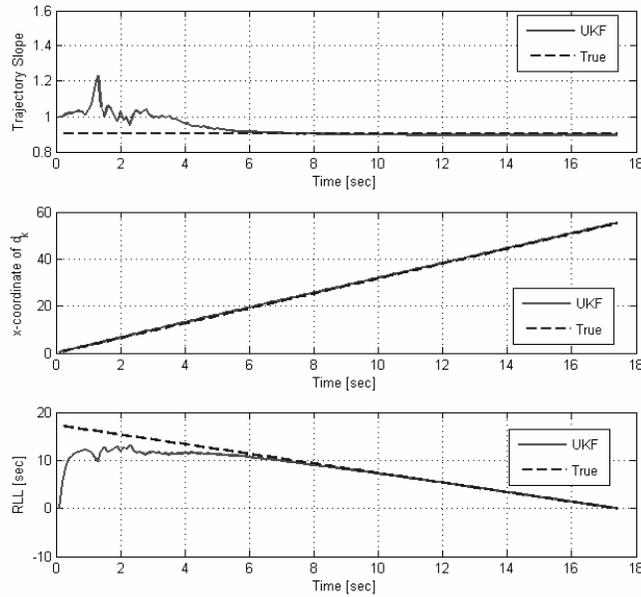


Figure 5-2: Time diagram of tracking the state variables and RLL.

5.3 Performance Evaluation

5.3.1 Performance of the Algorithm without Velocity Change

We first evaluate the performance of the UKF-based RLL-prediction algorithm when no velocity change takes place during the link lifetime. Each distance measurement contains a measurement error $\varepsilon_k \sim U(-\varepsilon_d, \varepsilon_d)$. Both Node 1 and Node 2 have a speed distribution $V \sim U(1, 10)[m/s]$, and a direction distribution $\theta \sim U(0, 2\pi)$. The transmission range $R=50[m]$. A predicted-RLL statistic is recorded when a RLL-prediction request arrives at Node 1 at a random time t_{req} and is compared with the actual RLL. In determining the distribution of t_{req} , note that regardless of how finely initialized the filter is, there is always an initial duration of divergence of the estimated states from the true states of the system. It will not be possible to obtain an accurate RLL prediction within this time duration. Therefore, a prediction request can be properly responded to only if it arrives after this initial fluctuating period. In this and subsequent simulation scenarios, the request arrival time t_{req} is uniformly distributed between $2[sec]$ and the full link lifetime, where the initial duration of $2[sec]$ is approximately equal to 10% of the time it takes Node 2 to traverse along the diameter of Node 1's circular neighborhood at the average node speed.

Figure 5-3 plots the statistical CDFs of RLL prediction inaccuracy of the above simulation scenario with $\varepsilon_d = 0.3\%R$, for two different values of Δt : $0.01[sec]$ and $0.1[sec]$, respectively. 30,000 statistics are generated to plot each curve in the figure.

For both values of Δt , it can be seen in the figure that between 80% and 86% of all RLL predictions made by UKF achieve an inaccuracy of 20% or less (correspondingly a prediction accuracy of 80% or more). This robust prediction performance is due to the fact that our UKF works well in estimating the states of the link lifetime at each time step. Of course, the algorithm performs better with a smaller Δt (i.e., more frequent invocations of the UKF). This can be explained by the fact that a smaller Δt leads to measuring more distances during the link lifetime. Given that the UKF recursively produces the estimated states with all the distance measurements it has accumulated up to the current time, it stands to reason that more distance measurements would naturally lead to a better estimate of the state of the link lifetime.

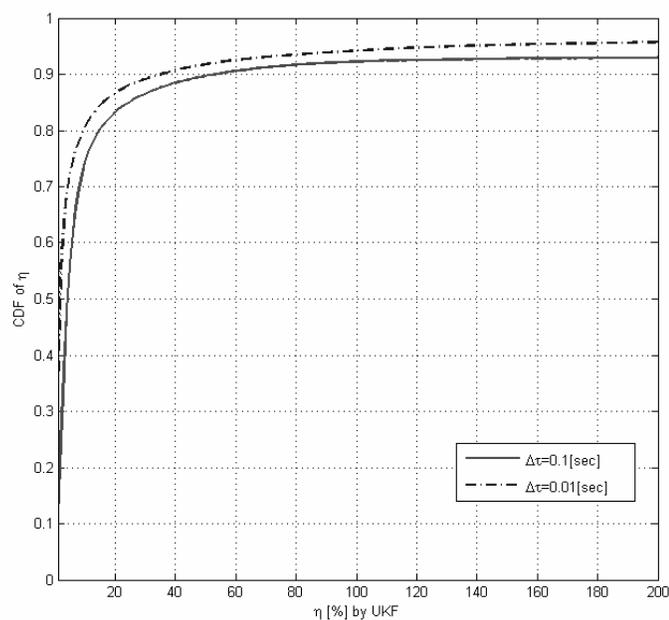


Figure 5-3: Statistical CDFs of η (constant velocity).

5.3.2 Performance of the Algorithm with Velocity Change

We next evaluate the performance of the UKF-based algorithm under two different mobility models with velocity change. The first type of trajectory is induced by the G-M mobility model, as introduced in Section 2.2.1. In the current simulation scenario, it is defined as follows:

$$\begin{aligned} V_n &= \alpha V_{n-1} + (1-\alpha)V_{avg} + \sqrt{1-\alpha^2}V_{x_{n-1}} \\ \Phi_n &= \alpha\Phi_{n-1} + (1-\alpha)\Phi_{avg} + \sqrt{1-\alpha^2}\Phi_{x_{n-1}} \end{aligned}, \quad (5-14)$$

where V_n and Φ_n denote the node speed and direction at the n -th epoch, respectively; α is the correlation factor between 0 and 1; V_{avg} and Φ_{avg} denote the average node speed and direction, respectively; $V_{x_{n-1}}$ and $\Phi_{x_{n-1}}$ denote the Gaussian-distributed random variables defined below. As for the values of these parameters, $V_{avg}= 5.5[m/s]$, $\Phi_{avg}\sim U(0, 2\pi)$, $\alpha=0.9$, $V_{x_{n-1}} \sim N(0, \sigma_V^2)$, and $\Phi_{x_{n-1}} \sim N(0, \sigma_\Phi^2)$, where $\sigma_V=V_{avg}/3$, and $\sigma_\Phi=\pi/6$.

Figure 5-4 plots the CDF of the RLL prediction inaccuracy, where Node 1 maintains constant velocity throughout the link lifetime, and Node 2's movements are governed by the G-M mobility model with different values of the Node-Velocity Update Interval, at $NVUI=5, 10, 15, 20[sec]$. As expected, with a longer epoch, velocity change occurs less frequently during the link lifetime. This enables the algorithm to measure more distances within a period of constant velocity, thus allowing the UKF to better estimate the state of the system, resulting in a better

prediction performance.

In the second, *Gradual-Turn* (G-T) mobility model, Node 2 initially moves with constant speed and direction for some duration after entering Node 1's transmission range. At some random time t_{vc} , Node 2 begins to incrementally change its direction over a time duration T_{vc} , at the end of which (i.e., $t_{vc}+T_{vc}$) it resumes constant velocity until the link breaks. This mobility model induces a smooth, gradually-turning trajectory that is more realistic than other mobility models that induce instantaneous or piecewise-linear trajectory changes.

We investigate how the time duration T_{vc} affects the algorithm's performance with velocity change. Figure 5-5 plots the CDF of the RLL prediction inaccuracy with the following parameters. Let t_{req} be uniformly distributed between 2.0[sec] and t_0+FLL , and t_{vc} be uniformly distributed between $t_0+\Delta t$ and t_0+FLL . The overall change in direction during T_{vc} is distributed as $U(-\Delta\Phi_{vc}, \Delta\Phi_{vc})$, where $\Delta\Phi_{vc}=30^\circ$ and 45° , and $T_{vc}=1$ and $3[sec]$. $\Delta t=0.1[sec]$. The figure demonstrates that a more gradual velocity change (i.e., a larger T_{vc}) leads to a better algorithm performance. This is because a more gradual change in velocity implies a smaller variance in the estimated state at each time step, and allows the UKF to better propagate its state estimate and the computed RLL as the link lifetime progresses. Of course, a large change in direction (i.e., a larger $\Delta\Phi_{vc}$) requires longer time for UKF to converge, resulting in performance degradation. It is worth noticing that for all the four curves, more than 70% of all RLL predictions achieve a η of 20% or less, demonstrating the robustness of the algorithm performance.

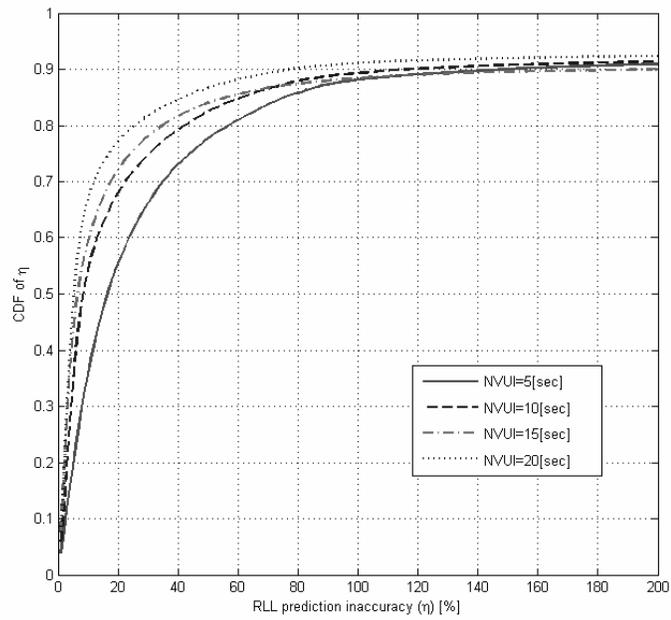


Figure 5-4: CDF of η under the G-M mobility model.

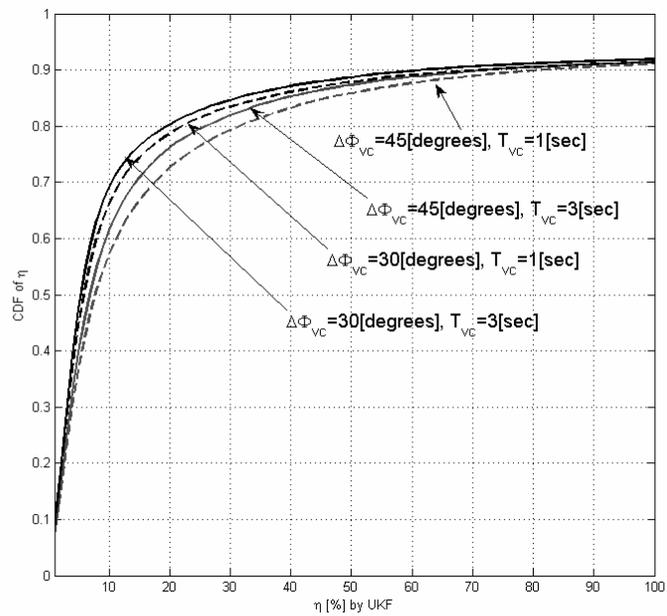


Figure 5-5: CDF of η under the G-T mobility model.

5.4 Summary

In Chapter 4, we proposed the MPT and MPT-VCD algorithms to compute the estimated RLL. However, when invoked, both algorithms employ only four periodical distance measurements to compute the residual link lifetime, while discarding a large quantity of distance measurements. Furthermore, the first time MPT or MPT-VCD is invoked is $3\Delta t[\text{sec}]$ after the link comes into being, and subsequently, it can only be invoked every $3\Delta t[\text{sec}]$ in order to maintain the periodicity of the distance measurements that it employs in RLL computations.

In this chapter, we proposed the UKF-based RLL-prediction algorithm that models the link lifetime as a non-linear dynamic system. The algorithm periodically measures distances between two nodes and recursively updates the state estimates of link lifetime at each time step. The first time the proposed algorithm is invoked is at $\Delta t[\text{sec}]$ after a link came into being, and is thereafter invoked every $\Delta t[\text{sec}]$ with the measuring of a new distance. All the distance measurements up to the current time step are utilized to update and refine the estimated system states. The algorithm also requires only very coarse initialization of the Unscented Kalman Filter for its estimated states to eventually converge to the true states.

Without velocity change, the proposed UKF-based algorithm has demonstrated a robust ability of bringing the estimated states into convergence with the true states of the link lifetime. Furthermore, a smaller sampling period measures more distances and leads to more frequent invocations of the UKF, since it gives the filter more distance measurement information to better estimate the states of the link lifetime. Performance evaluation was conducted in simulation under two types of trajectories with velocity changes. In the G-M mobility model, we have observed that the performance of the

algorithm improves when there are fewer velocity changes during the link lifetime. In the G-T mobility model, a more gradual change in velocity allows the UKF to better estimate the state. For both mobility models, the same accuracy can be achieved by 75% (with NVUI=20[sec]) and 70% of the RLL predictions, respectively.

Chapter 6 Conclusions and Discussions

6.1 What Is Done

There is a growing interest in the wireless research community to understand the characteristics of multi-hop path lifetime that may be exploited for QoS provisioning in the MANET. Some works proposed the use of the mean residual path lifetime as a parameter to measure path reliability. In Chapter 2, we examined the relationship between the mean RPL and each constituent link lifetime, and investigated the effects of mobility under three mobility models: Random Mobility, Random Waypoint, and Gauss-Markov. Through extensive simulations, we concluded that the mean RPL in these mobility models is an invariant quantity and unreliable parameter for predicting path lifetimes.

In Chapter 3, we studied the problem of intelligent best-path selection in the MANET. By associating the reliability of a multi-hop path with its lifetime, we proposed three path-selection algorithms based on link age that were aimed at choosing the best path from a set of available paths between a source-destination pair. We defined the “best path” as either a path among all available paths that would most likely meet a specified requirement for a desired minimal path lifetime, or one that would be the longest-living path among all available paths. Furthermore, we developed two corresponding performance metrics to evaluate these path-selection algorithms. The metrics also allow the algorithms to be compared with a baseline

random-selection algorithm, which arbitrarily chooses a path regardless of path reliability. Simulation results demonstrated that the performance of all three algorithms improved over the baseline algorithm as the size of the path set increased. Furthermore, these algorithms performed better in a high-mobility environment than in a low-mobility environment.

In Chapter 4, we first demonstrated that, under several popular mobility models such as RM, RWP, and G-M, the full link lifetime beyond a certain threshold can be modeled as a general exponential distribution. Given the memoryless property of the exponential distribution, we determined that link age would not be able to serve as a reliable link lifetime prediction parameter. We therefore took an alternative approach to address the problem of RLL prediction, one based on distance measurements between two mobile nodes of a link. We developed a two-node link model, in which nodes do not possess any knowledge of node velocity and position, and proposed the Mobile-Projected Trajectory algorithm to compute RLL by periodically measuring four distances to estimate the relative trajectory between two nodes. Furthermore, we derived an upper bound on the RLL prediction inaccuracy achievable by any distance measurement-based RLL prediction algorithm.

To account for the frequent occurrences of velocity change during the link lifetime, we devised a simple velocity-change detection test that was augmented to the MPT. We also derived the minimal detection threshold in the VCD test that guaranteed a zero probability of false alarm. This MPT-VCD algorithm eliminates the need for the algorithm to judiciously choose a sampling period. The performance evaluation of MPT-VCD demonstrated the improved effectiveness of the algorithm's prediction of RLL while encountering velocity changes during the link lifetime.

In Chapter 5, we proposed an Unscented Kalman Filter-based RLL prediction algorithm that cast the link lifetime as a non-linear dynamic system model. The UKF

can be initialized at the time when the second distance is measured. It then periodically measures distances and recursively estimates the states of the system at each time step by employing all distance measurements up to the current time. This is different from MPT-VCD, where only four distance measurements are involved in the RLL computations when it is invoked with a periodicity of $3\Delta t$. The performance evaluation of the UKF-based RLL prediction algorithm demonstrated a good ability of the UKF to produce the estimated states that converged to the true states of the system as time progresses. We also evaluated its performance under two types of trajectories induced by node movements, which also yielded robust results.

6.2 Some Thoughts and Future Work

The path-selection algorithms proposed in Chapter 3 all select a single best path from all available paths between a source-destination pair and discard all others. While those other paths may not be as good as the selected one, some, if not all, could still satisfy a specified path lifetime requirement and be useful in routing the data packets for the source. Therefore, a natural step forward from these path-selection algorithms is to develop a multi-path routing protocol that allows simultaneous data transmissions along multiple paths.

Multi-path routing has been actively explored as a candidate solution to realize QoS provisioning in the MANET. A routing protocol first discovers a multitude of paths between the source and the destination. The source then assigns the data packets evenly to all the paths for transmission. This approach, however, does not take into

account the reliability of each path. As a result, data packets that are assigned to a less reliable path may more likely be lost due to abrupt path breakage.

We envision a more intelligent multi-path routing protocol that takes into account the path reliability. Prior to assigning data packets to different paths, the routing protocol has obtained some predictive knowledge on the reliability of each path. With this knowledge, it computes the appropriate number of data packets to be assigned to each path for transmission. A more reliable path will be assigned more packets, and a less reliable path will be assigned fewer packets. Since the reliability information gives some predictive knowledge of the path lifetime, the routing protocol will be able to take proper measures to protect the data still in transmit along the path before it actually breaks (e.g., instructing the intermediate nodes to queue the packets, and initiating a search for a new path). This would have the benefits of minimizing data loss, reducing packet delivery delay, and achieving load balancing across the network.

Developing such a multi-path routing protocol requires a good parameter that can accurately indicate the reliability of each path. In this dissertation, we have associated path reliability exclusively with the residual path lifetime, which is mobility-induced. In practice, mobility alone is not sufficient for determining path reliability. For example, one path may persist for a very long time, but most of the time the received signal strength at one node from another is barely above the ambient noise level, whereas another path that has a shorter lifetime reports consistently strong received signal strength on its constituent links. Determining which of the two is better suited to carry more data packets may not be a straightforward decision. Therefore, a path-reliability parameter solely based on mobility is simplistic, and does not bode well in real-world implementations.

This calls into question the pursuit of developing a robust path-reliability parameter that withstands the more rigorous test of a realistic wireless environment.

This is still an open problem, and at the present, we can only offer some general observations as to how this problem may be addressed.

Firstly, the mobile node should strive to gain some knowledge about the particular mobile environment in which the network is deployed. Knowledge of the local environment could give the node some idea about the pattern of mobility. Suppose a node has counted an unusually high number of neighboring nodes in its neighborhood, this may be an indication that it is in a congested area. Commonsense dictates that each node is likely to move at a slower speed. As a result, the node may reasonably assume that the link lifetime between any pair of nodes is likely to remain long¹⁴. This knowledge would lead to some confidence regarding the survivability of the link. There are some works in the literature that have proposed routing protocols based on a particular mobile environment. For example, the ABR routing protocol targets a specific in-door mobile environment, where nodes are known to linger at some location for a long duration before moving again [82]. Another example is the Manhattan mobility model, which has often been employed to simulate an urban environment in which a MANET is deployed, where the nodes are expected to move in a more predictable pattern.

Secondly, the distance measurement-based RLL-prediction algorithms proposed in Chapter 4 and Chapter 5 make RLL predictions of the current link without any knowledge from past predictions. This may have discarded valuable information that could be beneficial to the current prediction. Recent history could offer the node some pattern about the likely robustness of the current prediction. Each node would therefore be able to accumulate and learn from the previous experience in order to

¹⁴ Of course, whether or not slower mobility translates to a suitable link remains to be seen, as a higher node density may generate a larger amount of MAC-layer traffic which could make the wireless channel busy much of the time.

improve its accuracy of the current prediction.

Once we have developed such a path-reliability parameter, the next task is to utilize this information and make the proper assignment of data packets to each path. In the literature, a work by Tsirigos and Haas proposed the Diversity Coding-based Multipath Routing protocol [85][86], which dynamically allocates different numbers of data packets to different paths based on individual path reliability, which the authors assumed to have been known *a priori*. Thus, a well-defined path-reliability parameter can be utilized at the input of the Diversity Coding-based routing protocol, allowing it to intelligently assign different numbers of data packets to different paths.

Some of the future directions of the research that immediately follow what is presented in this dissertation may include the following:

- A better defined prediction parameter that takes into consideration of other factors besides the link lifetime;
- Performance evaluation of the proposed RLL-prediction algorithms under more realistic mobility conditions;
- Improve the convergence time for the UKF-based RLL prediction algorithm.

Appendix A

Proof of Theorem on Minimal Number of Distance Measurements

We prove the theorem in Section 4.3.2. Denote the trajectory that Node 2 traverses in Node 1's transmission range, i.e., AF in Figure 4-4, as D_t , and $b=AB=BC=CD$. We first recognize that there exist exactly three possible scenarios for having four periodical distance measurements during the link lifetime:

- S1: d_0 and d_1 measured before d_{min} ; d_2 and d_3 after d_{min} ,
- S2: d_0, d_1 , and d_2 measured before d_{min} ; d_3 after d_{min} , and
- S3: d_0, d_1, d_2 , and d_3 all measured before d_{min} .

In S1, the nodes are in the receding state at d_2 . Figure 4-6 depicts the ranges of values the four d_i 's can take in this state. Along the trajectory (from 0 to D_t), d_0 can only be the transmission range R , d_1 can span the interval $[D_t/4, D_t/3]$, d_2 can span the interval $[D_t/2, 2D_t/3]$, and d_3 can span the interval $[3D_t/4, D_t]$. Each such interval is called the *feasible range regions for d_i* , denoted as A_i . S1 is therefore satisfied if and only if $D_t/4 < b < D_t/3$. This is possible if $d_3 > d_1$.

In S2, the nodes are in the approaching state at d_2 . Figure 4-7 illustrates the respective A_i 's for the d_i 's: $A_0=\{R\}$, $A_1=\{D_t/6, D_t/4\}$, $A_2=\{D_t/3, D_t/2\}$, and $A_3=\{D_t/2, 3D_t/4\}$. S2 is satisfied if $2b < D_t/2$ and $D_t/2 < 3b < D_t$, or $D_t/6 < b < D_t/4$. This corresponds to $d_0 > d_1 > d_2$ and $d_1 > d_3$.

In S3, the two nodes are also in the approaching state at d_2 . The respective A_i 's for

the d_i 's, illustrated in Figure 4-8, are as follows: $A_0=\{R\}$, $A_1=\{0, D/6\}$, $A_2=\{0, D/3\}$, and $A_3=\{0, D/2\}$. S3 is thus satisfied if and only if $0 < b < D/6$, and this corresponds to $d_0 > d_1 > d_2 > d_3$.

By comparing the four d_i 's, it is clear that S1 occurs only when $d_1 < d_3$, and S2 and S3 both occur when $d_1 > d_3$. In other words, to distinguish between the two states with four distance measurements, we only need to verify whether $d_1 > d_3$ holds: if it does, the two nodes are in approaching state at d_2 ; otherwise, they are in receding state at d_2 . d_3 is therefore responsible for determining which of the two states the nodes are in at d_2 , whereas only the first three measurements (d_0 through d_2) are needed to actually compute the RLL. Therefore, using four distance measurements completely eliminates the state ambiguity and always yields a unique solution for RLL, and the proof is completed. □

Appendix B

Derivations of MPT with N Distance Measurements

We now derive the results for MPT-N, where $N=5, 6, 7, 8.$, as introduced in Section 4.4.2. We begin with MPT-5. First, assume all distance measurements are error-free. Since it periodically makes five distance measurements, their x- and y-coordinates, denoted as (x_i, y_i) , $\forall i = 0, \dots, 4$, can be written as follows:

$$\begin{cases} (x_0, y_0) = (0, 0) & (x_1, y_1) = (x_1, \alpha x_1) \\ (x_2, y_2) = (2x_1, 2\alpha x_1) & (x_3, y_3) = (3x_1, 3\alpha x_1) \\ (x_4, y_4) = (4x_1, 4\alpha x_1) \end{cases} \quad (\text{B-1})$$

The relationships between d_i 's ($\forall i = 1, \dots, 4$) and their coordinates are described by the following system of equations:

$$\begin{cases} (x_1 - d_0)^2 + (\alpha x_1)^2 = d_1^2 \\ (2x_1 - d_0)^2 + (2\alpha x_1)^2 = d_2^2 \\ (3x_1 - d_0)^2 + (3\alpha x_1)^2 = d_3^2 \\ (4x_1 - d_0)^2 + (4\alpha x_1)^2 = d_4^2 \end{cases} \quad (\text{B-2})$$

By pairing the first equation in the above system with the second, third and fourth equation respectively, and multiplying the appropriate coefficient to factor out the term that contains α , we obtain the following three:

$$\begin{cases} 4(x_1 - d_0)^2 + 4\alpha^2 x_1^2 = 4d_1^2 \\ (2x_1 - d_0)^2 + 4\alpha^2 x_1^2 = d_2^2 \end{cases} \quad \begin{cases} 9(x_1 - d_0)^2 + 9\alpha^2 x_1^2 = 9d_1^2 \\ (3x_1 - d_0)^2 + 9\alpha^2 x_1^2 = d_3^2 \end{cases}$$

$$\begin{cases} 16(x_1 - d_0)^2 + 16\alpha^2 x_1^2 = 16d_1^2 \\ (4x_1 - d_0)^2 + 16\alpha^2 x_1^2 = d_4^2 \end{cases}$$

We can solve for x_1 as follows:

$$x_1 = \frac{3d_0^2 - 4d_1^2 + d_2^2}{4d_0} = \frac{8d_0^2 - 9d_1^2 + d_3^2}{12d_0} = \frac{15d_0^2 - 16d_1^2 + d_4^2}{24d_0} \quad (\text{B-3})$$

The three equalities in the above equation can be manipulated into the following equation:

$$\begin{aligned} 18d_0^2 - 24d_1^2 + 6d_2^2 &= 16d_0^2 - 218d_1^2 + 2d_3^2 = 15d_0^2 - 16d_1^2 + d_4^2 \\ \Rightarrow 2d_0^2 - 6d_1^2 + 6d_2^2 - 2d_3^2 &= d_0^2 - 2d_1^2 + 2d_3^2 - d_4^2 = 0 \\ \Rightarrow d_0^2 - 4d_1^2 + 6d_2^2 - 4d_3^2 + d_4^2 &= 0 \end{aligned} \quad (\text{B-4})$$

That is, in order for the five periodical distance measurements to be perfectly aligned along the linear trajectory, they must satisfy the last equation above.

With MPT-5, we wish to compute the estimated distance measurements \tilde{d}_i , $\forall i = 0, \dots, 4$, such that the \tilde{d}_i 's satisfy the above equation. Let $\tilde{d}_i = \hat{d}_i + e_i$. We formulate the minimization problem for MPT-5 as follows:

$$\text{Minimize: } \sum_{i=0}^4 e_i^2$$

$$\text{S. t: } \begin{aligned} & (\hat{d}_0 + e_0)^2 - 4(\hat{d}_1 + e_1)^2 + 6(\hat{d}_2 + e_2)^2 - \\ & 4(\hat{d}_3 + e_3)^2 + (\hat{d}_4 + e_4)^2 = 0 \end{aligned} \quad (\text{B-5})$$

The minimization is solved by applying the Lagrangian multiplier in the Lagrangian function $f(\underline{e}, \lambda)$ below:

$$f(\underline{e}, \lambda) = \sum_{i=0}^4 e_i^2 + \lambda \left[\begin{aligned} & (\hat{d}_0 + e_0)^2 - 4(\hat{d}_1 + e_1)^2 + \\ & 6(\hat{d}_2 + e_2)^2 - 4(\hat{d}_3 + e_3)^2 + (\hat{d}_4 + e_4)^2 \end{aligned} \right] \quad (\text{B-6})$$

The gradient of $f(\underline{e}, \lambda)$ is given by:

$$\nabla f = \begin{cases} \frac{\partial}{\partial e_0} f(\underline{e}, \lambda) = 2e_0 + 2\lambda(\hat{d}_0 + e_0) = 0 \Rightarrow e_0 = \frac{-\lambda}{1+\lambda} \hat{d}_0 \\ \frac{\partial}{\partial e_1} f(\underline{e}, \lambda) = 2e_1 - 8\lambda(\hat{d}_1 + e_1) = 0 \Rightarrow e_1 = \frac{4\lambda}{1-4\lambda} \hat{d}_1 \\ \frac{\partial}{\partial e_2} f(\underline{e}, \lambda) = 2e_2 + 12\lambda(\hat{d}_2 + e_2) = 0 \Rightarrow e_2 = \frac{-6\lambda}{1+6\lambda} \hat{d}_2 \\ \frac{\partial}{\partial e_3} f(\underline{e}, \lambda) = 2e_3 - 8\lambda(\hat{d}_3 + e_3) = 0 \Rightarrow e_3 = \frac{4\lambda}{1-4\lambda} \hat{d}_3 \\ \frac{\partial}{\partial e_4} f(\underline{e}, \lambda) = 2e_4 + 2\lambda(\hat{d}_4 + e_4) = 0 \Rightarrow e_4 = \frac{-\lambda}{1+\lambda} \hat{d}_4 \\ \frac{\partial}{\partial \lambda} f(\underline{e}, \lambda) = (\hat{d}_0 + e_0)^2 - 4(\hat{d}_1 + e_1)^2 + 6(\hat{d}_2 + e_2)^2 - 4(\hat{d}_3 + e_3)^2 + (\hat{d}_4 + e_4)^2 = 0 \end{cases}$$

Substituting the expressions for e_i into the last equation in the gradient, we obtain:

$$\begin{aligned}
\frac{\partial}{\partial \lambda} f(e, \lambda) &= \frac{\hat{d}_0^2}{(1+\lambda)^2} - \frac{4\hat{d}_1^2}{(1-4\lambda)^2} + \frac{6\hat{d}_2^2}{(1+6\lambda)^2} - \frac{4\hat{d}_3^2}{(1-4\lambda)^2} + \frac{\hat{d}_4^2}{(1+\lambda)^2} = 0 \\
& \left(\hat{d}_0^2 + \hat{d}_4^2 \right) \left[576\lambda^4 - 96\lambda^3 - 44\lambda^2 + 4\lambda + 1 \right] + \\
\Rightarrow & \left(\hat{d}_1^2 + \hat{d}_3^2 \right) \left[-144\lambda^4 - 336\lambda^3 - 244\lambda^2 - 56\lambda - 4 \right] + \\
& \hat{d}_2^2 \left[96\lambda^4 + 144\lambda^3 + 6\lambda^2 - 364\lambda + 6 \right] = 0
\end{aligned} \tag{B-7}$$

Re-arranging the above equation, we reach the following equation to solve for λ :

$$\begin{aligned}
& \lambda^4 \left[576(\hat{d}_0^2 + \hat{d}_4^2) - 144(\hat{d}_1^2 + \hat{d}_3^2) + 96\hat{d}_2^2 \right] + \lambda^3 \left[-96(\hat{d}_0^2 + \hat{d}_4^2) - 336(\hat{d}_1^2 + \hat{d}_3^2) + 144\hat{d}_2^2 \right] + \\
& \lambda^2 \left[-44(\hat{d}_0^2 + \hat{d}_4^2) - 244(\hat{d}_1^2 + \hat{d}_3^2) + 6\hat{d}_2^2 \right] + \lambda \left[4(\hat{d}_0^2 + \hat{d}_4^2) - 56(\hat{d}_1^2 + \hat{d}_3^2) - 36\hat{d}_2^2 \right] + \\
& \left[(\hat{d}_0^2 + \hat{d}_4^2) - 4(\hat{d}_1^2 + \hat{d}_3^2) + 6\hat{d}_2^2 \right] = 0
\end{aligned} \tag{B-8}$$

Note that of the four solutions for λ , two are complex-valued, and can be immediately eliminated. Of the remaining two real solutions, we note that one of them is always much closer to 0 than the other. Since we assume that the measurement error contained in each \hat{d}_i is very small compared with the true distance, the λ that is closer to 0 is the desired solution for the above equation.

Once λ is computed, we substitute it into ∇f to solve for the e_i 's, which would yield the \tilde{d}_i 's \tilde{x}_1 . Subsequently, we can compute the estimated trajectory slope $\tilde{\alpha}$ by the expression:

$$\tilde{\alpha} = \sqrt{\frac{\tilde{d}_1^2 - (\tilde{x}_1 - \tilde{d}_0)^2}{\tilde{x}_1^2}}, \quad \tilde{y}_1 = \tilde{\alpha}\tilde{x}_1 \tag{B-9}$$

The derivations for MPT-6 through MPT-8 are similar to those for MPT-5. For

MPT-6, the minimization is formulated as follows:

$$\begin{aligned}
& \text{Minimize: } \sum_{i=0}^5 e_i^2 \\
& \text{S. t.: } \begin{aligned} & (\hat{d}_0 + e_0)^2 - 5(\hat{d}_1 + e_1)^2 + 10(\hat{d}_2 + e_2)^2 - 10(\hat{d}_3 + e_3)^2 + \\ & 5(\hat{d}_4 + e_4)^2 - (\hat{d}_5 + e_5)^2 = 0 \end{aligned}
\end{aligned} \tag{B-10}$$

The e_i 's are solved with Lagrangian λ as follows:

$$\begin{aligned}
e_0 &= \frac{-\lambda}{1+\lambda} \hat{d}_0 & e_1 &= \frac{5\lambda}{1-5\lambda} \hat{d}_1 & e_2 &= \frac{-10\lambda}{1+10\lambda} \hat{d}_2 \\
e_4 &= \frac{-5\lambda}{1+5\lambda} \hat{d}_4 & e_5 &= \frac{\lambda}{1-\lambda} \hat{d}_5
\end{aligned} \tag{B-11}$$

λ is found by solving the following ten-degree polynomial expression:

$$\begin{aligned}
& \lambda^{10}T_{10} + \lambda^9T_9 + \lambda^8T_8 + \lambda^7T_7 + \lambda^6T_6 + \lambda^5T_5 + \lambda^4T_4 + \\
& \lambda^3T_3 + \lambda^2T_2 + \lambda T_1 + T_0 = 0
\end{aligned} \tag{B-12}$$

where:

$$\begin{aligned}
T_{10} &= 6250000(\hat{d}_0^2 - \hat{d}_5^2) - 1250000(\hat{d}_1^2 - \hat{d}_4^2) + 625000(\hat{d}_2^2 - \hat{d}_3^2) \\
T_9 &= -12500000(\hat{d}_0^2 + \hat{d}_5^2) - 5000000(\hat{d}_1^2 + \hat{d}_4^2) - 1250000(\hat{d}_2^2 + \hat{d}_3^2) \\
T_8 &= 5625000(\hat{d}_0^2 - \hat{d}_5^2) + 2475000(\hat{d}_1^2 - \hat{d}_4^2) - 1293750(\hat{d}_2^2 - \hat{d}_3^2) \\
T_7 &= 1250000(\hat{d}_0^2 + \hat{d}_5^2) - 1010000(\hat{d}_1^2 + \hat{d}_4^2) + 260000(\hat{d}_2^2 + \hat{d}_3^2) \\
T_6 &= -604375(\hat{d}_0^2 - \hat{d}_5^2) - 1199125(\hat{d}_1^2 - \hat{d}_4^2) + 713000(\hat{d}_2^2 - \hat{d}_3^2)
\end{aligned}$$

$$\begin{aligned}
T_5 &= -41250(\hat{d}_0^2 + \hat{d}_5^2) - 520050(\hat{d}_1^2 + \hat{d}_4^2) - 145000(\hat{d}_2^2 + \hat{d}_3^2) \\
T_4 &= 20375(\hat{d}_0^2 - \hat{d}_5^2) - 26755(\hat{d}_1^2 - \hat{d}_4^2) - 44740(\hat{d}_2^2 - \hat{d}_3^2) \\
T_3 &= 500(\hat{d}_0^2 + \hat{d}_5^2) + 10100(\hat{d}_1^2 + \hat{d}_4^2) + 10400(\hat{d}_2^2 + \hat{d}_3^2) \\
T_2 &= -249(\hat{d}_0^2 - \hat{d}_5^2) + 885(\hat{d}_1^2 - \hat{d}_4^2) + 480(\hat{d}_2^2 - \hat{d}_3^2) \\
T_1 &= -2(\hat{d}_0^2 + \hat{d}_5^2) - 50(\hat{d}_1^2 + \hat{d}_4^2) - 200(\hat{d}_2^2 + \hat{d}_3^2) \\
T_0 &= (\hat{d}_0^2 - \hat{d}_5^2) - 5(\hat{d}_1^2 - \hat{d}_4^2) + 10(\hat{d}_2^2 - \hat{d}_3^2)
\end{aligned}$$

Of the ten solutions for λ , eight are complex-valued, and can be immediately eliminated. Of the remaining two real solutions, the λ that is closer to 0 is the desired solution for the above equation. Subsequently, we can compute e_i 's, \tilde{d}_i 's, \tilde{x}_1 , and $\tilde{\alpha}$, and \tilde{y}_1 .

For MPT-7, the minimization is formulated as follows:

$$\begin{aligned}
&\text{Minimize: } \sum_{i=0}^6 e_i^2 \\
&\text{S. t.: } \begin{aligned}
&(\hat{d}_0 + e_0)^2 - 6(\hat{d}_1 + e_1)^2 + 15(\hat{d}_2 + e_2)^2 - 20(\hat{d}_3 + e_3)^2 + \\
&15(\hat{d}_4 + e_4)^2 - 6(\hat{d}_5 + e_5)^2 + (\hat{d}_6 + e_6)^2 = 0
\end{aligned}
\end{aligned} \tag{B-13}$$

The e_i 's are solved with Lagrangian λ as follows:

$$\begin{aligned}
e_0 &= \frac{-\lambda}{1+\lambda} \hat{d}_0 & e_1 &= \frac{6\lambda}{1-6\lambda} \hat{d}_1 & e_2 &= \frac{-15\lambda}{1+15\lambda} \hat{d}_2 & e_3 &= \frac{20\lambda}{1-20\lambda} \hat{d}_3 \\
e_4 &= \frac{-15\lambda}{1+15\lambda} \hat{d}_4 & e_5 &= \frac{6\lambda}{1-6\lambda} \hat{d}_5 & e_6 &= \frac{-\lambda}{1+\lambda} \hat{d}_5
\end{aligned} \tag{B-14}$$

λ is found by solving the following six-degree polynomial expression:

$$\lambda^6 T_6 + \lambda^5 T_5 + \lambda^4 T_4 + \lambda^3 T_3 + \lambda^2 T_2 + \lambda T_1 + T_0 = 0, \quad (\text{B-15})$$

where:

$$\begin{aligned} T_6 &= 3240000(\hat{d}_0^2 + \hat{d}_6^2) - 540000(\hat{d}_1^2 + \hat{d}_5^2) + 216000(\hat{d}_2^2 + \hat{d}_4^2) - 162000\hat{d}_3^2 \\ T_5 &= -972000(\hat{d}_0^2 + \hat{d}_6^2) - 1098000(\hat{d}_1^2 + \hat{d}_5^2) + 338400(\hat{d}_2^2 + \hat{d}_4^2) - 291600\hat{d}_3^2 \\ T_4 &= 33300(\hat{d}_0^2 + \hat{d}_6^2) - 572550(\hat{d}_1^2 + \hat{d}_5^2) + 42540(\hat{d}_2^2 + \hat{d}_4^2) - 95220\hat{d}_3^2 \\ T_3 &= 9540(\hat{d}_0^2 + \hat{d}_6^2) - 11040(\hat{d}_1^2 + \hat{d}_5^2) - 66900(\hat{d}_2^2 + \hat{d}_4^2) + 36000\hat{d}_3^2 \\ T_2 &= -419(\hat{d}_0^2 + \hat{d}_6^2) + 3564(\hat{d}_1^2 + \hat{d}_5^2) + 12195(\hat{d}_2^2 + \hat{d}_4^2) + 1240\hat{d}_3^2 \\ T_1 &= -22(\hat{d}_0^2 + \hat{d}_6^2) + 48(\hat{d}_1^2 + \hat{d}_5^2) - 750(\hat{d}_2^2 + \hat{d}_4^2) - 400\hat{d}_3^2 \\ T_0 &= (\hat{d}_0^2 + \hat{d}_6^2) - 6(\hat{d}_1^2 + \hat{d}_5^2) + 15(\hat{d}_2^2 + \hat{d}_4^2) - 20\hat{d}_3^2 \end{aligned}$$

Of the six solutions for λ , four are complex-valued, and can be immediately eliminated. Of the remaining two real solutions, the λ that is closer to 0 is the desired solution for the above equation. Subsequently, we can compute e_i 's, \tilde{d}_i 's, \tilde{x}_i , and $\tilde{\alpha}$, and \tilde{y}_i .

For MPT-8, the minimization is formulated as follows:

$$\begin{aligned} \text{Minimize: } & \sum_{i=0}^7 e_i^2 \\ \text{Such that: } & (\hat{d}_0 + e_0)^2 - 7(\hat{d}_1 + e_1)^2 + 21(\hat{d}_2 + e_2)^2 - 35(\hat{d}_3 + e_3)^2 + \\ & 35(\hat{d}_4 + e_4)^2 - 21(\hat{d}_5 + e_5)^2 + 7(\hat{d}_6 + e_6)^2 - (\hat{d}_7 + e_7)^2 = 0 \end{aligned} \quad (\text{B-16})$$

The e_i 's are solved with Lagrangian λ as follows:

$$\begin{aligned}
e_0 &= \frac{-\lambda}{1+\lambda} \hat{d}_0 & e_1 &= \frac{7\lambda}{1-7\lambda} \hat{d}_1 & e_2 &= \frac{-21\lambda}{1+21\lambda} \hat{d}_2 & e_3 &= \frac{35\lambda}{1-35\lambda} \hat{d}_3 \\
e_4 &= \frac{-35\lambda}{1+35\lambda} \hat{d}_4 & e_5 &= \frac{21\lambda}{1-21\lambda} \hat{d}_5 & e_6 &= \frac{-7\lambda}{1+7\lambda} \hat{d}_6 & e_7 &= \frac{\lambda}{1-\lambda} \hat{d}_7
\end{aligned} \tag{B-17}$$

λ is found by solving the following fourteen-degree polynomial expression:

$$\begin{aligned}
&\lambda^{14}T_{14} + \lambda^{13}T_{13} + \lambda^{12}T_{12} + \lambda^{11}T_{11} + \lambda^{10}T_{10} + \lambda^9T_9 + \lambda^8T_8 + \lambda^7T_7 + \\
&\lambda^6T_6 + \lambda^5T_5 + \lambda^4T_4 + \lambda^3T_3 + \lambda^2T_2 + \lambda T_1 + T_0 = 0
\end{aligned}$$

where:

$$\begin{aligned}
T_{14} &= 70071564550625(\hat{d}_0^2 - \hat{d}_7^2) - 100102166364375(\hat{d}_1^2 - \hat{d}_6^2) + 33367388788125(\hat{d}_2^2 - \hat{d}_5^2) - \\
&20020433272875(\hat{d}_3^2 - \hat{d}_4^2) \\
T_{13} &= -1401430329101250(\hat{d}_0^2 + \hat{d}_7^2) - 28600618961250(\hat{d}_1^2 + \hat{d}_6^2) - 3177846551250(\hat{d}_2^2 + \hat{d}_5^2) - \\
&1144024758450(\hat{d}_3^2 + \hat{d}_4^2) \\
T_{12} &= 667792674279675(\hat{d}_0^2 - \hat{d}_7^2) + 198778841561475(\hat{d}_1^2 - \hat{d}_6^2) - 68075526168825(\hat{d}_2^2 - \hat{d}_5^2) + \\
&40932479492415(\hat{d}_3^2 - \hat{d}_4^2) \\
T_{11} &= 65844980541900(\hat{d}_0^2 + \hat{d}_7^2) + 57377640833100(\hat{d}_1^2 + \hat{d}_6^2) + 6490589445900(\hat{d}_2^2 + \hat{d}_5^2) + \\
&2339932725900(\hat{d}_3^2 + \hat{d}_4^2) \\
T_{10} &= -32444986039319(\hat{d}_0^2 - \hat{d}_7^2) - 97239906431983(\hat{d}_1^2 - \hat{d}_6^2) + 36061817245461(\hat{d}_2^2 - \hat{d}_5^2) - \\
&21815065236755(\hat{d}_3^2 - \hat{d}_4^2) \\
T_9 &= -955008463262(\hat{d}_0^2 + \hat{d}_7^2) - 28953802671038(\hat{d}_1^2 + \hat{d}_6^2) - 3449176675182(\hat{d}_2^2 + \hat{d}_5^2) - \\
&1248485305550(\hat{d}_3^2 + \hat{d}_4^2) \\
T_8 &= 475318313211(\hat{d}_0^2 - \hat{d}_7^2) - 1448072212125(\hat{d}_1^2 - \hat{d}_6^2) - 1366598128329(\hat{d}_2^2 - \hat{d}_5^2) + \\
&914457189135(\hat{d}_3^2 - \hat{d}_4^2)
\end{aligned}$$

$$\begin{aligned}
T_7 &= 4371836840(\hat{d}_0^2 + \hat{d}_7^2) + 177159014312(\hat{d}_1^2 + \hat{d}_6^2) + 137973465000(\hat{d}_2^2 + \hat{d}_5^2) + \\
&53273868200(\hat{d}_3^2 + \hat{d}_4^2) \\
T_6 &= -2181733477(\hat{d}_0^2 - \hat{d}_7^2) + 11329319155(\hat{d}_1^2 - \hat{d}_6^2) + 12905210703(\hat{d}_2^2 - \hat{d}_5^2) - \\
&11470281865(\hat{d}_3^2 - \hat{d}_4^2) \\
T_5 &= -8369886(\hat{d}_0^2 + \hat{d}_7^2) - 378541758(\hat{d}_1^2 + \hat{d}_6^2) - 1541932686(\hat{d}_2^2 + \hat{d}_5^2) - 698933550(\hat{d}_3^2 + \hat{d}_4^2) \\
T_4 &= 4181513(\hat{d}_0^2 - \hat{d}_7^2) - 25895135(\hat{d}_1^2 - \hat{d}_6^2) + 13097133(\hat{d}_2^2 - \hat{d}_5^2) + 32118485(\hat{d}_3^2 - \hat{d}_4^2) \\
T_3 &= 6860(\hat{d}_0^2 + \hat{d}_7^2) + 326732(\hat{d}_1^2 + \hat{d}_6^2) + 2249100(\hat{d}_2^2 + \hat{d}_5^2) + 2405900(\hat{d}_3^2 + \hat{d}_4^2) \\
T_2 &= -3429(\hat{d}_0^2 - \hat{d}_7^2) + 22995(\hat{d}_1^2 - \hat{d}_6^2) - 44289(\hat{d}_2^2 - \hat{d}_5^2) - 8505(\hat{d}_3^2 - \hat{d}_4^2) \\
T_1 &= -2(\hat{d}_0^2 + \hat{d}_7^2) - 98(\hat{d}_1^2 + \hat{d}_6^2) - 882(\hat{d}_2^2 + \hat{d}_5^2) - 2450(\hat{d}_3^2 + \hat{d}_4^2) \\
T_0 &= (\hat{d}_0^2 - \hat{d}_7^2) - 7(\hat{d}_1^2 - \hat{d}_6^2) + 21(\hat{d}_2^2 - \hat{d}_5^2) - 35(\hat{d}_3^2 - \hat{d}_4^2)
\end{aligned}$$

Of the 14 solutions for λ , 12 are complex-valued, and can be immediately eliminated.

Of the remaining two real solutions, the λ that is closer to 0 is the desired solution for

the above equation. Subsequently, we can compute e_i 's, \tilde{d}_i 's, \tilde{x}_1 , and $\tilde{\alpha}$, and \tilde{y}_1 .

Finally, for each N , the predicted RLL τ seconds after \hat{d}_{N-1} is measured is therefore given by the formula:

$$R\hat{L}L(t_0 + (N-1)\Delta t + \tau) = \frac{2\tilde{d}_0\Delta t}{1 + \tilde{\alpha}^2} \sqrt{\frac{1 + \tilde{\alpha}^2}{\tilde{x}_1^2 + \tilde{y}_1^2}} - (N-1)\Delta t - \tau, \quad (\text{B-18})$$

where \tilde{d}_0 , \tilde{x}_1 , \tilde{y}_1 , and $\tilde{\alpha}$ are computed from the respective formulations.

BIBLIOGRAPHY

- [1] M. Allman and V. Paxson, "On Estimating End-to-End Network Path Properties," *Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pp.263-274, 1999.
- [2] F. Bai, N. Sadagopan, and A. Helmy, "IMPORTANT: A framework to systematically analyze the Impact of Mobility on Performance of Routing protocols for Adhoc Networks," *Conference on Computer Communications*, vol. 2, pp.825-835, March 2003.
- [3] F. Bai, N. Sadagopan, B. Krishnamachari, and A. Helmy, "Modeling Path Duration Distributions in MANETs and Their Impact on Reactive Routing Protocols," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 7, pp.1357-1373, September 2004.
- [4] D. Bertsekas and R. Gallager, *Data Networks*, 2nd Edition, Prentice-Hall, Inc., 1992.
- [5] J. Boleng and T. Camp, "Adaptive Location Aided Mobile Ad Hoc Network Routing," *IEEE International Conference on Performance, Computing, and Communications*, pp.423-432, 2004.
- [6] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," *ACM/IEEE International Conference on Mobile Computing and Networking*, pp.85-97, October 1998.
- [7] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *WCMC Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no. 5, pp. 483-502, 2002.
- [8] S. Capkun, M. Hamdi, and J.-P. Hubaux, "GPS-free Positioning in Mobile Ad-Hoc Networks," *IEEE Hawaii International Conference on System Sciences*, pp.1-10, January 2001.

- [9] S. Capkun, J.-P. Hubaux, and L. Buttyan, "Mobility Helps Security in Ad Hoc Networks," *ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp.46-56, June 2003.
- [10] S. Chakrabarti and A. Mishra, "QoS Issues in Ad Hoc Networks," *IEEE Communications Magazine*, pp.142-148, February 2001.
- [11] L. Chen and W. B. Heinzelman, "QoS-Aware Routing Based on Bandwidth Estimation for Mobile Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 3, pp.561-572, March 2005.
- [12] C.-C. Chong, F. Watanabe, and H. Inamura, "Potential of UWB Technology for the Next Generation Wireless Communications," *IEEE Ninth International Symposium on Spread Spectrum Techniques and Applications*, pp. 422-429, August 2006.
- [13] W. C. Chung and D. S. Ha, "An Accurate Ultra Wideband (UWB) Ranging For Precision Asset Location," *International Conference on Ultra Wideband Systems and Technologies*, pp.389-393, November 2003.
- [14] N. S. Correal, S. Kyperountas, Q. Shi, and M. Welborn, "An UWB Relative Location System," *IEEE Conference on Ultra Wideband Systems and Technologies*, pp.394-397, Nov. 2003.
- [15] C. K. Coursey, *Understanding Digital PCS: The TDMA Standard*, Artech House Publishers, 1999.
- [16] S. K. Das, A. Mukherjee, S. Bandyopadhyay, K. Paul, and D. Saha, "Improving Quality-of-Service in Ad Hoc Wireless Networks with Adaptive Multi-path Routing," *IEEE Global Telecommunications Conference*, vol.1, pp. 261-265, November 2000.
- [17] J. L. Devore, *Probability and Statistics for Engineering and the Sciences*, ed., Duxbury Press, 1995.
- [18] R. Dube, C. D. Rais, K.-Y. Wang, and S. K. Tripathi, "Signal Stability-Based Adaptive Routing (SSA) for Ad Hoc Mobile Networks," *Personal Communications*, vol. 4, no. 1, pp.36-45, February 1997.

- [19] J. Ellis, "UWB for Low Rate Precision Applications," *Ultra-wideband World*, September 2004.
- [20] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, "Age Matters: Efficient Route Discovery in Mobile Ad Hoc Networks Using Encounter Ages," *ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp.257-266, June 2003.
- [21] M. Gerharz, C. de Waal, M. Frank, and P. Martini, "Link Stability in Mobile Wireless Ad Hoc Networks," *IEEE Conference on Local Computer Networks*, pp.30-39, November 2002.
- [22] M. Gerharz, C. de Waal, P. Martini, and P. James, "Strategies for Finding Stable Paths in Mobile Wireless Ad Hoc Networks," *IEEE Conference on Local Computer Networks*, pp.130-139, October 2003.
- [23] J. H. Gove and D. Y. Hollinger, "Application of a Dual Unscented Kalman Filter for Simultaneous State and Parameter Estimation in Problems of Surface-atmosphere Exchange," *Journal of Geophysical Research*, vol. 111, pp.1-21, April 2006.
- [24] P. E. Greenwood and M. Nikulin, *A Guide to Chi-Squared Testing*, ed., John Wiley & Sons, 1996.
- [25] M. Grossglauser and D. N. C. Tse, "Mobility Increases the Capacity of Ad Hoc Wireless Networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp.477-486, August 2002.
- [26] I. Gruber and H. Li, "Link Expiration Times in Mobile Ad Hoc Networks," *IEEE Conference on Local Computer Networks*, pp.743-750. November 2002.
- [27] I. Gruber and H. Li, "Path Expiration Times in Mobile Ad Hoc Networks," *European Personal Mobile Communications Conference*, pp.27-31, April 2003.
- [28] R. A. Guerin and A. Orda, "QoS Routing in Networks with Inaccurate Information: Theory and Algorithms," *IEEE/ACM Transactions on Networking*, vol. 7, no. 3, pp.350-364, June 1999.

- [29] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp.388-404, Mar. 2000.
- [30] I. Guvenc, and Z. Sahinoglu, "Threshold-based TOA Estimation for Impulse Radio UWB Systems," *International Conference on Ultra-Wideband*, pp.420-425, September 2005.
- [31] Z. J. Haas and E. Y. Hua, "Residual Link Lifetime Prediction with Limited Information Input in Mobile Ad Hoc Networks," *IEEE Conference on Computer Communications*, pp.1867-1875, April 2008.
- [32] Y. Han, R. J. La, and A. M. Makowski, "Distribution of path durations in mobile ad-hoc networks – Palm's Theorem at work," *Technical Research Report, Center for Satellite and Hybrid Communication Networks*, 2003.
- [33] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher, "Range-free Localization and Its Impact on Large Scale Sensor Networks," *ACM Transactions on Embedded Computing Systems*, vol. 4, no. 4, pp.877-906, November 2005.
- [34] L. Hu and D. Evans, "Localization for Mobile Sensor Networks," *ACM International Conference on Mobile Computing and Networking*, pp.45-57, September 2004.
- [35] E. Y. Hua and Z. J. Haas, "Study of the Effects of Mobility on Residual Path Lifetime in Mobile Ad Hoc Networks," *IEEE Second Upstate New York Workshop on Communications and Networking*, pp.33-37, November 2005.
- [36] E. Y. Hua and Z. J. Haas, "Path Selection Algorithms in Homogeneous Mobile Ad Hoc Networks," *ACM International Wireless Communications and Mobile Computing Conference*, pp.275-280, July 2006.
- [37] E. Y. Hua and Z. J. Haas, "An Algorithm for Prediction of Link Lifetime in MANET Based on Unscented Kalman Filter," *accepted by Communications Letters*, 2009.
- [38] T. Imielinski and H. Korth, *Mobile Computing*, ed., Kluwer Academic Publishers, 1996.

- [39] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen, "Scalable Routing Strategies for Ad Hoc Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp.1369-1379, August 1999.
- [40] S. Jiang, D. He, and J. Rao, "A Prediction-Based Link Availability Estimation for Routing Metrics in MANETs," *IEEE/ACM Transactions on Networking*, vol. 13, no. 6, pp.1302-1312, December 2005.
- [41] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, Imielinski and Korth, Eds., Kluwer Academic Publications, vol. 353, 1996.
- [42] S. J. Julier and J. K. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp.401-422, March 2004.
- [43] A. Kamat and R. Prakash, "Effects of Link Stability and Directionality of Motion on Routing Algorithms in MANETs," *International Conference on Computer Communications and Networks*, pp.380-385, October 2000.
- [44] H. Kim, H. Sadjadpour, and J. J. Garcia-Luna-Aceves, "A closer look at the physical and protocol models for wireless ad hoc networks with multi-packet reception," *42nd Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, October 26-29, 2008.
- [45] R. Korsnes, K. Ovsthus, F. Y. Li, L. Landmark, and O. Kure, "Link Lifetime Prediction for Optimal Routing in Mobile Ad Hoc Networks," *Military Communications Conference*, vol. 2, pp.1245-1251, October 2005.
- [46] J.-Y. Lee and R. A. Scholtz, "Ranging in a Dense Multipath Environment Using an UWB Radio Link," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 9, pp.1677-1683, December 2002.
- [47] S.-B. Lee, G.-S. Ahn, X. Zhang, and A. T. Campbell, "INSIGNIA: An IP-Based Quality of Service Framework for Mobile Ad Hoc Networks," *Journal of Parallel and Distributed Computing*, vol. 60, pp.374-406, 2000.
- [48] S.-J. Lee and M. Gerla, "Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks," *IEEE International Conference on Communications*, vol. 10, pp.3201-3205, June 2001.

- [49] Z. Li, L. Sun, and E. C. Ifeachor, "Range-based Mobility Estimations in MANETs with Application to Link Availability Prediction," *IEEE International Conference on Communications*, June, pp.3376-3382, June 2007.
- [50] B. Liang and Z. J. Haas, "Predictive Distance-Based Mobility Management for Multidimensional PCS Networks," *ACM Transactions on Networking*, vol. 11, no. 5, pp.718-732, October 2003.
- [51] C. R. Lin and J.-S. Liu, "QoS Routing in Ad Hoc Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp.1426-1438, August 1999.
- [52] 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, no. 6, pp.922-936, August 1998.
- [53] A. B., McDonald and T. Znati, "A Mobility-Based Frame Work for Adaptive Clustering in Wireless Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp.1466-1487, August 1999.
- [54] A. B. McDonald and T. Znati, "A Path Availability Model for Wireless Ad-Hoc Networks," *IEEE Wireless Communications and Networking Conference*, vol. 1, pp.35-40, September 1999.
- [55] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust Distributed Network Localization with Noisy Range Measurements," *ACM Conference on Embedded Networked Sensor Systems*, pp.50-61, November 2004.
- [56] S. Murthy and J. J. Garcia-Luna-Aceves, "Congestion-Oriented Shortest Multipath Routing," *IEEE Joint Conference of Computer and Communications Societies*, vol. 3, pp.1028-1036, March 1996.
- [57] A. Nasipuri and S. R. Das, "On-Demand Multipath Routing for Mobile Ad Hoc Networks," *Computer Communications and Networks*, pp. 64-70, 1999.
- [58] A. Nasipuri, R. Castaneda, and S. R. Das, "Performance of Multipath Routing for On-Demand Protocols in Mobile Ad Hoc Networks," *Mobile Networks Application Journal*, vol. 6, no. 4, pp. 339-349, 2001.

- [59] W. Navidi and T. Camp, "Stationary Distributions for the Random Waypoint Mobility Model," *IEEE Transactions on Mobile Computing*, vol. 3, no. 1, pp.99-108, January 2004.
- [60] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) Using AoA," *IEEE Joint Conference of Computer and Communications Societies*, vol. 3, pp.1734-1743, March 2003.
- [61] P. Papadimitratos, Z. J. Haas, and E. G. Sirer, "Path Set Selection in Mobile Ad Hoc Networks," *ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp.1-11, June 2002.
- [62] V. D. Park and M. S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," *IEEE Joint Conference of Computer and Communications Societies*, vol. 3, pp.1405-1413, April 1997.
- [63] P. N. Pathirana, A. V. Savkin, and S. Jha, "Mobility Modeling and Trajectory Prediction for Cellular Networks with Mobile Base Stations," *ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp.213-221, June 2003.
- [64] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative Location Estimation in Wireless Sensor Networks," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp.2137-2148, August 2003.
- [65] N. B. Priyantha, H. Balakrishnan, E. D. Demaine, and S. Teller, "Mobile-Assisted Localization in Wireless Sensor Networks," *IEEE Conference on Computer Communications*, vol. 1, pp.172-183, Mar. 13-17, 2005.
- [66] J. Raju and J. J. Garcia-Luna-Aceves, "A New Approach to On-demand Loop-Free Multipath Routing," *International Conference on Computer Communications and Networks*, pp.522-527, October 1999.
- [67] T. S. Rappaport, *Wireless Communications: Principles and Practice (2nd Ed.)*, ed., Prentice Hall PTR, 2002.
- [68] H. Reed, *An Introduction to Ultra Wideband Communication Systems*, ed., Prentice Hall, March 2005.

- [69] D. S. Reeves and Hussein F. Salama, "A Distributed Algorithm for Delay-Constrained Unicast Routing," *IEEE/ACM Transactions on Networking*, vol. 8, no. 2, pp.239-250, April 2000.
- [70] I. Rubin and C. Choi, "Impact of the Location Area Structure on the Performance of Signaling Channels in Wireless Cellular Networks," *IEEE Communications Magazine*, vol. 35, no. 2, pp.108-115, February 1997.
- [71] N. Sadagopan, F. Bai, B. Krishnamachari, and A. Helmy, "PATHS: Analysis of PATH Duration Statistics and Their Impact on Reactive MANET Routing Protocols," *ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp.245-256, June 2003.
- [72] P. Samar and S. B. Wicker, "On the Behavior of Communication Links of a Node in a Multi-hop Mobile Environment," *Fifth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp.145-156, May 2004.
- [73] C. Savarese, J. M. Rabaey, and J. Beutel, "Locationing in Distributed Ad-Hoc Wireless Sensor Networks," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp.2037-2040, May 2001.
- [74] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic Fine-Grained Localization in Ad Hoc Networks of Sensors," *International Conference on Mobile Computing and Networking*, pp.166-179, July 2001.
- [75] S.-T. Sheu and J. Chen, "A Novel Delay-Oriented Shortest Path Routing Protocol for Mobile Ad Hoc Networks," *IEEE International Conference on Communications*, vol. 6, pp.1930-1934, June 2001.
- [76] D. Simon, *Optimal State Estimation*, ed., John Wiley & Sons, 2004.
- [77] R. Sivakumar, P. Sinha, and V. Bharghavan, "CEDAR: a Core-Extraction Distributed Ad hoc Routing algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp.1-12, August 1999.
- [78] J. Stine and G. de Veciana, "A Paradigm for Quality-of-Service in Wireless Ad Hoc Networks Using Synchronous Signaling and Node States," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 7, pp.1301-1321, September 2004.

- [79] W. Su, S.-J. Lee, and M. Gerla, "Mobility prediction and routing in ad hoc wireless networks," *International Journal of Network Management*, vol. 11, no. 1, pp.3-30, January 2001.
- [80] J. Sucec and I. Marsic, "Capacity Compatible Two-Level Link State Routing for Mobile Ad Hoc Networks," *Military Communications Conference*, vol. 1, pp.507-512, October 2002.
- [81] D. Tang and M. Baker, "Analysis of a Metropolitan-area Wireless Network," *ACM/IEEE International Conference on Mobile Computing and Networking*, pp.13-23, August 1999.
- [82] C.-K. Toh, "Associativity-based Routing for Ad Hoc Mobile Networks," *Wireless Personal Communications Journal: Special Issue on Mobile Networking and Computing Systems*, vol. 4, no. 2, pp.103-139, March 1997.
- [83] D. Turgut, S. K. Das, and M. Chatterjee, "Longevity of Routes in Mobile Ad hoc Networks," *IEEE Vehicular Technology Conference*, spring, vol. 4, pp.2833-2837, May 2001.
- [84] Y.-C. Tseng, Y.-F. Li, and Y.-C. Chang, "On Route Lifetime in Multihop Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp.366-376, October 2003.
- [85] A. Tsirigos and Z. J. Haas, "Multi-path Routing in the Presence of Frequent Topological Changes," *IEEE Communications*, vol. 39, no. 11, pp.3-30, November 2001.
- [86] A. Tsirigos and Z. J. Haas, "Analysis of Multipath Routing – Part I: The Effect on the Path Delivery Ratio," *IEEE Transactions on Wireless Communications*, vol. 3, no. 1, pp.138-146, January 2004.
- [87] Q. Wang and E. Y. Hua, "Centralized Transmission Power Scheduling in Wireless Sensor Networks," *ACM International Conference on Wireless Communications and Mobile Computing*, pp.290-295, August 2007.

- [88] H. Xie and D. J. Goodman, "Mobility Models and Biased Sampling Problem," *IEEE International Conference on Universal Personal Communications*, vol. 2, pp.803-807, October 1993.

- [89] J. Yoon, M. Liu, and B. Noble, "Random Waypoint Considered Harmful," *Joint Conference of the IEEE Computer and Communications*, vol. 2, pp.1312-1321, March 2003.

- [90] D. Yu, H. Li, and I. Gruber, "Path Availability in Ad Hoc Network," *International Conference on Telecommunications*, vol. 1, pp.383-387, February 2003.

- [91] Z. R. Zaidi and B. L. Mark, "A Two-tier Representation of Node Mobility in Ad Hoc Networks," *IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, pp.153-161, October 2004.