1-2014

The Impact of Adaptation Delays on Routing Protocols forMobile Ad-Hoc Networks (MANETs)

Yamin Al-Mousa

ROCHESTER INSTITUTE OF TECHNOLOGY

The Impact of Adaptation Delays on Routing Protocols for Mobile Ad-Hoc Networks (MANETs)

by

Yamin Al-Mousa

A Dissertation for the
Degree of Doctor of Philosophy

B. Thomas Golisano College of Computing and Information Sciences
Ph.D Program

August 2014

Director Approval

The Impact of Adaptation Delays on Routing Protocols for Mobile Ad-Hoc Networks (MANETs)

by

Yamin Al-Mousa

A dissertation submitted to Rochester Institute of Technology in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computing and Information Sciences

B. Thomas Golisano College of Computing and Information Sciences

Approved by Pengcheng Shi, Ph.D. Program Director

Signed: ______________________ Date: ____________

PhD Committee

Andres Kwasinski, Ph.D (Department of Computer Enginnering)
Signed: ______________________ Date: ______________

Sumita Mishra, Ph.D (Department of Computing Security)
Signed: ______________________ Date: ______________

John Hamilton, Ph.D (School of Mathematical Science)
Signed: ______________________ Date: ______________

David Ross, Ph.D (School of Mathematical Science)
Signed: ______________________ Date: ______________

Carl Lutzer, Ph.D (School of Mathematical Science)
Signed: ______________________ Date: ______________

ROCHESTER INSTITUTE OF TECHNOLOGY

Abstract

B. Thomas Golisano College of Computing and Information Sciences
Ph.D Program

Doctor of Philosophy

by Yamin Al-Mousa

MANETs are coping with major challenges such as the lack of infrastructure and mobility which causes networks topology to change dynamically. Due to limited resources, nodes have to collaborate and rely packets on the behalf of neighbors to reach their destinations forming multi-hop paths. The selection and maintenance of multi-hop paths is a challenging task as their stability and availability depend on the mobility of participating nodes, where paths used a few moments earlier would be rendered invalid due to ever changing topology. The purpose of a routing protocol is to establish and select valid paths between communicating nodes and repair or remove invalid ones. As mobility rate increases, routing protocols spend more time in path maintenance and less time in actual data communication, degrading network performance. This interaction among mobility, topology and routing performance is usually empirically studied through simulations. This dissertation will provide a novel deep analytical study of the root cause of performance degradation with mobility. This is accomplished by, firstly, studying how mobility impacts durations of topology paths called *Topological* modeling. Secondly, analyzing how routing protocols adapt to topology changes in *Adaptability* modeling which identifies *AdaptationDelays* representing the time taken by a routing protocol to translate a change in topology to logical information used in path selection. Combining the results from these two studies, performance models of routing protocols are obtained, which later is used to optimize its operation. This study is applied on two tree-based proactive routing protocols, the Optimized Link State Routing and the Multi-Meshed Tree.

Acknowledgements

I would like to acknowledge my dissertation advisor professor Andres Kwasinski for his professional support, guidance and his willingness to accommodate and help me in solving many of this research issues. I recognize his effort, time and significant contributions in making all of this possible. In addition, I would like to thank my committee members professors Sumita Mishra, David Ross and Carl Lutzer for their valuable inputs, insights and being role models and great mentors. I would like to extend special and warm acknowledgements to professor John Hamilton for his pivotal help in starting the analytical work and later in sharpening my skills in problem definition and tackling. In addition, I would like to thank professor Pengcheng Shi for his accommodations and unprecedented support. Finally, I would like to thank professor Nirmala Shenoy for her guidance during my beginning of this research and giving me the chance to start and finish.

Dedications

During the past years, I have received support on the personal level from countless individuals. Firstly, I dedicate this work to my wife, Amal, for her many sleepless nights and continuous motivation in meeting deadlines. I recognize your sacrifices, dedication and love in providing the ideal environment. At many crucial times, you were the influence and inspiration in taking the hard and right decisions, thank you for being the candle in my life. In the last three years, you were truly the perfect partner who I can depend on as we navigate through this life.

This journey started seven years ago for which my family members, parents and siblings, were definitely the igniting factor and the never ending source of motivation in this remarkable experience. More specifically, this work is dedicated to my mother, Hanan, and father, Samir. Your emotional support during set backs resulted in personal growth and were essential in writing this success story. I will never forget your contributions for which I am grateful for eternity.

Finally, a message to my son, Karam, I am glad that now we can spend more time together. I wish you a successful journey writing another successful chapter in the legacy of our extended family

. . .

Contents

List of Figures

List of Tables

Abbreviations and Definitions

$\lambda_{R \to A}$ The delay of node R signaling A as an *MPR*

$\Phi(i, j, t)$ The Topology Adjacency Matrix entry between the two nodes i and j at time t

$\Psi(i, j, t)$ The Logical Adjacency Matrix entry between the two nodes i and j at time t

ψ_1 *LLink* duration between two nodes as represented by logical information at the routing protocol

ψ_k *LPath* duration of k hops between two nodes as represented by logical information at the routing protocol

φ_1 *TLink* duration between two nodes i and j which is the time duration $t_2 - t_1, t_2 > t_1$ such that $\Phi(i, j, t_1) = \Phi(i, j, t_2) = 1$ and $\Phi(i, j, t_1 - \epsilon) = \Phi(i, j, t_2 + \epsilon) = 0$

φ_1^m refers to the duration, φ_1, of the m^{th} *TLink* of a *TPath*

$\varphi_{k_{avg}}$ The average of φ_k

φ_k *TPath* duration of k hops/*TLink*s and $k + 1$ nodes $\{1, 2, , k, k + 1\}$ which is the time duration $t_2 - t_1, t_2 > t_1$ such that $\prod_{i=1}^{k} \Phi(i, i + 1, t_1) = \prod_{i=1}^{k} \Phi(i, i + 1, t_2) = 1$ and $\prod_{i=1}^{k} \Phi(i, i + 1, t_1 - \epsilon) = \prod_{i=1}^{k} \Phi(i, i + 1, t_2 + \epsilon) = 0$

ϖ_k The time duration when packets from node A are successfully received at the root node R. It is when the corresponding entries in $\Phi(i, j, t)$ and $\Psi(i, j, t)$ agree to be TRUE; it is given by $\varphi_k \cap \psi_k$

$\xi_{k_{avg}}^{in}$ The average of ξ_k^{in}

$\xi^{out}_{k_{avg}}$ The average of ξ^{out}_{k}

ξ^{in}_{k} Delay in realizing in-contact over k hops path; it is given by $\psi^{in}_{T} - \varphi^{in}_{T}$

ξ^{out}_{k} Delay in realizing out-of-contact over k hops path; it is given by $\psi^{out}_{T} - \varphi^{out}_{T}$

Adaptability The ability of a routing protocol to adapt to topology changes with mobility in a timely manner

AdaptationDelays The time duration between the time when a change in topology happens and when the corresponding change is translated to logical information

D_{ij} The Euclidean distance between nodes i and j

D_{TX} The transmission range of a node

Disconnect A packet sent to declare the loss of a *VID*

hello A packet set every hello interval to announce node's *VIDList* in MMT, or other topology information in other protocols

k Number of hops

LID Leaf IDentification in a VID

LLink Logical Link(s)

LPath Logical Path(s)

maxChild Defines the maximum number of children/branches allowed to originate from a node

maxClient Defines the maximum number of tree clients

maxHop Defines the maximum number of hops allowed in a VID

maxVID Defines the maximum number of VIDs a node can have in its *VIDList*

*MPR*s Multi-Point Relays

P^{LnP} Logically not possible packet, which means that node A and the root node R cant talk as shown by logical adjacency matrix, $\Psi(i, j, t)$, at the packet generation time

P^{nR} A lost packet, not received

P^{TnP} Topologically not possible packet, which means that node A and the root node R were in two different graph components when the packet was generated

P_k^{LP} Logically possible packet, means that node A and the root node R can talk over k hops *LPath* based on logical adjacency matrix, $\Psi(i, j, t)$, at the packet generation time

P_k^{R} A received packet at the destination with k hops

P_k^{TP} Topologically possible packet with k hops, which means that node A and the root node R were members of the same graph component and the shortest *TPath* between them has k hops when the packet was generated

RegistrationReply A packet sent in response to *RegistrationRequest*

RegistrationRequest A packet sent to register a new *VID*

RID Root IDentification in a VID

Selector Refers to the node which selects an *MPR* in OLSR protocol

Sp_{avg} The average speed a node travels based on Sp_{min} and Sp_{max}

Sp_{min}, Sp_{max} The minimum and maximum speed a node can travel

T_T^{in} Logical in-contact time, is the time when the routing protocol at node A calculates a *LLink* or *LPath* to root node R

T_T^{in} Topological in-contact, is the time when node A becomes a member of the graph component which has the root node R

T_T^{out} Logical out-of-contact time, is the time when all *LLink*/*LPath* at node A to the root node R are removed

T_T^{out} Topological out-of-contact, is the time when node A just leaves the graph component which included the root node R.

TC Topology Control packet

Ti Rate of exchanging *LLink*s/*LPath*s information in routing protocols, such as the rate of sending *hello* and TC packets

*TLink*s Topology Links

*TPath*s Topology Paths

VID Virtual IDentification

ABR Associativity Based Routing

AODV Ad-hoc On-demand Distance Vector

CBR Constant Bit Rate packet generation model

CC Cluster Client

CH Cluster Head

Connectivity Probability It is is the probability that a node is in the same graph component of that of a root.

DARPA Defence Advanced Research Projects Agency

DBF Distributed Bellman-Form routing algorithm

DSDV Destination Sequenced Distance Vector

DSR Dynamic Source Routing

FSR Fisheye State Routing

GPS Global Positioning System

IARP IntrA-zone Routing Protocol

ID IDentification

IMAC Ideal Medium Access Control protocol

IPD Inverse Path Duration

MANETs Mobile Ad-hoc Networks

MMT Multi-Meshed Tree

MMTI A protocol stack of MMT routing protocol with IMAC

MMTW A protocol stack of MMT routing protocl with IEEE 802.11 MAC

MMTWm A protocol stack of MMT routing protocl with modified IEEE 802.11 MAC

N/A Not Applicable

OLSR Optimized Link State Routing

OLSRI A protocol stack of OLSR routing protocol with IMAC

OLSRW A protocol stack of OLSR routing protocl with IEEE 802.11 MAC

OLSRWm A protocol stack of OLSR routing protocl with modified IEEE 802.11 MAC

PDA Personal Data Assistant

QoS Quality of Service

RABR Route-lifetime Assessment Based Routing protocol

RREP Route REPly

RREQ Route REQuest

SSA Signal Stability-based Adaptive routing protocol

STAR Source Tree Adaptive Routing

TBRPF Topology Broadcast based on Reverse Path Forwarding

TCP Transport Control Protocol

TOPO A null protocol stack to study the statistics of topology change

TORA Temporally-Ordered Routing Algorithm

Transition Index It is an indication of topology change by monitoring nodes' transitions between joining and leaving the graph component containing a root. The index is calculated by normalizing the count of T_T^{in} and T_T^{out} logged during simulation over readings from same scenario.

WLAN Wireless Local Area Network

WRP Wireless Routing Protocol

ZRP Zone Routing Protocol

cdf cumulative density function, for a random variable x, it is denoted as $F(x)$

pdf probability density function, for a random variable x, it is denoted by $f(x)$

Chapter 1

Introduction

1.1 MANETs: definition and challenges

The availability of small and inexpensive wireless communicating devices with significant computing capability has played an important role in moving Mobile Ad-hoc Networks (MANETs) closer to reality placing them at researchers focal point [1–3]. Nodes in MANETs are expected to establish and maintain a network in an autonomous manner using wireless communication; for which they act as data sources, destinations and routers simultaneously. Unlike the wired counterparts, nodes in MANETs are deployed without infrastructure allowing them to move freely without being tethered by wires. The lack of infrastructure does not prevent the possibility of connecting to the Internet, when needed, through means of gateways. Nowadays, applications of MANETs are vast such as festival grounds, outdoor activities, sensing, emergency search and rescue operations, battlefields, defense and surveillance, or in any other scenario where networks should be deployed immediately or on temporary basis.

In addition to mobility, the lack of infrastructure and a central organizing entity, wireless links in MANETs are subject to fading and interference resulting in links' instability. Frequency allocation, security concerns, and random power outage add to MANETs' managing challenges. Other challenges are imposed by the nature of MANETs application and the guarantee of a required Quality of Service (QoS) in addition to large scale deployments which demand scalable networking solutions. Nodes in MANETs rely on limited power sources, such as batteries; which limits transmission range. For two remote nodes wishing to communicate, they should collaborate with others to relay packets resulting in a multi-hop path.

The selection of a multi-hop path is a fundamental problem in MANETs since its stability is dependent on the actions of the participating nodes, specially with mobility. Mobility is the biggest challenge in MANETs since it causes links and paths to be set up and torn down frequently making networks topology highly dynamic and difficult to manage. In general, higher mobility changes topology more frequently which degrades MANETs' performance [4–6]. Selecting stable multi-hop path is critical for achieving better performance with mobility because the less time spent in maintaining paths, the more time is available to communicate useful information. This interaction among mobility, topology and performance is essential in modeling MANETs and is seldom studied analytically. This work provides a fresher look at the impact of mobility on network's topology and performance through firstly studying and modeling the statistics of single links and multi-hop path durations. Secondly, we model how MANETs routing protocols are adapting to changes in topology. Finally, we combine these two models to derive performance models.

1.2 Motivation

Clearly, performance in MANETs is application-dependent which is measured by the ability to meet application's demands despite the limited resources. For example, file transfer is sensitive to packet loss while packet latency is tolerable. On the other hand, in streaming applications (voice or video) limited packet loss is acceptable while packet delays and jitter are problematic. We argue that the three performance metrics of packet delivery ratio, packet latency and jitter do overlap to a certain extent and all are affected by the ability of a MANET's routing protocol to adapt to topology changes due to mobility. In this context, we refer to topology as the ground truth of available links and paths. When two nodes R and A are in transmission range of each other, then a topology link *TLink* exists between them. Meanwhile, a topology path *TPath* between nodes R and A exists if there is a set of chained, two or more, *TLinks* connecting them. A routing protocol maintains a routing table containing the required information on how to reach other nodes in the network. Such information pieces are logical representations of the network's topology as perceived by the node. Hence, a *TLink* between node R

and A is perceived and stored as a logical link *LLink* at the routing layer of R with A and vise-versa. Usually, the routing algorithm is run on the collection of gathered *LLinks* to calculate logical paths *LPaths* between the node and other nodes in the network. As a result, a MANET's routing protocol adapts to topology changes by:

- Discovering new topology links and paths *TLink*s and *TPath*s
- Removing broken *TLink*s and *TPath*s

Indeed, some time is needed for the routing layer to realize a change in topology and modify the corresponding logical information. In dynamic topology, discovering new *TPath*s and *TLink*s quickly allows high packet delivery ratio and lowers packet latency. In addition, removing broken *TPath*s and *TLink*s quickly limits failed packet transmissions/retransmissions, which depletes precious resources (energy and bandwidth), lowers buffering delays and motivates routing protocols to find alternative *TPath*s, improving packet delivery ratio. While these two actions may look trivial as they take short time durations; however, they are non-negligible from the application point of view as they cause traffic disruption due to packet retries and eventually being dropped causing overall performance degradation. As a result, we conclude that MANET's performance is affected by the ability of a routing protocol to adapt to topology changes with mobility in a timely manner, we call it the routing protocol's *Adaptability*, which is measured by a set of *AdaptationDelays* representing the time needed for a MANET's routing protocol to propagate a change in ground truth topology, *TLinks* and *TPaths*, to logical change in logical information in its routing table as *LLinks* and *LPaths*. A routing protocol with higher *Adaptability* has lower *AdaptationDelays*; Hence it discovers new *TPath*s and *TLink*s then removes broken faster than others. Understanding the *Adaptability* of MANET's routing protocol is pivotal to model its performance under mobility; which demands studying the following:

- The behavior and durations of *TLink*s and *TPath*s between two nodes (*Topological* Modeling)
- The reaction of MANET's routing protocol to topology changes (*Adaptability* Modeling)

Understanding and categorizing the behavior and durations of *TLink*s/*TPath*s between two nodes is the essence of learning their impact on MANETs performance. Intuitively, *TPath* duration between two nodes depends on the participating *TLink*s; hence a comprehensive understanding of individual *TLink* behavior is the key to understand the bigger picture. Predicting *TPath* and *TLink* durations will be an easy task if all nodes in MANET have means of estimating locations and velocities, such as Global Positioning System (GPS), which is not a viable solution from hardware and application perspectives. As a result, probabilistic duration models are required.

On the other hand, a detailed study of routing protocol implementation is required to understand its *Adaptability*, measure its *AdaptationDelays* and model the impact on performance.

These models (*Topological* and *Adaptability* models) can be used to analyze the performance bounds of protocols and used to design new algorithms and protocols enabling efficient performance as the work in [7–10]. Models for *TPath* durations can be used in path selection to meet certain QoS requirements, in calculating cache timers in reactive protocols, in constructing alternative routes preemptive to failure of current ones, in selecting routes with longer durations to minimize path failures and recovery which adds unneeded overhead, in choosing proper route advertising intervals since advertising too often leads to wastage of resources and performance degradation; while infrequent advertising results leads to an incorrect picture of the network causing packet loss and routing loops.

To sum up, mobility causes topology to change in unpredictable manner by forming new *TLinks* and *TPaths* while rendering others invalid. A routing protocol stores discovered *TLinks* as *LLinks* in routing table which is used to calculate *LPaths* between nodes. As a change in network's topology occurs, it is translated by adding new *LLinks* and *LPaths* or removing old ones. Clearly, the translation of *TLinks* and *TPaths* to *LLinks* and *LPaths* is not immediate and takes time. We call the time duration between the time when a change in *TLinks* and *TPaths* happens and when the corresponding change is translated to *LLinks* and *LPaths* as *AdaptationDelays*. We use *AdaptationDelays* as a measure of MANET's routing protocol *Adaptability* which is the ability to adapt quickly to topology changes. The shorter the *AdaptationDelay*, the better *Adaptability*, the better the performance with mobility. Understanding the interaction between mobility and performance and producing performance models demands the development of two major models, *Topological* model for describing the behavior of *TLink*s and *TPath*s with mobility and *Adaptability* model to represent the routing protocols delay in adapting to topology changes. This work has the following contributions:

- Provides an analytical *Topological* model without prior assumptions such as known speeds or nodes' location.
- Presents an innovative *Adaptibility* modeling to show how the details of designing and implementing MANET's routing protocols impacts its performance.
- *AdaptationDelays* provides, to our knowledge, a unique in-depth insight at the true cause of performance degradation with mobility in MANET's protocols.
- Combines *Topological* and *Adaptability* models to provide analytical performance models. Such models are rare in literature and many are produced empirically.

1.3 Problem Statement and Objectives

The statement of this PhD dissertation is: "To model the interactions between topology changes under mobility and *Adaptability* of a routing protocol, then to model and optimize the performance of a routing protocol." As a result we define the following objectives:

- *Topological* modeling: to model the dynamics of MANETs topology with mobility. It provides models for *TLink*s and *TPath*s time durations.
- *Adaptability* modeling: to model the behavior of routing layers when topology changes occur, specifically by modeling their *AdaptationDelays*.
- Performance modeling: to produce a performance model in MANETs based on the interactions of *Topological* and *Adaptabiltiy* models. It will provide a clear insight why protocols, in general, have lower performance with mobility and why some perform better than others.
- Performance Enhancement: to use available models to optimize the operation of a routing protocol.

Chapter 2

Literature Review

The main purpose of this work is to study the impact of routing protocol's *Adaptability* on MANET's performance under mobility. We identify three main research areas related to this purpose: MANET's routing protocols, *Adaptability* and *Topological* modeling. To the best of our knowledge, literature lacks the foundations of *Adaptabiltiy* modeling leaving two research areas that will be surveyed in this Chapter. Section 2.1 presents a survey of popular MANET's protocols, while section 2.2 presents related work in *Topological* modeling and how it can be used to improve MANETs performance as found in literature.

2.1 Routing Protocols in MANETs

The purpose of routing protocols is to translate topology information *TLink*s and *TPath*s into logical information *LLink*s and *LPath*s. MANETs routing protocols can be divided, based on *LLink* and *LPath* selection criteria, into two classes: minimum-weight and stability-based [11]. Most protocols in the minimum-weight class base their selection on hop count, a measure of path delay, congestion and energy consumption. Minimum-weight protocols can further be categorized based on the way *LLink*s and *LPath*s are gathered and maintained as Proactive (table-driven), Reactive (on-demand) and Hybrid [12]. On the other hand, protocols in the stability-based class minimizes the impact and overhead of *LLink*s and *LPath*s maintenance and rediscovery by choosing those that are more likely to exist longer.

2.1.1 Routing Protocols in Minimum-Weight Class

2.1.1.1 Proactive (table-driven) routing protocols

In proactive protocols, each node gathers and maintains *LLink*s and *LPath*s to all known destinations, even when they are not used. Gathering and maintaining *LLink*s and *LPath*s is achieved through a combination of the three operations [13]:

- Periodically monitoring *LLink*s status,
- Triggering *LPath*s updates when changes in *LLink*s state is detected and
- Periodically announcing and updating available *LLink*s and *LPath*s

Proactive protocols can be further divided into two sub-categories:

- Updating periodically and
- Updating when a change is detected

The Defence Advanced Research Projects Agency (DARPA) packet radio network project [14], the IntrA-zone Routing Protocol (IARP) [15], the Optimized Link State Routing (OLSR) [16] and the Fisheye State Routing protocol (FSR) [17] are examples on the first sub-category. On the other hand, the Destination Sequenced Distance Vector (DSDV) [18], the Wireless Routing Protocol (WRP) [19], the Source Tree Adaptive Routing (STAR) [20] and the Topology Broadcast based on Reverse Path Forwarding routing protocol (TBRPF) [21] are examples on the second sub-category.

Sending *LLink*s and *LPath*s updates based on detected changes has the potential of producing larger overhead. One reason is that, in wireless networks, radio links between nodes may experience frequent disconnects and reconnects. In addition, a change in *LLink* or *LPath* may happen in quick succession due to mobility causing each change to be sent in its own update message. Instead, waiting some time and grouping all changes in a single update reduces overhead. In general, the main advantage of proactive protocols is the low lead latency since *LLink*s and *LPath*s to all possible destinations in the network are readily available at the time of making routing decisions; however, high overhead remains the main disadvantage especially in large dynamic networks.

OLSR [16] optimizes overhead over conventional *LLink* state proactive protocols. Each node selects a set of neighbors called Multi-Point Relays (*MPR*s). Only *LLink*s between an *MPR* and its selectors are reported in Topology Control (*TC*) packets, which are forwarded and diffuse throughout the network by *MPR*s only. *LPath*s between remote nodes (2 hops or more) are a sequence of *MPR*s. Hello packets are used for neighbor sensing and *MPR* selection. Later in section 2.1.3, we provide a deeper look at the design and operation of OLSR.

Guangyu et al. in [17] presented FSR, in which the burden of exchanging periodic *LLink*s state information is reduced using the concept of scopes. The scope is usually defined by the number of hops, in which a node exchanges *LLink*s state information with others within the scope more frequently than those outside. *LLink*s updates are solely time triggered and not event triggered. Broken *LLink*s is not reported in following updates. FSR is known for producing a less accurate *LPath*s to remote destination but accurate enough to allow packets to travel toward the destination. As the packet approaches the destination, the *LPath* becomes more accurate.

Perkins et al. in [18] proposed DSDV, which uses an improved Distributed version of Bellman-Ford (DBF) routing algorithm. The protocol is distance vector based, where each node maintains a local sequence number and *LPath* entry for every destination containing a next hop, hop count and a tagging sequence number assigned by destination to represent freshness. Periodically or as when significant change is detected, the local sequence number is incremented and sent along with the *LPath* entry for each destination containing hops count and the tagging sequence number. The *LPath* with higher sequence number and lower hops is chosen.

WRP was presented in [19], which also uses the DBF routing algorithm. However, it communicates the distance and second-to-last hop for each destination which reduces the cases in which a temporary routing loop can occur. If a change is detected, only information that reflects the change is sent.

Authors in [20] presented STAR which attempts to provide feasible *LPath*s that are not necessarily optimal through the use of least overhead routing approach. *LLink* is not updated periodically, rather it is updated conditionally. Updates are sent only when all *LPath*s to a destination or more are lost, when new destinations are detected or when *LLink* change in a way that might create loops. Deletion of *LLink*s is implicit when being replaced by others or explicit when the deletion causes the loss of all *LPath*s to a destination or more.

TBRPF [21] is a link state based routing protocols in which *LLink* state information is delivered to all nodes in the network. Each node broadcasts *LLink*s updates on its outgoing links that are part of a minimum hop broadcast tree rooted at the source. The tree is a collection of minimum hop *LPath*s from all nodes to the source. Its operation is based on the chicken-egg paradox: it computes the *LPath* that form the broadcast trees using information that is received along the trees themselves.

2.1.1.2 Reactive (on-demand) routing protocols

In reactive protocols, nodes construct and maintain *LLink*s and *LPath*s to a destination only when they are actually needed. The protocol operation usually consists of two phases: discovery and maintenance. In more details, when data is ready to be routed to a a destination, the discovery process is invoked by flooding the network with Route REQuest (RREQ) packets seeking the destination. When the destination is found, a Route REPly (RREP) packet containing information to construct *LLink*s and *LPath*s is sent back to the source. The *LLink* or *LPath* is maintained as needed and rediscovered when it fails. The main advantage of reactive protocols is the lower overhead in general; which is expected to increase as the network's topology becomes more dynamic due to frequent *LLink*s or *LPath*s errors and rediscoveries. One the contrary, the high lead latency to new destination is the main disadvantage.

Johnson et al. in [22] proposed Dynamic Source Routing (DSR), which is based on the concept of source routing. It is a reactive protocol which uses request/reply procedure during discovery process. As a node forwards RREQ, it appends its ID to the packets header. A destination replies to all RREQ it receives by reversing the order of IDs it reads from in the header to construct the *LPath*. Upon fowarding the RREPs, nodes cache *LPath* it reads from the header which can be used in subsequent RREQ to minimize overhead.

Temporally-Ordered Routing Algorithm (TORA) [23] is a reactive, distributed, highly adaptive, and loop free protocol. It is based on a link reversal algorithm and designed to provide multiple routes to a destination and minimize overhead by localizing the algorithmic reaction to topology changes. *LPath* is established by creating a directed acyclic graph rooted at the destination using a similar approach as request/reply (which are both flooded) processes. *LPath* optimality is considered to be of secondary importance.

Ad-hoc On-demand Distance Vector (AODV) [24] is based on distance vector, as its name indicates, which stores an entry in the routing table indicating the next node and how many hops are expected to reach a destination. It uses the conventional request/reply procedure to build a single *LPath* to requested destination. The destination replies to the first RREQ packet it receives and drops subsequent ones with the same source sequence number and broadcast ID. Unlike DSR, AODV doesn't append the ID of forwarding node of RREQ packets and it only stores a distance vectors pointing to the destination instead of ordered node IDs.

2.1.1.3 Hybrid routing protocols

Hybrid routing protocols combine the advantages of proactive and reactive protocols, where the network is divided into zones and every node performs different routing strategies depending on destination's location. In most of hybrid protocols, a node adopts a proactive routing strategy for destinations within its zone while reactive strategy is used for destinations outside the zone.

The zone routing protocol (ZRP) [25] is a pioneering concept in hybrid protocols which can be seen as a framework rather than a protocol. Each node defines its own zone by means of number of hops, where proactive schemes are used within local zone and reactive schemes are used otherwise to reach farther destinations.

The Multi-Meshed Tree protocol (MMT) [26–29] is another hybrid routing protocol based on clustering to address scalability. A cluster contains one cluster head (CH) node and several cluster clients (CCs) nodes. Proactive *LLink*s and *LPath*s are formed within a cluster, while *LPath*s across clusters are maintained reactively. Multiple redundant proactive *LPath*s are formed between a CC and its CH so that if one *LPath* is lost another is ready to use, thus accounting for dynamic topology. These *LPath*s are formed using the MMT algorithm which simplifies proactive *LPath* formation and maintenance thanks to its unique naming scheme called Virtual IDs (*VID*s). Later in section 3.1, we provide a deeper look at the design and operation of MMT cluster creation and operation. Reactive *LPath* is maintained as a sequence of clusters, hence, reactive discovery and maintenance are done at the cluster level. This adds resiliency against mobility since the *LPath* is not dependent on specific nodes, rather, the whole cluster. Clustering also avoids flooding control messages by keeping them within the cluster's boarders. Since a reactive *LPath* is a sequence of clusters and *LPath*s within a cluster are proactive ones; a reactive *LPath* is a concatenation of proactive *LPath*s which are continually updated with node mobility; hence, the probability of having a stale reactive *LPath* is lowered.

2.1.2 Routing Protocols in Stability-Based Class

Guenhwi et al. in [30] gave an insight of the impact of edge effect in scenarios with high node density. In protocols adopting minimum-hops *LPath* selection criterion, a node forwarding to another tends to select those at the edge of its transmission range in order to minimize number of hops. In mobile scenarios, these forwarding node leave the transmission range quickly which results in highly unstable *LPath*s. Hence, stability metrics should be used to allow *LPath* stability-based selection criterion to choose those nodes which have the potential to remain in range longer; saving extra overhead due to less maintenance.

Many stability based protocols adopt the reactive discovery process to construct *LPath*s. The work in [30] proposes the use of signal strength and differential signal strength (to determine closing or moving away neighbors) for reactive protocols. Simulation results show performance enhancement when using paths with longer lifetime and increased number of hops. Similar observations were also reported in [31].

Toh et al. proposed the Associativity Based Routing protocol (ABR) [32] where each node exchanges a pilot signal with neighbors and records the number of consecutive times a pilot signal is received called ticks. A link is considered stable if it has the number of ticks higher than a certain threshold. A requesting node selects the route having the highest degree of associativity along its containing nodes.

In Signal Stability-based Adaptive routing protocol (SSA) [33] a node exchanges beacon packets with neighbors and is able to measure received signal strength. A *LLink* with a neighbor is considered strong if packets are received with strong signal larger than a predefined threshold for several consecutive times, called clicks, and more than a certain clicks threshold. Route requests are forwarded only if were received over a strong link. If no replies were received, operation similar to conventional reactive protocols is assumed.

The Route-lifetime Assessment Based Routing protocol (RABR) [34] is also a reactive based routing protocol. It uses the concept of signal strength changes to estimate link life time called affinity. During the path discovery phase, the values of affinities along the path are added to the discovery packet; while the route with highest affinity is chosen.

Indeed, estimating and predicting topology changes can enhance the performance of many protocols as shown in [35], which uses GPS location information and motion prediction to enhance network's performance. Predicting a topology change helps in reducing overhead and limit traffic disruption by reconstructing paths proactively; However, this approach might be impractical since it requires the extra GPS hardware. Researchers depend on mobility models, specially the random way point mobility model, to simulate the performance of MANETs which represent realistic scenarios as shown in the survey [36]; despite its shortcomings of reaching steady state of average nodal speed [37] and inability to maintain a uniform node density throughout the network [38]. The same claim was reported in [39], which showed that real life data gathered for routes and link durations from 20 Personal Data Assistants (PDAs) connected with 802.11b have similar statistical properties as those exhibited by random way point mobility model and random reference point group mobility model whether the cause of link breakage is mobility or collisions and interference.

2.1.3 Optimized Link State Routing protocol OLSR, a deeper description

In addition to being a proactive routing protocol, OLSR [16] is an optimized version of the classical link state algorithm where a *LLink* change causes a flood of messages, *LLink* state messages, to inform all nodes in the network about the change. However, OLSR modifies that flooding process by adopting the following:

- The concept of Multi-Point Relays *MPRs* nodes which are selected by a subset of their neighbors (*MPRSelectors*). *MPR*s are responsible for forwarding *LLink* state messages, known as Topology Control *TC* packets, during the flooding process. This concept substantially reduces message overhead as compared to a classical flooding mechanism where every node retransmits the first copy of the flooding message.

- OLSR allows only elected *MPRs* to generate the flooding *TC* packets reducing the overall number of overhead messages generated and flooded in the network.
- An *MPR* node is required to only include the state of *LLink*s it has with its *MPRSelectors*. Additional available *LLink*s state information with other neighbors may be utilized for redundancy.

As a result and unlike classical *LLink* state routing, OLSR depends on partial *LLink*s state information to calculate *LPath*s for which it uses Dijkstra's algorithm. Dijkstra's algorithm is run by every node by considering itself as the root node, then constructing minimum-weight spanning tree to all other nodes in the network. In addition, the protocol is particularly suitable for large and dense networks as the technique of *MPRs* works well in that context.

LLink state information is gathered through the periodic exchange of *hello* packets which includes the ID of the originating node and the IDs of the neighboring nodes it has heard from. *hello* packets have three main purposes: *LLink* sensing, neighbor detection and *MPR* selection signaling. Note that a *hello* packet is broadcasted once and never forwarded. Next to each of a neighbor ID, two additional pieces of information are included as well. The first one represents the state of the *LLink* a node has with this neighbor, thus serving the *Link* sensing purpose, which can be one of the following:

- Asymmetric *LLink*: node *A* has asymmetric *LLink* with node *B* if it receives a *hello* packet from *B* which does not include *A* as one of the neighbors. This only means that *A* is able to hear from *B* and does not necessarily mean the opposite.
- Symmetric *LLink*: node *A* has a symmetric *LLink* with node *B* if it receives a *hello* packet from *B* which includes *A* as one of the neighbors. This indicates that *B* has heard from *A* in the past and *A* is able to hear from *B* as well.
- Lost *LLink*: indicates that the *LLink* have been lost with the neighbor after not hearing from him for 3 consecutive *hello* intervals.

The second neighbor information, to serve the purpose of neighbor detection and *MPR* selection signaling, is one of the following:

- Symmetric neighbor: node *A* has *B* as a symmetric neighbor if it has at least one symmetric *LLink* with *B*.
- *MPR* neighbor: node *A* has *B* as an *MPR* if it has at least one symmetric *LLink* with *B* and has selected *B* as an *MPR*. When node *B* receives such information, it knows that it became an *MPR* and *A* is one of its *MPRSelecotrs*. Note that an *MPR* node is always a symmetric neighbor to its *MPRSelector*.
- Not Neighbor: indicates that the node is either no longer of has not yet become symmetric neighbor.

A node *A* may have multiple selected *MPRs* where they cover in terms of transmission range all of *A*'s 2-hops neighbors. 2-hops neighbors are those nodes heard by *A*'s immediate neighbors and are identified by comparing the list of neighbors *A* has with the list of neighbors it receives in *hello* packets. The set of *MPR* is preferably kept small in order for the protocol to be efficient. A point worth mentioning is that the exchange of *hello* packets are sufficient to construct *LLinks* with its immediate neighbors and 2-hops *LPaths* reaching the 2-hops neighbors.

An mentioned before, a *MPR* node periodically sends a *TC* packet in which it includes the IDs of all of its *MPRSelectors*. *TC* packets also include a sequence number incremented by the originating *MPR* node to represent the freshness of the message and to avoid any loops that may occur due to information discrepancies. Unlike *hello* packet, *TC* packets are flooded throughout the network and only forwarded by *MPR* nodes. Hence, they are pivotal in providing information to build 3-hops *LPaths* or longer, as a result, all these *LPath*s contain *MPR* nodes only. In addition, since *MPR*s have symmetric *LLink*s with *MPRSelecots* and they are the only *TC* forwarding nodes, *LPath*s in OLSR are only constructed through symmetric *LLinks*. This avoids the problems associated with data packet transfer over asymmetric *LLink*s, such as the problem of not getting acknowledgments for data packets at each hop. Logical information received by *TC* packets are removed when not updated for 3 consecutive *TC* intervals.

2.2 Topology Modeling

MANETs suffer from performance degradation with mobility due its impact on network's topology. The impact of mobility is more severe when using *TPath*s with larger number of hops [5, 6]. In the last decade, researchers focused on this observation attempting to un-mangle the tight relationship between mobility, topology and performance. In [40], Deterministic and partially deterministic mobility model were adopted to model *TPath* duration distribution. It assumed that nodes are able to monitor location and velocity through GPS. Such information is passed to all nodes participating in a *LPath* during route discovery stage, which can be used later to predict *LPath* failure times and start the rediscovery in advance.

The work in [2, 4, 41] presented a statistical analysis of *TPath*s duration distribution based on simulation results. Results showed that some mobility models, such as reference point group and Manhattan grid mobility models may produce multi-modal duration distribution under low speeds; however, at moderate and high speeds and as number of hops increase, exponential distribution is a good approximation. The exponential decay is estimated based on the following observations:

- It increases with number of hops and speed
- It decreases with transmission range

The work also showed that the reciprocal of average *TPath* duration has a strong linear relationship with throughput and overhead. In more details, the reciprocal of average *TPath* duration has a negative correlation with throughput and a positive correlation with overhead. In [42], *TLink* duration was also found to follow exponential distribution.

Authors in [1, 43–45] attempted to explain the exponential distribution of *TLink* and *TPath* duration that appeared in [2, 4] using Palms theorem even when *TLink* durations are dependant and heterogeneous. The theorem requires the independence of involved variables; which was relaxed later in [1, 45] by assuming a *TLink* duration dependence that goes away asymptotically with increasing number of hops. The work also proves that the parameter of the exponential distribution of *TPath* duration is related to the means of *TLink* durations and is given by the sum of the inverses of the expected *TLink* durations. The authors claimed that the distribution of single *TLink* duration should be a non-increasing function, which contradicts with the results reported in [2, 4].

In [46], the authors collected statistical *TLink* durations from simulation. Unlike other studies, statistical durations were not restricted to curve-fit exponential distribution and the authors used a range of possible distributions, such as normal, Weibull and Lognormal. Through means of the Kolmogorov-Smirnof goodness-of-fit test (K-S test), they showed that Lognormal distribution is the best fit for the statistical distribution of *TLink* duration.

Tseng et al. presented in [47] a formal Markov model to estimate the duration of *TPath* in MANETs assuming that the nodes are moving based on discrete-time random walk model, which is used widely in personal communication services. The field is divided as cells where nodes have a multiple cells transmission range and they move from one cell to another in a single time making models accuracy dependant on cell sizes. This is one of the few models that consider *TLink* dependency (joint probability) when modeling *TPath* durations.

Modeling *TPath* remaining lifetime considering node density was presented in [48] in which routing protocol adopting minimum-weight *LPath* selection criterion tends to choose neighbors at the edge of each others transmission range. The model assumes that a *TPath* existed for some time in the past. In other words, the model focuses on the remaining *TPath* lifetime while its history is irrelevant. As a result, this poses an assumption on nodes' location that they are in range of each other forming the *TPath*. Such *TPath* remaining lifetime models are mainly used in optimizing reactive protocols where a *TPath* is used in the discovery process after nodes participating are already in range of each other and an estimation of its remaining lifetime is needed.

Authors in [49, 50] focused on modeling *TPath* duration of two hops only, involving three nodes, two of which were static while the middle node is moving according to random way point mobility model. The model considers the middle node to be placed randomly in the overlapping transmission area of the other two nodes and describes the time needed to break the two hops *TPath*. Then, a statistical model based on simulation results was used to derive *TPath* duration when all three nodes are moving and the overlapping area is changing over time, which was averaged for simplicity [50]. In [51], random walk mobility model was considered. A point worth mentioning is that these models are also focusing on the operation of reactive protocols.

Samar et al. produced extensive models in [7, 8] describing *TLink* dynamics between two nodes moving according to random way point mobility such as: *TLink* duration distribution, expected new *TLink* arrival (formation) rate and expected *TLink* breakage rate. In these models, the authors assumed that the exact speed of one of the node is known. Simulation results exhibited tight match with the analytical models except for the model of expected *TLink* lifetime which was attributed to simulation errors. Later, modifications to this models were presented by Nayebi et al. in [9, 10, 52]. The authors explained that the original authors assumed the node of interest is static which hid the discrepancies between simulation results and models in most cases. In addition, original authors assumed that the relative velocity of a particular node with respect to the node of interest is the same of any uniformly selected random node in the network which was proven invalid as the probability of encountering nodes at higher relative speeds is higher as will be shown in Figure 4.4. The work was later extended in [53, 54] where modeling *TLink* duration as a two state Markov model was proposed. The model is more suitable when the ratio of transmission range to node's speed is large, which means higher possibility of nodes changing direction of movement while still in range of one another. However, the models did not seem to exactly match the simulation results; however they are closer than those preseneted by Samar et al. in in [7, 8]. Chen et al. in [55] used similar methodology in [7, 8] to derive *TLink* duration model for nodes moving according to Manhattan grid mobility model. The *TLink* duration was estimated by considering three distinct scenarios: two nodes are moving in the same direction, opposite directions and perpendicular directions to each other.

Authors in [3] followed a distinct modeling approach where they found *TLink* and *TPath* availability probability over time using random direction mobility model, constant speed, and non-zero pause time in an infinite two dimensional field. Assuming that a *TPath* existed, availability probability depends on nodes' locations to each other as time proceeds. Hence, a model describing the evolution of nodes' spatial distribution was needed from which the probability that a *TLink* remains after some time can be estimated. In the case of *TPath* availability, the two cases of *TLinks* duration being dependant and independent were considered. Results showed that both cases are close enough making the difference insignificant. Estimates of *TLink* and *TPath* availability probability for random walk mobility model were presented in [56, 57], where the error margin in the simulation results was attributed to *TLink* dependency.

2.3 Usage of Topology Models in Optimizing MANETs Protocols

We dedicate this section to emphasize the potential of integrating topology models in the design and optimization of MANETs protocols to achieve better performance. In [58], an adaptive metric based on online statistical models of *TLink* durations and estimation of *TLink* remaining lifetime was used to identify stable *TLink*s. The authors in [59] studied how to maximize the rate of sending packets carrying information about network topology while preserving the connectivity in the network with high probability. The authors assumed a hypothetical protocol called Topology Control, TC, protocol which uses *hello* packets to exchange *LLink* information with neighbors.

The observation that average *TPath* duration is related to its *TLink*s average duration was also used to select routes with longest remaining lifetime [1, 43–45]. Average *TLink* durations were estimated from those established with neighbors. While forwarding route replies in the discovery phase of reactive protocols, each node adds the inverse of average *TLink* duration to a field called Inverse Path Duration (IPD). When the source receives all route replies, it chooses the route with lowest IPD value meaning the largest estimated expected duration. In other studies for reactive protocols, statistical models of path duration were used in [2, 4, 60] to configure the expiry timers of routing table entries resulting in significant overhead reduction.

Using models of *TLink* duration, Nayebi et al. in [9, 10] proposed adjusting some routing attributes such as the scheme for sending *hello* packets to increase probability of a neighbor hit before *TLink* breaks. Such considerations are pivotal in the operation of MANETs where every transmission should count due to limited power resources. Similarly, authors in [7, 8, 13] attempted choosing an optimal rate of sending *hello* packets in proactive protocols to reduce routing overhead while ensuring that the performance of the network does not deteriorate. Their goal was to find the largest *hello* sending interval such that the expected delay between the detection of a *TLink* change and the next broadcast of *hello* packet is small enough. They assumed that a *TLink* change is reflected immediately as *LLink* in routing layer and always appears in the following *hello* packet. This assumption is unrealistic as will be shown in Section 5 due to *AdaptationDelays*. Results show that the overhead decreased while the success rate was maintained; however, packets delays increased considerably. The increase in packet delay was due to the increased stale routing information causing more reroutes which can be solved by using Transport Control Protocol (TCP).

The *TLink* duration models in [3] were used to suggest an appropriate packet length that maximizes the probability of completing packet transmission before link breakage. In addition, the authors focused on balancing two concepts. On one hand, using *TPath* that has lower number of hops means less nodes involved; hence less *TPath* variability. On the other hand, using a longer *TPath* to the same destination means the involved nodes are closer to each other with shorter *TLink* distances which means longer time to travel outside the transmission range of each other with mobility and more *TPath* stability. The probability of *TPath* availability was used as a criterion for selection, where simulation results showed that *TPath*s with high number of hops have higher availability probability in the early stage of their lifetimes. On the other hand, as time progresses, *TPath*s with fewer hops have higher availability probability.

The work in [53, 54] is one of the few attempting to use topology models to improve the performance of MAC layers by optimizing the packet length considering the durations of *TLinks* it has to traverse. The concept is that longer packets require longer transmission time than others; Hence, frequent *TLink* breaks causes significant packet drops. On the other, shorter packets result in increased overheads, decrease channel utilization and waste more energy.

In [61], authors used models of *TLink* durations with cluster head to choose a suitable cluster maintenance intervals. Reducing neighbor detection time in OLSR was presented in [62]using either unicast based handshake or broadcast based handshake. Results show improvement in throughput and increase on overhead as well. This work is the closest, to our knowledge, to the concept of *AdaptationDelays* by attempting to minimize it.

Following are few points we observed from literature which supports the relevance of our work in the following chapters 3 through 7:

- Researchers depend on simulations to measure the performance of MANETs' protocols in which random mobility models, random way point mobility model and random direction mobility model, are used as they have similar statistical properties as in real life applications [39]. In our view, simulations should be as close as possible to real life scenarios; as a result, we adopt random direction mobility model as it also has the added benefit of maintaining a uniform spacial node distribution throughout the simulation time [38].

- Researchers realized the importance of *Topological* models which can be used to optimize performance. The bottom line is that basing route selection criterion on number of hops solely has the disadvantage of selecting unstable routes due to the edge effect [30].

- Most of *Topological* models in literature are derived from curve fitting statistical distributions of simulation observations [63]. Most analytical models are simple scenarios such as three nodes only or they are based on the assumptions of constant or known speed mobility models or predefined node locations. As a result, a comprehensive *Topological* mathematical model is still required.

- Many of *Topological* models are focusing on modeling the remaining lifetime of *TLink*s and *TPath*s assuming that a they existed for some time in the past and that time is well defined. This poses an assumption on nodes' location making these models suitable for optimizing reactive routing protocols only as was discussed earlier.

- Enhancements on proactive protocols performance are limited to adjusting the rate of updating topology information. Researchers also adopt the concept that a change in *TLink*s or *TPath*s is reflected immediately on the logical information *LLink*s or *LPath*s at the routing layer which is not accurate due to the *AdaptationDelays* as will be discussed in section 5. As a result, the impact of *AdaptationDelays* should be considered when tuning any protocol.

Chapter 3

Background Work

In this chapter, we present the background work and simulations performed to serve as an introduction to the analytical models listed in section 1.3. First we discuss the protocol stacks and scenarios we use to collect simulation results. Then we detail how we gauge the *Adaptability* or MANETs' routing protocols by measuring its *AdaptationDelays*. Finally, we present the performance results gathered from simulating two routing protocols MMT and OLSR. Analysis and relationships are derived from results to demonstrate the potential for future analytical models.

3.1 The Multi-Meshed Tree Algorithm

The MMT algorithm builds a meshed tree rooted at the root node *R*. The meshed tree can be thought of as multiple overlaid spanning trees, where combining the trees in Figure 3.1.a and Figure 3.1.b would result in the meshed tree in Figure 3.1.c. In MMT, multiple branches are allowed to mesh without resulting in loops thanks to the branch numbering scheme adopted by MMT. Due to the meshing of the tree branches, a node can reside on multiple branches simultaneously. The decisions to grow the tree and to extend its branches are done by each node locally.

The attachment of a node to the meshed tree is represented by a Virtual ID *VID*. A *VID* is a *LPath* which carries 3 pieces of information, (*RID*, *LID*, *hops*). *RID* is the ID of the root node, in this case it is *R*. The *hops* is the number of hops to travel between a node

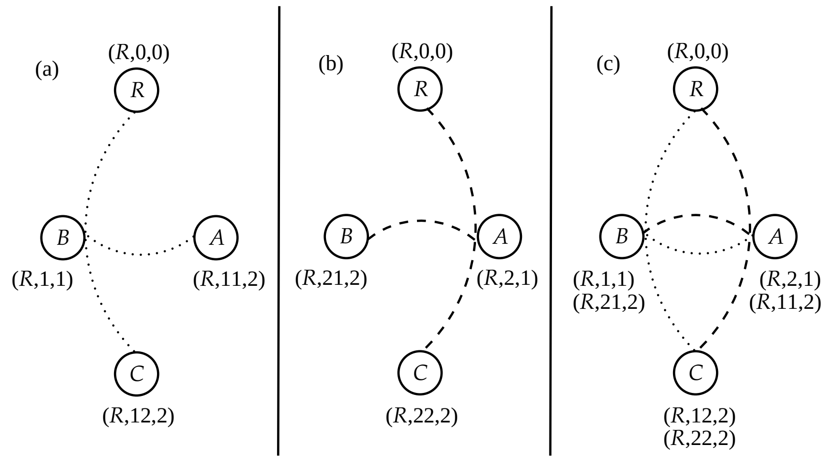

FIGURE 3.1: MMT Tree Creation

to R along that branch. Note that *hops* is used as a weight metric; however other weight metrics can be also considered. *LID* uniquely identifies the leaf or point of attachment of the node to the branch. The value of *LID* is derived from the parent's *VID* upstream (toward the root node R) in a branch. For example, node B *VID* $(R, 21, 2)$ is based on its connection via node A, the parent node in this case, which has the *VID* $(R, 2, 1)$. A node may have multiple *VIDs* derived from different parental *VIDs* upstream, thus allowing the node to reside on multiple branches. For example, node B has also *VID* $(R, 1, 1)$, derived from parent node R using its *VID* $(R, 0, 0)$. The *VID* numbering scheme helps in preventing loops and carries inherent *LPath* information. Nodes store and maintain their *VIDs* in a list called *VIDList*.

We notice that a new *VID* is formed by taking the parental *VID* and appending a single digit, known as Child Position *CPos*, to its right, then increase *hops* by 1. *CPos* is unique among node's children through maintaining a list called *ChildList* recording its children IDs, *CPos*, and their multiple *VIDs*. Lastly, the root node R maintains a list of its tree clients, *ClientList*, where it stores the IDs of all it clients and their multiple *VID*s.

Tree growth in MMT is controlled locally at each node by four parameters:

- *maxHop*: Defines the maximum number of hops allowed in a VID. It limits the length of a branch.

- *maxVID*: Defines the maximum number of VIDs a node can have in its *VIDList*. It controls number of branches a node can reside on, thus controlling the meshing of the branches.
- *maxChild*: Defines the maximum number of children a node allowed to have in its *ChildList*. It limits number of branches allowed to originate from a node.
- *maxClient*: Defines the maximum number of tree clients.

Algorithm 1 shows a simplified pseudo code for the MMT algorithm. This pseudo code is run by every node B that wishes to join a tree branch rooted at R. Table 3.1 explains the purpose and functionalities of functions used in MMT algorithm. In line 2, B initializes its *Neighbors* list to include all nodes in its transmission range. In lines 5 through 7, it iterates through each neighbor A and reads through its *VIDList*. Then in line 8, B excludes all *VIDs* from $VIDList_A$ which have *hops* equals to *maxHop*, thus enforcing the limitation parameter of *maxHop* for growing MMT branches. In line 9, each of the remaining *VIDs* in $VIDList_A$ is checked against all *VIDs* in $VIDList_B$ to determine whether it was derive from any of B's *VIDs*. As when a *VID* in $VIDList_A$ (VID_A) is found to be derived from another *VID* in $VIDList_B$ (VID_B), it is excluded from further processing. This check is pivotal in avoiding the creation of loops, in other words, to avoid deriving a *VID* from another *VID* which B already has. The algorithm used in derivation check is detailed later in section 3.1.

Line 10 gets the best *VID* from $VIDList_A$ based on *hops* value or any other cost metric. In lines 14 and 15, the algorithm enforces the limitation parameters of *maxChild* and *maxClient* for growing MMT branches. When passed previous limitations, a $newVID_B$ for B is derived from $BestVID_A$ in line 17 as was discussed earlier. Then, the newly derived $newVID_B$ is added to $VIDList_B$ (at the local node), $ChildList_A$ (at the parent) and $ClientList_R$ (at the root) in lines 18 through 20. Finally, the last check is performed in line 22 to enforce the last limiting parameter, *maxVID*, for growing MMT branches. If *maxVID* limit is reached, B gets the worst *VID* from $VIDList_B$ and removes it to free a slot for $newVID_B$ as shown in lines 23 and 24. Then, a cleanup of all other *VIDs* in all *VIDLists*, *ChildLists* and *ClientLists* that were derived from $WorstVID_B$ in line 25.

Derivation Check To check whether VID_A was derived from VID_B or not, we execute Algorithm 2. The derivation check algorithm is based on the comparison of the *LIDs* in

Algorithm 1 : MMT Algorithm

loop MMT
  Initialize *Neighbors*
  $VIDList_B \leftarrow VIDList$ in B
  $ClientList_R \leftarrow ClientList$ in R
  while $Neighbors \neq \emptyset$ **do**
    $A \leftarrow$ pop($Neighbors$)
    $VIDList_A \leftarrow VIDList$ in A
    removeMaxHop($VIDList_A$)
    removeDerived($VIDList_A$, $VIDList_B$)
    $BestVID_A \leftarrow$ getBestVID($VIDList_A$)
    $ChildList_A \leftarrow ChildList$ in A
    $ChildCount_A \leftarrow$ sizeOf($ChildList_A$)
    $ClientCount_R \leftarrow$ sizeOf($ClientList_R$)
    $Accept \leftarrow ChildCount_A < maxChild$ &
      $ClientCount_R < maxClient$
    if *Accept* **then**
      $newVID_B \leftarrow$ deriveVID($BestVID_A$)
      addVID($newVID_B$,$VIDList_B$)
      addVID($newVID_B$,$ChildList_A$)
      addVID($newVID_B$,$ClientList_R$)
      $VIDCount_B \leftarrow$ sizeOf($VIDList_B$)
      if $VIDCount_B > maxVID$ **then**
        $WorstVID_B \leftarrow$ getWorstVID($VIDList_B$)
        removeVID($WorstVID_B$, $VIDList_B$)
        removeDerivedAllLists($WorstVID_B$)
      end if
    end if
    delete A
  end while
end loop

both *VIDs* since the derivation process was nothing but appending digits to one of the *LIDs*. It starts by extracting the *LIDs* of both *VIDs* in lines 2 through 4, then finding the different in number of hops in line 5. If $diffHops \leq 0$, then it is impossible for VID_A to be derived from VID_B which is checked in line 6. The while loop in line 7 truncates a copy of the longer *LID* ($tempLID_A$) so it has the same number of digits as LID_B. Finally, the comparison and decision making occurs in line 11.

Algorithm 2 : Algorithm to check whether VID_A was derived from VID_B

function DERIVATIONCHECK(VID_A,VID_B)
    $LID_A \leftarrow VID_A \cdot LID$
    $tempLID_A \leftarrow LID_A$
    $LID_B \leftarrow VID_B \cdot LID$
    $diffHops \leftarrow VID_A \cdot hops - VID_B \cdot hops$
    if $diffHops > 0$ **then**
        while $diffHops > 0$ **do**
            $diffHops \leftarrow diffHops - 1$
            $tempLID_A \leftarrow (tempLID_A - (tempLID_A \%10))/10$
        end while
        if ($tempLID_A == LID_B$ **then**
            return TRUE
        else
            return False
        end if
    else
        return False
    end if
end function

TABLE 3.1: Functions in MMT Algorithm

Function	Purpose
pop(*Neighbors*)	Gets next node in set of *Neighbors*
removeMaxHop($VIDList_B$)	Removes *VID*s with *maxHop* limit from $VIDList_B$
removeDerived($VIDList_B$, $VIDList_A$)	Removes *VID*s in $VIDList_B$ that are derived from $VIDList_A$
getBestVID($VIDList_B$)	Gets *VID* with least *hops* in $VIDList_B$
sizeOf($ChildList_B$)	Gets number of entries in $ChildList_B$
deriveVID($BestVID_B$)	Gets a new *VID* derived from $BestVID_B$
addVID($newVID_A$,$VIDList_A$)	Adds $newVID_A$ to $VIDList_A$
getWorstVID($VIDList_A$)	Gets *VID* with largest *hops* in $VIDList_A$
removeVID($WorstVID_A$, $VIDList_A$)	Removes $WorstVID_A$ from $VIDList_A$
removeDerivedAllLists($WorstVID_A$)	Removes $WorstVID_A$ and all of its *VID* derivatives from *VIDList*s, *ChildList*s and *ClientList*s

3.2 MMT Protocol Implementation

In this section, we discuss how the MMT algorithm can be implemented as a routing protocol for MANETs. At every predefined hello interval, every node sends its *VIDList* in a *hello* packet which will be received by its neighbors, thus satisfying up to line 7 of Algorithm 1. Figure 3.2 shows a snippet of the message exchange, called registration process, which occurs during the creation of MMT tree in Figure 3.1. We notice that node B has one *VID* $(R, 1, 1)$ from the root node R derived from *VID* $(R, 0, 0)$. Thus, we observe that node B is present in $ChildList_R$ with *CPos* 1. Meanwhile, node A has two *VIDs* $(R, 2, 1)$ and $(R, 11, 2)$ derived from nodes R $(R, 0, 0)$ and B $(R, 1, 1)$, respectively. The first A's *VID* is in $ChildList_R$ with *CPos* 2 and the second is in $ChildList_B$ with *CPos* 1. Finally, we see that all *VIDs* of nodes A and B are in the $ClientList_R$ at the root node R.

At time T_0, node A attaches its $VIDList_A$ into a *hello* packet and broadcasts it to all nodes in range. As a result, node B receives a copy of $VIDList_A$ which is used to locally execute lines 8 through 10 in Algorithm 1, thus it realizes that *VID* $(R, 11, 2)$ from node A was already derived from its *VID* $(R, 1, 1)$ while the $BestVID_A$ is $(R, 2, 1)$. At time T_1, node B signals the selection of $BestVID_A$ by sending *RegistrationRequest* which will ultimately reach the root node R. As it traverses the tree branch, the parental node A will make sure that it has enough room in its $ChildList_A$ and derive $newVID_B$ using the newly assigned *CPos*, in this case 1. At this moment, we notice that node A has executed lines 14 and 17 in Algorithm 1.

At time T_2, the root node R receives the *RegistrationRequest* and makes sure that there is available space in the $ClientList_R$ in accordance to line 15 in Algorithm 1. We notice that the decision to accept the registration of $newVID_B$ happens at two levels, the parent and the root node, any of which can abort the registration process by simply sending a *RegistrationReject* packet to node B. In our case, $newVID_B$ is added to $ClientList_R$. After that, a *RegistrationAccept* is sent at time T_3 which will also traverse the same tree branch it came from allowing node A to add $newVID_B$ to $ChildList_A$. In addition, This ensures the existence of the branch in both directions. Finally, upon receiving the acceptance, node B adds the newly acquired *VID* to its $VIDList_B$. This is also mentioned in lines 18 through 20 in Algorithm 1.

Receiving a *VID* in periodic *hello* packets from a parent or a child indicates the existence of the link between the two nodes. On missing three consecutive announcements of a

VID, a node drops the corresponding *VID*. When a *VID* is dropped from the *VIDList*, a broadcast *Disconnect* packet is sent to the children nodes to dissolve their *VID*s derived from the dropped *VID*. On the other hand, a unicast *Disconnect* packet is sent to the root node *R* when a node notices that his child has dropped one of its *VID*s which is then used to clean the *ClientList* from the dropped *VID* and all of its derivatives. A similar behavior is followed as was mentioned in lines 24 and 25 in Algorithm 1. Note that all packet exchange are local except when informing the root node *R*; as a result, disseminating logical information is faster and there is no packet flooding.

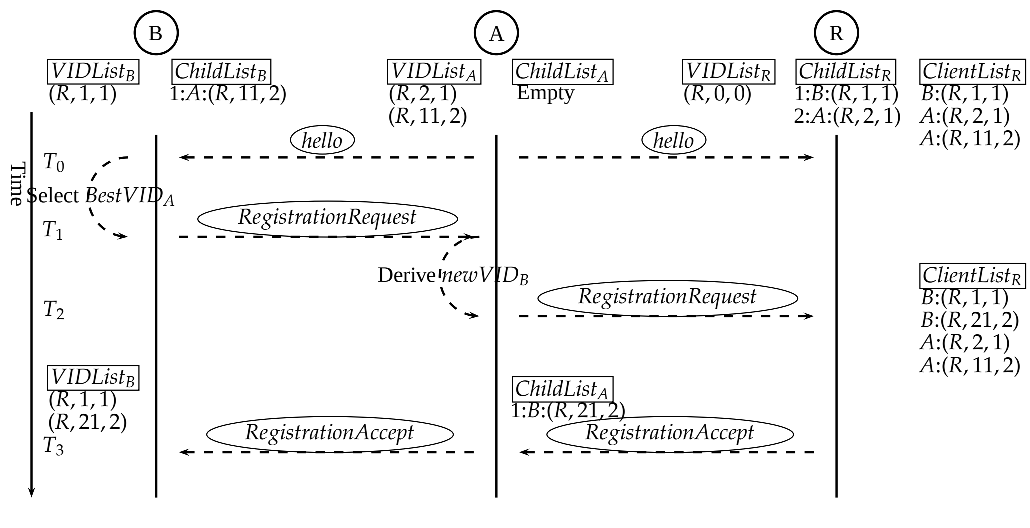

FIGURE 3.2: Registration Process in MMT Protocol

3.3 Field and Mobility Models

A set of N nodes, $V = 1, 2, , N$, have initial locations drawn from a two dimensional Poisson distribution in a field domain $F \in R^2$, after which each node picks a speed uniformly distributed on $[Sp_{min}, Sp_{max}]$ and a direction uniformly distributed on $[0, 2\pi]$. Speed and direction distributions are independent. When reaching the edge of F, a node makes a reflection angle equals to the angle of incidence. A new speed and direction is picked by a node every constant distance traveled called *StepSize*. This mobility model was chosen to maintain uniform node spatial density. We define $D_{ij}(t)$ as the Euclidean distance between nodes i and j at time t. A bidirectional link exists between two nodes i and j, with a topology adjacency matrix entry $\Phi(i, j, t) = 1$ when they become in transmission range D_{TX} of each other, that is $D_{ij}(t) \leq D_{TX}$; and $\Phi(i, j, t) = 0$ otherwise. Hence we can define the following:

- Topology: is a graph $G = (V, E)$, such that $|V| = N$ and at time t a *TLink* $(i, j) \in E$ iff $D_{ij}(t) \leq D_{TX}$
- φ_1: *TLink* duration between two nodes i and j, is the time duration $t_2 - t_1, t_2 > t_1$ such that $\Phi(i, j, t_1) = \Phi(i, j, t_2) = 1$ and $\Phi(i, j, t_1 - \epsilon) = \Phi(i, j, t_2 + \epsilon) = 0$
- φ_k: *TPath* duration of k hops (k *TLinks*) and $k + 1$ nodes $\{1, 2, , k, k + 1\}$, is the time duration $t_2 - t_1, t_2 > t_1$ such that $\prod_{i=1}^{k} \Phi(i, i + 1, t_1) = \prod_{i=1}^{k} \Phi(i, i + 1, t_2) = 1$ and $\prod_{i=1}^{k} \Phi(i, i + 1, t_1 - \epsilon) = \prod_{i=1}^{k} \Phi(i, i + 1, t_2 + \epsilon) = 0$

3.4 Protocol Stacks, Features and Simulation Variables

At the routing layer, we choose two proactive routing protocols based on tree creation, MMT and OLSR, the first uses a distributed algorithm and the latter is centralized. Even though we are analyzing only two protocols, same methodology and analysis presented in this dissertation can be applied to others. At the MAC layer, we design an ideal MAC (IMAC) which is able to avoid collisions at zero cost (time and overhead) while working on a channel with limited bandwidth. The reason for using IMAC is to show the true benefits of using one routing protocol over the other by removing the impact of MAC layers. Table 3.2 shows a list of protocol stack used and their purposes.

TABLE 3.2: Summary of Protocol Stacks

Protocol Stack	Routing Protocol	MAC Protocol	Purpose
TOPO	N/A	N/A	Enables the study of changes in topology
OLSRI	OLSR	IMAC	Simplifies studying the interactions between a routing layer and change in topology
MMTI	MMT	IMAC	

Clearly, many variables are involved in determining the interactions among mobility, topology change, *Adaptability* and performance. *Adaptability*, is dependent on the rate of updating topology information, using *hello* and topology related packets. MMT updates its topology information by sending *hello* packets periodically; However, OLSR uses two packets for that purpose, *hello* and *TC*. In our simulations, we identify one of the crucial simulation variables which is the interval exchanging *LLinks* or *LPaths* information in a routing protocol, *Ti*. We choose *Ti* to be the same for all packets involved in exchanging logical information, *hello* and *TC*, in order to make the comparison between MMT and OLSR fair.

Regarding mobility and topology, we identify two main variables involved in determining duration of *TLink* or *TPath*, φ_k, nodes' transmission range D_{TX} (in meters) and their speed range $Sp \in [Sp_{min}, Sp_{max}]$ (in meters/second). The bigger D_{TX}, the longer φ_k; and the faster the nodes, the shorter φ_k.

3.5 Measuring *Adaptability* and *AdaptationDelays*

Since *Adaptability* is the ability of a protocol stack to adapt to topology changes in a timely manner, it is measured as *Adaptationdelays*. *AdaptationDelays* are the time lag between a topology change and the corresponding logical information for which, we require two processes to monitor the following:

- $\Phi(i, j, t)$: The Topology Adjacency Matrix. Needed to monitor networks topology continually and
- $\Psi(i, j, t)$: The Logical Adjacency Matrix. Needed to monitor logical information as perceived by routing layers

3.5.1 Monitoring Topology Adjacency Matrix

If two nodes i and j come into or move out of transmission range D_{TX}, then the corresponding entries in the Topology Adjacency Matrix, $\Phi(i, j, t)$, are updated accordingly. The matrix gives the current networks topology at any instant, which is used to determine if node A could have a *TPath* to the root node R. Note that the concept of root node R applies to OLSR and MMT; in OLSR it refers to the node of interest executing Dijkstra's algorithm, while in MMT it refers to the Cluster Head CH node. Node A has a *TPath* to the root node R when the two nodes are in the same graph component. Changes to this matrix are monitored to record the following times with respect to any node A:

- T_T^{in}: Topological in-contact time, is the time when node A becomes a member of the graph component which has the root node R.
- T_T^{out}: Topological out-of-contact time, is the time when node A just leaves the graph component which includes the root node R.

As packets, intended to the root node R, are generated at node A they are categorized as:

- P_k^{TP}: Topologically possible packet with k hops, which means that node A and the root node R were members of the same graph component and the shortest *TPath* between them has k hops when the packet was generated.
- P^{TnP}: Topologically not possible packet, which means that node A and the root node R were in two different graph components when the packet was generated.

3.5.2 Monitoring Logical Adjacency Matrix

The monitoring of the Logical Adjacency Matrix, $\Psi(i, j, t)$, happens at a deeper level by tracking logical information as seen by the routing protocol indicating when node A can or can't talk to the root node R. Due to *AdaptationDelays*, changes in Topology Adjacency Matrix entries, $\Phi(i, j, t)$, usually precedes those in Logical Adjacency Matrix, $\Psi(i, j, t)$. Changes to $\Psi(i, j, t)$ are monitored to record the following with respect to any node A:

- T_L^{in}: Logical in-contact time, is the time when the routing protocol at node A calculates a *LLink* or *LPath* to root node R.
- T_L^{out}: Logical out-of-contact time, is the time when all *LLink* and *LPath* at node A to the root node R are removed.

To explain the time components defined above, Figure 3.3 is used where node A follows a trajectory as indicated by the arrow. The dotted circle indicates the transmission range D_{TX}. Node A starts moving at time T_0. T_T^{in} through T_L^{out} indicate four different time instants, where time difference between T_T^{out} and T_T^{in} is φ_1 between A and R, which is represented by the dark-shaded rectangle and time difference between T_L^{out} and T_L^{in} is the *LLink* time duration ψ_1 and is represented by the light-shaded rectangle. However, it is clear that the two time durations do not match due to *AdaptationDelays* of the routing protocol in responding to topology changes. As a result, one can identify the following two types of adaptation delays for a k hops path.

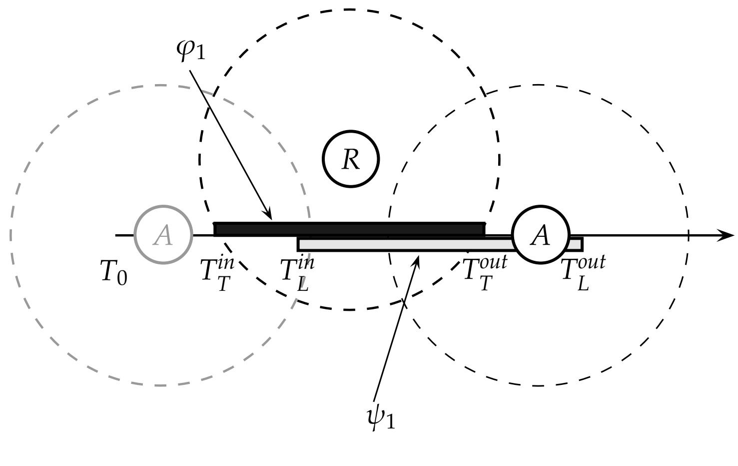

FIGURE 3.3: Defining Adaptation Delays

- ξ_k^{in}: Delay in realizing in-contact over k hops path. Routing protocols should minimize this delay in order to maximize the utilization of φ_k.

$$\xi_k^{in} = T_L^{in} - T_T^{in} \tag{3.1}$$

- ξ_k^{out}: Delay in realizing out-of-contact over k hops path. Minimizing this delay will decrease failed retransmissions on broken *TLink* and hence conserve energy. Additionally, realizing broken *TLink* faster improves routing performance by forcing the routing algorithm to calculate alternative *LPath* if possible.

$$\xi_k^{out} = T_L^{out} - T_T^{out} \tag{3.2}$$

As packets intended from node A to the root node R (coming from upper layer) reach the routing layer, it decides whether they will be passed to lower MAC layer or not based on available logical information. Hence, we define the following two packet categories:

- P_k^{LP}: Logically possible packet, means that node A and the root node R can talk over k hops *LPath* based on Logical Adjacency Matrix, $\Psi(i, j, t)$, at the packet generation time.
- P^{LnP}: Logically not possible packet, which means that node A and the root node R cant talk as shown by Logical Adjacency Matrix, $\Psi(i, j, t)$, at the packet generation time.

Table 3.3 concludes the relationship between the defined *AdaptationDelays* and the different packet categories. Note that packets from node A can only be received at the root node R when the corresponding entries in $\Phi(i, j, t)$ and $\Psi(i, j, t)$ are both True, we call this duration the usable duration ϖ_k which also can be calculated in (3.3). A received packet with k hops is represented as P_k^R, consequently, a lost packet (not received) is denoted by P^{nR}.

$$\varpi_k = \varphi_k - \xi_k^{in} \tag{3.3}$$

TABLE 3.3: Adaptation Delays and Packet Categories

Time Duration	Topological	Logical	Minimizing Duration	Received
ξ_k^{in}	P_k^{TP}	P_k^{LnP}	More φ_k utilization	P^{nR}
ξ_k^{out}	P^{TnP}	P^{LP}	Less failed transmissions	P^{nR}
ϖ_k	P_k^{TP}	P_k^{LP}	N/A	P_k^R
Others	P^{TnP}	P^{LnP}	N/A	P^{nR}

3.6 Base Line Performance Results

Results presented in this section will serve as an entry point to the proposed analytical models. Relationship and observations reveal directions on how analytical models will be derived and combined. Firstly, we will discuss *Topological* results in which we decide the values of variables that produce maximum topology changes in simulation and analytical models. Producing the maximum topology changes amplifies the impact of *AdaptationDelays* and emphasizes the importance of our *Adaptability* modeling. Then, we present *Adaptability* results of the two protocols, MMT and OLSR, in order to see how they differ when measuring their *AdaptationDelays* and how we can model them. Lastly, performance results are discussed in the light of previous *Topological* and *Adaptability* results; which also will guide the generation of performance models under mobility.

3.6.1 Topological Results

To study the impact of topology changes with mobility and *AdaptationDelays* on performance, we designed 3 simulation scenarios of network sized 10, 20 and 40, named as the scenarios Sc.10.Nodes, Sc.20.Nodes and Sc.40.Nodes, respectively. These 3 simulation scenarios are shown in Table 3.4. All scenarios have the same root and node density of 2.78×10^{-6} and 25×10^{-6} per m^2 respectively. Nodes are moving according the mobility model discussed in section 3.3. Since all scenarios have constrained field sizes, simulation parameters should be chosen carefully; for instance, choosing very high D_{TX} results in well-connected network with less topology changes defeating the purpose of the study. Meanwhile, very low D_{TX} means nodes spent most of simulation time stranded with few neighbors communicating.

TABLE 3.4: Full Scale Random Mobility Scenarios Summary

Scenario	Field Size	Roots	Total Nodes
Sc.10.Nodes	600m X 600m	1	10
Sc.20.Nodes	600m X 1200m	2	20
Sc.40.Nodes	1200m X 1200m	4	40

TABLE 3.5: Simulation Parameters for Identifying a Suitable D_{TX}

Parameter	Value(s)
D_{TX}	100m, 150m, 200m, 250m, 300m
Sp_{avg}	5m/s

As shown in Table 3.5, we ran the TOPO stack with 5 different $D_{TX} \in \{100m, 150m, 200m, 250m, 300m\}$ applied to the 3 scenarios to identify the most suited D_{TX} satisfying the study requirement of producing most topology changes to amplify the impact of *AdapatationDelays* and show the importance of *Adaptability* study. In all simulation runs, we fixed $[Sp_{min}, Sp_{max}] = [4, 6](m/s)$ while two metrics were collected:

- Transition Index: is an indication of topology changes produced by monitoring nodes' transitions between joining and leaving the graph component containing a root node. For each scenario, the index is calculated by normalizing the count of T_T^{in} and T_T^{out} logged during a simulation run across all runs.

- Connectivity Probability: which is the probability that a node is in the same graph component of that of the root during a simulation run.

In Figure 3.4, we plot transition index with D_{TX}, which shows that maximum topology changes occurred when $D_{TX} = 200m$ in all 3 scenarios and fewer topology changes otherwise. Figure 3.5 depicts the relationship between D_{TX} and connectivity probability for the 3 scenarios. We notice that when $D_{TX} < 200m$, the connectivity probability is low as a result of weakly connected network with fewer interactions among nodes. This agrees with Figure 3.4 where fewer topology changes were logged. On the other hand, increasing $D_{TX} > 200m$ results in well-connected network since nodes have a lower chance to escape each other's D_{TX}, this also results in fewer topology changes as shown in Figure 3.4. As a result, we adopt $D_{TX} = 200m$ in all future studies to produce maximum topology changes unless indicated otherwise.

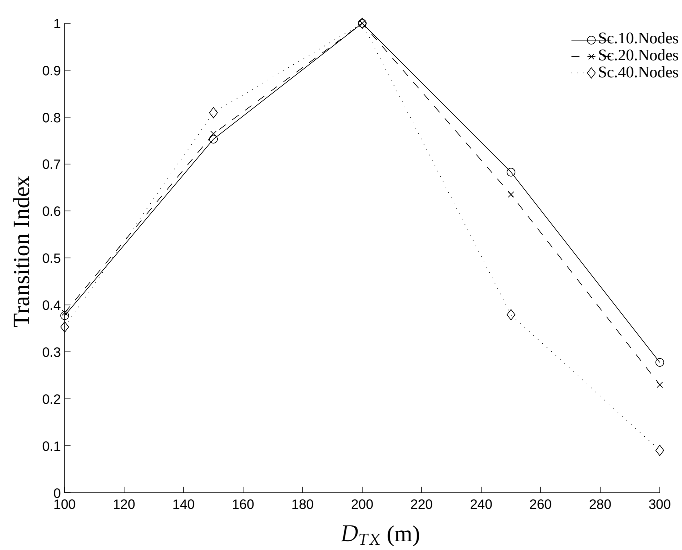

FIGURE 3.4: Plot of Transition Index with D_{TX}

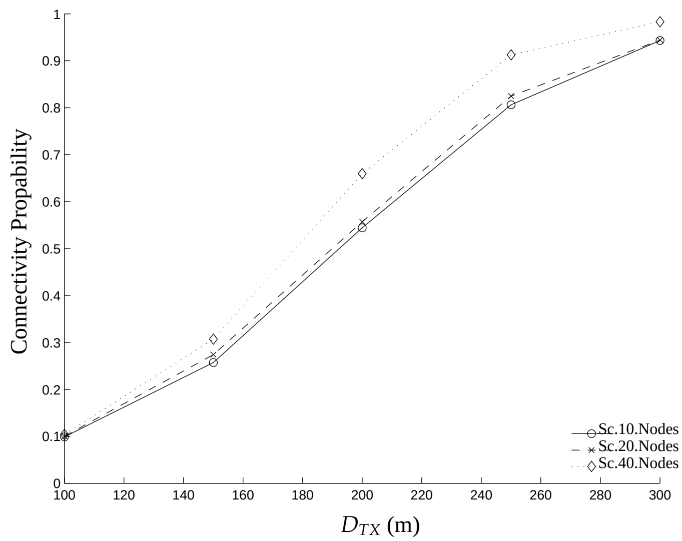

FIGURE 3.5: Plot of Connectivity Probability with D_{TX}

In addition, we collected same metrics from the 3 scenarios by fixing $D_{TX} = 200m$ and varying [1] $[Sp_{min}, Sp_{max}] \in \{[4,6]m/s, [9,11]m/s, [14,16]m/s, [19,21]m/s\}$ to study the impact. Simulation parameters are shown in Table 3.6, while Figure 3.6 shows the linear increase in Transition Index as we increase Sp_{avg}. However; as expected, Connectivity Probability remained the same for each scenario regardless of the change in Sp_{avg} and shown in Figure 3.7.

TABLE 3.6: Simulation Parameters for Studying the Impact of Increasing Sp_{avg}

Parameter	Value(s)
D_{TX}	200m
Sp_{avg}	5m/s, 10m/s, 15m/s 20m/s

[1]Values of $[Sp_{min}, Sp_{max}]$ were chosen with $\Delta Sp = 2$m/s as a convenient range to prevent overlap in Speed ranges. Meanwhile, Sp_{avg} was chosen at increments of 5m/s where maximum Sp_{avg} does not exceed 20m/s as it is the value where *LPath* utilization ratio approaches 50% as will be shown later

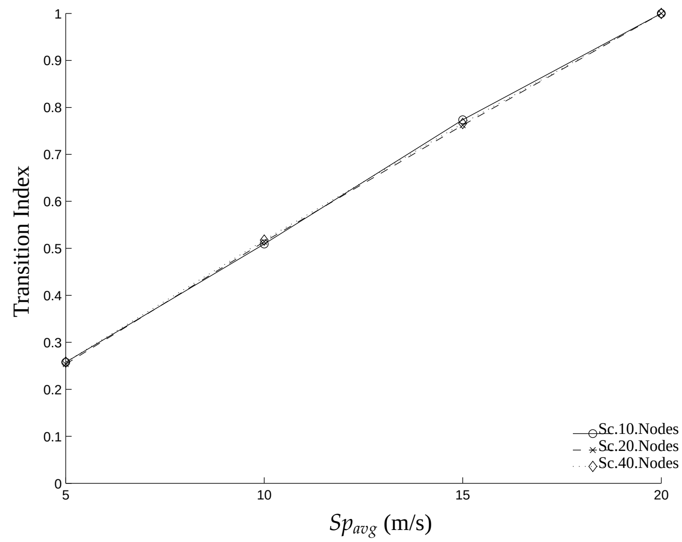

FIGURE 3.6: Plot of Transition Index with Sp_{avg}

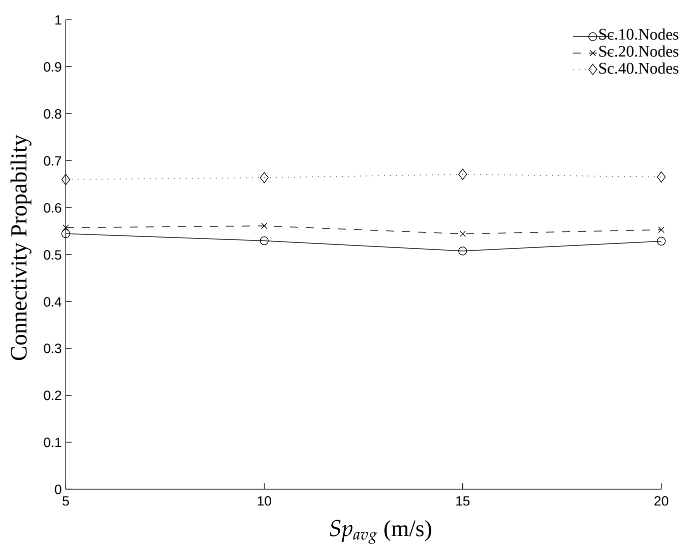

FIGURE 3.7: Plot of Connectivity Probability with Sp_{avg}

We collected values of the average φ_k, $\varphi_{k_{avg}}$, using the Scenarios in Table 3.4 and simulation parameters in Table 3.6. Figure 3.8 shows $\varphi_{k_{avg}}$ with varying Sp_{avg}. We observe that $\varphi_{k_{avg}}$ decreases when increasing Sp_{avg} or increasing number of hops, k. Referring to Table 3.7, we also notice that for any two values of $\varphi_{k^a_{avg}}$ and $\varphi_{k^b_{avg}}$ recorded with Sp^a_{avg} and Sp^b_{avg}, while number of hops k were k^a and k^b; they have the following approximation in (3.4). Note that this is just a mere approximation which only purpose is to demonstrate the trend in which $\varphi_{k_{avg}}$ changes with Sp_{avg} and k.

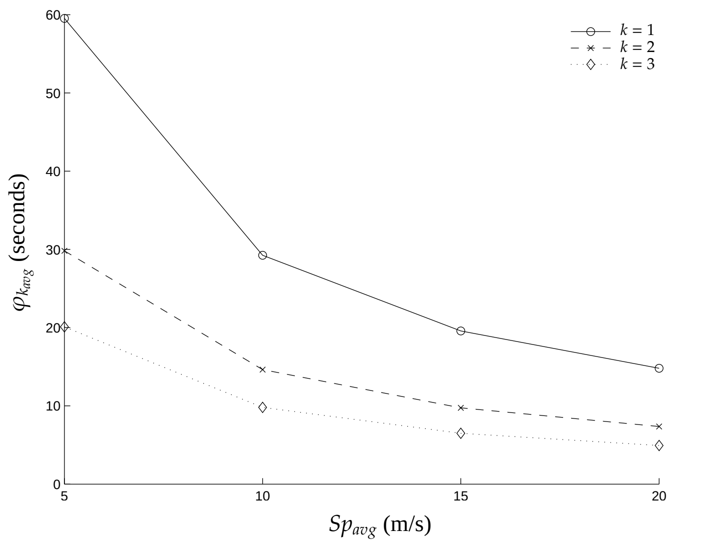

FIGURE 3.8: Plot of $\varphi_{k_{avg}}$ with Sp_{avg}

$$\left(k^a \times Sp^a_{avg}\right) \varphi_{k^a_{avg}} \approx \left(k^b \times Sp^b_{avg}\right) \times \varphi_{k^b_{avg}} \tag{3.4}$$

TABLE 3.7: Values of $\varphi_{k_{avg}}$ with Sp_{avg}

Number of hops k	Sp_{avg} 5m/s	10m/s	15m/s	20m/s
$k = 1$	59.53s	29.25s	19.57s	14.81s
$k = 2$	29.81s	14.63s	09.75s	07.36s
$k = 3$	20.10s	09.80s	06.49s	04.94s

Figures 3.9 depicts the probability density functions *pdf*s of φ_1, $f(\varphi_1)$, when Sp_{avg} changes. We notice that all *pdf*s have the maximum value according to (3.5). The reason for this will be explained in Section 4.1:

$$\varphi 1_{maxProb} = \frac{D_{TX}}{Sp_{avg}} \tag{3.5}$$

Referring to Figures 3.9, 3.10 and 3.11, increasing Sp_{avg} shifts $f(\varphi_k)$ to the left and narrows it which agrees with (3.4). In other words, increasing the Sp_{avg} by a factor of r while keeping D_{TX} the same, decreases the φ by a factor of r. Hence, if we let $f(\varphi_k)$ at Sp^a_{avg} be represented by $f(\varphi_k)\mid_{Sp^a_{avg}}$, then:

$$f(\varphi_k)\mid_{Sp^b_{avg}} \approx \frac{Sp^b_{avg}}{Sp^a_{avg}} \times f\left(\frac{Sp^b_{avg}}{Sp^a_{avg}}\varphi_k\right)\mid_{Sp^a_{avg}} \tag{3.6}$$

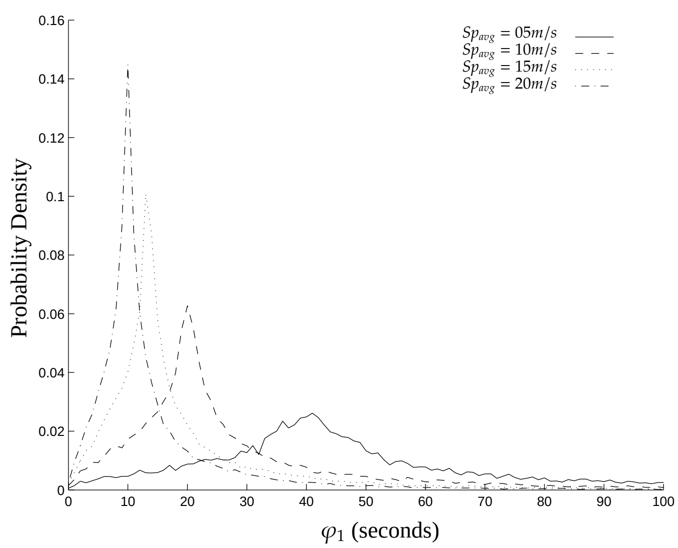

FIGURE 3.9: $f(\varphi_1)$ with Sp_{avg}

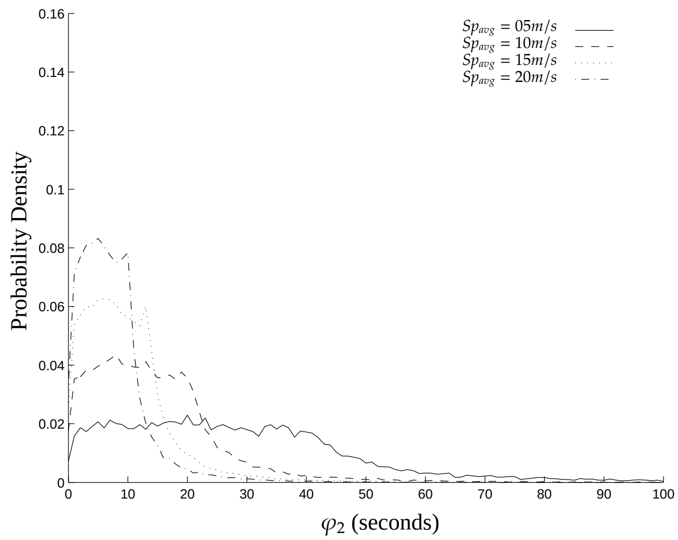

FIGURE 3.10: $f(\varphi_2)$ with Sp_{avg}

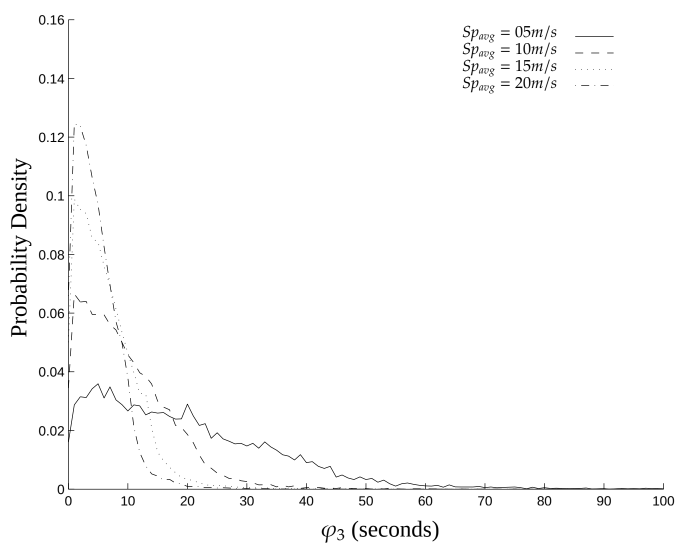

FIGURE 3.11: $f(\varphi_3)$ with Sp_{avg}

3.6.2 Adaptability Results

Results in this section were gathered by simulating MMTI and OLSRI protocol stacks using the Scenarios in Table 3.4 and Simulation Parameters in Table 3.8. In Figures 3.12, 3.13 and 3.14, we show $\xi^{in}_{k_{avg}}$ and $\xi^{out}_{k_{avg}}$ with respect to number of hops k as we change [2] $Ti \in \{1s, 2s, 3s\}$. we see that $\xi^{in}_{1_{avg}}$ in MMT is $\frac{Ti}{2}$ while in OLSR it is Ti and $\xi^{out}_{1_{avg}}$ for both protocols is around $\frac{5}{2}Ti$. These observations and more will be detailed and modeled in section 5. In addition, we observe that $\xi^{in}_{k_{avg}}$ increases as we increase the number of hops k, simply because more k means longer time to forward *LLink*s or *LPath*s information to other nodes.

TABLE 3.8: Simulation Parameters for Studying the Impact of Increasing Ti

Parameter	Value(s)
D_{TX}	200m
Sp_{avg}	5m/s
Ti	1s, 2s, 3s

In OLSR, we observe a faster increase in delay from $\xi^{in}_{2_{avg}}$ to $\xi^{in}_{3_{avg}}$. This increased delay is because OLSR nodes select and signal, using *hello* packet, *MPR* nodes which send and forward Topology Control *TC* packets in order to build 3 or more hops *LPath*s. This process takes longer time to accomplish than the simple *hello* packet exchange used to build 2 or less hops *LPaths*. We also observe that MMT exhibits linear increase in $\xi^{out}_{k_{avg}}$ as number of hops k increases due to the increased delay in resolving the associated branches and *VID*s in the dissemination of *Disconnect* packets.

[2]Values of Ti were chosen at increments of 1s where maximum Ti does not exceed 3s as it is the value when *LPath* utilization ratio approaches 50% as will be shown later

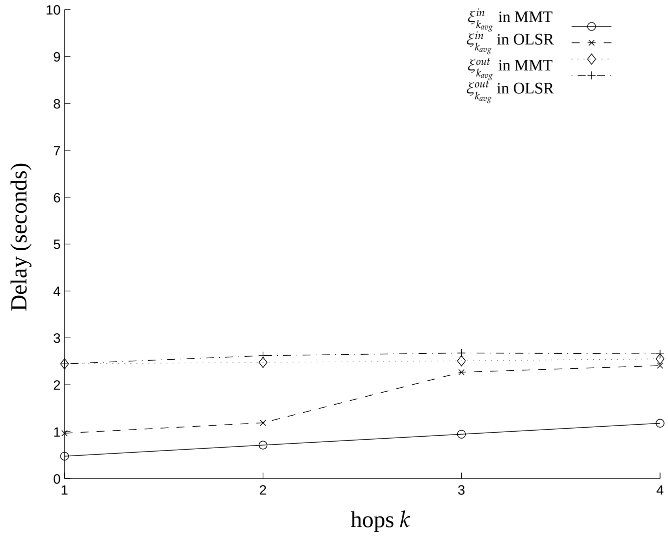

FIGURE 3.12: Plot of $\xi^{in}_{k_{avg}}$ and $\xi^{out}_{k_{avg}}$ in MMT and OLSR with hops k and $Ti = 1s$

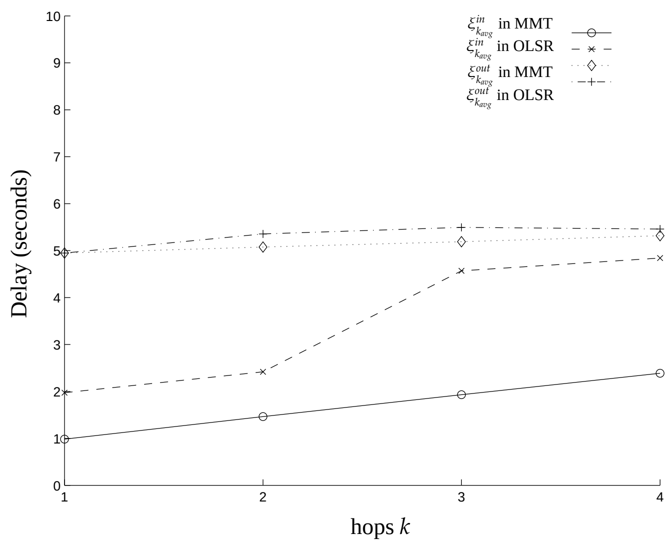

FIGURE 3.13: Plot of $\xi^{in}_{k_{avg}}$ and $\xi^{out}_{k_{avg}}$ in MMT and OLSR with hops k and $Ti = 2s$

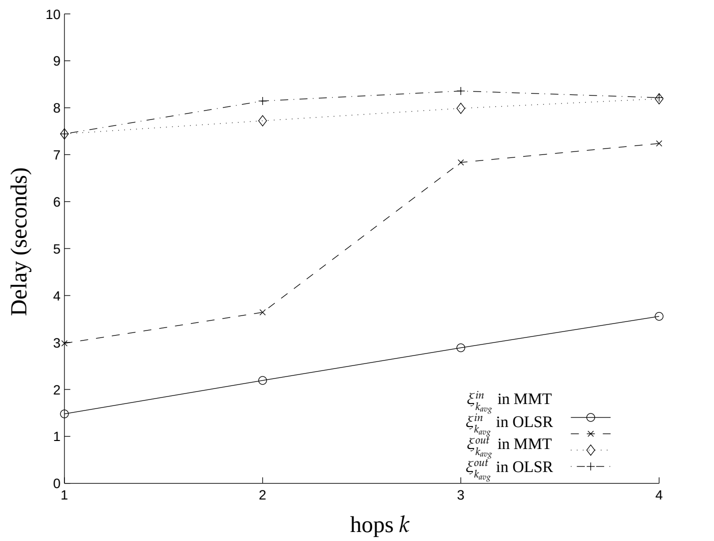

FIGURE 3.14: Plot of $\xi^{in}_{k_{avg}}$ and $\xi^{out}_{k_{avg}}$ in MMT and OLSR with hops k and $Ti = 3s$

3.6.3 Performance Results

In this section we simulate MMTI and OLSRI with Constant Bit Rate (CBR) packet generation model. We use the 3 scenarios in Table 3.4 with the Simulation Parameters shown in Table 3.9.

TABLE 3.9: Simulation Parameters for Collecting Performance Results

Parameter	Value(s)
D_{TX}	200m
Sp_{avg}	5m/s, 10m/s, 15m/s, 20m/s
Ti	1s, 2s, 3s

For a specific *LPath* with k hops, we define the following Utilization Ratio $\widetilde{\mathfrak{I}_k}$:

$$\widetilde{\mathfrak{I}_k} = \frac{\varpi_k}{\varphi_k}, \quad \varpi_k \geq 0 \tag{3.7}$$

Since the usable duration $\varpi_k = \varphi_k - \xi^{in}_k$, in (3.3), then:

$$\widetilde{\Im_k} = \begin{cases} 1 - \frac{\xi_k^{in}}{\varphi_k} & \xi_k^{in} \leq \varphi_k \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

$\widetilde{\Im_k}$ is a random variable that measures the ability of a routing protocol to utilize φ_k in delivering packets successfully to destination with respect to a specific *LPath*. The realizations of Utilization Ratio averaged over all *LPath*s of the same k hops is denoted as $\Im_k$. With the aid of Figure 3.3, one can find $\Im_k$ with respect to packet counts (#) as follows:

$$\Im_k = \frac{\#P_k^R}{\#P_k^{TP}} \tag{3.9}$$

We also define the overall $\Im$ of all *LPath*s up to maximum number of hops, k_{max}:

$$\Im = \frac{\sum_{k=1}^{k_{max}} \#P_k^R}{\sum_{k=1}^{k_{max}} \#P_k^{TP}} \tag{3.10}$$

$\Im$ measures the protocol's ability to utilize temporal paths in dynamic topology. This metric is different from the packet delivery ratio, which is usually the ratio between delivered and generated packets. $\Im$ takes into account the instantaneous networks ground truth topology and doesn't penalize the protocol during network segmentations. Protocols with higher $\Im$ are expected to have higher packet delivery ratio and lower packet latencies.

In table 3.10, we show an example of applying (3.9) and (3.10) for MMT and OLSR. Results of $\Im$ in MMT and OLSR are shown in Figures 3.16 through 3.24. Referring to these figure we conclude two observations. Firstly, Increasing the speed results in shorter φ_k; Hence, decreasing $\Im_k$. A *TPath* is only usable when it is logged at the routing layer as *LPath*. As a result, a *TPath* of k hops is unusable when $\varpi_k < 0$ and $\varphi_k \leq \xi_k^{in}$, using (3.3), which means it has a zero $\widetilde{\Im_k}$ in reference to (3.8). In Figure 3.15 we show $f(\varphi_2)$ with varying Sp_{avg}. To simplify the discussion, we assume that ξ_2^{in} is a constant value of 5s for which we plot a dashed rectangle representing the unusable φ_2. As a result we can find the probability of φ_2 being unusable:

$$P[\varphi_2^{in} \leq 5] = \int_0^5 f(\varphi_2) \mathrm{d}\varphi_2 \tag{3.11}$$

Notice that when we increase Sp_{avg}, more φ_k become unusable since $TPath$s have shorter durations, this results in decreasing $\widetilde{\mathfrak{I}_k}$ and eventually decreasing $\mathfrak{I}_k$ which agrees with the results shown in Figures 3.16 through 3.24.

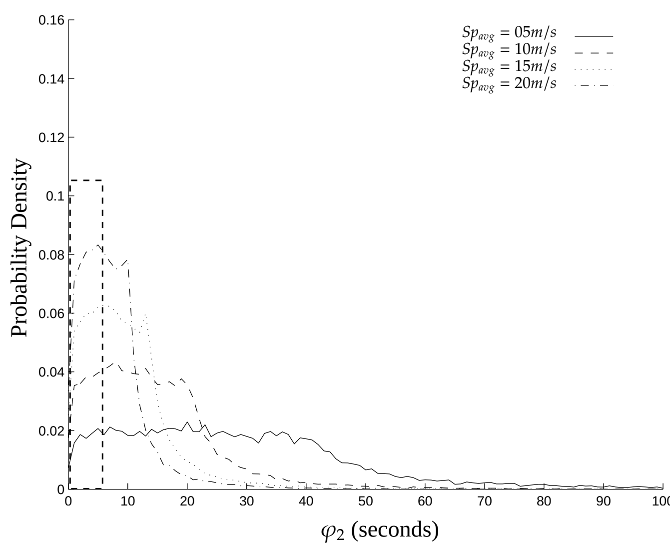

FIGURE 3.15: $f(\varphi_2)$ with Sp_{avg} and the Impact of ξ_2^{in}

The second observation is that decreasing ξ_k^{in} increases $\mathfrak{I}_k$ as evident in (3.8) then we conclude that the protocol which exhibits shorter ξ_k^{in} is expected to have higher $\mathfrak{I}_k$. In section 3.6.2, we showed that MMT has shorter ξ_k^{in} than OLSR; hence it will have higher $\mathfrak{I}_k$. Indeed, referring to Figures 3.16 through 3.24, we see that $\mathfrak{I}$ for MMT is always higher than OLSR regardless of Sp_{avg}. As we increase Ti, compare Figures 3.16 and 3.18, we also observe the drop of $\mathfrak{I}$ curves in both protocols due to the increase in ξ_k^{in}.

TABLE 3.10: Calculating $\mathfrak{I}_k$ and $\mathfrak{I}$ for MMT and OLSR in Scenario Sc.10.Nodes with $Sp_{avg} = 5m/s$ and $Ti = 1s$

Title	Number of Hops				
	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
# P_k^{TP}	1688099	977095	520475	220405	66548
# P_k^R in MMT	1672582	886548	438913	171515	48167
# P_k^R in OLSR	1659641	860799	365719	130951	33264
$\mathfrak{I}_k$ in MMT	0.991	0.907	0.843	0.778	0.724
$\mathfrak{I}_k$ in OLSR	0.981	0.881	0.703	0.594	0.500
$\mathfrak{I}$ in MMT	0.927				
$\mathfrak{I}$ in OLSR	0.878				

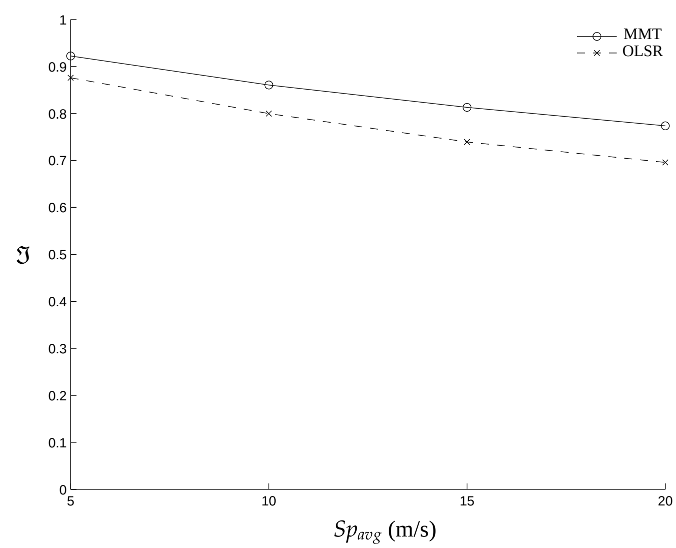

FIGURE 3.16: $\mathfrak{I}$ in scenario Sc.10.Nodes and Ti=1s with Sp_{avg}

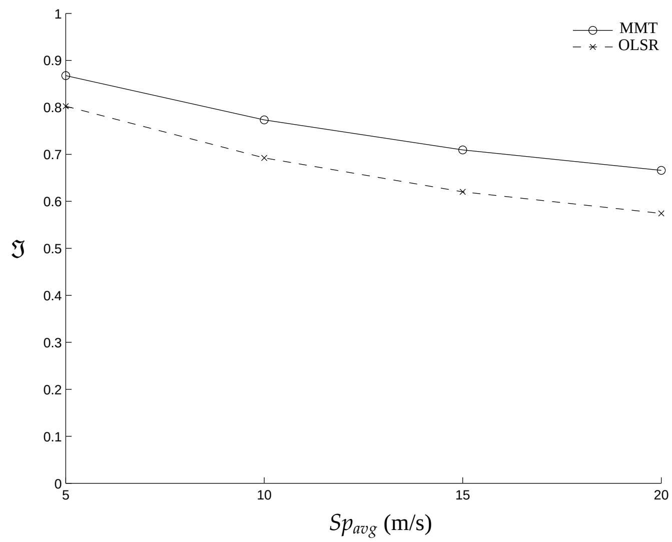

FIGURE 3.17: $\mathfrak{I}$ in scenario Sc.10.Nodes and Ti=2s with Sp_{avg}

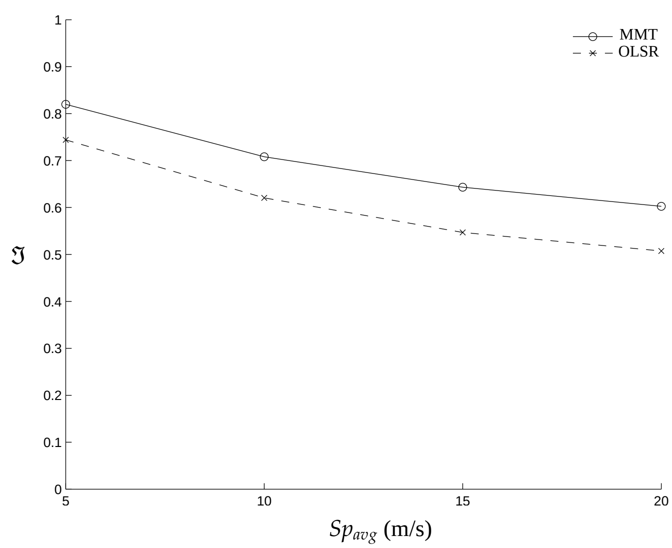

FIGURE 3.18: $\mathfrak{I}$ in scenario Sc.10.Nodes and Ti=3s with Sp_{avg}

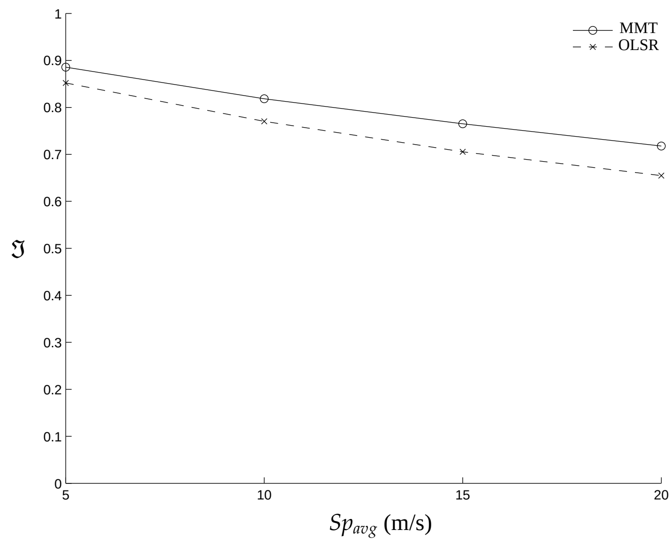

FIGURE 3.19: $\mathfrak{I}$ in scenario Sc.20.Nodes and Ti=1s with Sp_{avg}

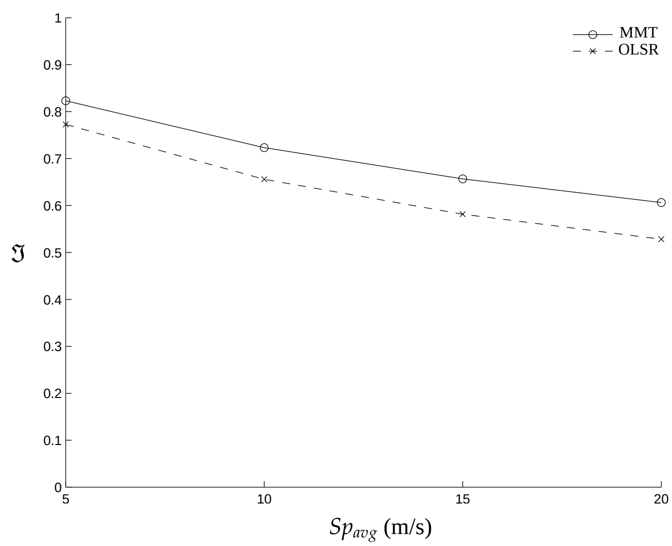

FIGURE 3.20: $\mathfrak{I}$ in scenario Sc.20.Nodes and Ti=2s with Sp_{avg}

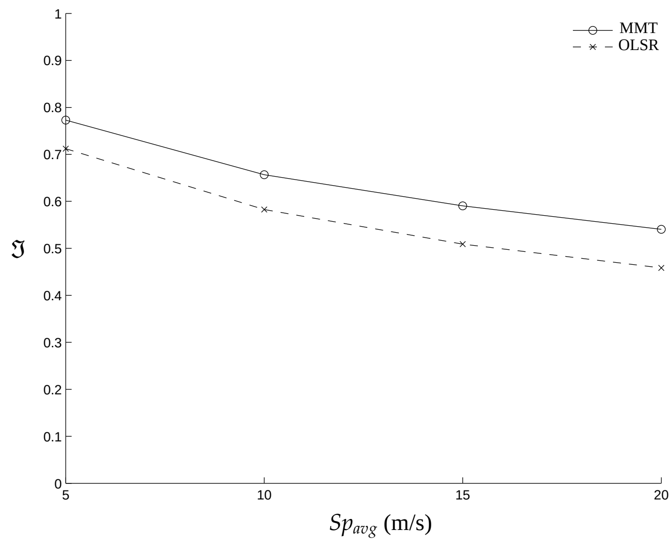

FIGURE 3.21: $\mathfrak{I}$ in scenario Sc.20.Nodes and Ti=3s with Sp_{avg}

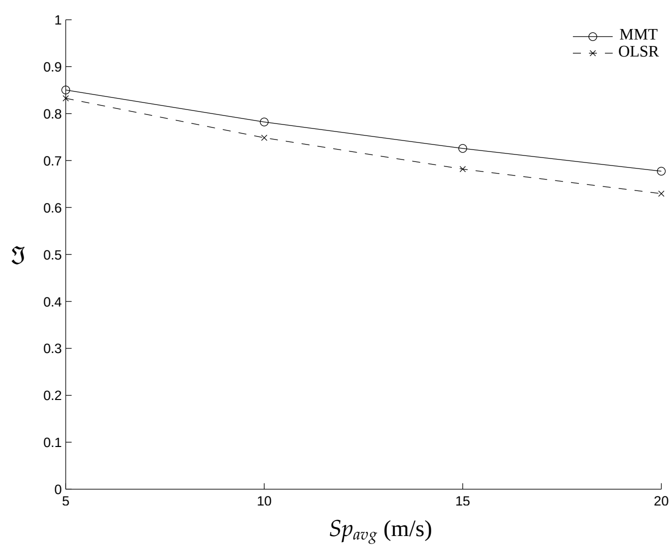

FIGURE 3.22: $\mathfrak{I}$ in scenario Sc.40.Nodes and Ti=1s with Sp_{avg}

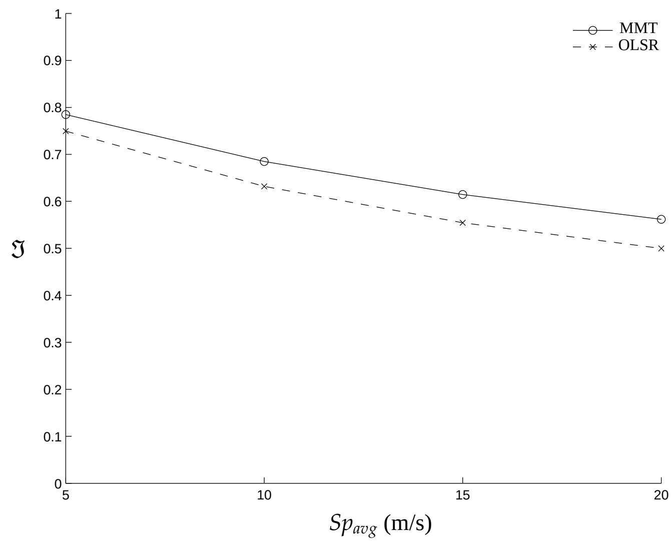

FIGURE 3.23: $\Im$ in scenario Sc.40.Nodes and Ti=2s with Sp_{avg}

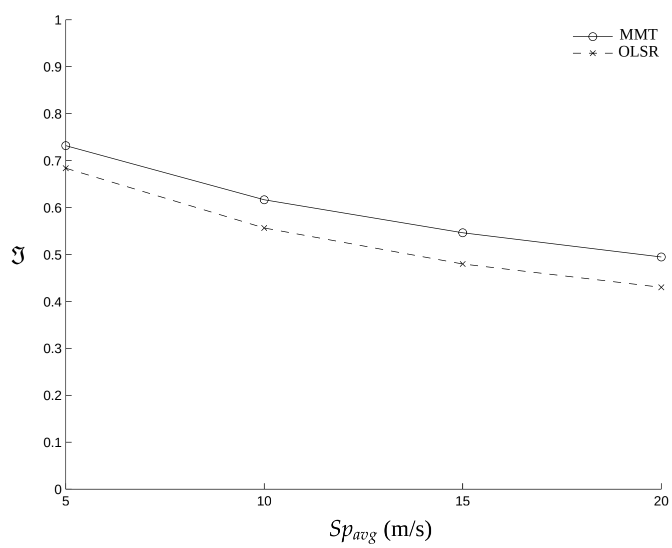

FIGURE 3.24: $\Im$ in scenario Sc.40.Nodes and Ti=3s with Sp_{avg}

P_k^{TP} and # P_k^R in MMT and OLSR shown in Table 3.10 can be normalized using formula 3.12. Normalized values are shown in Table 3.11 and plotted in Figure 3.25. $Norm(P_k^{TP})$ represents the distribution of P_k^{TP} packets with respect to number of hops k, while $Norm_{MMT}(P_k^R)$ and $Norm_{OLSR}(P_k^R)$ represents the fraction of those packets that were received in each protocol. Note that taking the sum over k gives the $\Im$ calculated previously as shown in the last column.

$$Norm(X) = \frac{X}{\sum_{k=1}^{k_{max}} P_k^{TP}} \tag{3.12}$$

TABLE 3.11: $Norm(P_k^{TP})$ and $Norm(P_k^R)$ for MMT and OLSR in Scenario Sc.10.Nodes with $Sp_{avg} = 5m/s$ and $Ti = 1s$

Title	Number of Hops					
	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$\sum_{k=0}^{\infty}$
$Norm(P_k^{TP})$	0.486	0.281	0.150	0.063	0.019	1.000
$Norm_{MMT}(P_k^R)$	0.482	0.255	0.126	0.049	0.014	0.927
$Norm_{OLSR}(P_k^R)$	0.478	0.248	0.105	0.038	0.010	0.878
$Norm_{MMT}(P_k^R) - Norm_{OLSR}(P_k^R)$	0.0037	0.0074	0.0211	0.0117	0.0043	0.0490

In Table 3.11, we notice that the difference between $Norm_{MMT}(P_k^R)$ and $Norm_{OLSR}(P_k^R)$ is not constant, as its maximum is when $k = 3$ and its minimum is when $k = 1$. This makes average hops for received packets, k_{avg}, in MMT higher than OLSR as shown in Figures 3.28 through 3.36. The sequential reason is that:

1. OLSR experiences a sudden increase from $\xi_{2_{avg}}^{in}$ to $\xi_{3_{avg}}^{in}$ as shown previously in Figures 3.12 through 3.14 due to the fact that OLSR nodes select and signal *MPR* nodes which send and forward Topology Control *TC* packets in order to build 3 or more hops *LPath*s. This process takes longer time to accomplish than the simple *hello* packet exchange used to build 2 or less hops *LPaths*. On the other hand, MMT has a near linear increase in $\xi_{k_{avg}}^{in}$.

2. Since $\varpi_k = \varphi_k - \xi_k^{in}$ from (3.3), then a sudden increase in ξ_3^{in} in OLSR decrease ϖ_3 for OLSR at a rate higher than MMT.

3. In Table 3.3, we notice that a packet is only received during ϖ_k. As a result, decreasing ϖ_3 in OLSR decreases # P_3^R for OLSR at larger rate than MMT.

4. According to (3.12), we see that decreasing # P_3^R for OLSR at larger rate than MMT causes the $Norm_{OLSR}(P_3^R)$ to decrease at larger rate than $Norm_{MMT}(P_3^R)$.

Moreover, we observe that k_{avg} for MMT and OLSR decreases as we increase Sp_{avg} in Figures 3.28 through 3.36 for the following sequential reason:

1. Increasing Sp_{avg} decreases the probability of longer φ_k, with greater impact when k is increasing due to the increased number of *TLinks* involved in the formation of longer *TPaths*, as shown in Figures 3.9 through 3.11.
2. Since $\varpi_k = \varphi_k - \xi_k^{in}$ from (3.3), then decreasing the probability of longer φ_k results in lower $P[\varpi_k > 0]$ with greater impact when k is increasing.
3. As we see in Table 3.3, we notice that # P_k^R is dependant on the duration of ϖ_k. Thus, decreasing ϖ_k, with greater impact when k is increasing, causes a corresponding decrease in # P_k^R, with greater impact when k is increasing. As a result, we notice a decrease in k_{avg}.

This is also evident when comparing Figures 3.25 and 3.26. Similar impact happens when we increase Ti, as comparing Figures 3.26 and 3.27 reveals the decrease in $Norm(P_k^R)$, with greater impact when k is increasing. Eventually, this decreases k_{avg} as depicted in Figures 3.28 through 3.36.

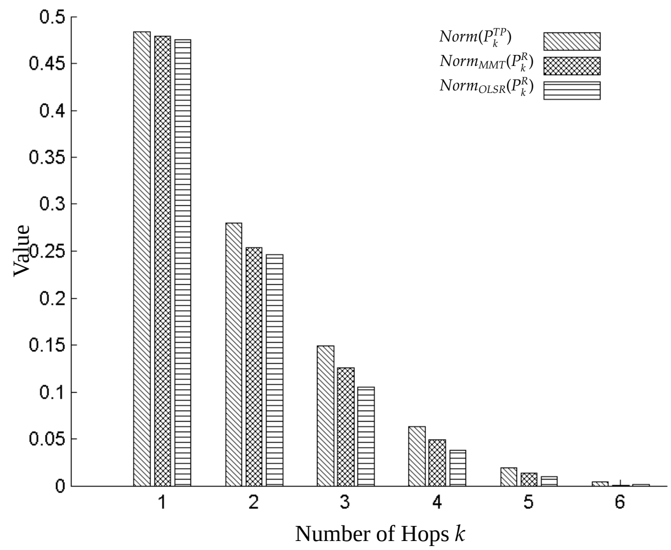

FIGURE 3.25: $Norm(P_k^{TP})$ and $Norm(P_k^R)$ for MMT and OLSR using scenario Sc.10.Nodes when $Ti = 1s$ and $Sp_{avg} = 5m/s$ with Number of Hops k

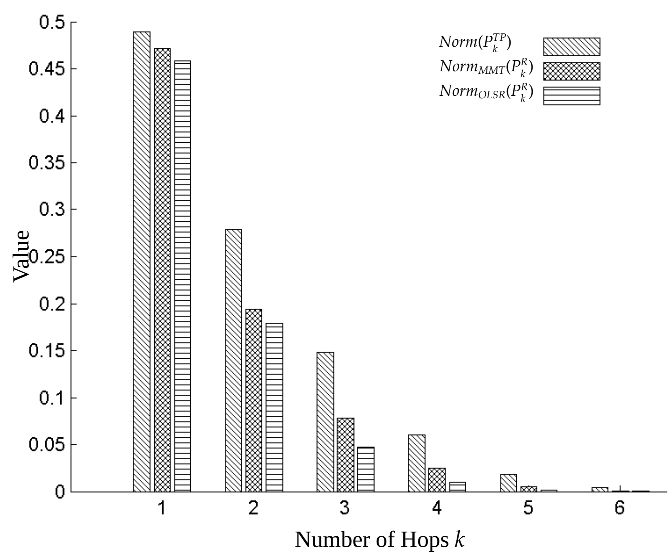

FIGURE 3.26: $Norm(P_k^{TP})$ and $Norm(P_k^R)$ for MMT and OLSR using scenario Sc.10.Nodes when $Ti = 1s$ and $Sp_{avg} = 20m/s$ with Number of Hops k

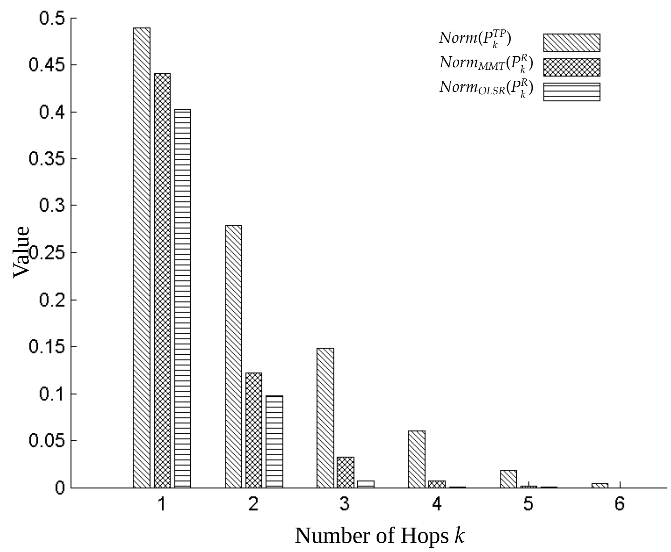

FIGURE 3.27: $Norm(P_k^{TP})$ and $Norm(P_k^R)$ for MMT and OLSR using scenario Sc.10.Nodes when $Ti = 3s$ and $Sp_{avg} = 20m/s$ with Number of Hops k

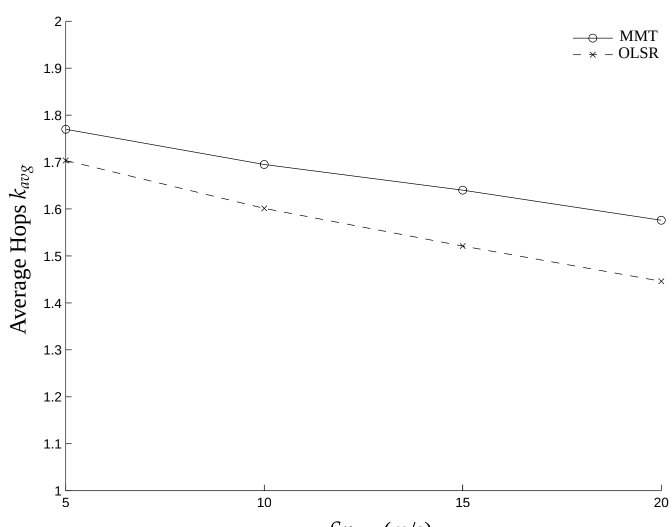

FIGURE 3.28: k_{avg} in scenario Sc.10.Nodes and Ti=1s with Sp_{avg}

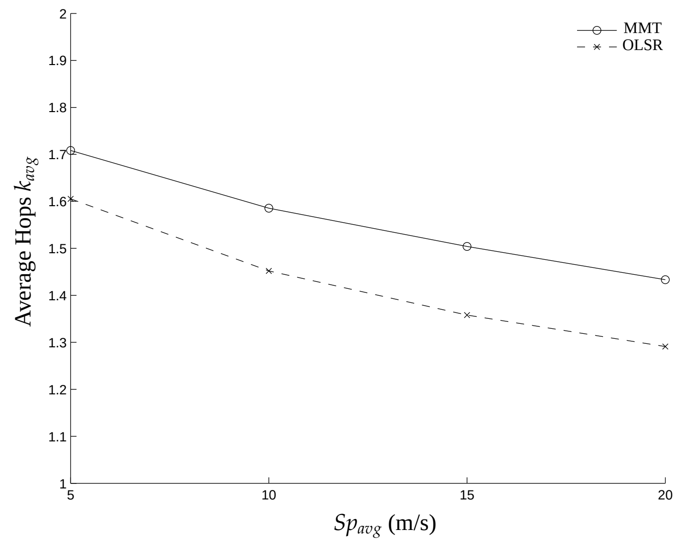

FIGURE 3.29: k_{avg} in scenario Sc.10.Nodes and Ti=2s with Sp_{avg}

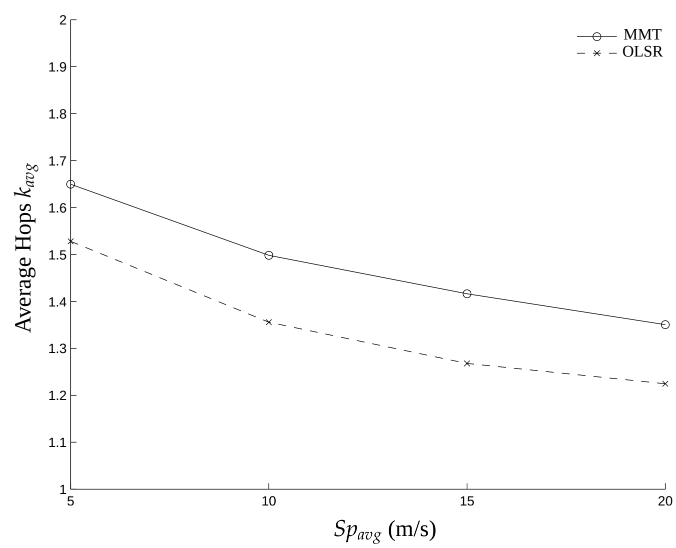

FIGURE 3.30: k_{avg} in scenario Sc.10.Nodes and Ti=3s with Sp_{avg}

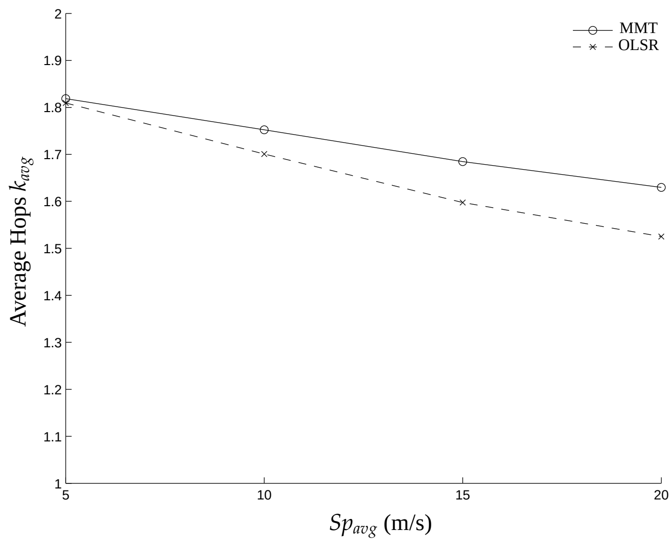

FIGURE 3.31: k_{avg} in scenario Sc.20.Nodes and Ti=1s with Sp_{avg}

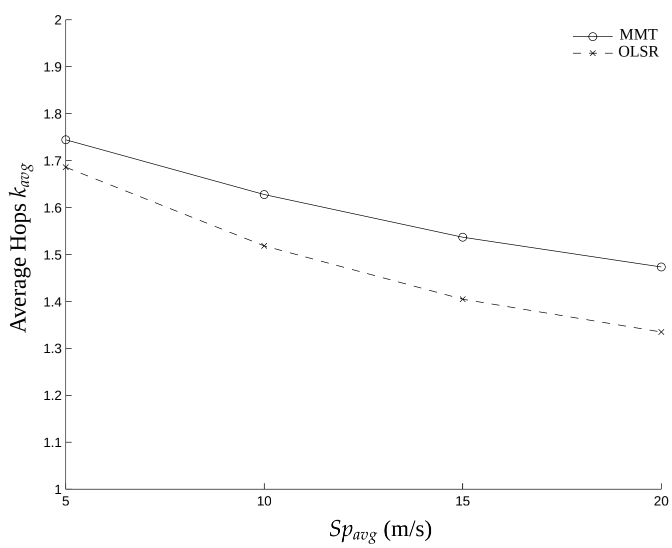

FIGURE 3.32: k_{avg} in scenario Sc.20.Nodes and Ti=2s with Sp_{avg}

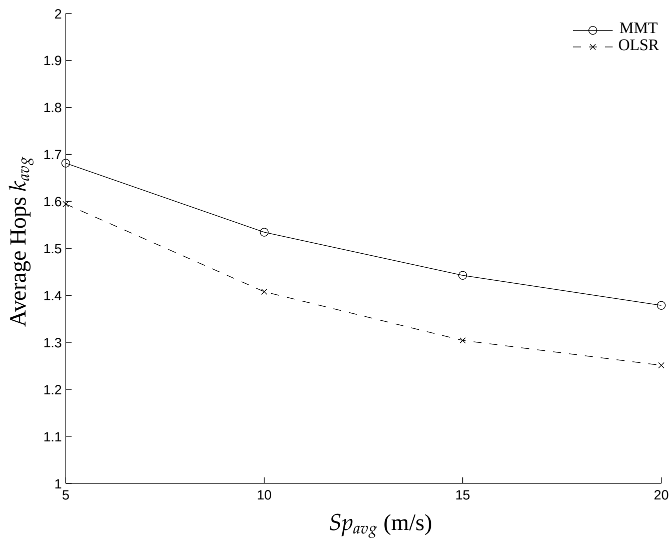

FIGURE 3.33: k_{avg} in scenario Sc.20.Nodes and Ti=3s with Sp_{avg}

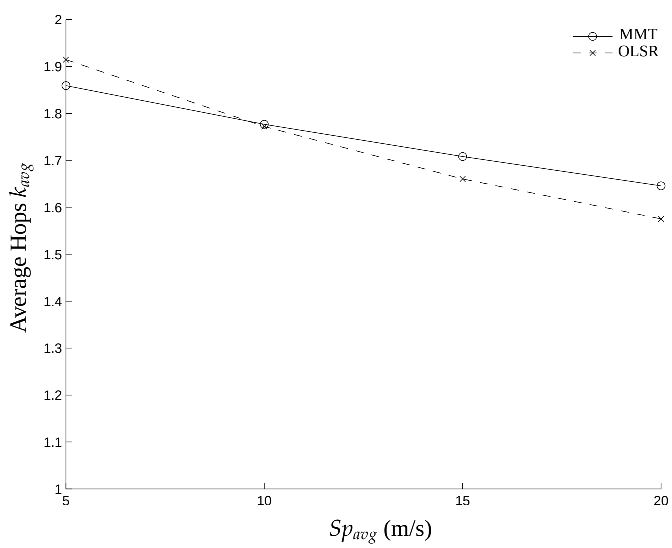

FIGURE 3.34: k_{avg} in scenario Sc.40.Nodes and Ti=1s with Sp_{avg}

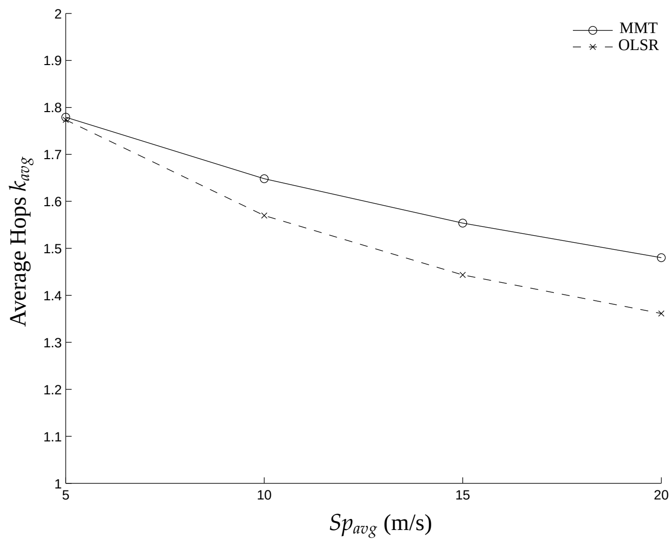

FIGURE 3.35: k_{avg} in scenario Sc.40.Nodes and Ti=2s with Sp_{avg}

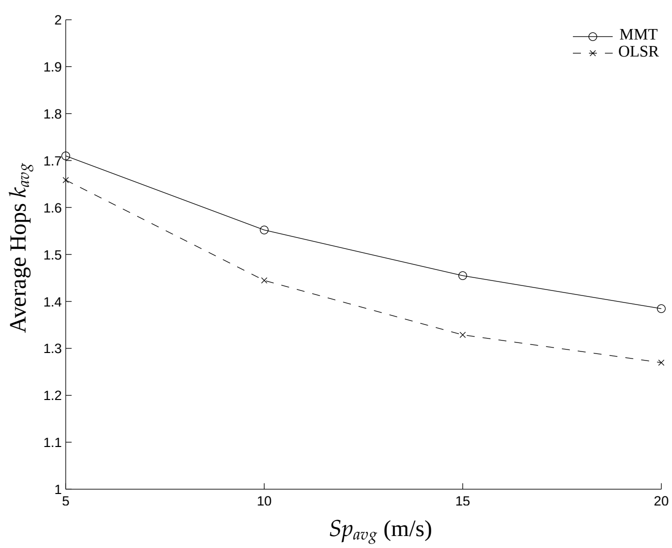

FIGURE 3.36: k_{avg} in scenario Sc.40.Nodes and Ti=3s with Sp_{avg}

MMT has several parameters controlling tree growth locally such as *maxChild* and *maxHop*. *maxChild* limits the number of 1-Hop *LLink*s with neighbors which is limited to 9. In case a node has more than 9 neighbors, some of the possible 1-Hop *TLink*s won't be built as 1-Hop *LLink*s. We refer to this case as *ChildrenSaturation*, which happens in MMT more frequently in large and dense networks, but doesn't exist in OLSR. Referring to Figure 3.37, we see the impact of *ChildrenSaturation* on MMT resulting in $Norm_{MMT}(P_1^R) \approx Norm_{OLSR}(P_1^R)$. At the same time, *maxHop* is set to 5; as a result, MMT won't build *LPath*s longer than 5 hops making $Norm_{MMT}(P_6^R) = 0$. These two reasons drive k_{avg} for OLSR higher than MMT as we see in Figure 3.34 when $Sp_{avg} = 5m/s$. As we increase Sp_{avg}, the impact of the previous 2 limitations diminishes as φ_k gets shorter and less usable making $Norm_{OLSR}(P_6^R) \approx 0$

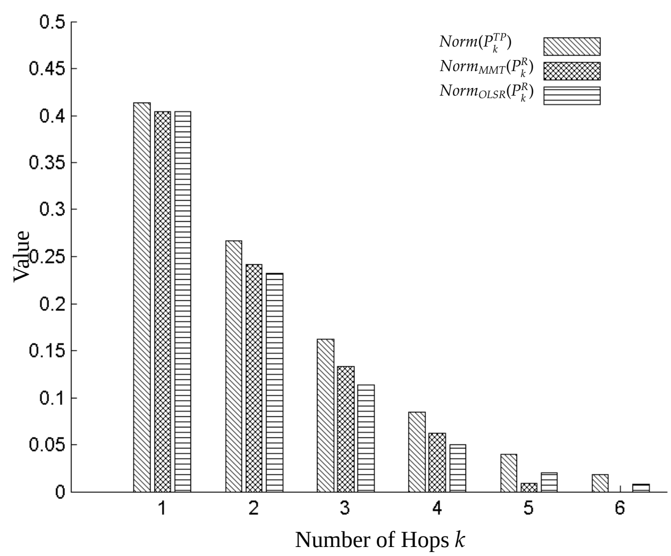

FIGURE 3.37: $Norm(P_k^{TP})$ and $Norm(P_k^R)$ for MMT and OLSR using Sc.40.Nodes scenario when $Ti = 1s$ and $Sp_{avg} = 5m/s$ with Number of Hops k

Chapter 4

Topological Modeling

This chapter attempts to answer the question of how long does a *TLink* and *TPath* between two nodes last for?

Models in this objective are based on probabilities and geometry. The difference of this objective from previous works in literature is that it derives comprehensive mathematical model, not based on empirical results, and without assuming extra hardware, such as GPS, to estimate velocities or have previous assumptions on node location. Since this objective is focusing on *Topological* modeling, we use TOPO protocol stack to verify its correctness. *Topological* Models are arranged in three sections shown next.

4.1 Modeling *TLink* Durations φ_1

This model assumes that D_{TX} is in the same order of magnitude as instantaneous speed of a node v which is uniformly distributed on $[Sp_{min}, Sp_{max}]$ according to the mobility model described in section 3.3. Hence, the probability that a node changes its direction while in range of another one is low. Let us assume that node A and root node R are moving with two velocities $\overrightarrow{V_A}$ and $\overrightarrow{V_R}$. By considering R fixed, A can be seen as moving with relative velocity of $\overrightarrow{V_r} = \overrightarrow{V_A} - \overrightarrow{V_R}$. This relative velocity has a magnitude of v_r. Let us also assume that A is travelling through the D_{TX} of R with cord length ℓ similar to Figure 4.1 which makes φ_1:

$$\varphi_1 = \frac{\ell}{v_r} \tag{4.1}$$

In (4.1), finding the *pdf* of φ_1, $f(\varphi_1)$, requires finding $f(\ell)$ and $f(v_r)$. To find $f(\ell)$, we refer again to Figure 4.1 where A is crossing the D_{TX} of R with cord length ℓ_0 which is $\Re_0$ away from R. Due to mobility model adopted in section 3.3, $\Re$ is a random variable uniformly distributed on $[0, D_{TX}]$. Hence its *pdf* is:

$$f(\Re) = \begin{cases} \frac{1}{D_{TX}} & 0 \leq \Re \leq D_{TX} \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

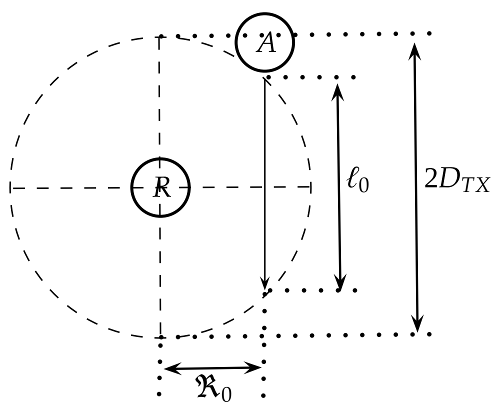

Figure 4.1: Link Duration Schematic

Then we write the cumulative density function *cdf* of $\Re$:

$$F(\Re) = \begin{cases} 0 & \Re < 0 \\ \frac{\Re}{D_{TX}} & 0 \leq \Re \leq D_{TX} \\ 1 & \Re > D_{TX} \end{cases} \tag{4.3}$$

Using Pythagoras theorem, $\Re$ can be rewritten as:

$$\mathfrak{R} = \sqrt{D_{TX}^2 - \left(\frac{\ell}{2}\right)^2} \tag{4.4}$$

Using (4.3) and (4.4), then *cdf* of ℓ:

$$\begin{aligned}
F(\ell_0) &= P[\ell \le \ell_0] = P\left[\left(\frac{\ell}{2}\right)^2 \le \left(\frac{\ell_0}{2}\right)^2\right] \\
&= P\left[D_{TX}^2 - \mathfrak{R}^2 \le \left(\frac{\ell_0}{2}\right)^2\right] = P\left[D_{TX}^2 - \left(\frac{\ell_0}{2}\right)^2 \le \mathfrak{R}^2\right] \\
&= P[\mathfrak{R}_0^2 \le \mathfrak{R}^2] = P[\mathfrak{R}_0 \le \mathfrak{R}] \\
&= 1 - F(\mathfrak{R}_0)
\end{aligned} \tag{4.5}$$

$$F(\ell) = 1 - \frac{\mathfrak{R}}{D_{TX}} = \begin{cases} 0 & \ell < 0 \\ 1 - \sqrt{1 - \left(\frac{\ell}{2D_{TX}}\right)^2} & 0 < \ell < 2D_{TX} \\ 1 & \ell > 2D_{TX} \end{cases} \tag{4.6}$$

Taking the derivative we get $f(\ell)$, which is the first required *pdf*:

$$f(\ell) = \frac{\mathrm{d}F(\ell)}{\mathrm{d}\ell} = \begin{cases} \frac{\ell}{2D_{TX}\sqrt{4D_{TX}^2 - \ell^2}} & 0 < \ell < 2D_{TX} \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

In Figure 4.2, we show $f(\ell)$ with $D_{TX} \in \{100m, 150m, 200m\}$.

To find $f(v_r)$, let us assume that the angle between the two velocities $\overrightarrow{V_R}$ and $\overrightarrow{V_A}$ is θ_r, which is a random variable uniformly distributed on $[0, \pi]$ [1] according to mobility model adopted in section 3.3:

[1]When the angle between the two velocities $\overrightarrow{V_R}$ and $\overrightarrow{V_A} > \pi$, it can be viewed from another perspective and measured to be $< \pi$. As a result, it is appropriate to consider $\theta_r \in [0, \pi]$

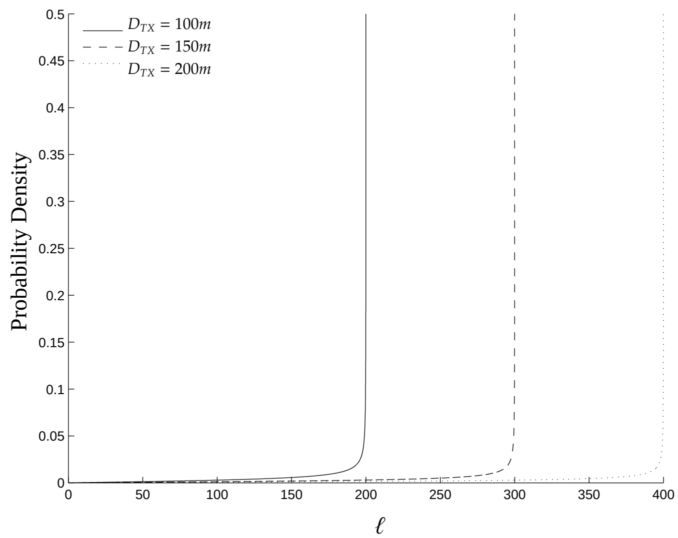

FIGURE 4.2: Plot of $f(\ell)$ as D_{TX} changes

$$f(\theta_r) = \begin{cases} \frac{1}{\pi} & 0 \leq \theta_r \leq \pi \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

Using the law of cosines, relative speed is given by:

$$v_r^2 = v_R^2 + v_A^2 - 2v_R v_A \cos(\theta_r) \tag{4.9}$$

Assuming we know v_R and v_A, then:

$$\theta_r \mid_{v_R, v_A} = \cos^{-1}\left(\frac{v_R^2 + v_A^2 - v_r^2}{2v_R v_A}\right) \tag{4.10}$$

Referring to Figure 4.3, we notice that decreasing θ_r increases $\cos(\theta)$. At the same time, in (4.9), we see that increasing increases $\cos(\theta)$ decreases v_r. As a result, we can write:

$$
\begin{aligned}
F(v_{r_0}\ |_{v_R,v_A}) &= P[v_r \leq v_{r_0}\ |_{v_R,v_A}] \\
&= P[v_R^2 + v_A^2 - 2v_R v_A \cos(\theta_r) \leq v_{r_0}^2\ |_{v_R,v_A}] \\
&= P[2v_R v_A \cos(\theta_r) \geq v_R^2 + v_A^2 - v_{r_0}^2\ |_{v_R,v_A}] \\
&= P[\cos(\theta_r) \geq \frac{v_R^2 + v_A^2 - v_{r_0}^2}{2v_R v_A}\ |_{v_R,v_A}] \\
&= P[\cos(\theta_r) \geq \cos(\theta_{r_0})\ |_{v_R,v_A}] \\
&= P[\theta_r \leq \theta_{r_0}\ |_{v_R,v_A}] \\
&= \int_0^{\theta_{r_0}} \frac{1}{\pi} \mathrm{d}\theta_r = \frac{\theta_{r_0}}{\pi}
\end{aligned}
\tag{4.11}
$$

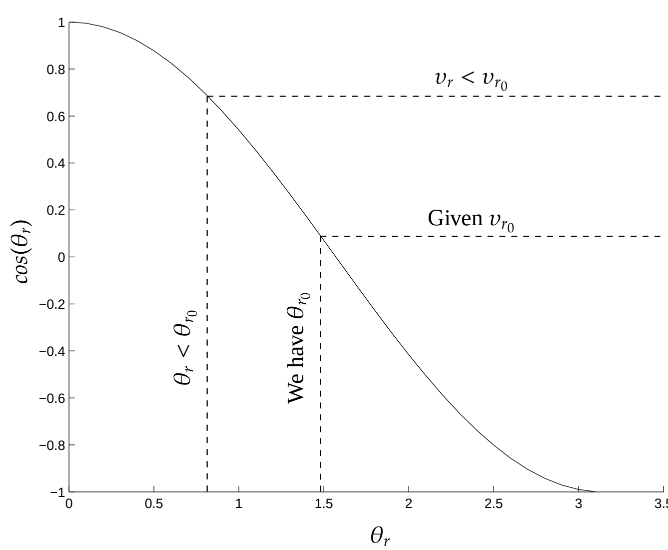

FIGURE 4.3: Plot of $cos(\theta_r)$

Using (4.10), we get $F(v_r\ |_{v_R,v_A})$:

$$F(v_r\mid_{v_R,v_A}) = \frac{\theta_r}{\pi} = \begin{cases} 0 & v_r\mid_{v_R,v_A} < |v_R - v_A| \\ \frac{1}{\pi}\cos^{-1}\left(\frac{v_R^2+v_A^2-v_r^2}{2v_Rv_A}\right) & |v_R - v_A| \leq v_r\mid_{v_R,v_A} \leq v_R + v_A \\ 1 & v_r\mid_{v_R,v_A} > v_R + v_A \end{cases} \tag{4.12}$$

Taking its derivative, we get $f(v_r\mid_{v_R,v_A})$:

$$\begin{aligned} f(v_r\mid_{v_R,v_A}) &= \frac{\mathrm{d}F(v_r\mid_{v_R,v_A})}{\mathrm{d}v_r} \\ &= \begin{cases} \frac{v_r}{\pi v_R v_A\sqrt{1-\left(\frac{v_R^2+v_A^2-v_r^2}{2v_Rv_A}\right)^2}} & |v_R - v_A| \leq v_r\mid_{v_R,v_A} \leq v_R + v_A \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{4.13}$$

Finally, we get the $f(v_r)$, the second required *pdf*:

$$f(v_r) = \begin{cases} \int\limits_{Sp_{min}}^{Sp_{max}}\int\limits_{Sp_{min}}^{Sp_{max}} f(v_r\mid_{v_R,v_A})f(v_R)f(v_A)\mathrm{d}v_R\mathrm{d}v_A & \begin{array}{l} 0 \leq v_r \leq 2Sp_{max} \\ |v_R - v_A| \leq v_r\mid_{v_R,v_A} \leq v_R + v_A \end{array} \\ 0 & \text{otherwise} \end{cases} \tag{4.14}$$

Using numerical evaluation, Figure 4.4 depicts $f(v_r)$ as $[Sp_{min}, Sp_{max}] \in \{[4,6]m/s, [9,11]m/s, [14,16]m/s, [19,21]m/s\}$.

In Figure 4.4, we notice that the maximum probability of v_r occurs around $2Sp_{avg}$ unlike the assumption made by the authors in [7, 8], we are agreeing with the discussion presented in [52]. Here we observe that the probability of two nodes having heads on encounter happens with higher probability than any other. As the two nodes encounter each other at higher speed (around $2Sp_{avg}$) they have more chance to encounter each other again or other nodes in the future, thus increasing the occurrences of high v_r.

To find $f(\varphi_1)$, we need to find the joint probability of the two random variables ℓ and v_r, with their *pdf*s derived in (4.7) and (4.14) respectively. Figure 4.5 shows the Cartesian

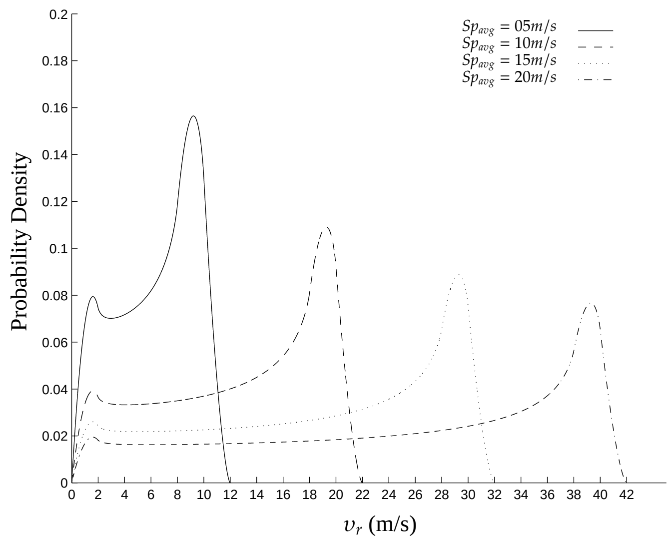

FIGURE 4.4: Plot of $f(v_r)$ with Sp_{avg}

field with x and y axis renamed to the independent random variables v_r and ℓ. Also, let us assume that the field is occupied by a surface resulting from the product of the two *pdf*s, $f(v_r)$ and $f(\ell)$. The Line $\varphi_{1_0} = \frac{\ell}{v_r}$ passes through all the values of ℓ and v_r which result in φ_{1_0}. The other line represent all the possible values of ℓ and v_r which produce $\varphi_1 < \varphi_{1_0}$. As a result, to find $F(\varphi_{1_0})$ one can integrate the the product of the two *pdf*s, $f(v_r)$ and $f(\ell)$ over the area below the line $\varphi_{1_0} = \frac{\ell}{v_r}$:

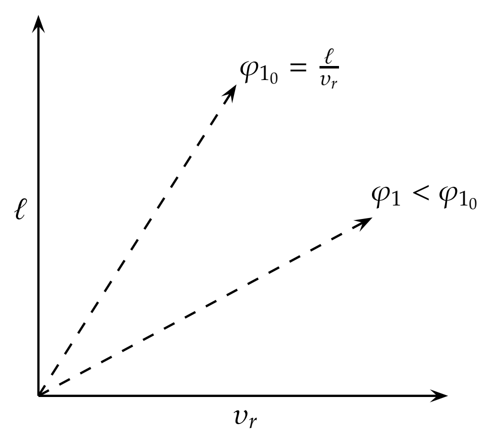

FIGURE 4.5: Joint Probability of v_r and ℓ

$$F(\varphi_{1_0}) = P[\varphi_1 \leq \varphi_{1_0}] = \int_0^{2Sp_{max}} \int_0^{v_r \varphi_{1_0}} f(v_r) f(\ell) \mathrm{d}\ell \mathrm{d}v_r \quad v_r \varphi_{1_0} < 2D_{TX} \tag{4.15}$$

$$F(\varphi_1) = \begin{cases} 0 & \varphi_1 < 0 \\ \int_0^{2Sp_{max}} \int_0^{v_r \varphi_1} f(v_r) f(\ell) \mathrm{d}\ell \mathrm{d}v_r & 0 \leq \varphi_1 \; \& \; v_r \varphi_1 < 2D_{TX} \end{cases} \tag{4.16}$$

Figure 4.6 shows $F(\varphi_1)$ when $D_{TX} = 200m$ and $[Sp_{min}, Sp_{max}] \in \{[4,6]m/s, [9,11]m/s, [14,16]m/s, [19,21]m/s\}$. We notice that $F(\varphi_1)$ approaches probability of 1 faster as we increase Sp_{avg}. Indeed, this is because the probability of shorter φ_1 increases with speed. Taking the derivatives, we get $f(\varphi_1)$ as depicted in Figure 4.7.

Simulation results are collected by running the scenarios in 3.4 using the field and mobility model described in section 3.3 and simulation parameters in Table 3.6. Simulation results were produced as detailed in section 3.5.1. Overlaying modeling and simulation results is shown in Figures 4.8, 4.9, 4.10 and 4.11 where we notice that our models tightly agrees with simulation results.

Referring to Figure 4.2, we notice that the maximum $f(\ell)$ happens around $2D_{TX}$. In addition, we observe that $f(v_r)$ in Figure 4.4 has a maximum probability density around $2Sp_{avg}$. Using (4.1), we expect φ_1 to occur with maximum probability around $\frac{2D_{TX}}{2Sp_{avg}}$ which agrees with presentations in Figures 4.8 through 4.11 and was presented in (3.5).

In addition, we notice that in the range when $\varphi_1 < \varphi_{1_{maxProb}}$, φ_1 from simulation have higher probability density than model counterparts. On the other range, when $\varphi_1 > \varphi_{1_{maxProb}}$, φ_1 from simulation have lower probability density than model. The reason is that our model allows for very long φ_1 where nodes can have v_r extremely small for very long times, which is not possible in simulation as it is time bounded. Another reason is that the mobility model adopted allows nodes to change speed and direction while in D_{TX} range of each other making the probability of longer φ_1 lower.

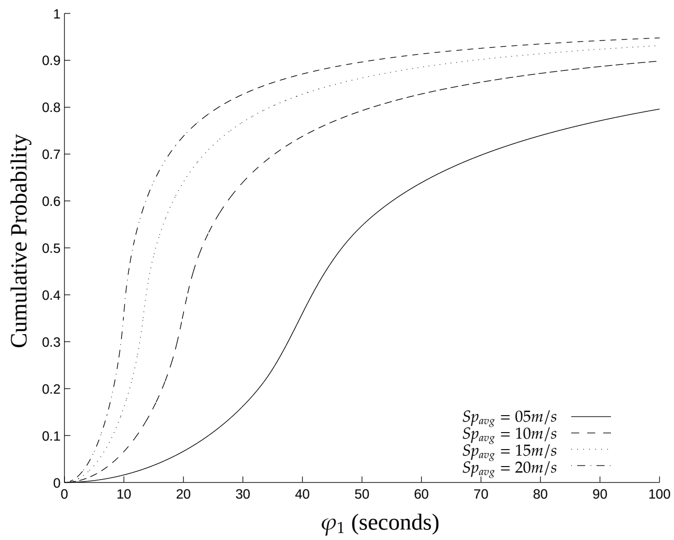

FIGURE 4.6: Model of $F(\varphi_1)$ with Sp_{avg} and $D_{TX} = 200m$

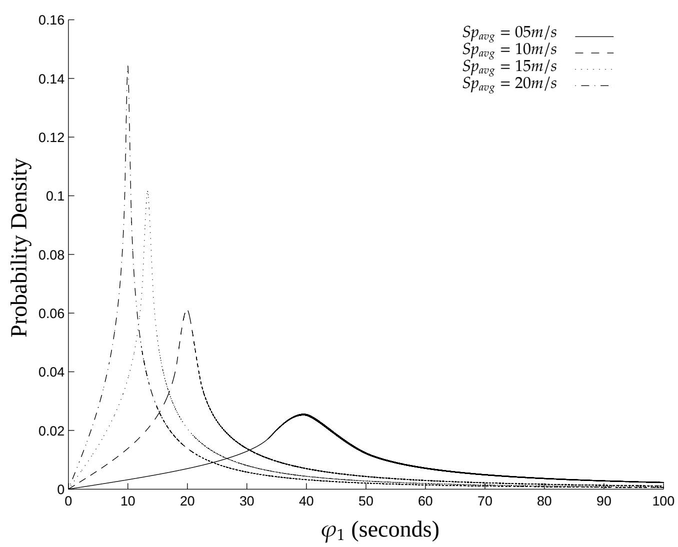

FIGURE 4.7: Model of $f(\varphi_1)$ with Sp_{avg} and $D_{TX} = 200m$

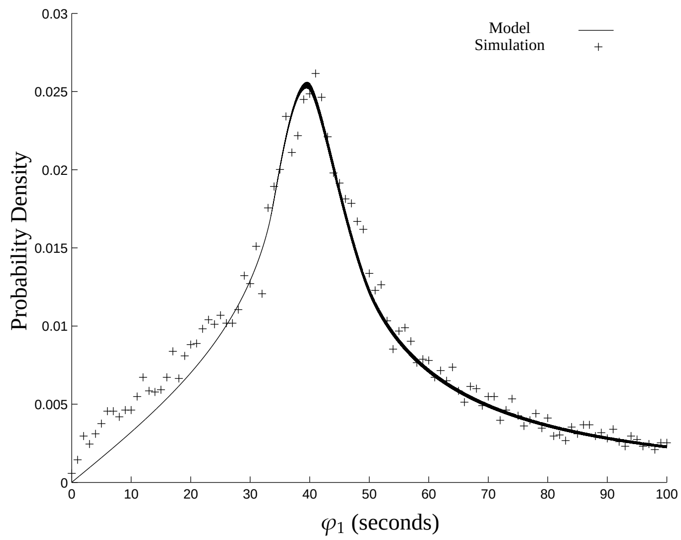

FIGURE 4.8: Model vs Simulation of $f(\varphi_1)$ with $Sp_{avg} = 05m/s$ and $D_{TX} = 200m$

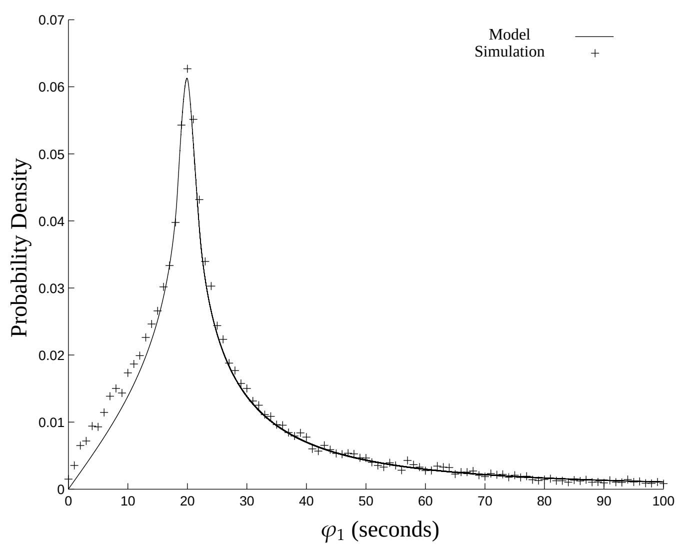

FIGURE 4.9: Model vs Simulation of $f(\varphi_1)$ with $Sp_{avg} = 10m/s$ and $D_{TX} = 200m$

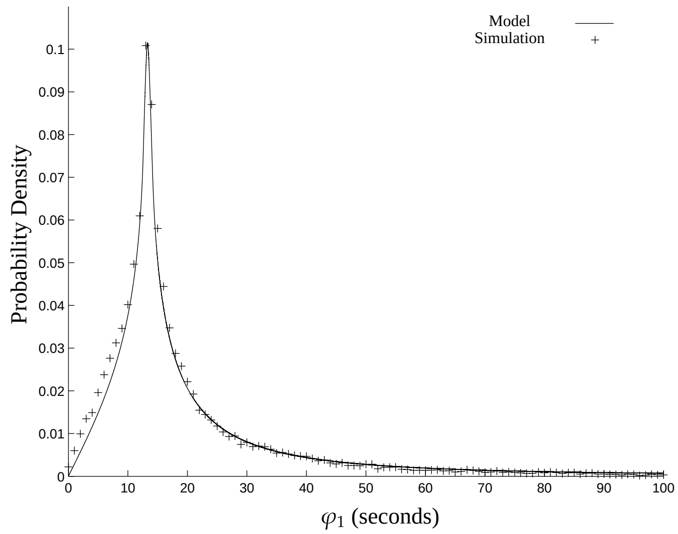

FIGURE 4.10: Model vs Simulation of $f(\varphi_1)$ with $Sp_{avg} = 15m/s$ and $D_{TX} = 200m$

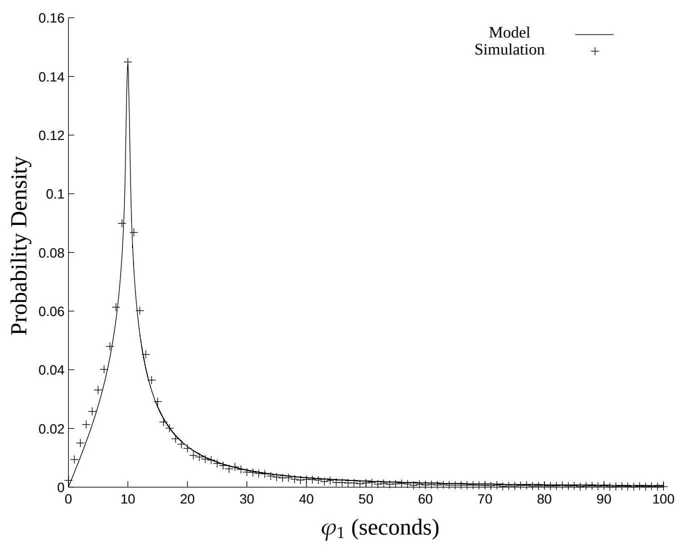

FIGURE 4.11: Model vs Simulation of $f(\varphi_1)$ with $Sp_{avg} = 20m/s$ and $D_{TX} = 200m$

4.2 Modeling *TPath* Durations φ_k

A *TPath* is formed when the last one of its *TLink*s is just formed. On the other hand, the rest of *TLink*s existed for a fraction of their φ_1 making their remaining duration found by multiplying δ by φ_1. δ is a random variable with *pdf* found empirically [2]. For a path with $k+1$ nodes and k links or hops, we will refer to the duration of the *mth TLink* as φ_1^m. Then, *TPath* duration, φ_k, can be calculated by taking the minimum of the duration of the last *TLink* which was formed and the remaining duration of the rest of $k-1$ *TLinks*:

$$\varphi_k = min\left\{\varphi_1^1, \delta\times\varphi_1^2, \delta\times\varphi_1^3, ..., \delta\times\varphi_1^k\right\} \tag{4.17}$$

Note that this model assumes that duration of *TLink*s are independent of each other which was proven valid [3].

4.2.1 Modeling $f(\varphi_2)$

To find $f(\varphi_2)$, we generate two arrays [3] of *TLink* durations following the *cdf* formulated in 4.16. Referring to (4.17), the durations of second array are modified by multiplying with δ which is found empirically to be uniformly distributed on $[0, 1.3]$. Figure 4.12 shows the resulting $f(\varphi_2)$ with various Sp_{avg}. Simulation results are collected by running the scenarios in 3.4 using the field and mobility model described in section 3.3 and simulation parameters in Table 3.6. Overlaying model and simulation results are shown in Figures 4.13, 4.14, 4.15 and 4.16.

[2]Finding the model of δ is only possible when considering the past of nodes' spacial locations and velocities which is outside the scope of this work

[3]This is due to the fact that the model of φ_k in (4.17) depends on finding δ from empirical results

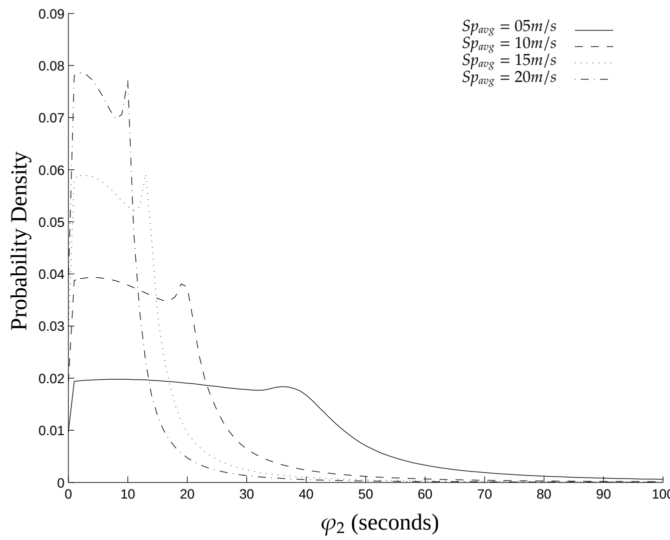

FIGURE 4.12: Model of $f(\varphi_2)$ with Sp_{avg} and $D_{TX} = 200m$

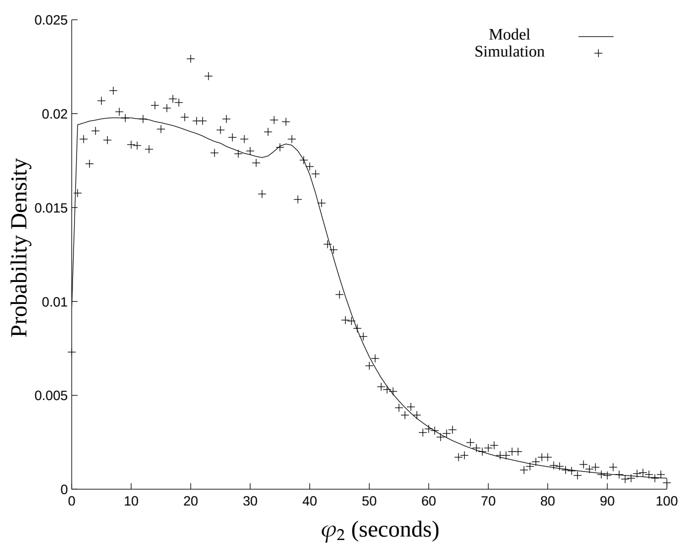

FIGURE 4.13: Model vs Simulation of $f(\varphi_2)$ with $Sp_{avg} = 05m/s$ and $D_{TX} = 200m$

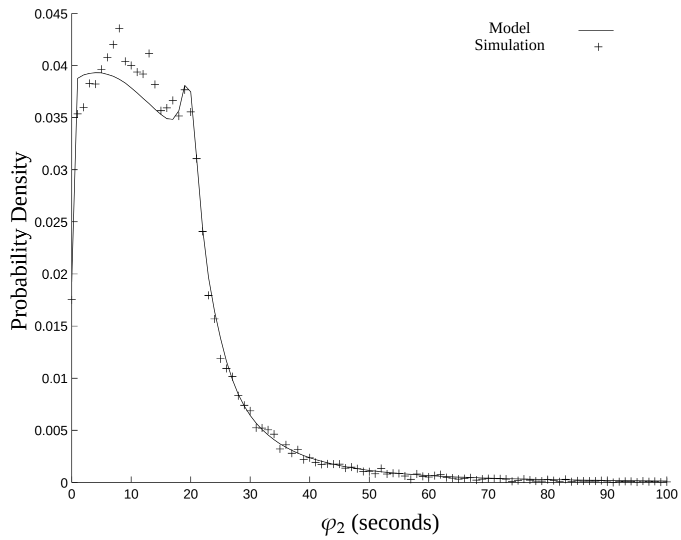

FIGURE 4.14: Model vs Simulation of $f(\varphi_2)$ with $Sp_{avg} = 10m/s$ and $D_{TX} = 200m$

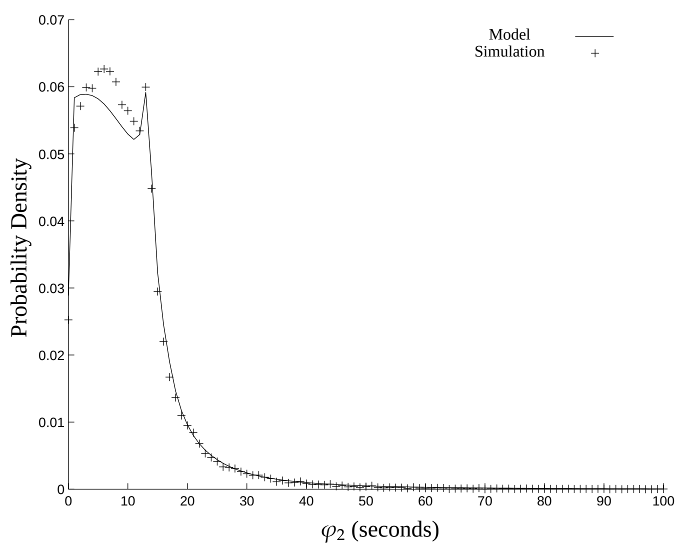

FIGURE 4.15: Model vs Simulation of $f(\varphi_2)$ with $Sp_{avg} = 15m/s$ and $D_{TX} = 200m$

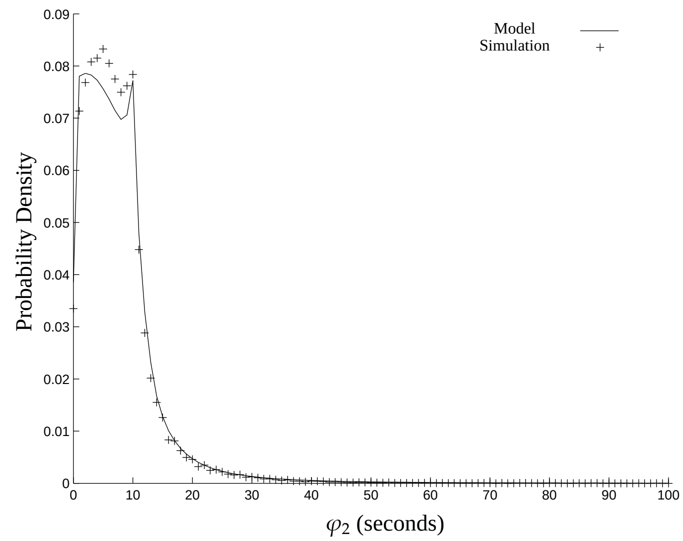

FIGURE 4.16: Model vs Simulation of $f(\varphi_2)$ with $Sp_{avg} = 20m/s$ and $D_{TX} = 200m$

4.2.2 Modeling $f(\varphi_3)$

Similarly, to find $f(\varphi_3)$, we generate three arrays of *TLink* durations following the *CDF* shown in 4.16. Referring to (4.17), the durations of second and third arrays are modified by multiplying with δ which is found to be uniformly distributed on $[0, 1.3]$. Figure 4.17 shows the resulting $f(\varphi_3)$ with various Sp_{avg}. Simulation results are collected by running the scenarios in 3.4 using the field and mobility model described in section 3.3 and simulation parameters in Table 3.6. Overlaying modeling and simulation results are shown in Figures 4.18, 4.19, 4.20 and 4.21.

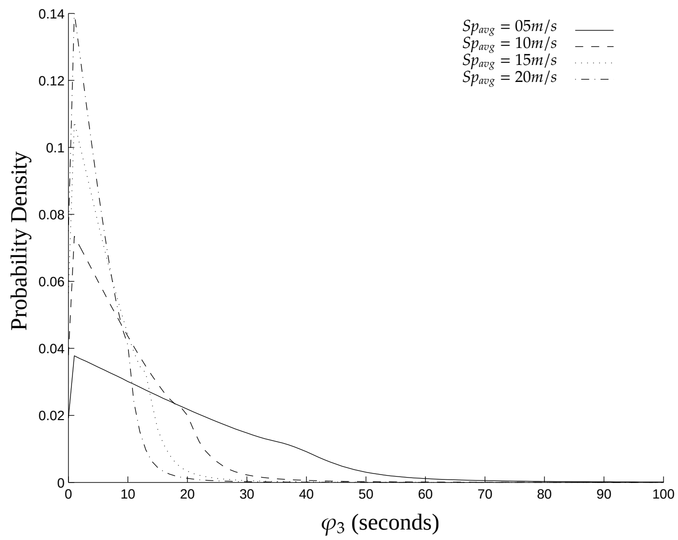

FIGURE 4.17: Model of $f(\varphi_3)$ with Sp_{avg} and D_{TX} = 200m

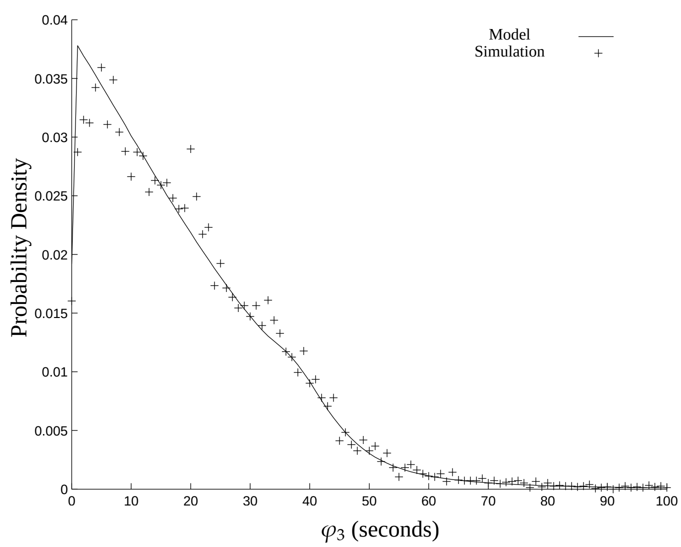

FIGURE 4.18: Model vs Simulation of $f(\varphi_3)$ with Sp_{avg} = 05m/s and D_{TX} = 200m

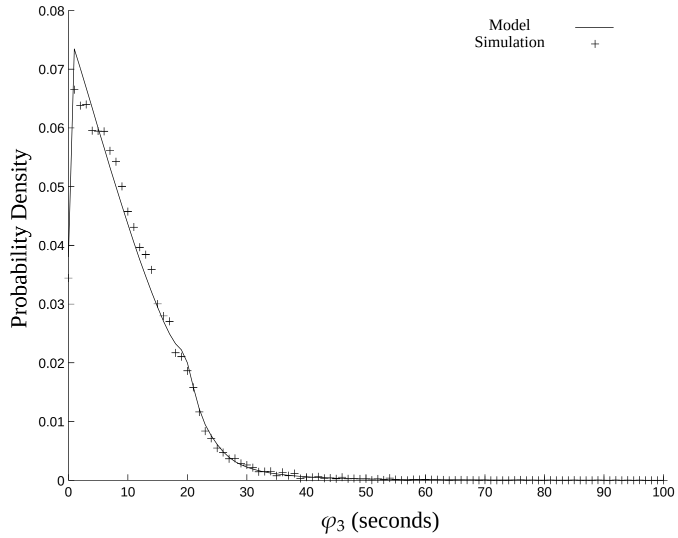

FIGURE 4.19: Model vs Simulation of $f(\varphi_3)$ with $Sp_{avg} = 10m/s$ and $D_{TX} = 200m$

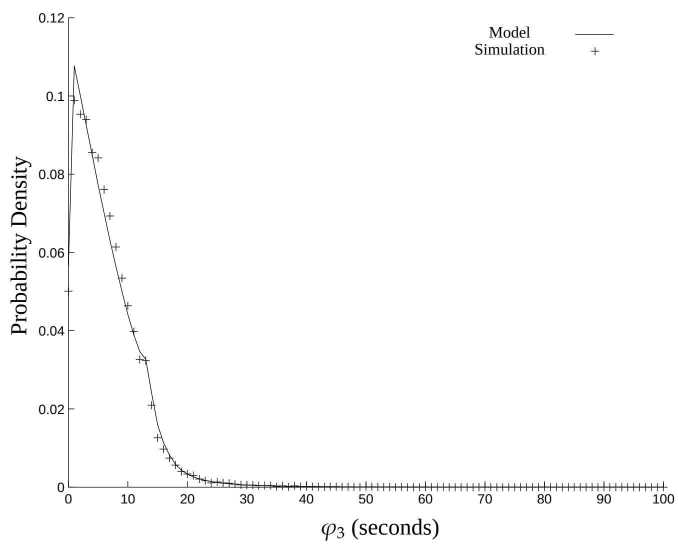

FIGURE 4.20: Model vs Simulation of $f(\varphi_3)$ with $Sp_{avg} = 15m/s$ and $D_{TX} = 200m$

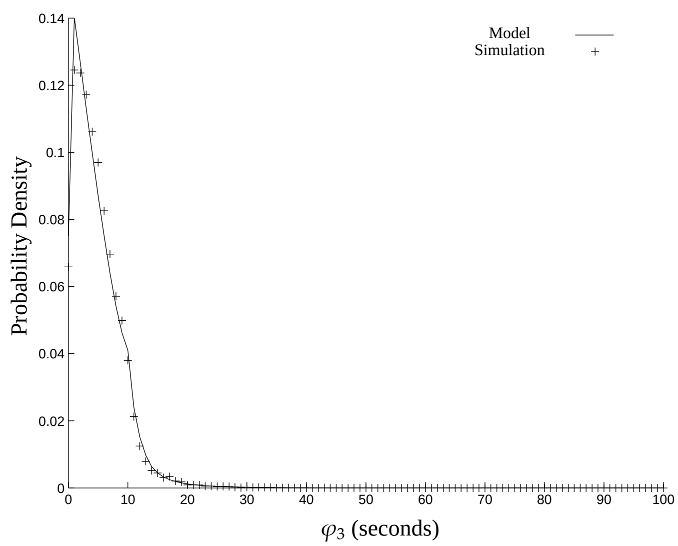

FIGURE 4.21: Model vs Simulation of $f(\varphi_3)$ with $Sp_{avg} = 20m/s$ and $D_{TX} = 200m$

Chapter 5

Adaptability Modeling

This chapter focuses on modeling the behavior of a protocol stack in adapting to topology changes. As mentioned earlier, a change in topology takes some time, *AdaptationDelay*, to take effect at the routing layer logical information. This interaction occurs between the network's topology (ground truth) and the routing layer; hence we use the protocol stacks OLSRI and MMTI. Analyzing this interaction is not found in literature because the impact of a topology change is assumed to have an instantaneous impact on logical information. This objective will answer the question of what is the *AdaptationDelay* of a routing protocol when creating *LPath* information after *TPath* is formed?

5.1 Modeling ξ_k^{in}

In addition to the time period of updating topology information, *Ti*, another factor involved in determining *AdaptationDelays* for building *LPath*s is the nature of the routing algorithm and the routing protocol implementation details. For example, centralized routing protocols depend on gathering logical link information from distant nodes resulting in longer delays than distributed protocols where routing decisions are executed based on locally available information. We design a collection of scenarios in order to simplify the analysis and factors involved in *AdaptationDelays*.

5.1.1 Core Probability Formulation

The modeling of *AdaptationDelays* requires the computation of probabilities for time durations between different intervals. These probabilities are expressed as several inequalities involving several time instances τ_a, τ_b, τ_c, τ_d, τ_e and τ_x each of which follows identical independent distributions (i.i.d) of uniform distribution on $[0, Ti]$, $\sim U(0, Ti)$. This assumption was based on the uniformly distributed random timers adopted by most routing protocols, specifically MMT and OLSR, in order to send and update logical information. Then, we find the probabilities of the following inequalities where each formulation is given an ID shown at the left hand side. In (5.1), we are computing the probability that τ_a is less than an τ_x; meanwhile, (5.2) is calculating the complement of that probability.

$$P_{2A}(\tau_x) = P[\tau_a < \tau_x] = \int_0^{\tau_x} \frac{1}{Ti} \mathrm{d}\tau_a = \frac{\tau_x}{Ti} \tag{5.1}$$

$$P_{2B}(\tau_x) = P[\tau_a > \tau_x] = \int_{\tau_x}^{Ti} \frac{1}{Ti} \mathrm{d}\tau_a = 1 - \frac{\tau_x}{Ti} \tag{5.2}$$

The formula in (5.3) is representing the probability that τ_a is less than τ_b which in turn is less than τ_x:

$$P_{3A}(\tau_x) = P[\tau_a < \tau_b < \tau_x] = \int_0^{\tau_x} \int_0^{\tau_b} \frac{1}{Ti^2} \mathrm{d}\tau_a \mathrm{d}\tau_b = \frac{\tau_x^2}{2Ti^2} \tag{5.3}$$

In (5.4), we show the probability that τ_a is less than τ_x while τ_b is larger than τ_x:

$$P_{3B}(\tau_x) = P[\tau_a < \tau_x < \tau_b] = \int_0^{\tau_x} \frac{1}{Ti} \mathrm{d}\tau_a \times \int_{\tau_x}^{Ti} \frac{1}{Ti} \mathrm{d}\tau_b = \frac{\tau_x}{Ti} - \frac{\tau_x^2}{Ti^2} \tag{5.4}$$

The probability that τ_b is larger than τ_a which in turn is larger than τ_x is calculated next:

$$P_{3C}(\tau_x) = P[\tau_x < \tau_a < \tau_b] = \int_{\tau_x}^{Ti} \int_{\tau_a}^{Ti} \frac{1}{Ti^2} \mathrm{d}\tau_b \mathrm{d}\tau_a = \frac{1}{2} - \frac{\tau_x}{Ti} + \frac{\tau_x^2}{2Ti^2} \tag{5.5}$$

Next, we calculate the probability that τ_a is less than τ_b which in turn is less that τ_c which is less than τ_x:

$$P_{4A}(\tau_x) : P[\tau_a < \tau_b < \tau_c < \tau_x] = \int_0^{\tau_x} \int_0^{\tau_c} \int_0^{\tau_b} \frac{1}{Ti^3} \mathrm{d}\tau_a \mathrm{d}\tau_b \mathrm{d}\tau_c = \frac{\tau_x^3}{6Ti^3} \tag{5.6}$$

In the following formulas (5.7) through (5.9), we show the probability of different orders involving τ_a, τ_b, τ_c and τ_X:

$$P_{4B}(\tau_x) = P[\tau_a < \tau_b < \tau_x < \tau_c] = \int_0^{\tau_x} \int_0^{\tau_b} \frac{1}{Ti^2} \mathrm{d}\tau_a \mathrm{d}\tau_b \times \int_{\tau_x}^{Ti} \frac{1}{Ti} \mathrm{d}\tau_c = \frac{\tau_x^2}{2Ti^2} - \frac{\tau_x^3}{2Ti^3} \tag{5.7}$$

$$\begin{aligned} P_{4C}(\tau_x) = P[\tau_a < \tau_x < \tau_b < \tau_c] &= \int_0^{\tau_x} \frac{1}{Ti} \mathrm{d}\tau_a \times \int_{\tau_x}^{Ti} \int_{\tau_b}^{Ti} \frac{1}{Ti^2} \mathrm{d}\tau_c \mathrm{d}\tau_b \\ &= \frac{\tau_x}{2Ti} - \frac{\tau_x^2}{Ti^2} + \frac{\tau_x^3}{2Ti^3} \end{aligned} \tag{5.8}$$

$$P_{4D}(\tau_x) = P[\tau_x < \tau_a < \tau_b < \tau_c] = \int_{\tau_x}^{Ti}\int_{\tau_a}^{Ti}\int_{\tau_b}^{Ti} \frac{1}{Ti^3}\mathrm{d}\tau_c\mathrm{d}\tau_b\mathrm{d}\tau_a$$
$$= \frac{1}{6} - \frac{\tau_x}{2Ti} + \frac{\tau_x^2}{2Ti^2} - \frac{\tau_x^3}{6Ti^3} \quad (5.9)$$

Formulas (5.10) through (5.14) shows the probabilities for different orderings of five random variables τ_a, τ_b, τ_c, τ_d and τ_x:

$$P_{5A}(\tau_x) = P[\tau_a < \tau_b < \tau_c < \tau_d < \tau_x] = \int_0^{\tau_x}\int_0^{\tau_d}\int_0^{\tau_c}\int_0^{\tau_b} \frac{1}{Ti^4}\mathrm{d}\tau_a\mathrm{d}\tau_b\mathrm{d}\tau_c\mathrm{d}\tau_d = \frac{\tau_x^4}{24Ti^4} \quad (5.10)$$

$$P_{5B}(\tau_x) = P[\tau_a < \tau_b < \tau_c < \tau_x < \tau_d] = \int_0^{\tau_x}\int_0^{\tau_c}\int_0^{\tau_b} \frac{1}{Ti^3}\mathrm{d}\tau_a\mathrm{d}\tau_b\mathrm{d}\tau_c \times \int_{\tau_x}^{Ti} \frac{1}{Ti}\mathrm{d}\tau_d$$
$$= \frac{\tau_x^3}{6Ti^3} - \frac{\tau_x^4}{6Ti^4} \quad (5.11)$$

$$P_{5C}(\tau_x) = P[\tau_a < \tau_b < \tau_x < \tau_c < \tau_d] = \int_0^{\tau_x}\int_0^{\tau_b} \frac{1}{Ti^2}\mathrm{d}\tau_a\mathrm{d}\tau_b \times \int_{\tau_x}^{Ti}\int_{\tau_c}^{Ti} \frac{1}{Ti^2}\mathrm{d}\tau_d\mathrm{d}\tau_c$$
$$= \frac{\tau_x^2}{4Ti^2} - \frac{\tau_x^3}{Ti^3} + \frac{\tau_x^4}{4Ti^4} \quad (5.12)$$

$$P_{5D}(\tau_x) = P[\tau_a < \tau_x < \tau_b < \tau_c < \tau_d] = \int_0^{\tau_x} \frac{1}{Ti}\mathrm{d}\tau_a \times \int_{\tau_x}^{Ti}\int_{\tau_b}^{Ti}\int_{\tau_c}^{Ti} \frac{1}{Ti^3}\mathrm{d}\tau_d\mathrm{d}\tau_c\mathrm{d}\tau_b$$
$$= \frac{\tau_x}{6Ti} - \frac{\tau_x^2}{2Ti^2} + \frac{\tau_x^3}{2Ti^3} - \frac{\tau_x^4}{6Ti^4} \quad (5.13)$$

$$P_{5E}(\tau_x) = P[\tau_x < \tau_a < \tau_b < \tau_c < \tau_d] = \int_{\tau_x}^{Ti}\int_{\tau_a}^{Ti}\int_{\tau_b}^{Ti}\int_{\tau_c}^{Ti} \frac{1}{Ti^4} \mathrm{d}\tau_d \mathrm{d}\tau_c \mathrm{d}\tau_b \mathrm{d}\tau_a$$
$$= \frac{1}{24} - \frac{\tau_x}{6Ti} + \frac{\tau_x^2}{4Ti^2} - \frac{\tau_x^3}{6Ti^3} + \frac{\tau_x^4}{24Ti^4} \quad (5.14)$$

In the next two formulas, we consider the existence of six random variables τ_a, τ_b, τ_c, τ_d, τ_e and τ_x and only calculate two possible orders of $\tau_a < \tau_b < \tau_c < \tau_d < \tau_e < \tau_x$ and $\tau_a < \tau_b < \tau_c < \tau_d < \tau_x < \tau_e$, as we will see later that remaining orders are not required for the modeling of *AdaptationDelays*:

$$P_{6A}(\tau_x) = P[\tau_a < \tau_b < \tau_c < \tau_d < \tau_e < \tau_x] = \int_0^{\tau_x}\int_0^{\tau_e}\int_0^{\tau_d}\int_0^{\tau_c}\int_0^{\tau_b} \frac{1}{Ti^5} \mathrm{d}\tau_a \mathrm{d}\tau_b \mathrm{d}\tau_c \mathrm{d}\tau_d \mathrm{d}\tau_e$$
$$= \frac{\tau_x^5}{120Ti^5} \quad (5.15)$$

$$P_{6B}(\tau_x) = P[\tau_a < \tau_b < \tau_c < \tau_d < \tau_x < \tau_e] = \int_0^{\tau_x}\int_0^{\tau_d}\int_0^{\tau_c}\int_0^{\tau_b} \frac{1}{Ti^4} \mathrm{d}\tau_a \mathrm{d}\tau_b \mathrm{d}\tau_c \mathrm{d}\tau_d \times \int_{\tau_x}^{Ti} \frac{1}{Ti} \mathrm{d}\tau_e$$
$$= \frac{\tau_x^4}{24Ti^4} - \frac{\tau_x^5}{24Ti^5} \quad (5.16)$$

5.1.2 Designing Scenarios for Adaptability Modeling

To simplify the analysis of *AdaptationDelays*, we design several scenarios to control the formation of *TLinks* and *TPaths* between nodes. In these scenarios, nodes are placed from left to right in the order of columns in Table 5.1. Some of the nodes are static while others are moving to form a line topology. For instance, scenario Sc.2.B is composed of static nodes *R* and *A* in range of each other while node *B* is moving (to the right of *A*) to come in range with *A* forming the line topology. Note that 2 and *B* in Sc.2.B refer to the number of hops and the ID of the moving nodes. It is worth mentioning that using this naming convention, it is possible to have two scenarios with identical impact on the formation of *TLinks* and *TPaths* between nodes such as Sc.1.R and Sc.1.A, also, Sc.3.BC and Sc.3.RA.

TABLE 5.1: Summary of *AdapationDelays* Scenarios

Scenario	Node C	Node B	Node A	Node R
Sc.1.R	N/A	N/A	Fixed	Moving
Sc.2.A	N/A	Fixed	Moving	Fixed
Sc.2.R	N/A	Fixed	Fixed	Moving
Sc.2.B	N/A	Moving	Fixed	Fixed
Sc.3.R	Fixed	Fixed	Fixed	Moving
Sc.3.A	Fixed	Fixed	Moving	Fixed
Sc.3.B	Fixed	Moving	Fixed	Fixed
Sc.3.C	Moving	Fixed	Fixed	Fixed
Sc.3.AB	Fixed	Moving	Moving	Fixed
Sc.3.AC	Moving	Fixed	Moving	Fixed
Sc.3.BC	Moving	Moving	Fixed	Fixed

5.1.3 Modeling ξ_1^{in} in MMT

5.1.3.1 Scenario Sc.1.R

Figure 5.1 shows this one-hop scenario in which the root node R is moving into the range of node A. Referring to Figure 5.2, we assume that nodes R and A are running MMT and come within D_{TX} of each other at time T_T^{in} from which we draw a time reference every Ti seconds as a vertical dashed line. Each node has it own internal timer which times out every Ti seconds to send a *hello* packet as indicated by $\boxed{h}$. The internal timers of R and A are skewed from the time reference of T_T^{in} by α_R and α_A seconds respectively. Both α_R and α_A are random variables distributed uniformly on $[0, Ti]$. This assumption was based on the fact that random timers to update and send logical information adopted by MMT and OLSR are uniformly distributed on $[0, Ti]$.

The first *hello* packet is sent by node A and received by R which is shown as the first upward arrow crossing the time axis. A is neither a root node or part of the MMT tree yet; as a result, the *hello* packet doesn't cause anything to happen. Later, A receives a *hello* packet from the root node R which is a root node. This *hello* packet includes the *VID* of R, (R,0,0) which A uses to initiate a registration process as explained in section 3.2. The process is concluded by the following:

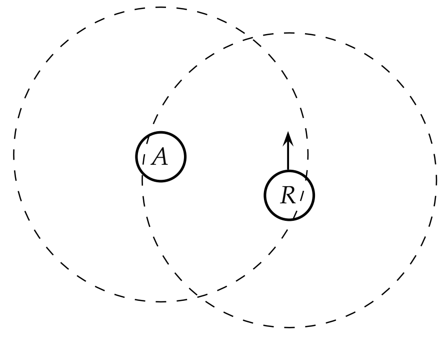

FIGURE 5.1: Scenario Sc.1.R

- A having the *VID* (R,1,1) in its *VIDList*, which serves as a two-way *LLink* with R and denoted by $R_{0c}^{\Leftrightarrow}$
- The *VID* (R,1,1) is also stored in the *ClientList* and *ChildList* at node R, which serves as a two-way *LLink* with A and denoted by $A_{0c}^{\Leftrightarrow}$

We use the notation $^{\Leftrightarrow}$ to represent that the *LLink* is a two-way logical link. The subscript 'c' in *LLink* stands for created. Logical link, *LLinks*, are kept alive by the periodic reception of *hello* packets from each other. As when a *hello* packet is not received, the numbered subscript in *LLink* is incremented by 1 till it is removed. For example, when A misses three consecutive *hello* packets from R, it is indicated by $R_1^{\Leftrightarrow}$, then $R_2^{\Leftrightarrow}$ before being removed at $R_3^{\Leftrightarrow}$. In Figure 5.2 we see that the reception of *hello* packets maintained the numbered subscript to 0.

The instant of building *LLinks* is called T_L^{in}, as mentioned before, and using (3.1) we find that $\xi_1^{in} = T_L^{in} - T_T^{in} = \alpha_R$. Since α_R is uniformly distributed on $[0, Ti]$, we write (5.17) for the *pdf* of ξ_1^{in} in MMT:

$$f_{MMT}(\xi_1^{in}) = \begin{cases} \frac{1}{Ti}, & 0 \le \xi_1^{in} \le Ti, \\ 0, & \text{otherwise}, \end{cases} \quad = \sim U(0, Ti) \tag{5.17}$$

Scenario Sc.1.R is the only possible scenario to form one hop *LPath* in Random Way Point mobility model. Figure 5.3 depicts the model of $f_{MMT}(\xi_1^{in})$ against simulation results

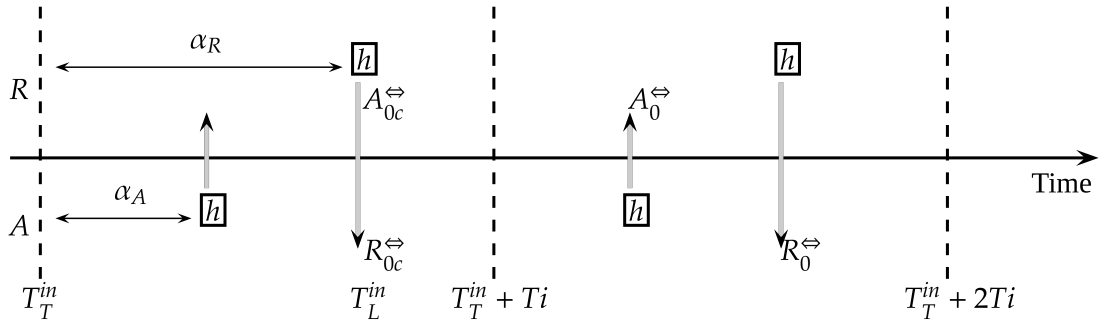

FIGURE 5.2: Modeling ξ_1^{in} in MMT

gathered from running scenarios shown in Table 3.4 adopting the mobility model in section 3.3 and using simulation parameters in Table 5.2.

TABLE 5.2: Simulation Parameters for Adaptability Modeling

Parameter	Value(s)
D_{TX}	200m
Sp_{avg}	5m/s, 10m/s, 15m/s, 20m/s
Ti	2s

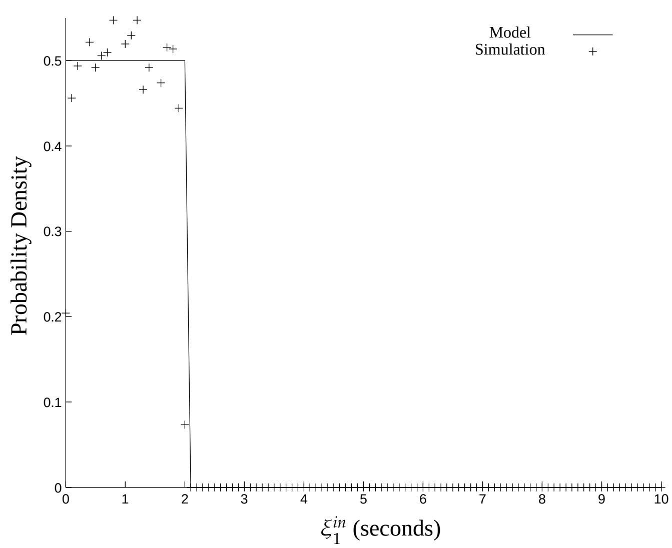

FIGURE 5.3: $f_{MMT}(\xi_1^{in})$ with $Ti = 2s$

5.1.4 Modeling ξ_1^{in} in OLSR

5.1.4.1 Scenario Sc.1.R

This scenario is considered where OLSR is running on the two nodes R and A shown in Figure 5.1. *LLinks* in OLSR are categorized either as one-way or two-way, while only the two-way *LLink*s are assumed reliable for data communication. In Figure 5.4 and after coming within distance D_{TX} of each other at time T_T^{in}, a two-way logical link, *LLink*, is built between two nodes R and A as follows:

- R sends a *hello* packet with originator ID R.
- A receives the *hello* packet and knows that it can hear R but the other direction is not necessarily true, represented by $R_{0c}^{\Rightarrow}$.
- At time $T_{L_R}^{in}$, A sends a *hello* packet with originator ID A and neighbor IDs R.
- R receives the *hello* packet and knows that it can hear A and the other direction is true; hence R forms a two-way *LLink* with A, represented by $A_{0c}^{\Leftrightarrow}$.
- At time $T_{L_A}^{in}$, R sends a *hello* packet with originator ID R and neighbor ID A.
- A receives the *hello* packet and knows that both directions are true; Hence A forms a two-way *LLink* with R, represented by $R_{0c}^{\Leftrightarrow}$.

Using (3.1), we find that ξ_1^{in} as observed by R forming *LLink* to A, $\xi_{1_R}^{in} = T_{L_R}^{in} - T_T^{in} = \alpha_A$ which is uniformly distributed on $[0, Ti]$, also $\xi_{1_A}^{in} = T_{L_A}^{in} - T_T^{in} = Ti + \alpha_R$ which is uniformly distributed on $[Ti, 2Ti]$. Switching the roles of nodes R and A, as in Figure 5.5, we can conclude that for any two nodes:

$$\xi_{1_R}^{in} \text{in OLSR} = \begin{cases} \alpha_A, & \alpha_R < \alpha_A, \\ Ti + \alpha_A, & \alpha_R > \alpha_A. \end{cases} \tag{5.18}$$

From (5.18), we notice that $\xi_{1_R}^{in}$ has two different ranges, the first is $[0, Ti]$ and the second is $[Ti, 2Ti]$. When $\xi_{1_R}^{in} \in [0, Ti]$, we can derive the *pdf* of $\xi_{1_R}^{in}$ using (5.1):

$$
\begin{aligned}
f_{OLSR}(\xi_{1_R}^{in}) &= P[\xi_{1_R}^{in} = \alpha_{A_0}] \times P[\alpha_R < \alpha_{A_0}] \\
&= \tfrac{1}{Ti} P_{2A}(\xi_{1_R}^{in}), \quad 0 \leq \xi_{1_R}^{in} \leq Ti.
\end{aligned} \tag{5.19}
$$

When $\xi_{1_R}^{in} \in [Ti, 2Ti]$, we can simplify the problem by introducing a new random variable denoted by, $\xi_{1_R}^{in'}$, and it equals α_A when $\alpha_R > \alpha_A$. This makes $\xi_{1_R}^{in'} \in [0, Ti]$, and using (5.2):

$$
\begin{aligned}
f_{OLSR}(\xi_{1_R}^{in'}) &= P[\xi_{1_R}^{in'} = \alpha_{A_0}] \times P[\alpha_R > \alpha_{A_0}] \\
&= \tfrac{1}{Ti} P_{2B}(\xi_{1_R}^{in'}), \quad 0 < \xi_{1_R}^{in'} \leq Ti.
\end{aligned} \tag{5.20}
$$

When $\xi_{1_R}^{in} \in [Ti, 2Ti]$, we assumed $\xi_{1_R}^{in'} = \alpha_A$ and in reality $\xi_{1_R}^{in} = Ti + \alpha_A$. This makes $\xi_{1_R}^{in'} = \xi_{1_R}^{in} - Ti$ which is used to relax the assumption made in (5.20) by replacing $\xi_{1_R}^{in'}$ with $\xi_{1_R}^{in} - Ti$. Then, by combining with (5.19), we can write (5.21) for the *pdf* of ξ_1^{in} in OLSR:

$$
f_{OLSR}(\xi_1^{in}) = \begin{cases}
\frac{1}{Ti} P_{2A}(\xi_1^{in}), & 0 \leq \xi_1^{in} \leq Ti, \\
\frac{1}{Ti} P_{2B}(\xi_1^{in} - Ti), & Ti < \xi_1^{in} \leq 2Ti, \\
0, & \text{otherwise,}
\end{cases} \tag{5.21}
$$

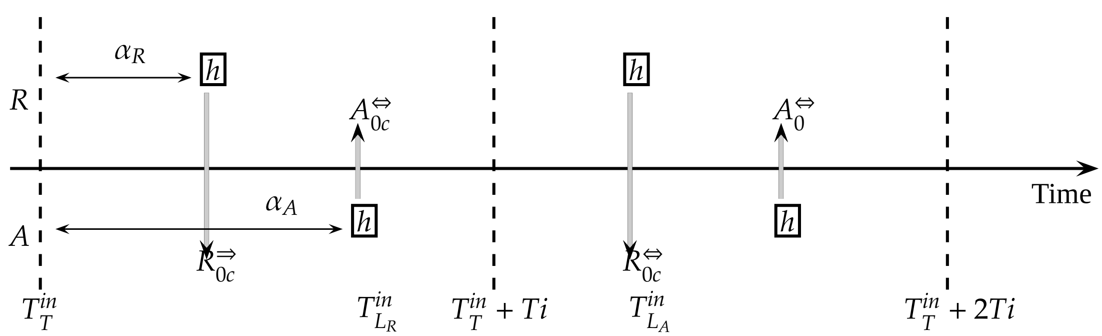

FIGURE 5.4: Modeling ξ_1^{in} in OLSR when $\alpha_R \leq \alpha_A$

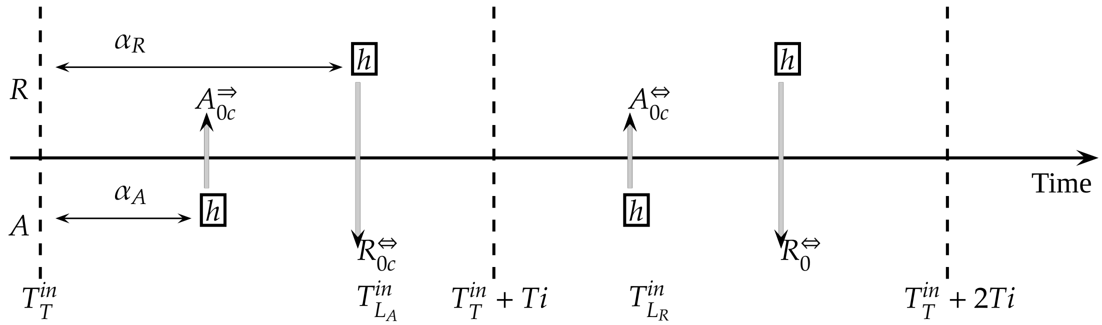

FIGURE 5.5: Modeling ξ_1^{in} in OLSR when $\alpha_R > \alpha_A$

Figure 5.6 depicts the model of $f_{OLSR}(\xi_1^{in})$ in Scenario Sc.1.R against simulation results gathered from running scenarios in Table 3.4 and using simulation parameters in Table 5.2. Note that Scenario Sc.1.R which is the only possible scenario to form one hop *LPath* in the mobility model adopted in section 3.3.

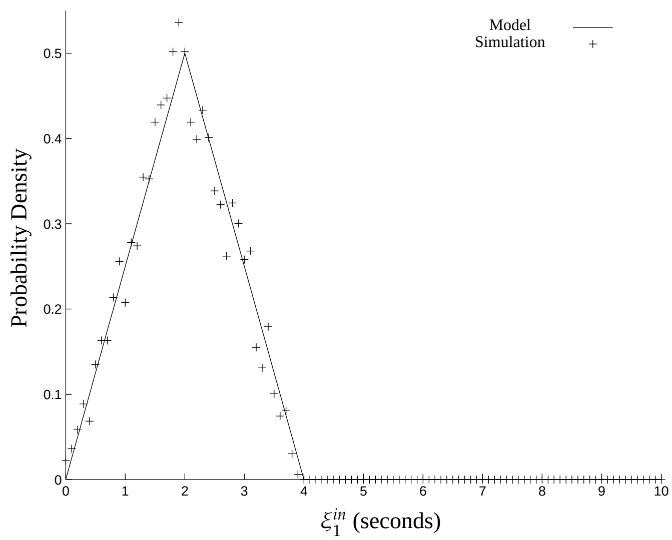

FIGURE 5.6: $f_{OLSR}(\xi_1^{in})$ with $Ti = 2s$

5.1.5 Modeling ξ_2^{in} in MMT

Similar *AdaptionDelays* modeling methodology can be applied in two-hops *LPath* scenarios. Depending on nodes locations and their mobility, two-hops *LPaths* can be modeled as following several scenarios.

5.1.5.1 Scenario Sc.2.A

This scenario is shown in Figure 5.7 which is one of the two hops scenarios where node A is moving in range of both node B and root R and all are running MMT.

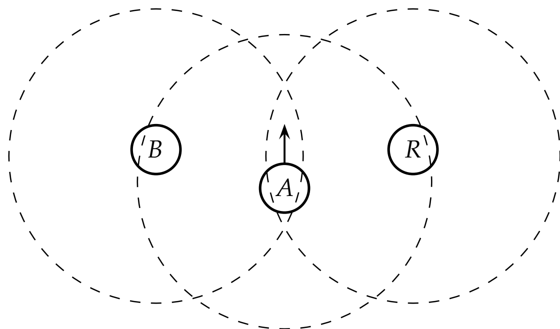

FIGURE 5.7: Scenario Sc.2.A

In Figure 5.8, we see that the second *hello* packet sent by the root R creates two-way *LLinks*, between R and A, similar to what was discussed in Figure 5.2. When the *LLink* is created, A has the *VID* (R,1,1). As a result, when the third *hello* packet is sent from node A containing the newly acquired *VID* (R,1,1), it triggers the registration process at node B at time T_L^{in} which is concluded by:

- B having a *VID* (R,11,2) in its *VIDList* which serves as a two-way *LPath* with R and *LLink* with A, indicated by $R_{0c}^{\Leftrightarrow}$ and $A_{0c}^{\Leftrightarrow}$ respectively.
- R stores *VID* (R,11,2) in its *ClientList* which serves as a two-way *LPath* with B, indicated by $B_{0c}^{\Leftrightarrow}$ at R.
- A stores *VID* (R,11,2) in its *ChildList* which serves as a two-way *LLink* with B, indicated by $B_{0c}^{\Leftrightarrow}$ at A.

To summarize, when the root R sends its *hello* packet, it triggers the registration process at node A. Consequently, when A sends its *hello* packet, it gives node B enough information to start its own registration process. Referring to Figure 5.8, we find ξ_2^{in} for MMT in scenario Sc.2.A when $\alpha_R \leq \alpha_A$ equals to α_A.

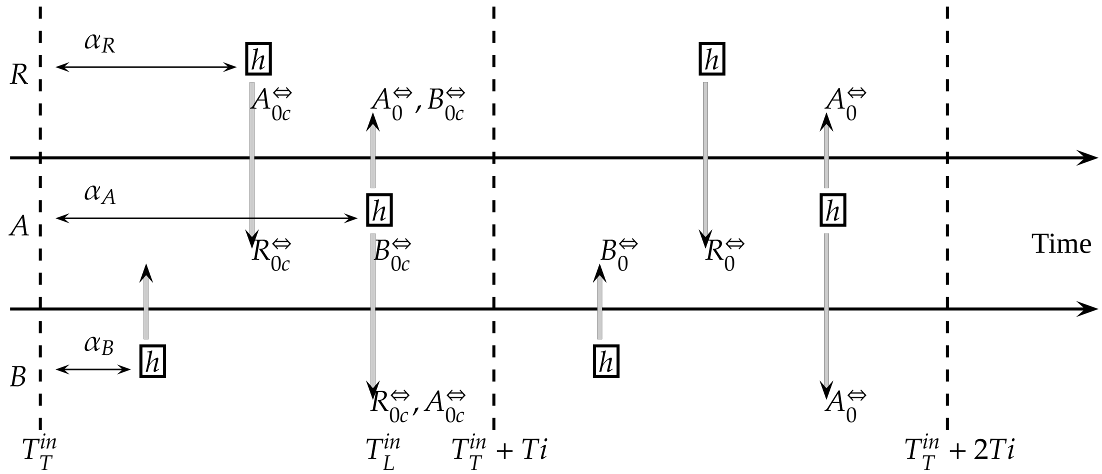

FIGURE 5.8: Modeling ξ_2^{in} in MMT using Scenario Sc.2.A and $\alpha_R \leq \alpha_A$

In Figure 5.9 we show the other case which corresponds to $\alpha_R > \alpha_A$. The third *hello* packet sent by R causes A to be part of the MMT tree by acquiring the *VID* (R,1,1). Node B is included in the MMT tree at time T_L^{in} which is the time A sends its *hello* packet including its newly acquired *VID*, the fifth *hello* packet. As a result, ξ_2^{in} for MMT in Scenario Sc.2.R when $\alpha_R > \alpha_A$ is $Ti + \alpha_A$. Summarizing the two cases of Scenario Sc.2.A:

$$\xi_2^{in} \text{in MMT for Sc.2.A} = \begin{cases} \alpha_A, & \alpha_R < \alpha_A, \\ Ti + \alpha_A, & \alpha_R > \alpha_A. \end{cases} \tag{5.22}$$

From (5.22), we notice that ξ_2^{in} has similar formulations as in (5.18). As a result, using same derivation methodology from section 5.1.4.1, we can write (5.23) for the *pdf* of ξ_2^{in} in MMT Scenario Sc.2.A:

$$f_{MMT}^{Sc.2.A}(\xi_2^{in}) = \begin{cases} \frac{1}{Ti} P_{2A}(\xi_2^{in}), & 0 \leq \xi_2^{in} \leq Ti, \\ \frac{1}{Ti} P_{2B}(\xi_2^{in} - Ti), & Ti < \xi_2^{in} \leq 2Ti, \\ 0, & \text{otherwise}, \end{cases} \tag{5.23}$$

Figure 5.10 depicts the model of $f_{MMT}^{Sc.2.A}(\xi_2^{in})$ with $Ti = 2s$ against simulation results gathered from simulating Scenario Sc.2.A.

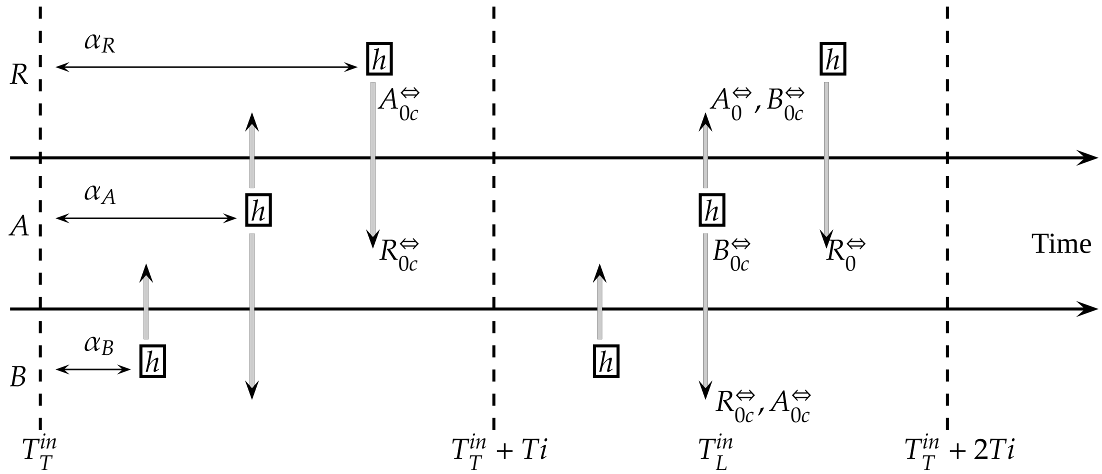

FIGURE 5.9: Modeling ξ_2^{in} in MMT using Scenario Sc.2.A and $\alpha_R > \alpha_A$

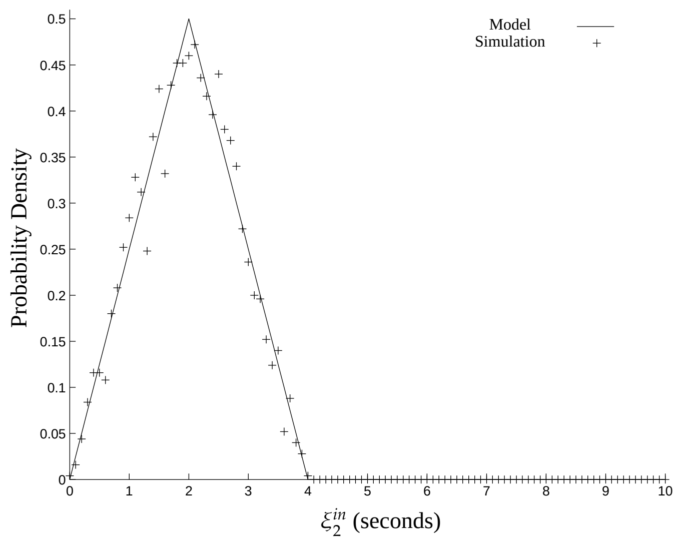

FIGURE 5.10: $f_{MMT}^{Sc.2.A}(\xi_2^{in})$ with $Ti = 2s$

5.1.5.2 Scenario Sc.2.R

Figure 5.11 shows another two-hops scenario where node R, the root, is moving in range of node A. Notice that A and B are already in range of each other; However, they lack MMT tree data and have no logical links, *LLinks*, information since the root R is not part of their line topology yet. As a result, the analysis for this scenario is similar to what was discussed in Scenario Sc.2.A. The reason is that the impact of node A joining the line topology in Scenario Sc.2.A is, effectively, the same as R joining the line topology. Thus, we write (5.24) using (5.23) for the *pdf* of ξ_2^{in} in the MMT Scenario Sc.2.R. Figure 5.12 depicts the model of $f_{MMT}^{Sc.2.R}(\xi_2^{in})$ with $Ti = 2s$ against the simulation results collected in running the Scenario Sc.2.R.

$$f_{MMT}^{Sc.2.R}(\xi_2^{in}) = f_{MMT}^{Sc.2.A}(\xi_2^{in}) \tag{5.24}$$

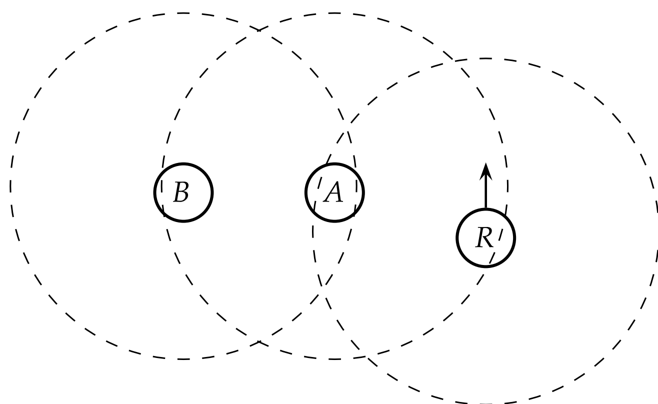

FIGURE 5.11: Scenario Sc.2.R for MMT

5.1.5.3 Scenario Sc.2.B

Figure 5.13 shows this scenario which is the last of the two hops scenarios for MMT, where only node B is moving to come in range of node A which already has a two-way logical link, *LLink*, with root R indicated by $R_0^{\Leftrightarrow}$.

The three nodes are running MMT, R has *VID* (R,0,0), while A has (R,1,1) in its *VIDList* which is also stored in the *ClientList* and *ChildList* at node R. In Figure 5.14, we show that nodes R and A have two-way *LLink*s before the time T_T^{in} indicated by $A_0^{\Leftrightarrow}$ and $R_0^{\Leftrightarrow}$. Trying other cases of ordering α_R, α_A and α_B shows that T_L^{in} is always when A sends its *hello* packet. Because A has already acquired its *VID* (R,1,1) before T_T^{in} and including it in a *hello* packet is sufficient for B to start its registration process. As a result, the ξ_2^{in} in MMT for Scenario Sc.2.B is always α_A which is uniformly distributed on $[0, Ti]$. In

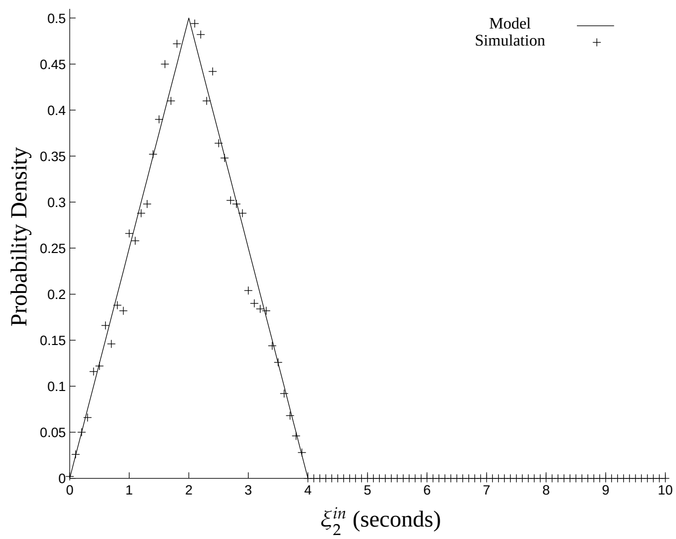

FIGURE 5.12: $f_{MMT}^{Sc.2.R}(\xi_2^{in})$ with $Ti = 2s$

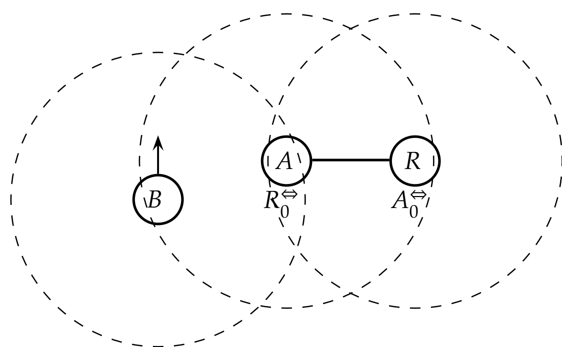

FIGURE 5.13: Scenario Sc.2.B

other words, the MMT tree has to extend one hop only to B by acquiring VID (R,11,2). This concept is similar to the discussion in section 5.1.3 which results in:

$$f_{MMT}^{Sc.2.B}(\xi_2^{in}) = f_{MMT}^{Sc.1.R}(\xi_1^{in}) = \begin{cases} \frac{1}{Ti}, & 0 \leq \xi_2^{in} \leq Ti, \\ 0, & \text{otherwise}, \end{cases} =\sim U(0, Ti) \tag{5.25}$$

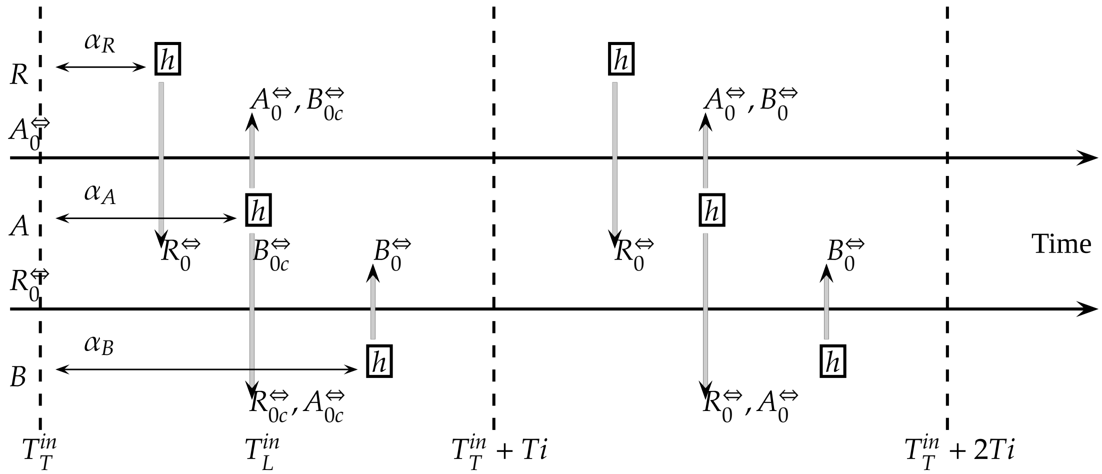

FIGURE 5.14: Modeling ξ_2^{in} in MMT using Scenario Sc.2.B

Figure 5.15 depicts the model of $f_{MMT}^{Sc.2.B}(\xi_2^{in})$ with $Ti = 2s$ against results from simulating The Scenario Sc.2.B.

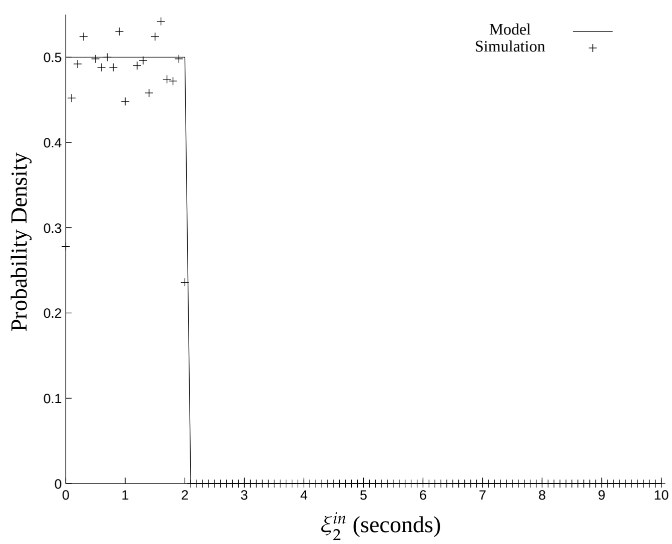

FIGURE 5.15: $f_{MMT}^{Sc.2.B}(\xi_2^{in})$ with $Ti = 2s$

In the mobility model detailed in section 3.3, the occurrence of the Scenario Sc.2.A is very unlikely in forming 2-hops *TPath* since it requires that node A comes in range of two nodes R and B at exactly the same instant. This means that a 2-hops *TPath* might form according to either Scenario Sc.2.R or Sc.2.B with equal probabilities. As a result, we write $f_{MMT}(\xi_2^{in})$ in (5.26) from (5.24) and (5.25). In Figure 5.16, we show the model of $f_{MMT}(\xi_2^{in})$ against simulation results collected from running the scenarios detailed in Table 3.4 using mobility model in 3.3 and simulation parameters in Table 5.2.

$$f_{MMT}(\xi_2^{in}) = \frac{f_{MMT}^{Sc.2.R}(\xi_2^{in}) + f_{MMT}^{Sc.2.B}(\xi_2^{in})}{2} \tag{5.26}$$

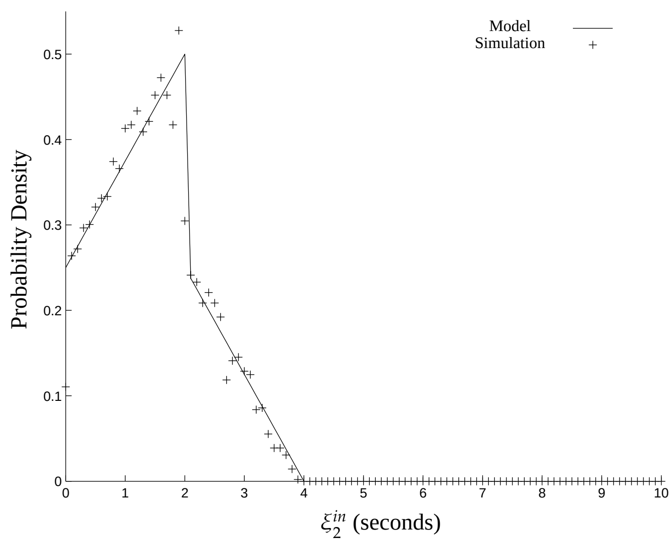

FIGURE 5.16: $f_{MMT}(\xi_2^{in})$ with $Ti = 2s$

5.1.6 Modeling ξ_2^{in} in OLSR

5.1.6.1 Scenario Sc.2.A

The scenario in Figure 5.7 is now applied to nodes running OLSR. Referring to Figure 5.17, two-hops two-way logical paths, *LPaths*, are created as follow:

- Before time T_L^{in} and similar to what was discussed in Figures 5.5 and 5.4, A has created two-way *LLink*s with B, $B_{0c}^{\Leftrightarrow}$, and R, $R_{0c}^{\Leftrightarrow}$. Also, two-way *LLink*s were created with A, $A_{0c}^{\Leftrightarrow}$, at R and B.
- At time T_L^{in}, A sends a *hello* packet, the fifth *hello* packet, with originator ID A and neighbor IDs R and B.
- R receives the *hello* packet and by comparing its neighbor list against the list received from A, R knows that B is a two-hops neighbor through A. Similarly, B learns that R is a two-hops neighbor through A.
- At the same instant, two-way *LPath*s are built, $B_{0c}^{\Leftrightarrow}$ at R and $R_{0c}^{\Leftrightarrow}$ at B.

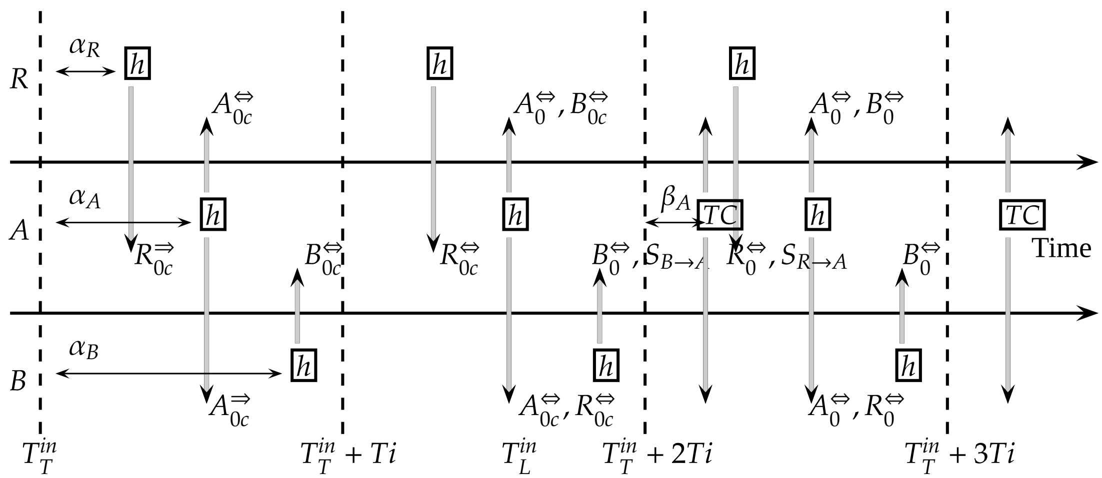

FIGURE 5.17: Modeling ξ_2^{in} in OLSR when A is moving in and $\alpha_R < \alpha_A < \alpha_B$

This makes $\xi_2^{in} = Ti + \alpha_A$ regardless of how α_R, α_A and α_B are ordered in time; another example with a different order of α_R, α_A and α_B is provided in Figure 5.19. The reason for $\xi_2^{in} = Ti + \alpha_A$ is that when A appears as the middle node between B and R and then sends its *hello* packet between times $T_T^{in} + Ti$ and $T_T^{in} + 2Ti$, A always builds a two-way *LLink* with any neighbor, say X, through one of two possibilities:

- When $\alpha_A < \alpha_X$: A has sent a *hello* packet once and received once from X containing ID A which is what happened between A and B in Figure 5.17.
- When $\alpha_A > \alpha_X$: A has sent a *hello* packet once and received twice from X with at least one of them contains ID A which is what happened between A and R in Figure 5.17.

This makes the *pdf* of ξ_2^{in} for OLSR in Scenario Sc.2.A as:

$$f_{OLSR}^{Sc.2.A}(\xi_2^{in}) = \begin{cases} \frac{1}{Ti}, & Ti \leq \xi_2^{in} \leq 2Ti, \\ 0, & \text{otherwise,} \end{cases} \quad =\sim U(0, Ti) \tag{5.27}$$

This result is shown in Figure 5.18 and compared to simulation results with $Ti = 2s$ using the Scenario Sc.2.A:

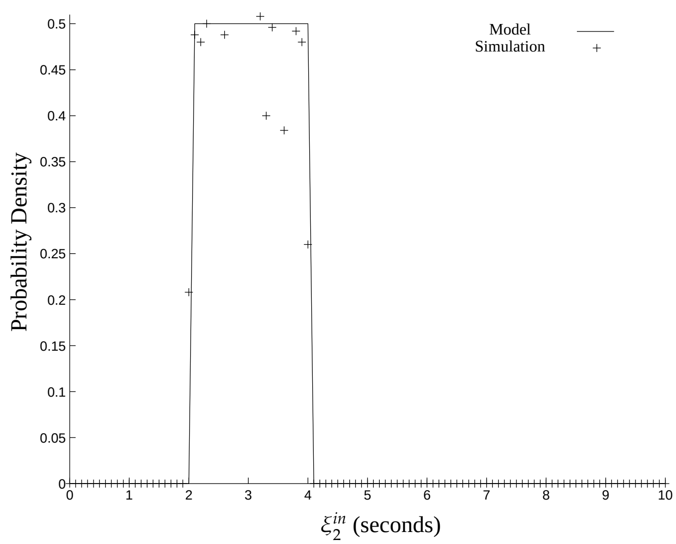

FIGURE 5.18: $f_{OLSR}^{Sc.2.A}(\xi_2^{in})$ with $Ti = 2s$

In the OLSR protocol, a node has to select a subset of its one-hop neighbors known as *MPR*s that their transmission ranges cover all of its two-hops neighbors. This selection is signaled to *MPR*s using *hello* packets. A node selected as *MPR* enables a timer, called *TC* timer, to send *TC* packets which times out every *Ti* seconds identical to *hello* timer but not necessarily in synch. As explained earlier in section 2.1.3, an *MPR* node sends *TC* packets containing at least the IDs of nodes that selected him as an *MPR*; these nodes are called *MPRSelectors*. *TC* packets are only forwarded by other *MPR*s to reach all nodes in the network announcing the availability of *LLink*s between *MPR*s and their *MPRSelectors*. Information in *TC* packets can be used to build two-way *LPath*s of two hops or more. In reference to Figure 5.17, *MPR* selection, signaling and operation happens as follow:

- At time T_L^{in}, nodes R and B learn that they are two-hops neighbors of each other and select A as their *MPR* since its transmission range meets the criterion of covering all of node's two-hops neighbors.
- Later, B sends a *hello* packet to A, the sixth *hello* packet, including the information that A was selected as *MPR* by B, indicated by $S_{B\to A}$.
- A receives the *hello* packet, learns that it was selected as *MPR* and enables its *TC* timer which times out every Ti seconds as indicated by $\boxed{TC}$. Note that *TC* timer is skewed from the time reference, the vertical dashed lines, by a time β_A which is a random variable uniformly distributed on $[0, Ti]$.
- Next, the first *TC* packet containing B as *MPRSelector* is sent; However, it does not trigger any event as it does not provide any new information.
- R sends a *hello* packet to A, the seventh *hello* packet, including the information that A was selected as *MPR* by R, indicated by $S_{R\to A}$.
- The node A receives the *hello* packet, learns that it was selected also as *MPR* by R. When the *TC* timer fires again, the new *TC* packet contains B and R as *MPRSelectors*.

We refer to the delay when node R signals node A its selection as *MPR* as $\lambda_{R\to A}$. In Figure 5.17, $\lambda_{B\to A} = Ti + \alpha_B$ and $\lambda_{R\to A} = 2Ti + \alpha_R$ when $\alpha_R < \alpha_A < \alpha_B$. Compared to Figure 5.19 where the change is $\lambda_{R\to A} = Ti + \alpha_R$ and $\alpha_A < \alpha_R < \alpha_B$; we find the following:

$$\lambda_{R\to A}\text{in OLSR for Sc.2.A} = \begin{cases} Ti + \alpha_R, & \alpha_A \leq \alpha_R, \\ 2Ti + \alpha_R, & \alpha_A > \alpha_R. \end{cases} \tag{5.28}$$

$$\lambda_{B\to A}\text{in OLSR for Sc.2.A} = \begin{cases} Ti + \alpha_B, & \alpha_A \leq \alpha_B, \\ 2Ti + \alpha_B, & \alpha_A > \alpha_B. \end{cases} \tag{5.29}$$

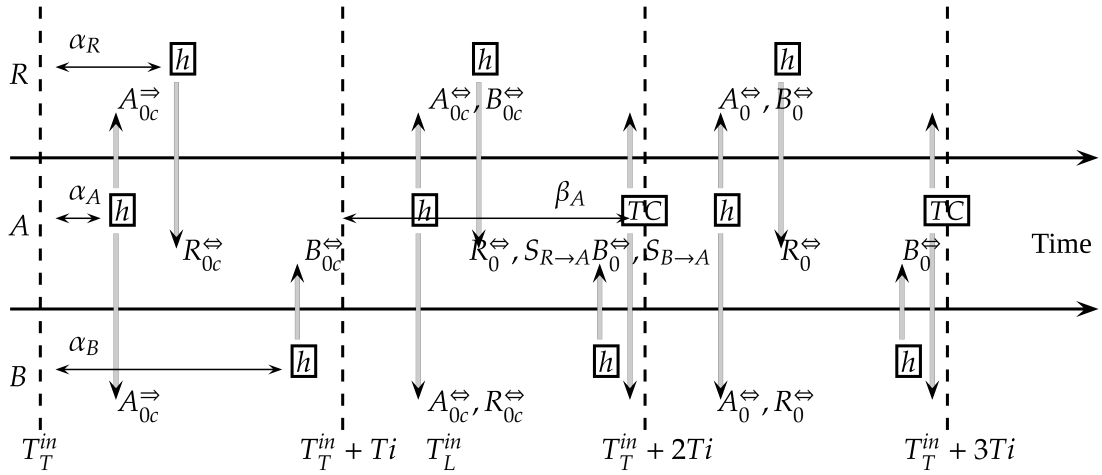

FIGURE 5.19: Modeling ξ_2^{in} in OLSR using Scenario Sc.2.A and $\alpha_A < \alpha_R < \alpha_B$

5.1.6.2 Scenario Sc.2.B

Figure 5.13 shows another scenario for forming two-hops *TPath* where all nodes are running OLSR. Note that R and A have two-way *LLink*s between each other before the time T_T^{in}. Figures 5.20 and 5.21 show the cases when $\alpha_A < \alpha_R < \alpha_B$ and $\alpha_A < \alpha_B < \alpha_R$. Following the protocol for OLSR operation mentioned before, we observe that $\xi_2^{in} = Ti + \alpha_A$, $\lambda_{R\to A} = Ti + \alpha_R$ and $\lambda_{B\to A} = Ti + \alpha_B$ in both cases. In the case when $\alpha_R < \alpha_A < \alpha_B$, seen in Figure 5.22, the times ξ_2^{in} and $\lambda_{B\to A}$ are the same as in the previous two cases in Figures 5.20 and 5.21 but the difference is that $\lambda_{R\to A} = 2Ti + \alpha_R$ instead of $\lambda_{R\to A} = Ti + \alpha_R$.

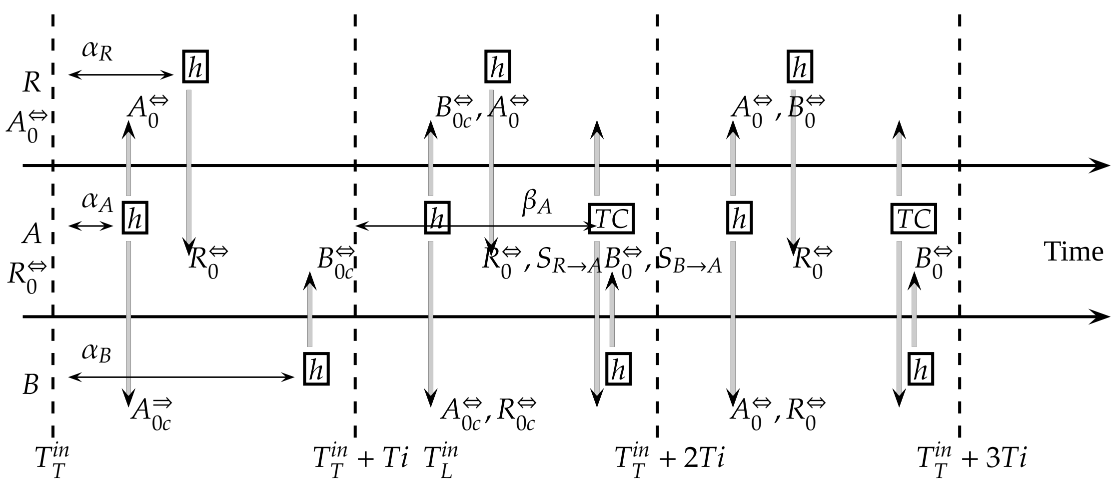

FIGURE 5.20: Modeling ξ_2^{in} in OLSR using Scenario Sc.2.B and $\alpha_A < \alpha_R < \alpha_B$

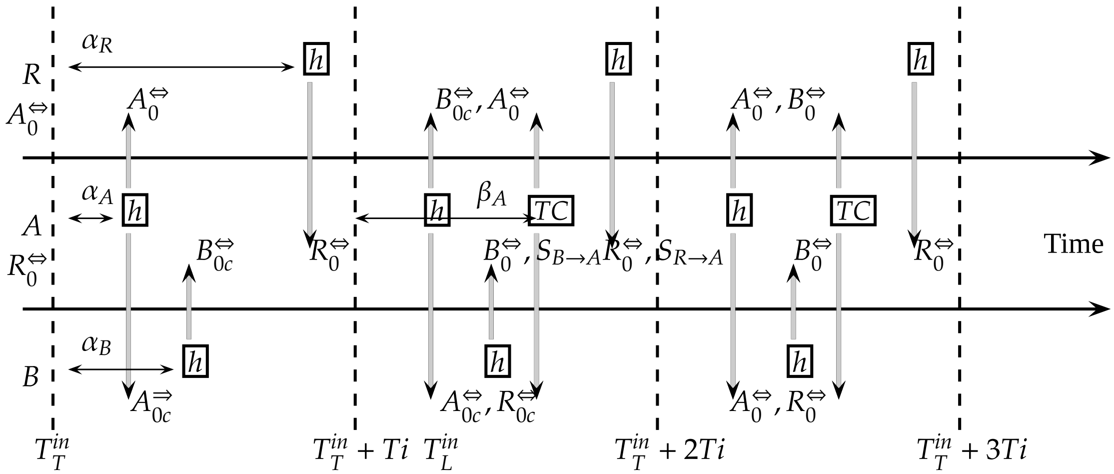

FIGURE 5.21: Modeling ξ_2^{in} in OLSR using Scenario Sc.2.B and $\alpha_A < \alpha_B < \alpha_R$

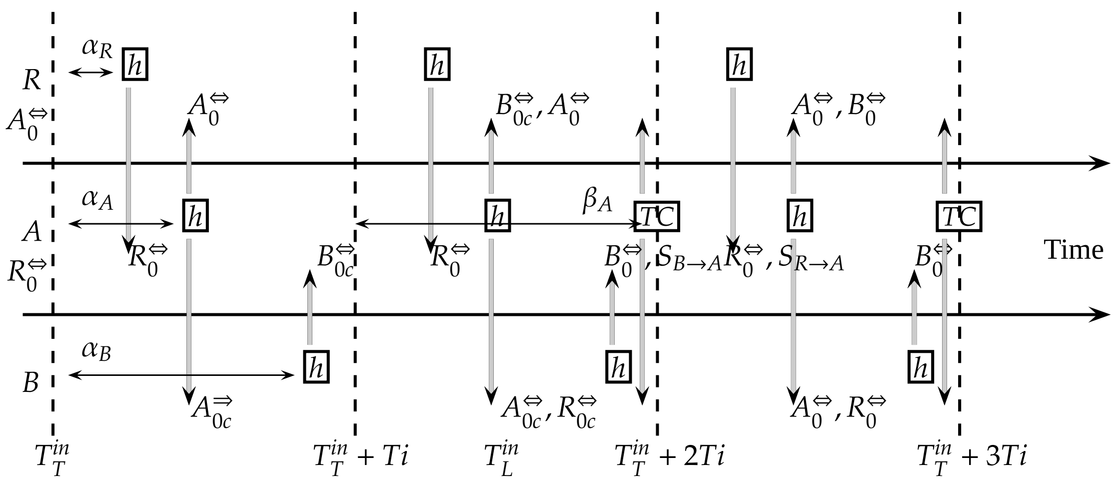

FIGURE 5.22: Modeling ξ_2^{in} in OLSR using Scenario Sc.2.B and $\alpha_R < \alpha_A < \alpha_B$

The case when $\alpha_B < \alpha_A < \alpha_R$ is shown in Figure 5.23, in which we see that B builds a two-way *LPath* with R at time $T_{L_B}^{in}$, similarly, R builds its two-way *LPath* with B at time $T_{L_R}^{in}$. This makes ξ_2^{in} as experienced by B and R, $\xi_{2_B}^{in}$ and $\xi_{2_R}^{in}$, equals to α_A and $Ti + \alpha_A$ respectively. Meanwhile, $\lambda_{R\to A} = Ti + \alpha_R$ and $\lambda_{B\to A} = Ti + \alpha_B$. The same case of $\alpha_B < \alpha_A < \alpha_R$ is shown in Figure 5.24 where R builds a two-way *LPath* with B as follow:

- At the time of the fourth *hello* packet being sent by B, it included the signaling of *MPR* selection to A indicated by $S_{B\to A}$.

- A receives the *hello* packet and knows it was selected as *MPR*; hence, it enables its *TC* timer which will time out before A sends its *hello* packet, the fifth *hello* packet.
- The *TC* timer at A times out and sends a *TC* packet with originator ID A and containing B as *MPRSelector*.
- At time $T^{in}_{L_R}$, R receives the *TC* packet and knows that B has selected A as an *MPR*; hence, they must have two-way logical links, *LLinks*, built between each other. The root node R uses the new *LLinks* between A and B to build a two-way logical path, *LPath*, with B, indicated by $B^{\Leftrightarrow}_{0c}$.

Notice that $T^{in}_{L_R}$ in Figure 5.24 happens earlier than Figure 5.23, making $\xi^{in}_{2_R} = Ti + \beta_A$ under the condition $\alpha_B < \beta_A < \alpha_A$. Figures 5.25 and 5.26, case $\alpha_R < \alpha_B < \alpha_A$, exhibit similar findings as the case when $\alpha_B < \beta_A < \alpha_A$ with the difference that $\lambda_{R\rightarrow A} = 2Ti + \alpha_R$.

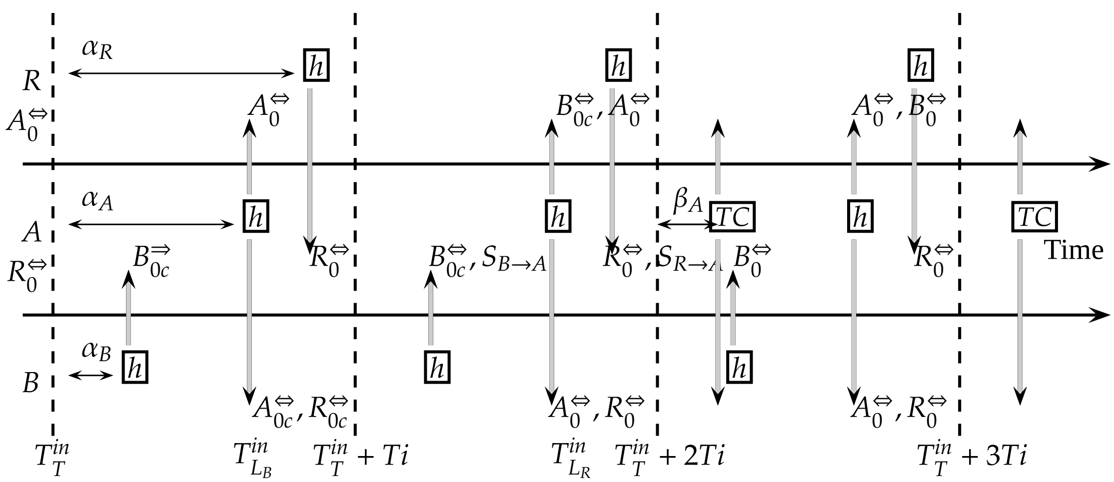

FIGURE 5.23: Modeling ξ^{in}_2 in OLSR using Scenario Sc.2.B and $\alpha_B < \alpha_A < \alpha_R$

The final case, $\alpha_B < \alpha_R < \alpha_A$ is shown in Figures 5.27 and 5.28 where satisfying the condition $\alpha_B < \beta_A < \alpha_A$ doesn't only change $\xi^{in}_{2_R}$ from $Ti + \alpha_A$ to $Ti + \beta_A$, but also $\lambda_{R\rightarrow A}$ from $2Ti + \alpha_R$ to $Ti + \alpha_R$. Table 5.3 presents a summary of previous discussions where the symbol "|" denotes a logical "or" between conditions. The column labeled "&" represents the extra conditions applied to the condition shown in the first column. The expansion of an extra condition is shown in the footer of Table 5.3.

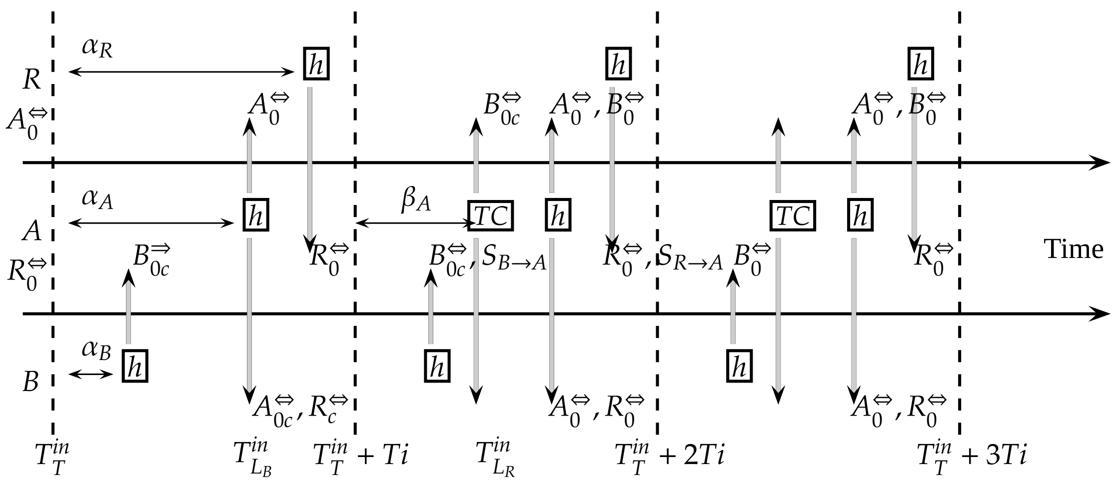

FIGURE 5.24: Modeling ξ_2^{in} in OLSR using Scenario Sc.2.B and $\alpha_B < \alpha_A < \alpha_R$ with TC

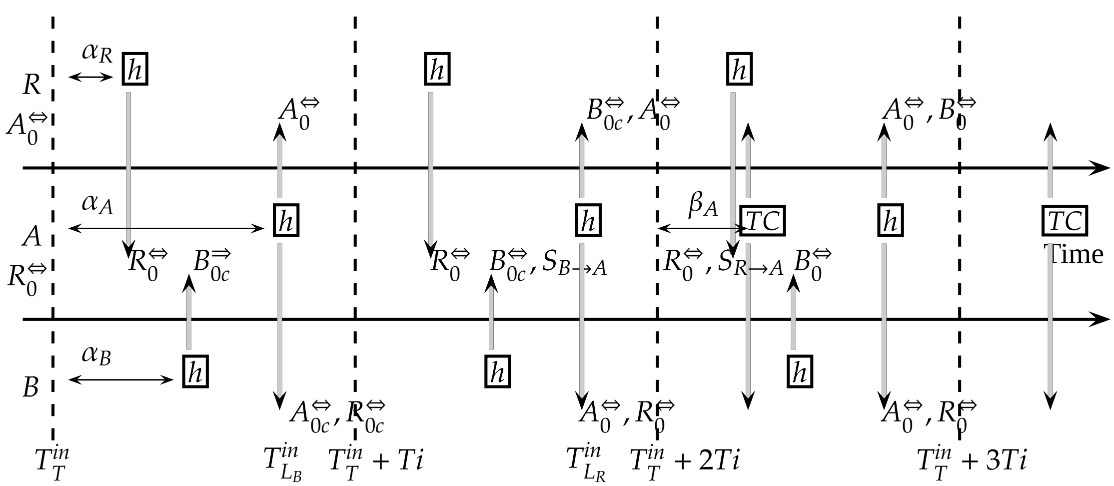

FIGURE 5.25: Modeling ξ_2^{in} in OLSR using Scenario Sc.2.B and $\alpha_R < \alpha_B < \alpha_A$

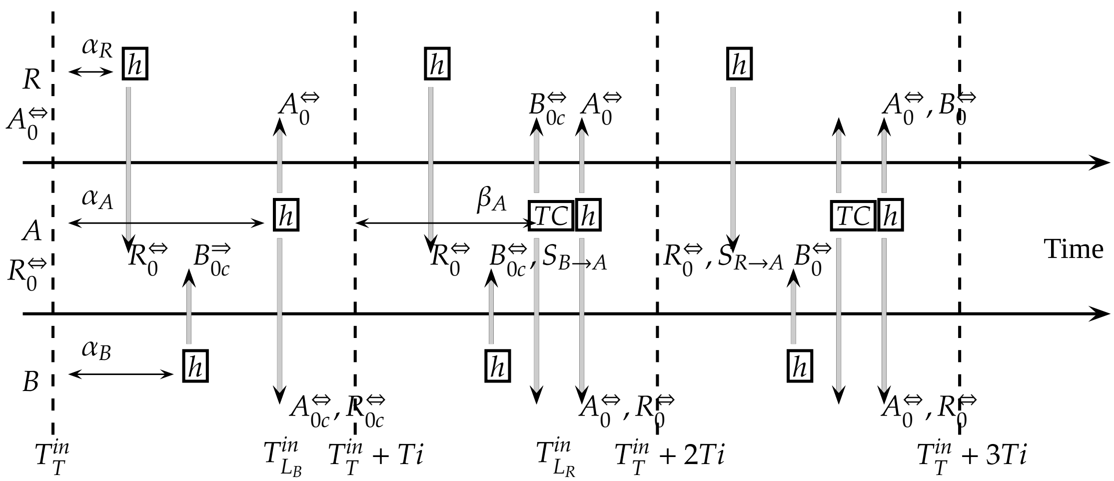

FIGURE 5.26: Modeling ξ_2^{in} in OLSR using Scenario Sc.2.B and $\alpha_R < \alpha_B < \alpha_A$ with TC

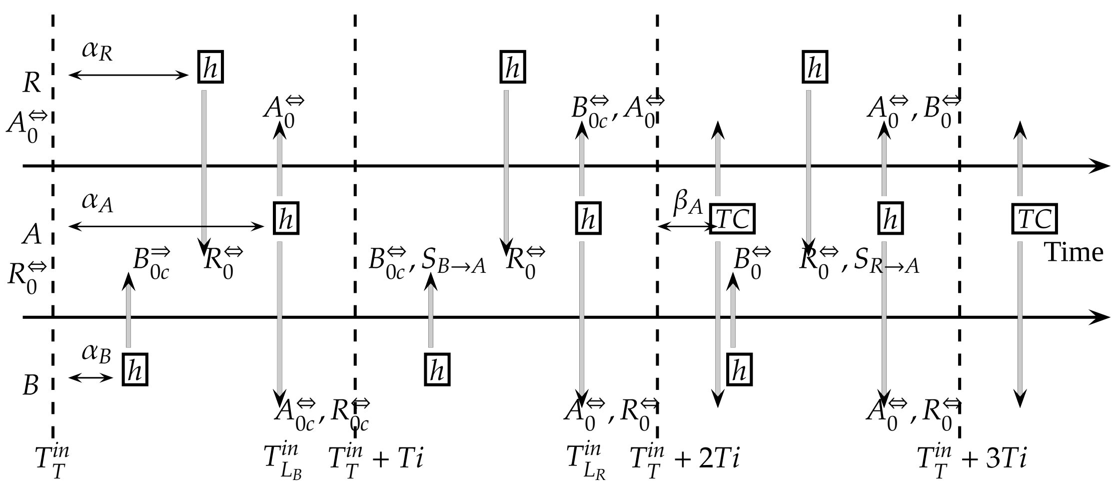

FIGURE 5.27: Modeling ξ_2^{in} in OLSR using Scenario Sc.2.B and $\alpha_B < \alpha_R < \alpha_A$

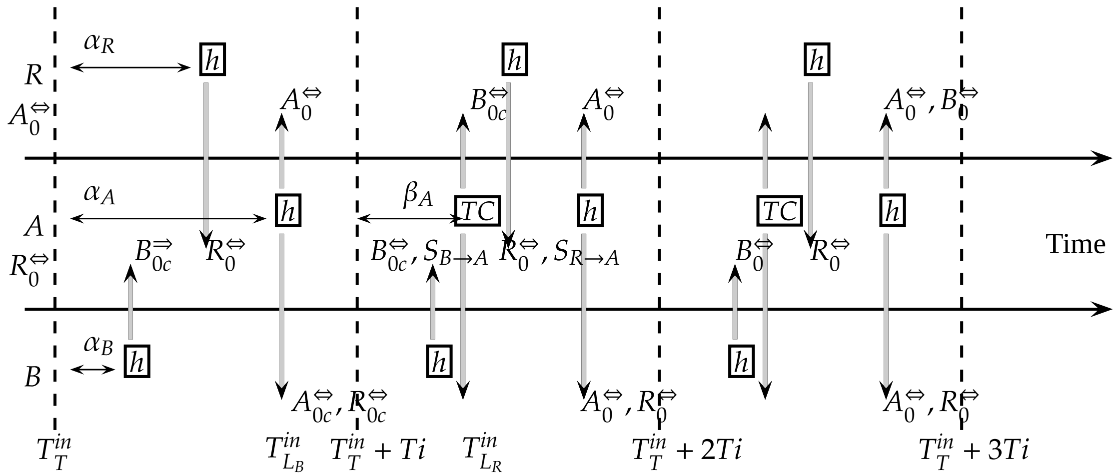

FIGURE 5.28: Modeling ξ_2^{in} in OLSR using Scenario Sc.2.B and $\alpha_B < \alpha_R < \alpha_A$ with TC

TABLE 5.3: Summary of Cases in Scenario Sc.2.B for OLSR

Condition	&	$\xi_{2_R}^{in}$	$\xi_{2_B}^{in}$	&	$\lambda_{R\to A}$	$\lambda_{B\to A}$
$\alpha_A < \alpha_R < \alpha_B$	n/a	$Ti + \alpha_A$	$Ti + \alpha_A$	n/a	$Ti + \alpha_R$	$Ti + \alpha_B$
$\alpha_A < \alpha_B < \alpha_R$	n/a	$Ti + \alpha_A$	$Ti + \alpha_A$	n/a	$Ti + \alpha_R$	$Ti + \alpha_B$
$\alpha_R < \alpha_A < \alpha_B$	n/a	$Ti + \alpha_A$	$Ti + \alpha_A$	n/a	$2Ti + \alpha_R$	$Ti + \alpha_B$
$\alpha_B < \alpha_A < \alpha_R$	w	$Ti + \alpha_A$	α_A	n/a	$Ti + \alpha_R$	$Ti + \alpha_B$
	x	$Ti + \beta_A$				
$\alpha_R < \alpha_B < \alpha_A$	w	$Ti + \alpha_A$	α_A	n/a	$2Ti + \alpha_R$	$Ti + \alpha_B$
	x	$Ti + \beta_A$				
$\alpha_B < \alpha_R < \alpha_A$	w	$Ti + \alpha_A$	α_A	y	$2Ti + \alpha_R$	$Ti + \alpha_B$
	x	$Ti + \beta_A$		z	$Ti + \alpha_R$	
w is $\beta_A < \alpha_B \vert \beta_A > \alpha_A$, x is $\alpha_B < \beta_A < \alpha_A$,y is $\beta_A < \alpha_B \vert \beta_A > \alpha_R$, z is $\alpha_B < \beta_A < \alpha_R$						

From Table 5.3 we notice that $\xi_{2_B}^{in}$ in Scenario Sc.2.B has the following values:

$$\xi_{2_B}^{in} \text{ in OLSR for Sc.2.B} = \begin{cases} \alpha_A, & \alpha_B < \alpha_A, \\ Ti + \alpha_A, & \alpha_B > \alpha_A. \end{cases} \tag{5.30}$$

From (5.30), we notice that $\xi_{2_B}^{in}$ has similar formulations as in (5.18) when replacing α_B with α_R. As a result, using similar derivation methodology as in section 5.1.4.1, we can write (5.31) for the *pdf* of $\xi_{2_B}^{in}$ in OLSR for Scenario Sc.2.B:

$$f_{OLSR}^{Sc.2.B}(\xi_{2_B}^{in}) = \begin{cases} \frac{1}{Ti} P_{2A}(\xi_{2_B}^{in}), & 0 \leq \xi_{2_B}^{in} \leq Ti, \\ \frac{1}{Ti} P_{2B}(\xi_{2_B}^{in} - Ti), & Ti < \xi_{2_B}^{in} \leq 2Ti, \\ 0, & \text{otherwise}, \end{cases} \tag{5.31}$$

Figure 5.29 depicts the model of $f_{OLSR}^{Sc.2.B}(\xi_2^{in})$ with $Ti = 2s$ against simulation results.

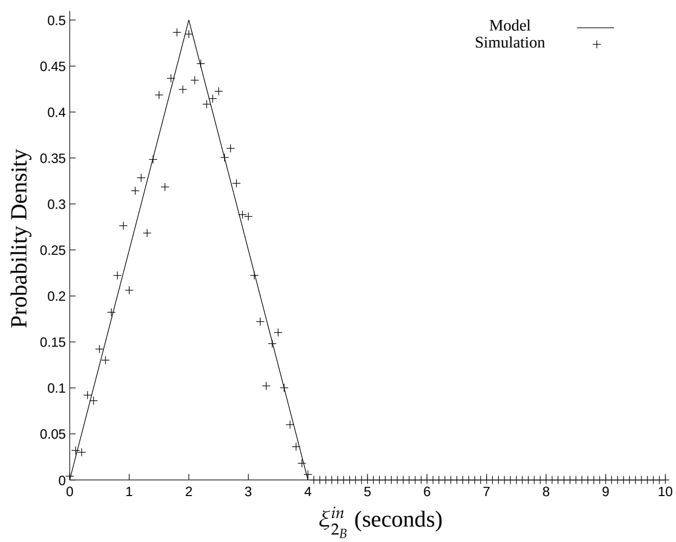

FIGURE 5.29: $f_{OLSR}^{Sc.2.B}(\xi_{2_B}^{in})$ with $Ti = 2s$

The values of $\xi_{2_R}^{in}$ in Scenario Sc.2.B shown in Table 5.3 are simplified in three rounds as presented in Table 5.4 which shows that $\xi_{2_R}^{in}$ has a single range of $[Ti, 2Ti]$. For the case when $\xi_{2_R}^{in} = Ti + \alpha_A$ and to simplify the derivation, we will introduce a new random variable $\xi_{2_R}^{in'} = \alpha_A$ and use (5.2), (5.3) and (5.4):

$$\begin{aligned} f_{OLSR}^{Sc.2.B}(\xi_{2_R}^{in'}) &= P[\xi_{2_R}^{in'} = \alpha_{A_0}] \times \Big(P[\alpha_B > \alpha_{A_0}] + P[\beta_A < \alpha_B < \alpha_{A_0}] + P[\alpha_B < \alpha_{A_0} < \beta_A]\Big) \\ &= \frac{1}{Ti}\Big(P_{2B}(\xi_{2_R}^{in'}) + P_{3A}(\xi_{2_R}^{in'}) + P_{3B}(\xi_{2_R}^{in'})\Big), \quad 0 \le \xi_{2_R}^{in'} \le Ti. \end{aligned} \tag{5.32}$$

Table 5.4: Simplifying $\xi_{2_R}^{in}$ in Scenario Sc.2.B for OLSR

$\xi_{2_R}^{in}$	Condition
	Simplification Round 1
$Ti + \alpha_A$	$(\alpha_A < \alpha_R < \alpha_B)\vert(\alpha_A < \alpha_B < \alpha_R)\vert(\alpha_R < \alpha_A < \alpha_B)\vert[(\alpha_B < \alpha_A < \alpha_R)$ & $(\beta_A < \alpha_B\vert\beta_A > \alpha_A)]\vert[(\alpha_R < \alpha_B < \alpha_A)$ & $(\beta_A < \alpha_B\vert\beta_A > \alpha_A)]\vert[(\alpha_B < \alpha_R < \alpha_A)$ & $(\beta_A < \alpha_B\vert\beta_A > \alpha_A)]$
$Ti + \beta_A$	$[(\alpha_B < \alpha_A < \alpha_R)$ & $(\alpha_B < \beta_A < \alpha_A)]\vert[(\alpha_R < \alpha_B < \alpha_A)$ & $(\alpha_B < \beta_A < \alpha_A)]\vert[(\alpha_B < \alpha_R < \alpha_A)$ & $(\alpha_B < \beta_A < \alpha_A)]$
	Simplification Round 2
$Ti + \alpha_A$	$(\alpha_A < \alpha_B)\vert[(\alpha_B < \alpha_A)$ & $(\beta_A < \alpha_B\vert\beta_A > \alpha_A)]$
$Ti + \beta_A$	$\alpha_B < \alpha_A$ & $\alpha_B < \beta_A < \alpha_A$
	Simplification Round 3
$Ti + \alpha_A$	$\alpha_A < \alpha_B\vert\beta_A < \alpha_B < \alpha_A\vert\alpha_B < \alpha_A < \beta_A$
$Ti + \beta_A$	$\alpha_B < \beta_A < \alpha_A$

On the other hand, when $\xi_{2_R}^{in} = Ti + \beta_A$, we will introduce another random variable $\xi_{2_R}^{in''} = \beta_A$ and use (5.4):

$$\begin{aligned} f_{OLSR}^{Sc.2.B}(\xi_{2_R}^{in''}) &= P[\xi_{2_R}^{in''} = \beta_{A_0}] \times P[\alpha_B < \alpha_{A_0} < \beta_A] \\ &= \frac{1}{Ti}P_{3B}(\xi_{2_R}^{in''}), \quad 0 \le \xi_{2_R}^{in''} \le Ti. \end{aligned} \tag{5.33}$$

The assumptions made in (5.32) and (5.33) are relaxed by replacing $\xi_{2_R}^{in'}$ and $\xi_{2_R}^{in''}$ by $\xi_{2_R}^{in} - Ti$. Then, by combining the two equations, we write (5.34) for the *pdf* of $\xi_{2_R}^{in}$ in OLSR Scenario Sc.2.B as:

$$f^{Sc.2.B}_{OLSR}(\xi^{in}_{2_R}) = \begin{cases} \frac{1}{Ti}Q(\xi^{in}_{2_R}), & Ti \leq \xi^{in}_{2_R} \leq 2Ti, \\ 0, & \text{otherwise,} \end{cases} \tag{5.34}$$

where $Q(\xi^{in}_{2_R}) = P_{2B}(\xi^{in}_{2_R} - Ti) + P_{3A}(\xi^{in}_{2_R} - Ti) + 2P_{3B}(\xi^{in}_{2_R} - Ti)$. This *pdf*, $f^{Sc.2.B}_{OLSR}(\xi^{in}_{2_R})$, is shown in Figure 5.30 and compared to simulation results with $Ti = 2s$.

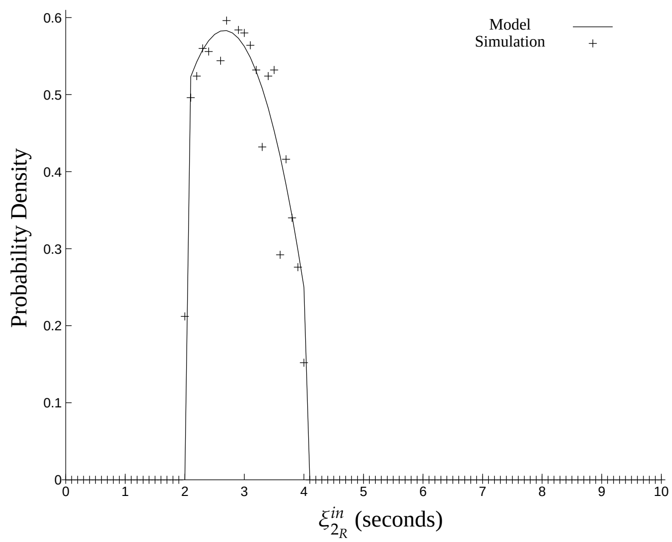

FIGURE 5.30: $f^{Sc.2.B}_{OLSR}(\xi^{in}_{2_R})$ with $Ti = 2s$

As explained before, the adopted mobility model detailed in section 3.3 makes the occurrence of the Scenario Sc.2.A very unlikely in forming 2-hops *TPath* since it requires that node A comes in range of two nodes R and B at exactly the same instant. This means that the *AdaptationDelay* for a 2-hops topological path, *TPath*, occurs based on the *pdf*s of $f^{Sc.2.B}_{OLSR}(\xi^{in}_{2_B})$ or $f^{Sc.2.B}_{OLSR}(\xi^{in}_{2_R})$ with equal probability; as a result, we write $f_{OLSR}(\xi^{in}_{2})$ in (5.35) from (5.31) and (5.34). In Figure 5.31, we compare the model of $f_{OLSR}(\xi^{in}_{2})$ against simulation results using mobility model in section 3.3 and simulation parameters in Table 5.2.

$$f_{OLSR}(\xi_2^{in}) = \frac{f_{OLSR}^{Sc.2.B}(\xi_{2_B}^{in}) + f_{OLSR}^{Sc.2.B}(\xi_{2_R}^{in})}{2} \tag{5.35}$$

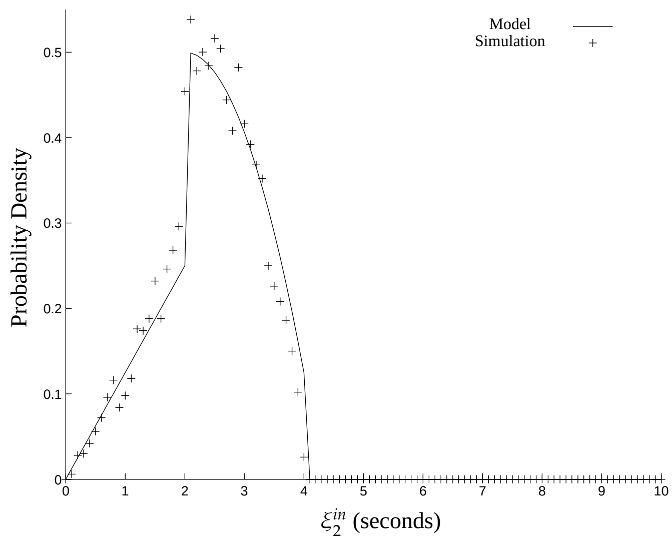

FIGURE 5.31: $f_{OLSR}(\xi_2^{in})$ with $Ti = 2s$

5.1.7 Modeling ξ_3^{in} in MMT

Following similar methodology, this sections presents all possible scenarios associated with 3-hops topological paths, *TPath*, to facilitate the study of *AdaptationDelays* of building 3-hops logical paths, *LPaths*.

5.1.7.1 Scenario Sc.3.R

The scenario in Figure 5.32 shows the first of the three hops scenarios where nodes A, B and C are forming a line topology and root R is coming in range of A at one end of the network. Figure 5.33 shows the communication scenario when $\alpha_R < \alpha_A < \alpha_B$ in which the second *hello* packet is sent from the root node R creating two-way logical links, *LLinks*, between R and A similar to what was discussed in Figure 5.2. Then, the following *hello* packet extends the MMT tree creation to node B as explained previously in Figure 5.8. Finally, node B sends the forth *hello* packet containing its newly acquired *VID* (R,2,11) which when received by node C, it starts a registration process at time T_L^{in} making ξ_3^{in}, in this case, equals to α_B seconds. The registration process is concluded by:

- C having a *VID* (R,111,3) in its *VIDList* which serves as a two-way *LPath* with R and *LLink* with A, indicated by $R_{0c}^{\Leftrightarrow}$ and $B_{0c}^{\Leftrightarrow}$ respectively.
- R stores *VID* (R,111,3) in its *ClientList* which serves as a two-way *LPath* with C, indicated by $C_{0c}^{\Leftrightarrow}$ at R.
- B stores *VID* (R,111,3) in its *ChildList* which serves as a two-way *LLink* with C, indicated by $C_{0c}^{\Leftrightarrow}$ at B.

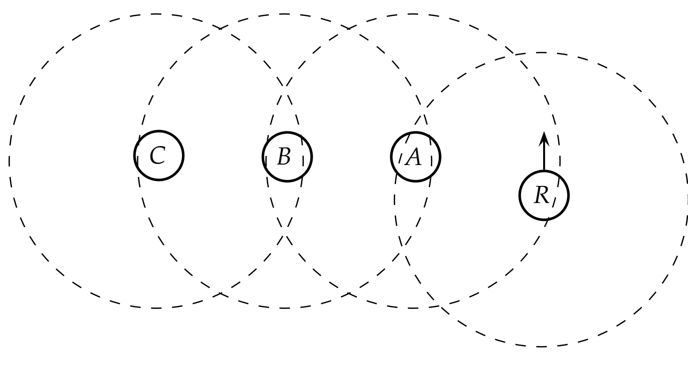

FIGURE 5.32: Scenario Sc.3.R for MMT

In contrast, when the order is $\alpha_B < \alpha_A < \alpha_R$ as shown in Figure 5.34, we notice that B waits $Ti + \alpha_A$ to become part of the MMT tree since $\alpha_A < \alpha_R$. In addition, C has to wait at most another Ti seconds to join the MMT tree since $\alpha_B < \alpha_A$ making $\xi_3^{in} = 2Ti + \alpha_B$.

In the remaining ordering cases for α_R, α_A and α_B, shown in Figures 5.35 through 5.38, we notice that one of the two inequalities, $\alpha_A < \alpha_R$ or $\alpha_B < \alpha_A$ is satisfied making $\xi_3^{in} = Ti + \alpha_B$. Table 5.5 shows a summary of conditions and associated values for ξ_3^{in} in Scenario Sc.3.R. Notice that the coefficient of Ti in the second column equals the number of times that either $\alpha_A < \alpha_R$ or $\alpha_B < \alpha_A$ are found in the condition column.

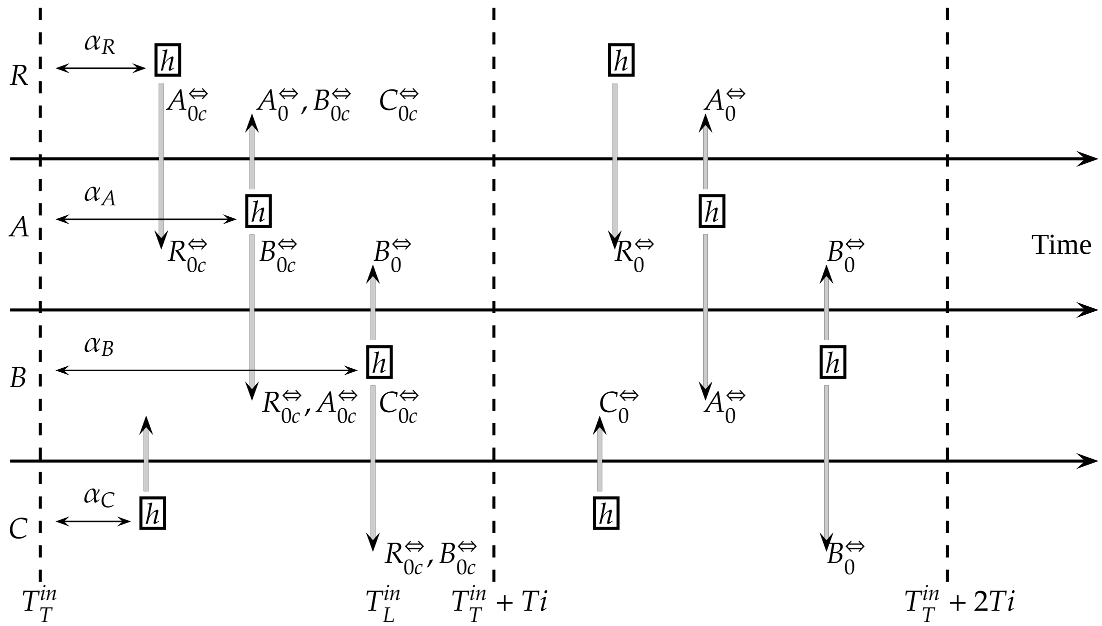

FIGURE 5.33: Modeling ξ^{in}_3 in MMT using Scenario Sc.3.R and $\alpha_R < \alpha_A < \alpha_B$

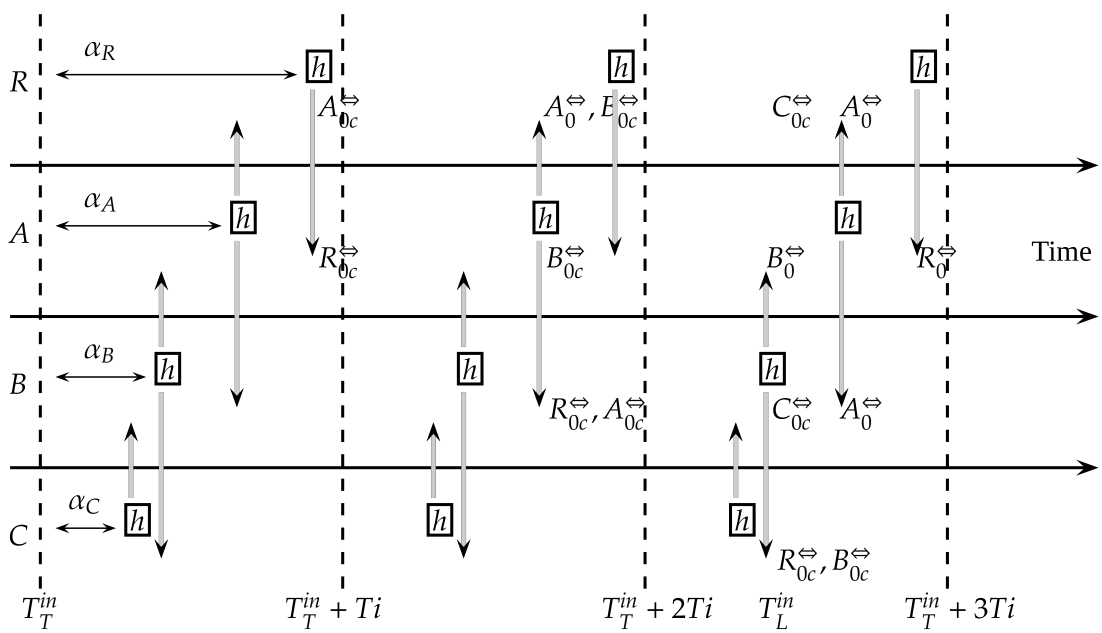

FIGURE 5.34: Modeling ξ^{in}_3 in MMT using Scenario Sc.3.R and $\alpha_B < \alpha_A < \alpha_R$

TABLE 5.5: Summary of ξ_3^{in} in Scenario Sc.3.R for MMT

Condition	ξ_3^{in}
$\alpha_R < \alpha_A < \alpha_B$	α_B
$\alpha_R < \alpha_B < \alpha_A$	$Ti + \alpha_B$
$\alpha_A < \alpha_R < \alpha_B$	$Ti + \alpha_B$
$\alpha_A < \alpha_B < \alpha_R$	$Ti + \alpha_B$
$\alpha_B < \alpha_R < \alpha_A$	$Ti + \alpha_B$
$\alpha_B < \alpha_A < \alpha_R$	$2Ti + \alpha_B$

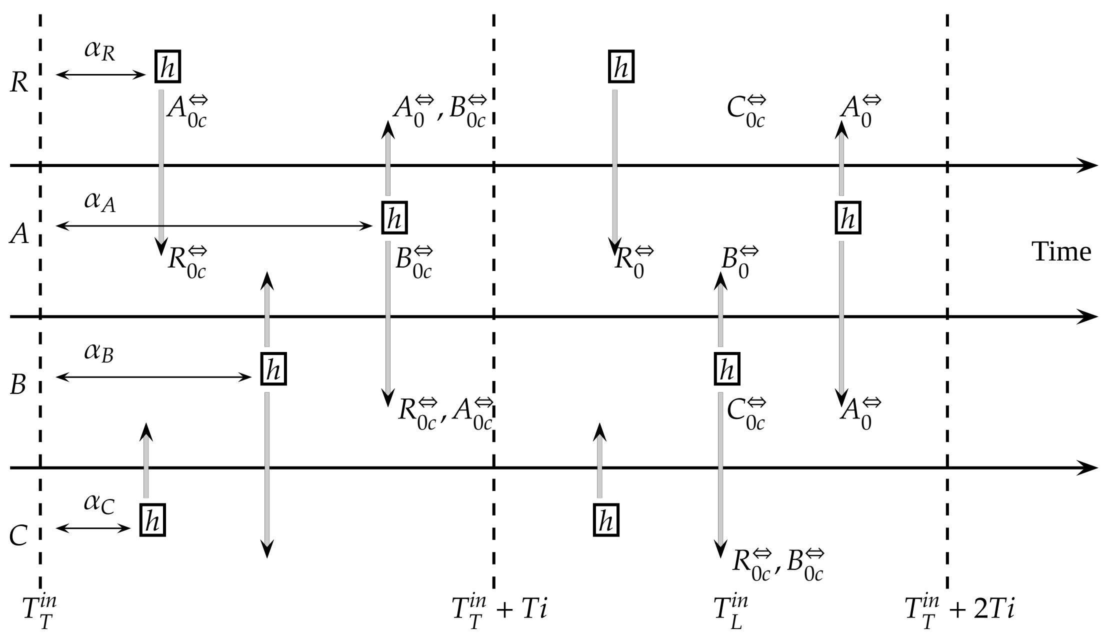

FIGURE 5.35: Modeling ξ_3^{in} in MMT using Scenario Sc.3.R and $\alpha_R < \alpha_B < \alpha_A$

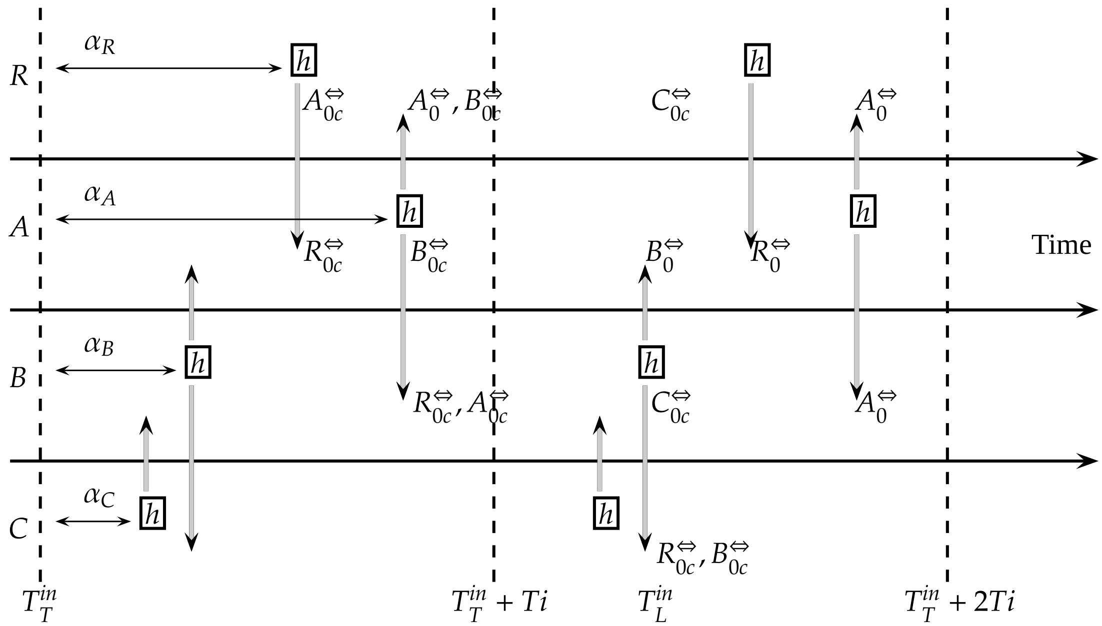

FIGURE 5.36: Modeling ξ_3^{in} in MMT using Scenario Sc.3.R and $\alpha_B < \alpha_R < \alpha_A$

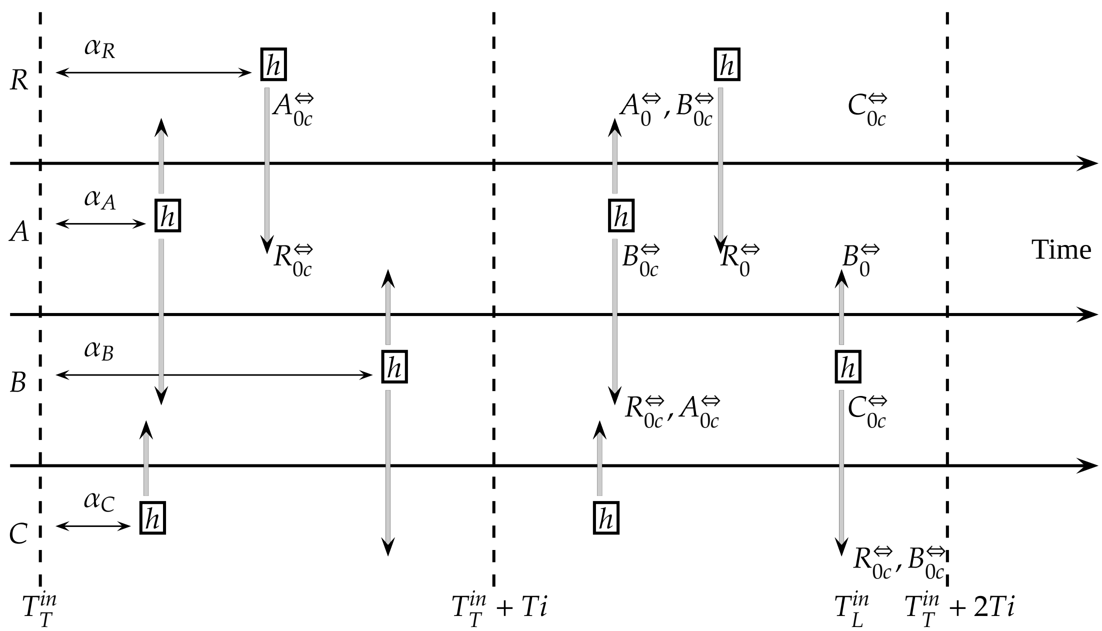

FIGURE 5.37: Modeling ξ_3^{in} in MMT using Scenario Sc.3.R and $\alpha_A < \alpha_R < \alpha_B$

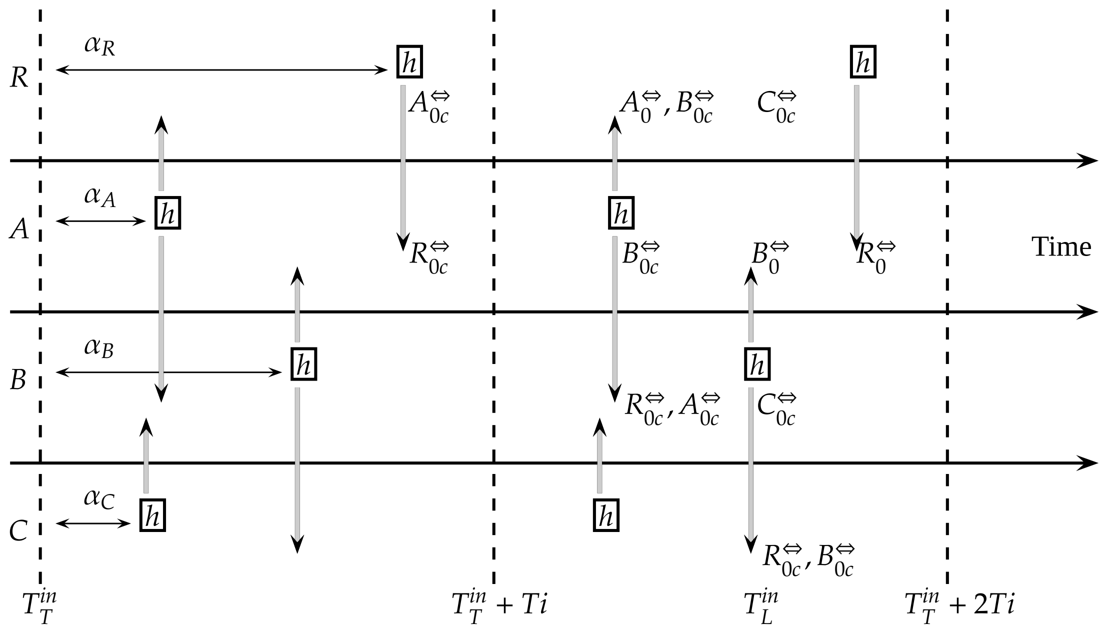

FIGURE 5.38: Modeling ξ_3^{in} in MMT using Scenario Sc.3.R and $\alpha_A < \alpha_B < \alpha_R$

From Table 5.5, we notice that ξ_3^{in} has different behaviors over three different ranges which are $[0, Ti]$, $[Ti, 2Ti]$ and $[2Ti, 3Ti]$. When $\xi_3^{in} \in [0, Ti]$, we use (5.3) to derive the *pdf* of ξ_3^{in} for Scenario Sc.3.R as:

$$\begin{aligned} f_{MMT}^{Sc.3.R}(\xi_3^{in}) &= P[\xi_3^{in} = \alpha_{B_0}] \times P[\alpha_R < \alpha_A < \alpha_{B_0}] \\ &= \frac{1}{Ti} P_{3A}(\xi_3^{in}), \quad 0 \leq \xi_3^{in} \leq Ti. \end{aligned} \tag{5.36}$$

When $\xi_3^{in} \in [2Ti, 3Ti]$, we can simplify the derivation problem by introducing a new random variable $\xi_3^{in''} = \alpha_B$. Then, using (5.5) we get:

$$\begin{aligned} f_{MMT}^{Sc.3.R}(\xi_3^{in''}) &= P[\xi_3^{in''} = \alpha_{B_0}] \times P[\alpha_{B_0} < \alpha_A < \alpha_R] \\ &= \frac{1}{Ti} P_{3C}(\xi_3^{in''}), \quad 0 < \xi_3^{in''} \leq Ti. \end{aligned} \tag{5.37}$$

Lastly when $\xi_3^{in} \in [Ti, 2Ti]$, we can introduce another random variable $\xi_3^{in'} = \alpha_B$ in order to simplify the derivation problem. Note that this range consists of the complements of the probabilities associated with the previous two ranges in (5.36) and (5.37). Hence we write:

$$\begin{aligned} f_{MMT}^{Sc.3.R}(\xi_3^{in'}) &= P[\xi_3^{in'} = \alpha_{B_0}] \times (1 - P[\alpha_R < \alpha_A < \alpha_{B_0}] - P[\alpha_{B_0} < \alpha_A < \alpha_R]) \\ &= \frac{1}{Ti}(1 - P_{3A}(\xi_3^{in'}) - P_{3C}(\xi_3^{in'})), \quad 0 < \xi_3^{in'} \leq Ti. \end{aligned} \tag{5.38}$$

The assumptions made in (5.37) and (5.38) are relaxed by replacing $\xi_3^{in'}$ and $\xi_3^{in''}$ by $\xi_3^{in} - Ti$ and $\xi_3^{in} - 2Ti$ respectively. Then, by combining with (5.36), we write (5.39) for the *pdf* of ξ_3^{in} in MMT's Scenario Sc.3.R:

$$f_{MMT}^{Sc.3.R}(\xi_3^{in}) = \begin{cases} \frac{1}{Ti} P_{3A}(\xi_3^{in}), & 0 \leq \xi_3^{in} \leq Ti, \\ \frac{1}{Ti}(1 - P_{3A}(\xi_3^{in} - Ti) - P_{3C}(\xi_3^{in} - Ti)), & Ti < \xi_3^{in} \leq 2Ti, \\ \frac{1}{Ti} P_{3C}(\xi_3^{in} - 2Ti), & 2Ti < \xi_3^{in} \leq 3Ti. \end{cases} \tag{5.39}$$

5.1.7.2 Scenarios Sc.3.A, Sc.3.AB and Sc.3.AC

These scenarios are shown in Figures 5.39, 5.40 and 5.41, which are effectively the same as Scenario Sc.3.R where the MMT tree creation does not start till A is within the range of R to allow the MMT tree creation to B; hence we find:

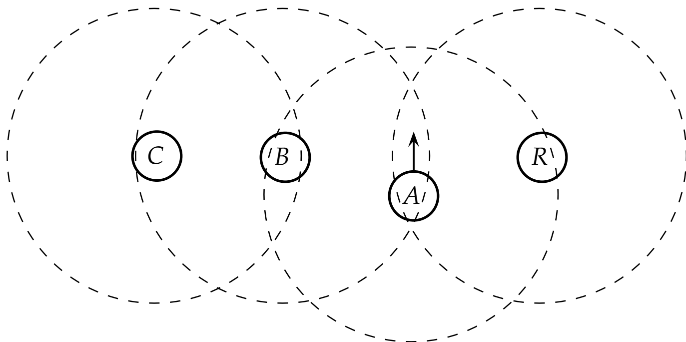

FIGURE 5.39: Scenario Sc.3.A for MMT

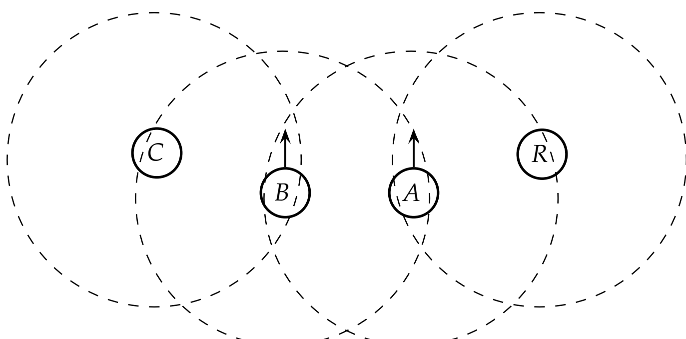

FIGURE 5.40: Scenario Sc.3.AB for MMT

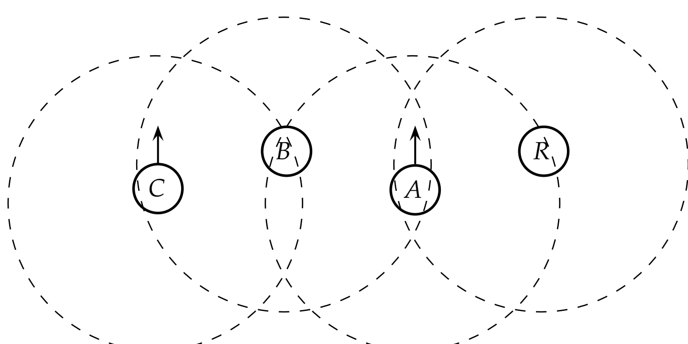

FIGURE 5.41: Scenario Sc.3.AC

$$f_{MMT}^{Sc.3.A}(\xi_3^{in}) = f_{MMT}^{Sc.3.R}(\xi_3^{in}) \tag{5.40}$$

$$f_{MMT}^{Sc.3.AB}(\xi_3^{in}) = f_{MMT}^{Sc.3.R}(\xi_3^{in}) \tag{5.41}$$

$$f_{MMT}^{Sc.3.AC}(\xi_3^{in}) = f_{MMT}^{Sc.3.R}(\xi_3^{in}) \tag{5.42}$$

Figure 5.42 depicts the model of $f_{MMT}^{Sc.3.R}(\xi_3^{in})$, $f_{MMT}^{Sc.3.A}(\xi_3^{in})$, $f_{MMT}^{Sc.3.AB}(\xi_3^{in})$ and $f_{MMT}^{Sc.3.AC}(\xi_3^{in})$ with $Ti = 2s$ against simulation results.

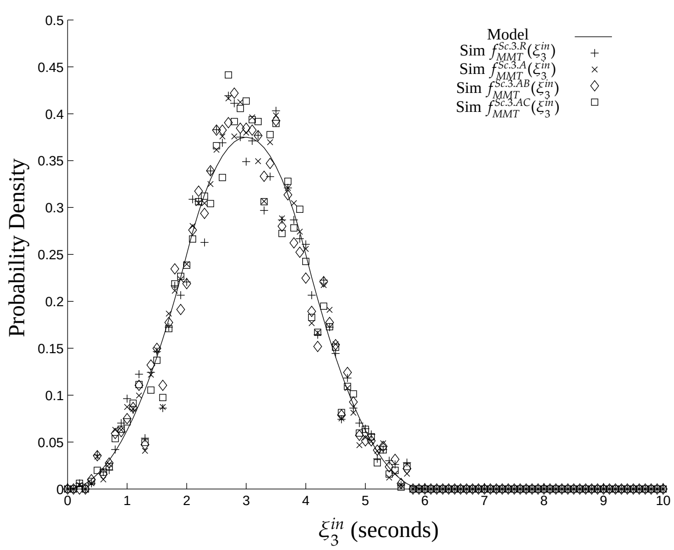

FIGURE 5.42: $f_{MMT}^{Sc.3.R}(\xi_3^{in})$, $f_{MMT}^{Sc.3.A}(\xi_3^{in})$, $f_{MMT}^{Sc.3.AB}(\xi_3^{in})$ and $f_{MMT}^{Sc.3.AC}(\xi_3^{in})$ with $Ti = 2s$

5.1.7.3 Scenarios Sc.3.B and Sc.3.BC

These scenarios are shown in Figures 5.43 and 5.44, which are effectively the same where the root R and node A have built the MMT tree between each other, as indicated by $A_0^{\Leftrightarrow}$ and $R_0^{\Leftrightarrow}$. Then, node B comes within the range of A. As a result, ξ_3^{in} reflects the time taken to extend the MMT tree to nodes B and C; in other words to two hops. This is a similar concept to what was tackled in section 5.1.5 considering Scenario Sc.2.A and Sc.2.R. Thus, we write (5.43) and (5.44):

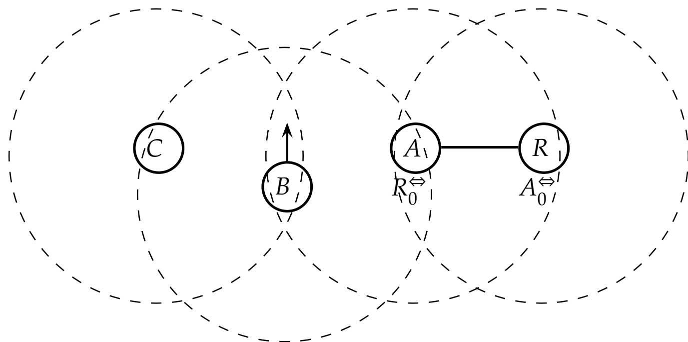

FIGURE 5.43: Scenario Sc.3.B

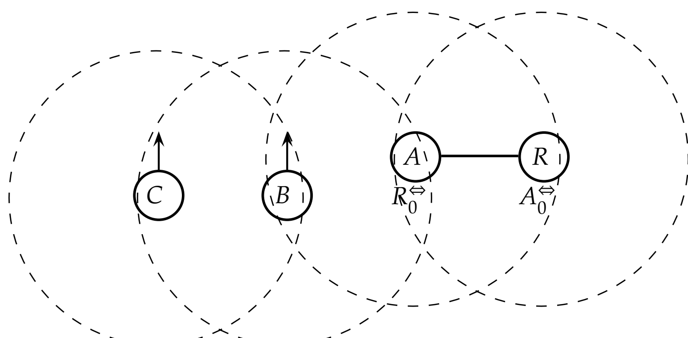

FIGURE 5.44: Scenario Sc.3.BC for MMT

$$f_{MMT}^{Sc.3.B}(\xi_3^{in}) = f_{MMT}^{Sc.2.R}(\xi_2^{in}) \tag{5.43}$$

$$f_{MMT}^{Sc.3.BC}(\xi_3^{in}) = f_{MMT}^{Sc.2.R}(\xi_2^{in}) \tag{5.44}$$

Figure 5.45 depicts the model of $f_{MMT}^{Sc.2.A}(\xi_2^{in})$, $f_{MMT}^{Sc.2.R}(\xi_2^{in})$, $f_{MMT}^{Sc.3.B}(\xi_3^{in})$ and $f_{MMT}^{Sc.3.BC}(\xi_3^{in})$ with $Ti = 2s$ against simulation results.

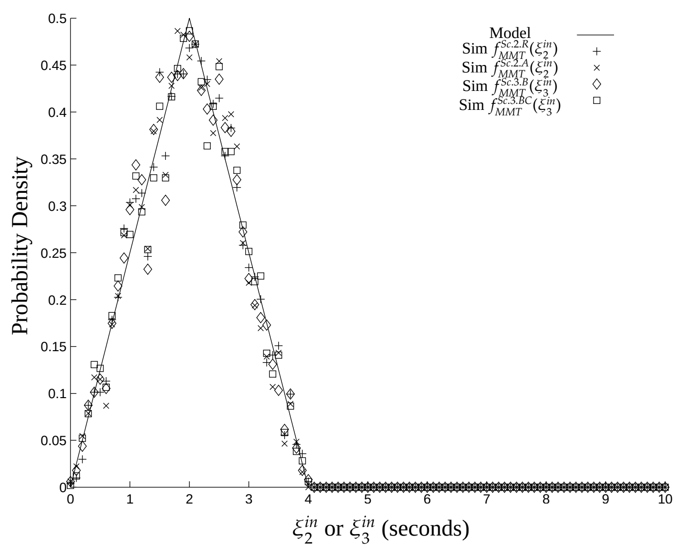

FIGURE 5.45: $f_{MMT}^{Sc.2.A}(\xi_2^{in})$, $f_{MMT}^{Sc.2.R}(\xi_2^{in})$, $f_{MMT}^{Sc.3.B}(\xi_3^{in})$ and $f_{MMT}^{Sc.3.BC}(\xi_3^{in})$ with $Ti = 2s$

5.1.7.4 Scenario Sc.3.C

Figure 5.46 shows the last possible scenario for forming three-hops topological path, *TPath*. In this scenario, nodes A and B already have the *VID*s $(R, 1, 1)$ and $(R, 11, 2)$, respectively, before extending the MMT tree another hop to include node C. This means that this scenario is similar to Scenario Sc.1.R and Sc.2.B discussed in sections 5.1.3 and 5.1.5. Then we conclude:

$$f_{MMT}^{Sc.3.C}(\xi_3^{in}) = f_{MMT}^{Sc.1.R}(\xi_1^{in}) \tag{5.45}$$

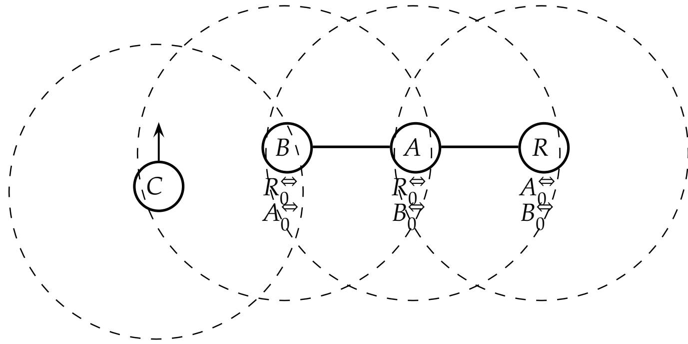

FIGURE 5.46: Scenario Sc.3.C

Figure 5.47 depicts the model of $f_{MMT}^{Sc.1.R}(\xi_1^{in})$, $f_{MMT}^{Sc.2.B}(\xi_2^{in})$ and $f_{MMT}^{Sc.3.C}(\xi_3^{in})$ with $Ti = 2s$ against simulation results.

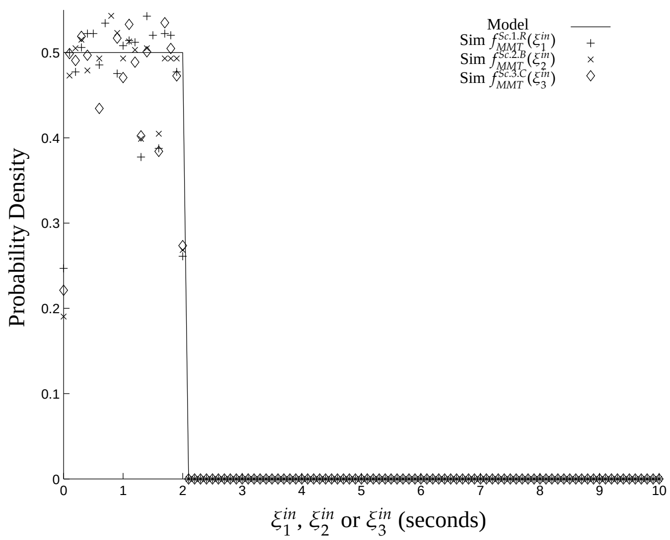

FIGURE 5.47: $f_{MMT}^{Sc.1.R}(\xi_1^{in})$, $f_{MMT}^{Sc.2.B}(\xi_2^{in})$ and $f_{MMT}^{Sc.3.C}(\xi_3^{in})$ with $Ti = 2s$

In the mobility model in section 3.3, a 3-hops *TPath* might form according to either Scenario Sc.3.R, Sc.3.BC or Sc.3.C with equal probabilities while Scenarios Sc.3.A, Sc.3.AB, Sc.3.AC and Sc.3.B have a probabilities close to zero. As mentioned before, this is due to the fact that the latter scenarios require the formation of two topological links, *TLinks*, at the same exact instant; For example, in Scenario Sc.3.A, *Tlinks* are formed at the same instant between nodes A and B and between A and R. As a result, $f_{MMT}(\xi_3^{in})$ is derived in (5.46) from (5.39), (5.44) and (5.45). In Figure 5.48, we show the model of $f_{MMT}(\xi_3^{in})$ against simulation results using mobility model in section 3.3 and simulation parameters in Table 5.2.

$$f_{MMT}(\xi_3^{in}) = \frac{f_{MMT}^{Sc.3.R}(\xi_3^{in}) + f_{MMT}^{Sc.3.BC}(\xi_3^{in}) + f_{MMT}^{Sc.3.C}(\xi_3^{in})}{3} \tag{5.46}$$

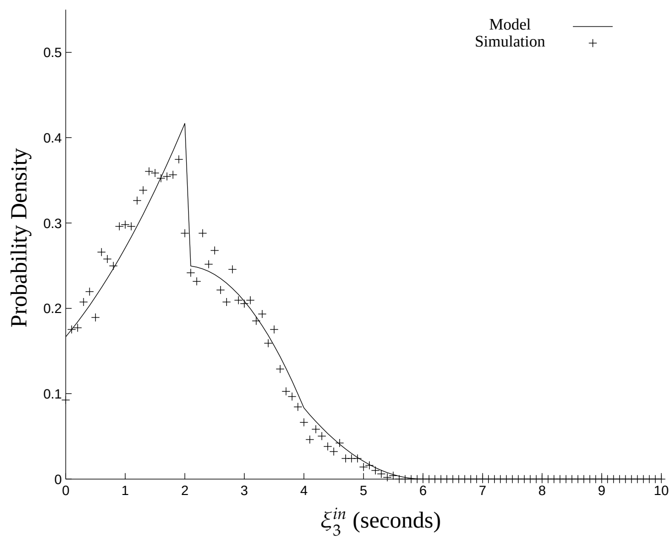

FIGURE 5.48: $f_{MMT}(\xi_3^{in})$ with $Ti = 2s$

5.1.8 Modeling ξ_3^{in} in OLSR

In any three hops OLSR scenario with nodes R, A, B and C, as shown in Figure 5.46, node R establishes a 3-hops *LPath* to node C after:

- Nodes R, A, B and C, exchange several *hello* packets to establish logical links, *LLinks*, with 1-hop neighbors and discover 2-hops neighbors.
- C selects B as an *MPR*, $\lambda_{C \to B}$, which is selected and signaled using *hello* packets exchanged between nodes C and B.

- B selects A as an MPR, $\lambda_{B \to A}$, through the exchange of *hello* packets between node B and A.
- Then, B sends a TC packet with C as $MPRSelector$.

In the previous discussion, we notice that establishing 3-hops *LPath* is achieved through *hello* packet exchange among all nodes in the scenario and TC packet sent by node B. Since the timing of sending *hello* and TC packets are controlled by the random variables α and β; the random variables involved in determining $\xi^{in}_{3_R}$ are α_R, α_A, α_B, α_C and β_B. Similarly, node C establishes a 3-hops *LPath* to node R after:

- Nodes R, A, B and C establish *LLinks* with 1-hop neighbors and discover 2-hops neighbors.
- R selects A as an MPR, $\lambda_{R \to A}$.
- A selects B as an MPR, $\lambda_{A \to B}$.
- Then, A sends a TC packet with R as $MPRSelector$.

Hence, the random variables involved in determining $\xi^{in}_{3_C}$ are α_R, α_A, α_B, α_C and β_A.

5.1.8.1 Scenario Sc.3.C

This scenario is shown in Figure 5.46 which is one of the scenarios for forming 3-hops logical paths, *LPaths*, for OLSR routing protocol. We notice that nodes A, B and C are similar in role to those in Figure 5.13 after renaming R, A and B as A, B and C, respectively. Referring to Table 5.3 and applying the renaming, we form Table 5.6 applied to the behavior of nodes A, B and C in the present scenario. To model $\xi^{in}_{3_R}$, we form the Table 5.7, which we call the combination table as it combines the behavior of several nodes and the associated events of selecting *MPRs* and sending TC packets. The first column has all the possible ordering of α_R, α_A, α_B and α_C. The second column is filled with the data shown in Table 5.6. For example, the second column of the first condition in Table 5.7, $\alpha_R < \alpha_A < \alpha_B < \alpha_C$, is filled with the value of $\lambda_{C \to B}$ for the third condition, $\alpha_A < \alpha_B < \alpha_C$, in Table 5.6. Notice that we are ignoring α_R from the condition in Table 5.7 to get the corresponding condition in Table 5.6. The third column in Table

5.7 is always zero since nodes R, A and B already have formed a line topology before the time lapse being modeled. In Table 5.7, The column labeled "biggest" holds the biggest value of all λ, in this case, $\lambda_{A\to B}$ or $\lambda_{C\to B}$. Notice that the last two columns are representing the delay $\xi^{in}_{3_R}$; which is the time elapsed from the instant T^{in}_T till the time when B sends a TC packet containing C as $MPRSelector$. The value of this delay is $(xTi)+\beta_B$ when $\alpha_C<\beta_B$ where x is the coefficient of Ti in the column labeled "biggest"; otherwise the TC packet has to wait another Ti seconds making the delay $((x+1)Ti)+\beta_B$.

TABLE 5.6: Renaming Instance C

Condition	&	$\lambda_{A\to B}$	$\lambda_{C\to B}$
$\alpha_B<\alpha_A<\alpha_C$	n/a	$Ti+\alpha_A$	$Ti+\alpha_C$
$\alpha_B<\alpha_C<\alpha_A$	n/a	$Ti+\alpha_A$	$Ti+\alpha_C$
$\alpha_A<\alpha_B<\alpha_C$	n/a	$2Ti+\alpha_A$	$Ti+\alpha_C$
$\alpha_C<\alpha_B<\alpha_A$	n/a	$Ti+\alpha_A$	$Ti+\alpha_C$
$\alpha_A<\alpha_C<\alpha_B$	n/a	$2Ti+\alpha_A$	$Ti+\alpha_C$
$\alpha_C<\alpha_A<\alpha_B$	y	$2Ti+\alpha_A$	$Ti+\alpha_C$
	z	$Ti+\alpha_A$	
y is $\beta_B<\alpha_C\vert\beta_B>\alpha_A$, z is $\alpha_C<\beta_B<\alpha_A$			

Therefore, from Table 5.7 we notice that $\xi^{in}_{3_R}$ in Scenario Sc.3.C has the following values:

$$\xi^{in}_{3_R}\text{in OLSR for Sc.3.C} = \begin{cases} Ti+\beta_B, & \alpha_C<\beta_B, \\ 2Ti+\beta_B, & \alpha_C>\beta_B. \end{cases} \tag{5.47}$$

From (5.47), we notice that $\xi^{in}_{3_R}$ has support over two different ranges of values, the first is $[Ti,2Ti]$ and the second is $[2Ti,3Ti]$. When $\xi^{in}_{3_R}\in[Ti,2Ti]$, we simplify the derivation problem by introducing a new random variable $\xi^{in'}_{3_R}=\beta_B$. Then using (5.1), we get the following:

$$\begin{aligned} f^{Sc.3.C}_{OLSR}(\xi^{in'}_{3_R}) &= P[\xi^{in'}_{3_R}=\beta_{B_0}]\times P[\alpha_C<\beta_{B_0}] \\ &= \tfrac{1}{Ti}P_{2A}(\xi^{in'}_{3_R}), \quad 0\le\xi^{in'}_{3_R}\le Ti. \end{aligned} \tag{5.48}$$

TABLE 5.7: Deriving $\xi_{3_R}^{in}$ in Scenario Sc.3.C for OLSR

Condition	$\lambda_{C\to B}$	$\lambda_{B\to A}$	biggest	$\xi_{3_R}^{in}$	
				$\alpha_C < \beta_B$	$\alpha_C > \beta_B$
$\alpha_R < \alpha_A < \alpha_B < \alpha_C$	$Ti + \alpha_C$	0.0	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_R < \alpha_A < \alpha_C < \alpha_B$	$Ti + \alpha_C$	0.0	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_R < \alpha_B < \alpha_A < \alpha_C$	$Ti + \alpha_C$	0.0	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_R < \alpha_B < \alpha_C < \alpha_A$	$Ti + \alpha_C$	0.0	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_R < \alpha_C < \alpha_A < \alpha_B$	$Ti + \alpha_C$	0.0	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_R < \alpha_C < \alpha_B < \alpha_A$	$Ti + \alpha_C$	0.0	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_A < \alpha_R < \alpha_B < \alpha_C$	$Ti + \alpha_C$	0.0	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_A < \alpha_R < \alpha_C < \alpha_B$	$Ti + \alpha_C$	0.0	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_A < \alpha_B < \alpha_R < \alpha_C$	$Ti + \alpha_C$	0.0	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_A < \alpha_B < \alpha_C < \alpha_R$	$Ti + \alpha_C$	0.0	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_A < \alpha_C < \alpha_R < \alpha_B$	$Ti + \alpha_C$	0.0	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_A < \alpha_C < \alpha_B < \alpha_R$	$Ti + \alpha_C$	0.0	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_B < \alpha_R < \alpha_A < \alpha_C$	$Ti + \alpha_C$	0.0	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_B < \alpha_R < \alpha_C < \alpha_A$	$Ti + \alpha_C$	0.0	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_B < \alpha_A < \alpha_R < \alpha_C$	$Ti + \alpha_C$	0.0	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_B < \alpha_A < \alpha_C < \alpha_R$	$Ti + \alpha_C$	0.0	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_B < \alpha_C < \alpha_R < \alpha_A$	$Ti + \alpha_C$	0.0	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_B < \alpha_C < \alpha_A < \alpha_R$	$Ti + \alpha_C$	0.0	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_C < \alpha_R < \alpha_A < \alpha_B$	$Ti + \alpha_C$	0.0	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_C < \alpha_R < \alpha_B < \alpha_A$	$Ti + \alpha_C$	0.0	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_C < \alpha_A < \alpha_R < \alpha_B$	$Ti + \alpha_C$	0.0	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_C < \alpha_A < \alpha_B < \alpha_R$	$Ti + \alpha_C$	0.0	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_C < \alpha_B < \alpha_R < \alpha_A$	$Ti + \alpha_C$	0.0	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_C < \alpha_B < \alpha_A < \alpha_R$	$Ti + \alpha_C$	0.0	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$

When $\xi_{3_R}^{in} \in [2Ti, 3Ti]$, we introduce a new random variable $\xi_{3_R}^{in''} = \beta_B$. Then using (5.2), we get:

$$\begin{aligned} f_{OLSR}^{Sc.3.C}(\xi_{3_R}^{in''}) &= P[\xi_{3_R}^{in''} = \beta_{B_0}] \times P[\alpha_C > \beta_{B_0}] \\ &= \tfrac{1}{Ti} P_{2B}(\xi_{3_R}^{in''}), \quad 0 < \xi_{3_R}^{in''} \leq Ti. \end{aligned} \tag{5.49}$$

Next, the assumption made in (5.48) and (5.49) are relaxed by replacing $\xi_{3_R}^{in'}$ and $\xi_{3_R}^{in''}$ by $\xi_{3_R}^{in} - Ti$ and $\xi_{3_R}^{in} - 2Ti$, respectively. Then, by combining (5.48) and (5.49) we see that the *pdf* of $\xi_{3_R}^{in}$ in Scenario Sc.3.C running OLSR can be written as:

$$f_{OLSR}^{Sc.3.C}(\xi_{3_R}^{in}) = \begin{cases} \frac{1}{Ti}P_{2A}(\xi_{3_R}^{in} - Ti), & Ti \leq \xi_{3_R}^{in} \leq 2Ti, \\ \frac{1}{Ti}P_{2B}(\xi_{3_R}^{in} - 2Ti), & 2Ti < \xi_{3_R}^{in} \leq 3Ti, \\ 0, & \text{otherwise}, \end{cases} \tag{5.50}$$

Figure 5.49 depicts the comparison of the modeled $f_{OLSR}^{Sc.3.C}(\xi_{3_R}^{in})$ with $Ti = 2s$ and simulation results.

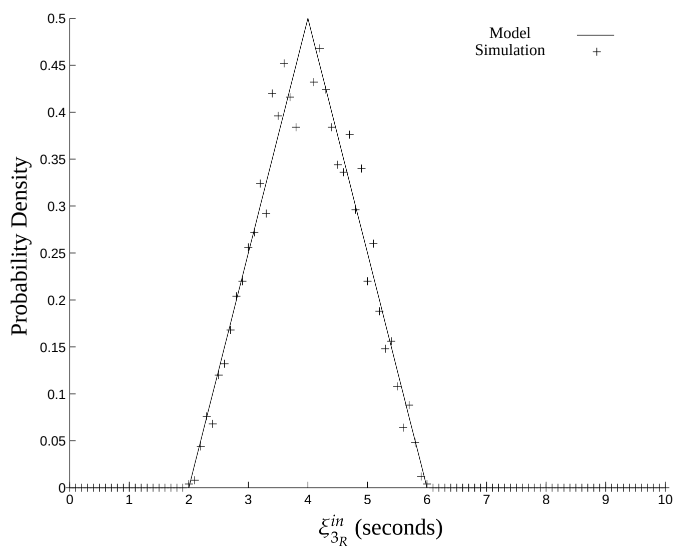

FIGURE 5.49: $f_{OLSR}^{Sc.3.C}(\xi_{3_R}^{in})$ with $Ti = 2s$

The Scenario Sc.3.C is not symmetrical and viewed differently from nodes' R and C perspectives. As a result, to model $\xi_{3_C}^{in}$, we form the combination Table 5.8. Here the column of $\lambda_{R\to A}$ is always zero, while the values of $\lambda_{A\to B}$ are filled from Table 5.6. The biggest column holds the biggest value between $\lambda_{R\to A}$ or $\lambda_{A\to B}$. The last two columns represents $\xi_{3_C}^{in}$ which is $(xTi) + \beta_A$ when $\alpha_A < \beta_A$ where x is the coefficient of Ti in the biggest column. Otherwise the delay is $((x+1)Ti) + \beta_B$.

TABLE 5.8: Deriving $\xi^{in}_{3_C}$ in Scenario Sc.3.C for OLSR

Condition	&	$\lambda_{R\to A}$	$\lambda_{A\to B}$	biggest	$\xi^{in}_{3_C}$	
					$\alpha_A < \beta_A$	$\alpha_A > \beta_A$
$\alpha_R < \alpha_A < \alpha_B < \alpha_C$	n/a	0.0	$2Ti + \alpha_A$	$2Ti + \alpha_A$	$2Ti + \beta_A$	$3Ti + \beta_A$
$\alpha_R < \alpha_A < \alpha_C < \alpha_B$	n/a	0.0	$2Ti + \alpha_A$	$2Ti + \alpha_A$	$2Ti + \beta_A$	$3Ti + \beta_A$
$\alpha_R < \alpha_B < \alpha_A < \alpha_C$	n/a	0.0	$Ti + \alpha_A$	$Ti + \alpha_A$	$Ti + \beta_A$	$2Ti + \beta_A$
$\alpha_R < \alpha_B < \alpha_C < \alpha_A$	n/a	0.0	$Ti + \alpha_A$	$Ti + \alpha_A$	$Ti + \beta_A$	$2Ti + \beta_A$
$\alpha_R < \alpha_C < \alpha_A < \alpha_B$	y	0.0	$2Ti + \alpha_A$	$2Ti + \alpha_A$	$2Ti + \beta_A$	$3Ti + \beta_A$
	z	0.0	$Ti + \alpha_A$	$Ti + \alpha_A$	$Ti + \beta_A$	$2Ti + \beta_A$
$\alpha_R < \alpha_C < \alpha_B < \alpha_A$	n/a	0.0	$Ti + \alpha_A$	$Ti + \alpha_A$	$Ti + \beta_A$	$2Ti + \beta_A$
$\alpha_A < \alpha_R < \alpha_B < \alpha_C$	n/a	0.0	$2Ti + \alpha_A$	$2Ti + \alpha_A$	$2Ti + \beta_A$	$3Ti + \beta_A$
$\alpha_A < \alpha_R < \alpha_C < \alpha_B$	n/a	0.0	$2Ti + \alpha_A$	$2Ti + \alpha_A$	$2Ti + \beta_A$	$3Ti + \beta_A$
$\alpha_A < \alpha_B < \alpha_R < \alpha_C$	n/a	0.0	$2Ti + \alpha_A$	$2Ti + \alpha_A$	$2Ti + \beta_A$	$3Ti + \beta_A$
$\alpha_A < \alpha_B < \alpha_C < \alpha_R$	n/a	0.0	$2Ti + \alpha_A$	$2Ti + \alpha_A$	$2Ti + \beta_A$	$3Ti + \beta_A$
$\alpha_A < \alpha_C < \alpha_R < \alpha_B$	n/a	0.0	$2Ti + \alpha_A$	$2Ti + \alpha_A$	$2Ti + \beta_A$	$3Ti + \beta_A$
$\alpha_A < \alpha_C < \alpha_B < \alpha_R$	n/a	0.0	$2Ti + \alpha_A$	$2Ti + \alpha_A$	$2Ti + \beta_A$	$3Ti + \beta_A$
$\alpha_B < \alpha_R < \alpha_A < \alpha_C$	n/a	0.0	$Ti + \alpha_A$	$Ti + \alpha_A$	$Ti + \beta_A$	$2Ti + \beta_A$
$\alpha_B < \alpha_R < \alpha_C < \alpha_A$	n/a	0.0	$Ti + \alpha_A$	$Ti + \alpha_A$	$Ti + \beta_A$	$2Ti + \beta_A$
$\alpha_B < \alpha_A < \alpha_R < \alpha_C$	n/a	0.0	$Ti + \alpha_A$	$Ti + \alpha_A$	$Ti + \beta_A$	$2Ti + \beta_A$
$\alpha_B < \alpha_A < \alpha_C < \alpha_R$	n/a	0.0	$Ti + \alpha_A$	$Ti + \alpha_A$	$Ti + \beta_A$	$2Ti + \beta_A$
$\alpha_B < \alpha_C < \alpha_R < \alpha_A$	n/a	0.0	$Ti + \alpha_A$	$Ti + \alpha_A$	$Ti + \beta_A$	$2Ti + \beta_A$
$\alpha_B < \alpha_C < \alpha_A < \alpha_R$	n/a	0.0	$Ti + \alpha_A$	$Ti + \alpha_A$	$Ti + \beta_A$	$2Ti + \beta_A$
$\alpha_C < \alpha_R < \alpha_A < \alpha_B$	y	0.0	$2Ti + \alpha_A$	$2Ti + \alpha_A$	$2Ti + \beta_A$	$3Ti + \beta_A$
	z	0.0	$Ti + \alpha_A$	$Ti + \alpha_A$	$Ti + \beta_A$	$2Ti + \beta_A$
$\alpha_C < \alpha_R < \alpha_B < \alpha_A$	n/a	0.0	$Ti + \alpha_A$	$Ti + \alpha_A$	$Ti + \beta_A$	$2Ti + \beta_A$
$\alpha_C < \alpha_A < \alpha_R < \alpha_B$	y	0.0	$2Ti + \alpha_A$	$2Ti + \alpha_A$	$2Ti + \beta_A$	$3Ti + \beta_A$
	z	0.0	$Ti + \alpha_A$	$Ti + \alpha_A$	$Ti + \beta_A$	$2Ti + \beta_A$
$\alpha_C < \alpha_A < \alpha_B < \alpha_R$	y	0.0	$2Ti + \alpha_A$	$2Ti + \alpha_A$	$2Ti + \beta_A$	$3Ti + \beta_A$
	z	0.0	$Ti + \alpha_A$	$Ti + \alpha_A$	$Ti + \beta_A$	$2Ti + \beta_A$
$\alpha_C < \alpha_B < \alpha_R < \alpha_A$	n/a	0.0	$Ti + \alpha_A$	$Ti + \alpha_A$	$Ti + \beta_A$	$2Ti + \beta_A$
$\alpha_C < \alpha_B < \alpha_A < \alpha_R$	n/a	0.0	$Ti + \alpha_A$	$Ti + \alpha_A$	$Ti + \beta_A$	$2Ti + \beta_A$
y is $\beta_B < \alpha_C \vert \beta_B > \alpha_A$, z is $\alpha_C < \beta_B < \alpha_A$						

The values of $\xi^{in}_{3_C}$ for Scenario Sc.3.C are shown in Table 5.8 which are simplified in several rounds in Table 5.9 showing that $\xi^{in}_{3_C}$ has three different ranges $[Ti, 2Ti]$, $[2Ti, 3Ti]$ and $[3Ti, 4Ti]$. When $\xi^{in}_{3_C} \in [Ti, 2Ti]$, we can simplify the derivation by introducing a new random variable $\xi^{in'}_{3_C} = \beta_A$. Using (5.3), (5.10) and (5.11), we find that:

$$\begin{aligned} f^{Sc.3.C}_{OLSR}(\xi^{in'}_{3_C}) &= P[\xi^{in'}_{3_C} = \beta_{A_0}] \times \Big(P[\alpha_B < \alpha_A < \beta_{A_0}] + P[\alpha_C < \beta_B < \alpha_A < \alpha_B < \beta_{A_0}] \\ &\quad + P[\alpha_C < \beta_B < \alpha_A < \beta_{A_0} < \alpha_B]\Big) \\ &= \frac{1}{Ti}\Big(P_{3A}(\xi^{in'}_{3_C}) + P_{5A}(\xi^{in'}_{3_C}) + P_{5B}(\xi^{in'}_{3_C})\Big), \quad 0 \le \xi^{in'}_{3_C} \le Ti. \end{aligned} \tag{5.51}$$

TABLE 5.9: Simplifying $\xi^{in}_{3_C}$ in Scenario Sc.3.C for OLSR

$\xi^{in}_{3_C}$	Condition
	Simplification Round 1
$Ti + \beta_A$	$(\alpha_B < \alpha_A < \alpha_C \,\&\, \alpha_A < \beta_A) \vert (\alpha_B < \alpha_C < \alpha_A \,\&\, \alpha_A < \beta_A) \vert (\alpha_C < \alpha_B < \alpha_A \,\&\, \alpha_A < \beta_A) \vert (\alpha_C < \alpha_A < \alpha_B \,\&\, \alpha_C < \beta_B < \alpha_A \,\&\, \alpha_A < \beta_A)$
$3Ti + \beta_A$	$(\alpha_A < \alpha_B < \alpha_C \,\&\, \alpha_A > \beta_A) \vert (\alpha_A < \alpha_C < \alpha_B \,\&\, \alpha_A > \beta_A) \vert (\alpha_C < \alpha_A < \alpha_B \,\&\, \alpha_A > \beta_A \,\&\, (\beta_B < \alpha_C \vert \beta_B > \alpha_A))$
$2Ti + \beta_A$	otherwise
	Simplification Round 2
$Ti + \beta_A$	$(\alpha_B < \alpha_A \,\&\, \alpha_A < \beta_A) \vert (\alpha_C < \beta_B < \alpha_A < \alpha_B \,\&\, \alpha_A < \beta_A)$
$3Ti + \beta_A$	$(\alpha_A < \alpha_B < \alpha_C \,\&\, \alpha_A > \beta_A) \vert (\alpha_A < \alpha_C < \alpha_B \,\&\, \alpha_A > \beta_A) \vert [(\beta_A < \alpha_C < \alpha_A < \alpha_B \vert \alpha_C < \beta_A < \alpha_A < \alpha_B) \,\&\, (\beta_B < \alpha_C \vert \beta_B > \alpha_A)]$
$2Ti + \beta_A$	otherwise
	Simplification Round 3
$Ti + \beta_A$	$\alpha_B < \alpha_A < \beta_A \vert \alpha_C < \beta_B < \alpha_A < \alpha_B < \beta_A \vert \alpha_C < \beta_B < \alpha_A < \beta_A < \alpha_B$
$3Ti + \beta_A$	$\beta_A < \alpha_A < \alpha_B < \alpha_C \vert \beta_A < \alpha_A < \alpha_C < \alpha_B \vert \beta_B < \beta_A < \alpha_C < \alpha_A < \alpha_B \vert \beta_A < \beta_B < \alpha_C < \alpha_A < \alpha_B \vert \beta_B < \alpha_C < \beta_A < \alpha_A < \alpha_B \vert \beta_A < \alpha_C < \alpha_A < \alpha_B < \beta_B \vert \beta_A < \alpha_C < \alpha_A < \beta_B < \alpha_B \vert \alpha_C < \beta_A < \alpha_A < \alpha_B < \beta_B \vert \alpha_C < \beta_A < \alpha_A < \beta_B < \alpha_B$
$2Ti + \beta_A$	otherwise

Before we consider the case when $\xi^{in}_{3_C} \in [2Ti, 3Ti]$, we will consider the case when $\xi^{in}_{3_C} \in [3Ti, 4Ti]$. This out of order derivation will become handy as the number of conditions associated with the case when $\xi^{in}_{3_C} \in [2Ti, 3Ti]$ is larger than those in the case when $\xi^{in}_{3_C} \in [3Ti, 4Ti]$; then deriving for the case when $\xi^{in}_{3_C} \in [2Ti, 3Ti]$ is just a matter of taking the complements of the probabilities associated with the cases when $\xi^{in}_{3_C} \in [Ti, 2Ti]$ and $\xi^{in}_{3_C} \in [3Ti, 4Ti]$.

Hence, $\xi^{in}_{3_C} \in [3Ti, 4Ti]$, we can simplify the derivation by introducing another random variable $\xi^{in'''}_{3_C} = \beta_A$. Using (5.9), (5.12), (5.13) and (5.14), we obtain:

$$
\begin{aligned}
f^{Sc.3.C}_{OLSR}(\xi^{in'''}_{3_C}) = P[\xi^{in'''}_{3_C} = \beta_{A_0}] \times \Big(& P[\beta_{A_0} < \alpha_A < \alpha_B < \alpha_C] + P[\beta_{A_0} < \alpha_A < \alpha_C < \alpha_B] \\
& + P[\beta_B < \beta_{A_0} < \alpha_C < \alpha_A < \alpha_B] + P[\beta_{A_0} < \beta_B < \alpha_C < \alpha_A < \alpha_B] \\
& + P[\beta_B < \alpha_C < \beta_{A_0} < \alpha_A < \alpha_B] + P[\beta_{A_0} < \alpha_C < \alpha_A < \alpha_B < \beta_B] \\
& + P[\beta_{A_0} < \alpha_C < \alpha_A < \beta_B < \alpha_B] + P[\alpha_C < \beta_{A_0} < \alpha_A < \alpha_B < \beta_B] \\
& + P[\alpha_C < \beta_{A_0} < \alpha_A < \beta_B < \alpha_B]\Big) \\
& = \frac{1}{Ti}\Big(2P_{4D}(\xi^{in'''}_{3_C}) + P_{5C}(\xi^{in'''}_{3_C}) \\
& + 3P_{5D}(\xi^{in'''}_{3_C}) + 3P_{5E}(\xi^{in'''}_{3_C})\Big), \quad 0 \le \xi^{in'''}_{3_C} \le Ti.
\end{aligned} \tag{5.52}
$$

Lastly when $\xi^{in}_{3} \in [2Ti, 3Ti]$, we can introduce a third random variable $\xi^{in''}_{3} = \beta_A$ used to aid with the derivation problem. Note that this range consists of the complements of the probabilities associated with the previous two ranges in (5.51) and (5.52). Hence we write:

$$
\begin{aligned}
f^{Sc.3.C}_{OLSR}(\xi^{in''}_{3_C}) = P[\xi^{in''}_{3_C} = \beta_{A_0}] \times \Big(& 1 - P[\alpha_B < \alpha_A < \beta_{A_0}] - P[\alpha_C < \beta_B < \alpha_A < \alpha_B < \beta_{A_0}] \\
& - P[\alpha_C < \beta_B < \alpha_A < \beta_{A_0} < \alpha_B] - P[\beta_{A_0} < \alpha_A < \alpha_B < \alpha_C] \\
& - P[\beta_{A_0} < \alpha_A < \alpha_C < \alpha_B] - P[\beta_B < \beta_{A_0} < \alpha_C < \alpha_A < \alpha_B] \\
& - P[\beta_{A_0} < \beta_B < \alpha_C < \alpha_A < \alpha_B] - P[\beta_B < \alpha_C < \beta_{A_0} < \alpha_A < \alpha_B] \\
& - P[\beta_{A_0} < \alpha_C < \alpha_A < \alpha_B < \beta_B] - P[\beta_{A_0} < \alpha_C < \alpha_A < \beta_B < \alpha_B] \\
& - P[\alpha_C < \beta_{A_0} < \alpha_A < \alpha_B < \beta_B] - P[\alpha_C < \beta_{A_0} < \alpha_A < \beta_B < \alpha_B]\Big) \\
& = \frac{1}{Ti}\Big(1 - P_{3A}(\xi^{in''}_{3_C}) - P_{5A}(\xi^{in''}_{3_C}) - P_{5B}(\xi^{in''}_{3_C}) - 2P_{4D}(\xi^{in''}_{3_C}) \\
& - P_{5C}(\xi^{in''}_{3_C}) - 3P_{5D}(\xi^{in''}_{3_C}) - 3P_{5E}(\xi^{in''}_{3_C})\Big), \quad 0 \le \xi^{in''}_{3_C} \le Ti.
\end{aligned} \tag{5.53}
$$

The assumptions made in (5.51), (5.52) and (5.53) are relaxed by replacing $\xi^{in'}_{3_C}$, $\xi^{in''}_{3_C}$ and $\xi^{in'''}_{3_C}$ by $\xi^{in}_{3_C} - Ti$, $\xi^{in}_{3_C} - 2Ti$ and $\xi^{in}_{3_C} - 3Ti$ respectively. Then, by combining them, we write (5.54) for the *pdf* of $\xi^{in}_{3_C}$ in Scenario Sc.3.C for OLSR:

$$f_{OLSR}^{Sc.3.C}(\xi_{3_C}^{in}) = \begin{cases} \frac{1}{Ti}R(\xi_{3_C}^{in}), & Ti < \xi_{3_C}^{in} \leq 2Ti, \\ \frac{1}{Ti}\Big(1 - R(\xi_{3_C}^{in} - Ti) - S(\xi_{3_C}^{in} + Ti)\Big), & 2Ti < \xi_{3_C}^{in} \leq 3Ti, \\ \frac{1}{Ti}S(\xi_{3_C}^{in}), & 3Ti < \xi_{3_C}^{in} \leq 4Ti, \\ 0, & \text{otherwise}, \end{cases} \tag{5.54}$$

where $R(\xi_{3_C}^{in}) = P_{3A}(\xi_{3_C}^{in} - Ti) + P_{5A}(\xi_{3_C}^{in} - Ti) + P_{5B}(\xi_{3_C}^{in} - Ti)$ and $S(\xi_{3_C}^{in}) = 2P_{4D}(\xi_{3_C}^{in} - 3Ti) + 3P_{5D}(\xi_{3_C}^{in} - 3Ti) + 3P_{5E}(\xi_{3_C}^{in} - 3Ti) + P_{5C}(\xi_{3_C}^{in} - 3Ti)$. The *pdf* $f_{OLSR}^{Sc.3.C}(\xi_{3_C}^{in})$ is shown in Figure 5.50 and compared to simulation results with $Ti = 2s$.

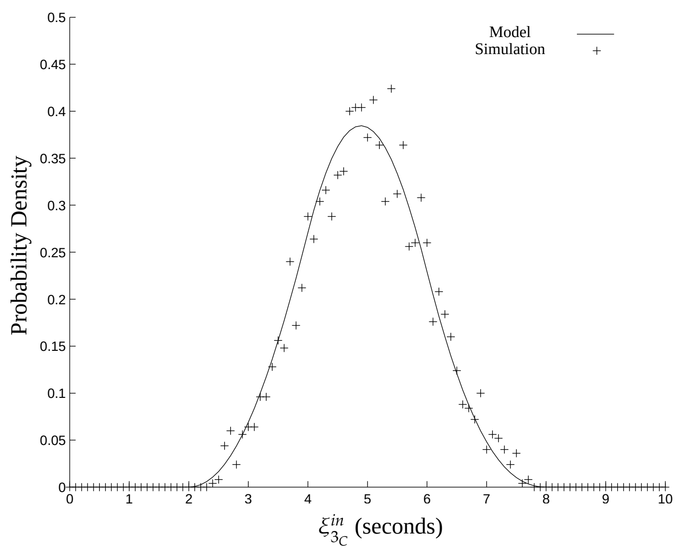

FIGURE 5.50: $f_{OLSR}^{Sc.3.C}(\xi_{3_C}^{in})$ with $Ti = 2s$

5.1.8.2 Scenario Sc.3.B

This scenario is shown in Figure 5.43 where nodes R, A and B are in the same setup as in Figure 5.13. Considering nodes A,B and C, we observe that they play similar role to those in Figure 5.7 after renaming R, A and B as A, B and C, respectively. Referring to the findings in (5.28) and (5.29) and applying the renaming we obtain (5.55) and (5.56).

Then from Table 5.3, we form the combination Table 5.10 to aid the derivation of $\xi^{in}_{3_R}$ in Scenario Sc.3.B.

$$\lambda_{A\to B}\text{in OLSR} = \begin{cases} Ti + \alpha_R, & \alpha_B \leq \alpha_A, \\ 2Ti + \alpha_R, & \alpha_B > \alpha_A. \end{cases} \tag{5.55}$$

$$\lambda_{C\to B}\text{in OLSR} = \begin{cases} Ti + \alpha_C, & \alpha_B \leq \alpha_C, \\ 2Ti + \alpha_C, & \alpha_B > \alpha_C. \end{cases} \tag{5.56}$$

The values of $\xi^{in}_{3_R}$ for Scenario Sc.3.B shown in Table 5.10 are simplified in Table 5.11 which shows that the support of $\xi^{in}_{3_R}$ can be divided into three different ranges $[Ti, 2Ti]$, $[2Ti, 3Ti]$ and $[3Ti, 4Ti]$.

When $\xi^{in}_{3_R} \in [Ti, 2Ti]$, we can simplify the derivation problem by introducing a new random variable $\xi^{in'}_{3_R} = \beta_B$. Then, we use (5.3) to derive the following:

$$\begin{aligned} f^{Sc.3.B}_{OLSR}(\xi^{in'}_{3_R}) &= P[\xi^{in'}_{3_R} = \beta_{B_0}] \times P[\alpha_B < \alpha_C < \beta_{B_0}] \\ &= \tfrac{1}{Ti} P_{3A}(\xi^{in'}_{3_R}), \quad 0 \leq \xi^{in'}_{3_R} \leq Ti. \end{aligned} \tag{5.57}$$

When $\xi^{in}_{3_R} \in [3Ti, 4Ti]$, we will introduce a second random variable $\xi^{in'''}_{3_R} = \beta_B$ and using (5.5), we get:

$$\begin{aligned} f^{Sc.3.B}_{OLSR}(\xi^{in'''}_{3_R}) &= P[\xi^{in'''}_{3_R} = \beta_{B_0}] \times P[\beta_{B_0} < \alpha_C < \alpha_B] \\ &= \tfrac{1}{Ti} P_{3C}(\xi^{in'''}_{3_R}), \quad 0 \leq \xi^{in'''}_{3_R} \leq Ti. \end{aligned} \tag{5.58}$$

When $\xi^{in}_{3_R} \in [2Ti, 3Ti]$, we base the derivation on the fact that the associated probabilities in the present case consist of the complements of the probabilities shown in previous two cases, (5.57) and (5.58). To derive for this case, we assume the existence of a third variable $\xi^{in''}_{3_R} = \beta_B$. Hence, we find:

TABLE 5.10: Deriving $\xi_{3_R}^{in}$ in Scenario Sc.3.B for OLSR

Condition	$\lambda_{C\to B}$	$\lambda_{B\to A}$	biggest	$\xi_{3_R}^{in}$	
				$\alpha_C < \beta_B$	$\alpha_C > \beta_B$
$\alpha_R < \alpha_A < \alpha_B < \alpha_C$	$Ti+\alpha_C$	$Ti+\alpha_B$	$Ti+\alpha_C$	$Ti+\beta_B$	$2Ti+\beta_B$
$\alpha_R < \alpha_A < \alpha_C < \alpha_B$	$2Ti+\alpha_C$	$Ti+\alpha_B$	$2Ti+\alpha_C$	$2Ti+\beta_B$	$3Ti+\beta_B$
$\alpha_R < \alpha_B < \alpha_A < \alpha_C$	$Ti+\alpha_C$	$Ti+\alpha_B$	$Ti+\alpha_C$	$Ti+\beta_B$	$2Ti+\beta_B$
$\alpha_R < \alpha_B < \alpha_C < \alpha_A$	$Ti+\alpha_C$	$Ti+\alpha_B$	$Ti+\alpha_C$	$Ti+\beta_B$	$2Ti+\beta_B$
$\alpha_R < \alpha_C < \alpha_A < \alpha_B$	$2Ti+\alpha_C$	$Ti+\alpha_B$	$2Ti+\alpha_C$	$2Ti+\beta_B$	$3Ti+\beta_B$
$\alpha_R < \alpha_C < \alpha_B < \alpha_A$	$2Ti+\alpha_C$	$Ti+\alpha_B$	$2Ti+\alpha_C$	$2Ti+\beta_B$	$3Ti+\beta_B$
$\alpha_A < \alpha_R < \alpha_B < \alpha_C$	$Ti+\alpha_C$	$Ti+\alpha_B$	$Ti+\alpha_C$	$Ti+\beta_B$	$2Ti+\beta_B$
$\alpha_A < \alpha_R < \alpha_C < \alpha_B$	$2Ti+\alpha_C$	$Ti+\alpha_B$	$2Ti+\alpha_C$	$2Ti+\beta_B$	$3Ti+\beta_B$
$\alpha_A < \alpha_B < \alpha_R < \alpha_C$	$Ti+\alpha_C$	$Ti+\alpha_B$	$Ti+\alpha_C$	$Ti+\beta_B$	$2Ti+\beta_B$
$\alpha_A < \alpha_B < \alpha_C < \alpha_R$	$Ti+\alpha_C$	$Ti+\alpha_B$	$Ti+\alpha_C$	$Ti+\beta_B$	$2Ti+\beta_B$
$\alpha_A < \alpha_C < \alpha_R < \alpha_B$	$2Ti+\alpha_C$	$Ti+\alpha_B$	$2Ti+\alpha_C$	$2Ti+\beta_B$	$3Ti+\beta_B$
$\alpha_A < \alpha_C < \alpha_B < \alpha_R$	$2Ti+\alpha_C$	$Ti+\alpha_B$	$2Ti+\alpha_C$	$2Ti+\beta_B$	$3Ti+\beta_B$
$\alpha_B < \alpha_R < \alpha_A < \alpha_C$	$Ti+\alpha_C$	$Ti+\alpha_B$	$Ti+\alpha_C$	$Ti+\beta_B$	$2Ti+\beta_B$
$\alpha_B < \alpha_R < \alpha_C < \alpha_A$	$Ti+\alpha_C$	$Ti+\alpha_B$	$Ti+\alpha_C$	$Ti+\beta_B$	$2Ti+\beta_B$
$\alpha_B < \alpha_A < \alpha_R < \alpha_C$	$Ti+\alpha_C$	$Ti+\alpha_B$	$Ti+\alpha_C$	$Ti+\beta_B$	$2Ti+\beta_B$
$\alpha_B < \alpha_A < \alpha_C < \alpha_R$	$Ti+\alpha_C$	$Ti+\alpha_B$	$Ti+\alpha_C$	$Ti+\beta_B$	$2Ti+\beta_B$
$\alpha_B < \alpha_C < \alpha_R < \alpha_A$	$Ti+\alpha_C$	$Ti+\alpha_B$	$Ti+\alpha_C$	$Ti+\beta_B$	$2Ti+\beta_B$
$\alpha_B < \alpha_C < \alpha_A < \alpha_R$	$Ti+\alpha_C$	$Ti+\alpha_B$	$Ti+\alpha_C$	$Ti+\beta_B$	$2Ti+\beta_B$
$\alpha_C < \alpha_R < \alpha_A < \alpha_B$	$2Ti+\alpha_C$	$Ti+\alpha_B$	$2Ti+\alpha_C$	$2Ti+\beta_B$	$3Ti+\beta_B$
$\alpha_C < \alpha_R < \alpha_B < \alpha_A$	$2Ti+\alpha_C$	$Ti+\alpha_B$	$2Ti+\alpha_C$	$2Ti+\beta_B$	$3Ti+\beta_B$
$\alpha_C < \alpha_A < \alpha_R < \alpha_B$	$2Ti+\alpha_C$	$Ti+\alpha_B$	$2Ti+\alpha_C$	$2Ti+\beta_B$	$3Ti+\beta_B$
$\alpha_C < \alpha_A < \alpha_B < \alpha_R$	$2Ti+\alpha_C$	$Ti+\alpha_B$	$2Ti+\alpha_C$	$2Ti+\beta_B$	$3Ti+\beta_B$
$\alpha_C < \alpha_B < \alpha_R < \alpha_A$	$2Ti+\alpha_C$	$Ti+\alpha_B$	$2Ti+\alpha_C$	$2Ti+\beta_B$	$3Ti+\beta_B$
$\alpha_C < \alpha_B < \alpha_A < \alpha_R$	$2Ti+\alpha_C$	$Ti+\alpha_B$	$2Ti+\alpha_C$	$2Ti+\beta_B$	$3Ti+\beta_B$

$$
\begin{aligned}
f_{OLSR}^{Sc.3.B}(\xi_{3_R}^{in''}) &= P[\xi_{3_R}^{in''} = \beta_{B_0}] \times (1 - P[\alpha_B < \alpha_C < \beta_{B_0}] - P[\beta_{B_0} < \alpha_C < \alpha_B]) \\
&\frac{1}{Ti}\left(1 - P_{3A}(\xi_{3_R}^{in''}) - P_{3C}(\xi_{3_R}^{in''})\right), \quad 0 \le \xi_{3_R}^{in''} \le Ti. \qquad (5.59)
\end{aligned}
$$

The assumptions made in (5.57), (5.58) and (5.59) are relaxed by replacing $\xi_{3_R}^{in'}$, $\xi_{3_R}^{in''}$ and $\xi_{3_R}^{in'''}$ by $\xi_{3_R}^{in} - Ti$, $\xi_{3_R}^{in} - 2Ti$ and $\xi_{3_R}^{in} - 3Ti$ respectively. Then, by combining them, we obtain the *pdf* of $\xi_{3_R}^{in}$ in Scenario Sc.3.B running OLSR as:

TABLE 5.11: Simplifying $\xi^{in}_{3_R}$ in Scenario Sc.3.B for OLSR

$\xi^{in}_{3_R}$	Condition
Simplification Round 1	
$Ti+\beta_B$	$(\alpha_A<\alpha_B<\alpha_C\&\alpha_C<\beta_B)\vert(\alpha_B<\alpha_A<\alpha_C\&\alpha_C<\beta_B)\vert(\alpha_B<\alpha_C<\alpha_A\&\alpha_C<\beta_B)$
$3Ti+\beta_B$	$(\alpha_C<\alpha_A<\alpha_B\&\alpha_C>\beta_B)\vert(\alpha_A<\alpha_C<\alpha_B\&\alpha_C>\beta_B)\vert(\alpha_C<\alpha_B<\alpha_A\&\alpha_C>\beta_B)$
$2Ti+\beta_B$	otherwise
Simplification Round 2	
$Ti+\beta_B$	$\alpha_B<\alpha_C$ & $\alpha_C<\beta_B$
$3Ti+\beta_B$	$\alpha_C<\alpha_B$ & $\alpha_C>\beta_B$
$2Ti+\beta_B$	otherwise
Simplification Round 3	
$Ti+\beta_B$	$\alpha_B<\alpha_C<\beta_B$
$3Ti+\beta_B$	$\beta_B<\alpha_C<\alpha_B$
$2Ti+\beta_B$	otherwise

$$f^{Sc.3.B}_{OLSR}(\xi^{in}_{3_R})=\begin{cases}\frac{1}{Ti}P_{3A}(\xi^{in}_{3_R}-Ti), & Ti\le\xi^{in}_{3_R}\le 2Ti,\\ \frac{1}{Ti}\left(1-P_{3A}(\xi^{in}_{3_R}-2Ti)-P_{3C}(\xi^{in}_{3_R}-2Ti)\right), & 2Ti<\xi^{in}_{3_R}\le 3Ti,\\ \frac{1}{Ti}P_{3C}(\xi^{in}_{3_R}-3Ti), & 3Ti<\xi^{in}_{3_R}\le 4Ti,\\ 0, & \text{otherwise,}\end{cases}\tag{5.60}$$

The model of $f^{Sc.3.B}_{OLSR}(\xi^{in}_{3_R})$ is shown in Figure 5.51 and compared to simulation results with $Ti=2s$.

To model $\xi^{in}_{3_C}$ in Scenario Sc.3.B, we form the combination Table 5.12. Here the values for $\xi^{in}_{3_C}$ have four columns. When the biggest column is filled from the third column, we fill the sixth and seventh columns as in the conditions shown in the first and second row; otherwise, we fill the last two columns. The value $\xi^{in}_{3_C}$ is $(xTi)+\beta_A$ when α_A or α_R are $<\beta_A$; otherwise it is $((x+1)Ti)+\beta_A$ where x is the coefficient of Ti in the biggest column.

Table 5.13 shows the simplified values of $\xi^{in}_{3_C}$ which has three ranges $[Ti,2Ti]$, $[2Ti,3Ti]$ and $[3Ti,4Ti]$. When $\xi^{in}_{3_C}\in[Ti,2Ti]$, we will introduce a new random variable $\xi^{in'}_{3_C}=\beta_A$ to aid the derivation in this case. Using (5.6), we derive:

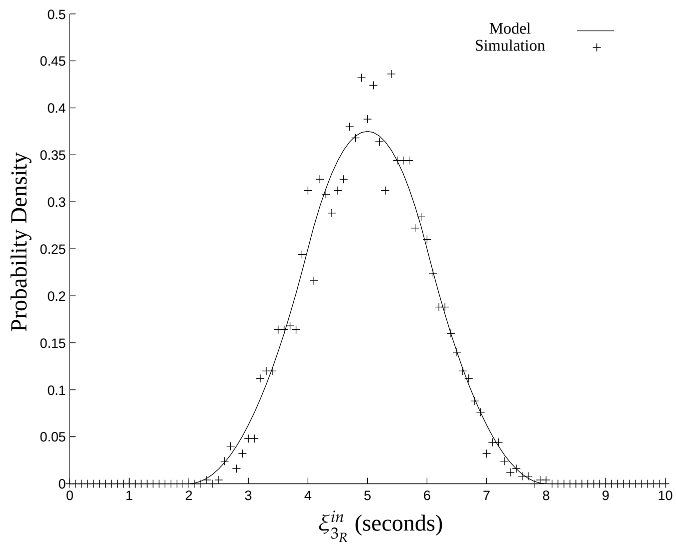

FIGURE 5.51: $f^{Sc.3.B}_{OLSR}(\xi^{in}_{3_R})$ with $Ti = 2s$

$$\begin{aligned} f^{Sc.3.B}_{OLSR}(\xi^{in'}_{3_C}) &= P[\xi^{in'}_{3_C} = \beta_{A_0}] \times P[\alpha_B < \alpha_A < \alpha_R < \beta_{A_0}] \\ &= \frac{1}{Ti} P_{4A}(\xi^{in'}_{3_C}), \quad 0 \le \xi^{in'}_{3_C} \le Ti. \end{aligned} \tag{5.61}$$

When $\xi^{in}_{3_C} \in [3Ti, 4Ti]$, we will introduce another new random variable $\xi^{in'''}_{3_C} = \beta_A$ and using (5.5) and (5.9), we get:

$$\begin{aligned} f^{Sc.3.B}_{OLSR}(\xi^{in'''}_{3_C}) = P[\xi^{in'''}_{3_C} = \beta_{A_0}] \times \Big(& P[\beta_{A_0} < \alpha_B < \alpha_R < \alpha_A] + P[\beta_{A_0} < \alpha_R < \alpha_B < \alpha_A] \\ & + P[\beta_{A_0} < \alpha_A < \alpha_B] \Big) \\ = \ & \frac{1}{Ti} \Big(2P_{4D}(\xi^{in'''}_{3_C}) + P_{3C}(\xi^{in'''}_{3_C}) \Big), \quad 0 \le \xi^{in'''}_{3_C} \le Ti. \end{aligned} \tag{5.62}$$

Lastly when $\xi^{in}_{3_C} \in [3Ti, 4Ti]$, we can derive for this case by introducing a third random variable $\xi^{in''}_{3_C} = \beta_A$ and considering the fact that this case complements the probabilities in (5.61) and (5.61). As a result, we find the following:

TABLE 5.12: Deriving $\xi^{in}_{3_C}$ in Scenario Sc.3.B for OLSR

Condition	&	$\lambda_{R\to A}$	$\lambda_{A\to B}$	biggest	$\xi^{in}_{3_C}$			
					$\alpha_A<\beta_A$	$\alpha_A>\beta_A$	$\alpha_R<\beta_A$	$\alpha_R>\beta_A$
$\alpha_R<\alpha_A<\alpha_B<\alpha_C$	n/a	$2Ti+\alpha_R$	$2Ti+\alpha_A$	$2Ti+\alpha_A$	$2Ti+\beta_A$	$3Ti+\beta_A$		
$\alpha_R<\alpha_A<\alpha_C<\alpha_B$	n/a	$2Ti+\alpha_R$	$2Ti+\alpha_A$	$2Ti+\alpha_A$	$2Ti+\beta_A$	$3Ti+\beta_A$		
$\alpha_R<\alpha_B<\alpha_A<\alpha_C$	n/a	$2Ti+\alpha_R$	$Ti+\alpha_A$	$2Ti+\alpha_R$			$2Ti+\beta_A$	$3Ti+\beta_A$
$\alpha_R<\alpha_B<\alpha_C<\alpha_A$	n/a	$2Ti+\alpha_R$	$Ti+\alpha_A$	$2Ti+\alpha_R$			$2Ti+\beta_A$	$3Ti+\beta_A$
$\alpha_R<\alpha_C<\alpha_A<\alpha_B$	n/a	$2Ti+\alpha_R$	$2Ti+\alpha_A$	$2Ti+\alpha_A$	$2Ti+\beta_A$	$3Ti+\beta_A$		
$\alpha_R<\alpha_C<\alpha_B<\alpha_A$	n/a	$2Ti+\alpha_R$	$Ti+\alpha_A$	$2Ti+\alpha_R$			$Ti+\beta_A$	$2Ti+\beta_A$
$\alpha_A<\alpha_R<\alpha_B<\alpha_C$	n/a	$Ti+\alpha_R$	$2Ti+\alpha_A$	$2Ti+\alpha_A$	$2Ti+\beta_A$	$3Ti+\beta_A$		
$\alpha_A<\alpha_R<\alpha_C<\alpha_B$	n/a	$Ti+\alpha_R$	$2Ti+\alpha_A$	$2Ti+\alpha_A$	$2Ti+\beta_A$	$3Ti+\beta_A$		
$\alpha_A<\alpha_B<\alpha_R<\alpha_C$	n/a	$Ti+\alpha_R$	$2Ti+\alpha_A$	$2Ti+\alpha_A$	$2Ti+\beta_A$	$3Ti+\beta_A$		
$\alpha_A<\alpha_B<\alpha_C<\alpha_R$	n/a	$Ti+\alpha_R$	$2Ti+\alpha_A$	$2Ti+\alpha_A$	$2Ti+\beta_A$	$3Ti+\beta_A$		
$\alpha_A<\alpha_C<\alpha_R<\alpha_B$	n/a	$Ti+\alpha_R$	$2Ti+\alpha_A$	$2Ti+\alpha_A$	$2Ti+\beta_A$	$3Ti+\beta_A$		
$\alpha_A<\alpha_C<\alpha_B<\alpha_R$	n/a	$Ti+\alpha_R$	$2Ti+\alpha_A$	$2Ti+\alpha_A$	$2Ti+\beta_A$	$3Ti+\beta_A$		
$\alpha_B<\alpha_R<\alpha_A<\alpha_C$	y	$2Ti+\alpha_R$	$Ti+\alpha_A$	$2Ti+\alpha_R$			$2Ti+\beta_A$	$3Ti+\beta_A$
	z	$Ti+\alpha_R$	$Ti+\alpha_A$	$Ti+\alpha_A$	$Ti+\beta_A$	$2Ti+\beta_A$		
$\alpha_B<\alpha_R<\alpha_C<\alpha_A$	y	$2Ti+\alpha_R$	$Ti+\alpha_A$	$2Ti+\alpha_R$			$2Ti+\beta_A$	$3Ti+\beta_A$
	z	$Ti+\alpha_R$	$Ti+\alpha_A$	$Ti+\alpha_A$	$Ti+\beta_A$	$2Ti+\beta_A$		
$\alpha_B<\alpha_A<\alpha_R<\alpha_C$	n/a	$Ti+\alpha_R$	$Ti+\alpha_A$	$Ti+\alpha_R$			$Ti+\beta_A$	$2Ti+\beta_A$
$\alpha_B<\alpha_A<\alpha_C<\alpha_R$	n/a	$Ti+\alpha_R$	$Ti+\alpha_A$	$Ti+\alpha_R$			$Ti+\beta_A$	$2Ti+\beta_A$
$\alpha_B<\alpha_C<\alpha_R<\alpha_A$	y	$2Ti+\alpha_R$	$Ti+\alpha_A$	$2Ti+\alpha_R$			$2Ti+\beta_A$	$3Ti+\beta_A$
	z	$Ti+\alpha_R$	$Ti+\alpha_A$	$Ti+\alpha_A$	$Ti+\beta_A$	$2Ti+\beta_A$		
$\alpha_B<\alpha_C<\alpha_A<\alpha_R$	n/a	$Ti+\alpha_R$	$Ti+\alpha_A$	$Ti+\alpha_R$			$Ti+\beta_A$	$2Ti+\beta_A$
$\alpha_C<\alpha_R<\alpha_A<\alpha_B$	n/a	$2Ti+\alpha_R$	$2Ti+\alpha_A$	$2Ti+\alpha_A$	$2Ti+\beta_A$	$3Ti+\beta_A$		
$\alpha_C<\alpha_R<\alpha_B<\alpha_A$	n/a	$2Ti+\alpha_R$	$Ti+\alpha_A$	$2Ti+\alpha_R$			$2Ti+\beta_A$	$3Ti+\beta_A$
$\alpha_C<\alpha_A<\alpha_R<\alpha_B$	n/a	$Ti+\alpha_R$	$2Ti+\alpha_A$	$2Ti+\alpha_A$	$2Ti+\beta_A$	$3Ti+\beta_A$		
$\alpha_C<\alpha_A<\alpha_B<\alpha_R$	n/a	$Ti+\alpha_R$	$2Ti+\alpha_A$	$2Ti+\alpha_A$	$2Ti+\beta_A$	$3Ti+\beta_A$		
$\alpha_C<\alpha_B<\alpha_R<\alpha_A$	y	$2Ti+\alpha_R$	$Ti+\alpha_A$	$2Ti+\alpha_R$			$2Ti+\beta_A$	$3Ti+\beta_A$
	z	$Ti+\alpha_R$	$Ti+\alpha_A$	$Ti+\alpha_A$	$Ti+\beta_A$	$2Ti+\beta_A$		
$\alpha_C<\alpha_B<\alpha_A<\alpha_R$	n/a	$Ti+\alpha_R$	$Ti+\alpha_A$	$Ti+\alpha_R$			$Ti+\beta_A$	$2Ti+\beta_A$
y is $\beta_A<\alpha_B \mid \beta_A>\alpha_R$, z is $\alpha_B<\beta_A<\alpha_R$								

$$
\begin{aligned}
f^{Sc.3.B}_{OLSR}(\xi^{in''}_{3_C}) &= P[\xi^{in''}_{3_C}=\beta_{A_0}]\times\Big(1-P[\alpha_B<\alpha_A<\alpha_R<\beta_{A_0}]-P[\beta_{A_0}<\alpha_B<\alpha_R<\alpha_A] \\
&\quad -P[\beta_{A_0}<\alpha_R<\alpha_B<\alpha_A]-P[\beta_{A_0}<\alpha_A<\alpha_B]\Big) \\
&= \frac{1}{Ti}\Big(1-P_{4A}(\xi^{in''}_{3_C})-2P_{4D}(\xi^{in''}_{3_C}) \\
&\quad -P_{3C}(\xi^{in''}_{3_C})\Big),\quad 0\le\xi^{in''}_{3_C}\le Ti.
\end{aligned}
\tag{5.63}
$$

TABLE 5.13: Simplifying $\xi^{in}_{3_C}$ in Scenario Sc.3.B for OLSR

$\xi^{in}_{3_C}$	Condition
Simplification Round 1	
$Ti+\beta_A$	$(\alpha_B<\alpha_R<\alpha_A \,\&\, \alpha_A<\beta_A \,\&\, \alpha_B<\beta_A<\alpha_R)\vert(\alpha_B<\alpha_A<\alpha_R \,\&\, \alpha_R<\beta_A)$
$3Ti+\beta_A$	$(\alpha_B<\alpha_R<\alpha_A \,\&\, \alpha_R>\beta_A \,\&\, (\beta_A<\alpha_B\vert\beta_A>\alpha_R))\vert(\alpha_R<\alpha_B<\alpha_A \,\&\, \alpha_R>\beta_A)\vert(\alpha_A<\alpha_B \,\&\, \alpha_A>\beta_A)$
$2Ti+\beta_A$	otherwise
Simplification Round 2	
$Ti+\beta_A$	$\alpha_B<\alpha_A<\alpha_R<\beta_A$
$3Ti+\beta_A$	$\beta_A<\alpha_B<\alpha_R<\alpha_A\vert\beta_A<\alpha_R<\alpha_B<\alpha_A\vert\beta_A<\alpha_A<\alpha_B$
$2Ti+\beta_A$	otherwise

The assumptions made in (5.61), (5.62) and (5.63) are relaxed by replacing $\xi^{in'}_{3_C}$, $\xi^{in''}_{3_C}$ and $\xi^{in'''}_{3_C}$ by $\xi^{in}_{3_C}-Ti$, $\xi^{in}_{3_C}-2Ti$ and $\xi^{in}_{3_C}-3Ti$ respectively. Then, by combining them, we write (5.64) for the *pdf* of $\xi^{in}_{3_C}$ in Scenario Sc.3.B running OLSR:

$$f^{Sc.3.B}_{OLSR}(\xi^{in}_{3_C}) = \begin{cases} \frac{1}{Ti}P_{4A}(\xi^{in}_{3_C}-Ti), & Ti<\xi^{in}_{3_C}\le 2Ti,\\ \frac{1}{Ti}\left(1-P_{4A}(\xi^{in}_{3_C}-2Ti)-T(\xi^{in}_{3_C}-2Ti)\right), & 2Ti<\xi^{in}_{3_C}\le 3Ti,\\ \frac{1}{Ti}T(\xi^{in}_{3_C}-3Ti), & 3Ti<\xi^{in}_{3_C}\le 4Ti,\\ 0, & \text{otherwise}, \end{cases} \tag{5.64}$$

where $T(\xi^{in}_{3_C})=2P_{4D}(\xi^{in}_{3_C})+P_{3C}(\xi^{in}_{3_C})$. The *pdf* of $f^{Sc.3.C}_{OLSR}(\xi^{in}_{3_C})$ is shown in Figure 5.52 and compared to simulation results with $Ti=2s$.

5.1.8.3 Scenario Sc.3.BC

This scenario is shown in Figure 5.53, shows that nodes R, A and B are of the same exact arrangement as in Figure 5.13, while the findings in Table 5.3 summarizes their behavior. However, nodes B, A and R in Figure 5.13 can be renamed as A, B and C respectively; then we apply the renaming to form Table 5.14 expressing the operation in this scenario. Finally, we create the combination Table 5.15.

The values of $\xi^{in}_{3_R}$ for Scenario Sc.3.BC in Table 5.15 are simplified in Table 5.16 which shows that $\xi^{in}_{3_R}$ has three different ranges $[Ti,2Ti]$, $[2Ti,3Ti]$ and $[3Ti,4Ti]$.

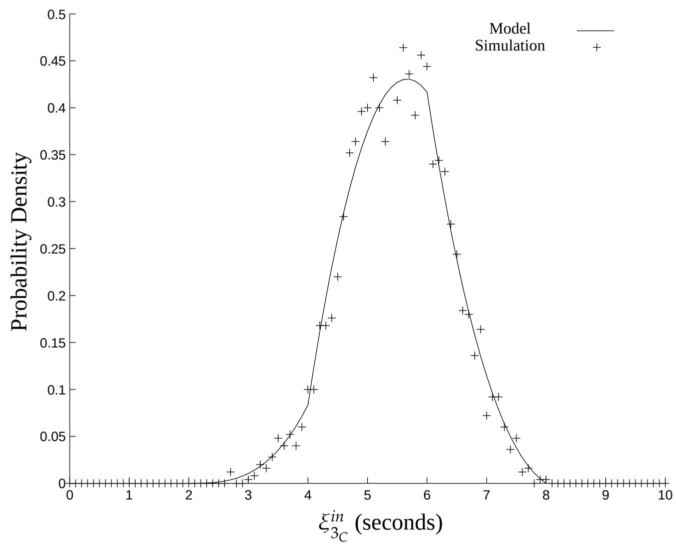

FIGURE 5.52: $f^{Sc.3.B}_{OLSR}(\xi^{in}_{3_C})$ with $Ti = 2s$

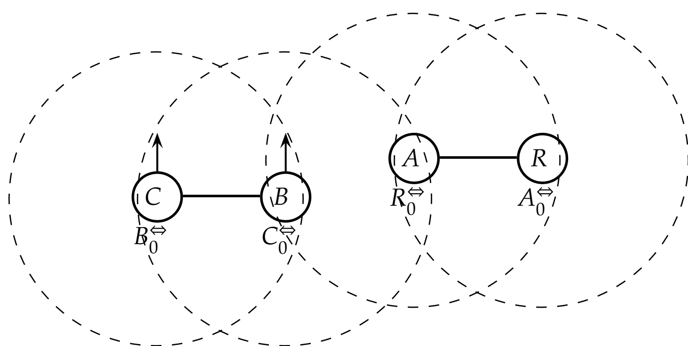

FIGURE 5.53: Scenario Sc.3.BC

When $\xi^{in}_{3_R} \in [Ti, 2Ti]$, we will introduce a new random variable $\xi^{in'}_{3_R} \in [0, Ti]$ which equals to β_B. Then, we use (5.3) to derive the following:

$$\begin{aligned} f^{Sc.3.BC}_{OLSR}(\xi^{in'}_{3_R}) &= P[\xi^{in'}_{3_R} = \beta_{B_0}] \times P[\alpha_B < \alpha_C < \beta_{B_0}] \\ &= \frac{1}{Ti} P_{3A}(\xi^{in'}_{3_R}), \quad 0 \leq \xi^{in'}_{3_R} \leq Ti. \end{aligned} \tag{5.65}$$

TABLE 5.14: Renaming Instance E

Condition	&	$\lambda_{C\to B}$	$\lambda_{A\to B}$
$\alpha_B < \alpha_C < \alpha_A$	n/a	$Ti + \alpha_C$	$Ti + \alpha_A$
$\alpha_B < \alpha_A < \alpha_C$	n/a	$Ti + \alpha_C$	$Ti + \alpha_A$
$\alpha_C < \alpha_B < \alpha_A$	n/a	$2Ti + \alpha_C$	$Ti + \alpha_A$
$\alpha_A < \alpha_B < \alpha_C$	n/a	$Ti + \alpha_C$	$Ti + \alpha_A$
$\alpha_C < \alpha_A < \alpha_B$	n/a	$2Ti + \alpha_C$	$Ti + \alpha_A$
$\alpha_A < \alpha_C < \alpha_B$	y	$2Ti + \alpha_C$	$Ti + \alpha_A$
	z	$Ti + \alpha_C$	
y is $\beta_B < \alpha_A \vert \beta_B > \alpha_C$, z is $\alpha_A < \beta_B < \alpha_C$			

When $\xi_{3_R}^{in} \in [3Ti, 4Ti]$, we introduce a new random variable $\xi_{3_R}^{in'''} = \beta_B$ and then we use (5.9) to get:

$$\begin{aligned} f_{OLSR}^{Sc.3.BC}(\xi_{3_R}^{in'''}) = {} & P[\xi_{3_R}^{in'''} = \beta_{B_0}] \times (P[\beta_{B_0} < \alpha_A < \alpha_C < \alpha_B] \\ & + P[\beta_{B_0} < \alpha_C < \alpha_A < \alpha_B] + P[\beta_{B_0} < \alpha_C < \alpha_B < \alpha_A]) \\ = {} & \tfrac{3}{Ti} P_{4D}(\xi_{3_R}^{in'''}), \quad 0 \le \xi_{3_R}^{in'''} \le Ti. \end{aligned} \tag{5.66}$$

Finally, when $\xi_{3_R}^{in} \in [2Ti, 3Ti]$ we will introduce a new random variable $\xi_{3_R}^{in''} = \beta_B$ then we use the complements of the probabilities presented in (5.65) and (5.66) to get:

$$\begin{aligned} f_{OLSR}^{Sc.3.BC}(\xi_{3_R}^{in''}) = {} & P[\xi_{3_R}^{in''} = \beta_{B_0}] \times (1 - P[\alpha_B < \alpha_C < \beta_{B_0}] - P[\beta_{B_0} < \alpha_A < \alpha_C < \alpha_B]) \\ & - P[\beta_{B_0} < \alpha_C < \alpha_A < \alpha_B] - P[\beta_{B_0} < \alpha_C < \alpha_B < \alpha_A]) \\ & \tfrac{1}{Ti}\left(1 - P_{3A}(\xi_{3_R}^{in''}) - 3P_{4D}(\xi_{3_R}^{in''})\right), \quad 0 \le \xi_{3_R}^{in''} \le Ti. \end{aligned} \tag{5.67}$$

The assumptions made in (5.65), (5.66) and (5.67) are relaxed by replacing $\xi_{3_R}^{in'}$, $\xi_{3_R}^{in''}$ and $\xi_{3_R}^{in'''}$ by $\xi_{3_R}^{in} - Ti$, $\xi_{3_R}^{in} - 2Ti$ and $\xi_{3_R}^{in} - 3Ti$ respectively. Then, by combining them, we write (5.68) for the *pdf* of $\xi_{3_R}^{in}$ in Scenario Sc.3.BC running OLSR:

TABLE 5.15: Deriving $\xi_{3_R}^{in}$ in Scenario Sc.3.BC for OLSR

Condition	&	$\lambda_{C\to B}$	$\lambda_{B\to A}$	biggest	$\xi_{3_R}^{in}$			
					$\alpha_C < \beta_B$	$\alpha_C > \beta_B$	$\alpha_B < \beta_B$	$\alpha_B > \beta_B$
$\alpha_R < \alpha_A < \alpha_B < \alpha_C$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$		
$\alpha_R < \alpha_A < \alpha_C < \alpha_B$	y	$2Ti + \alpha_C$	$Ti + \alpha_B$	$2Ti + \alpha_C$	$2Ti + \beta_B$	$3Ti + \beta_B$		
	z	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_B$			$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_R < \alpha_B < \alpha_A < \alpha_C$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$		
$\alpha_R < \alpha_B < \alpha_C < \alpha_A$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$		
$\alpha_R < \alpha_C < \alpha_A < \alpha_B$	n/a	$2Ti + \alpha_C$	$Ti + \alpha_B$	$2Ti + \alpha_C$	$2Ti + \beta_B$	$3Ti + \beta_B$		
$\alpha_R < \alpha_C < \alpha_B < \alpha_A$	n/a	$2Ti + \alpha_C$	$Ti + \alpha_B$	$2Ti + \alpha_C$	$2Ti + \beta_B$	$3Ti + \beta_B$		
$\alpha_A < \alpha_R < \alpha_B < \alpha_C$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$		
$\alpha_A < \alpha_R < \alpha_C < \alpha_B$	y	$2Ti + \alpha_C$	$Ti + \alpha_B$	$2Ti + \alpha_C$	$2Ti + \beta_B$	$3Ti + \beta_B$		
	z	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_B$			$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_A < \alpha_B < \alpha_R < \alpha_C$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$		
$\alpha_A < \alpha_B < \alpha_C < \alpha_R$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$		
$\alpha_A < \alpha_C < \alpha_R < \alpha_B$	y	$2Ti + \alpha_C$	$Ti + \alpha_B$	$2Ti + \alpha_C$	$2Ti + \beta_B$	$3Ti + \beta_B$		
	z	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_B$			$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_A < \alpha_C < \alpha_B < \alpha_R$	y	$2Ti + \alpha_C$	$Ti + \alpha_B$	$2Ti + \alpha_C$	$2Ti + \beta_B$	$3Ti + \beta_B$		
	z	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_B$			$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_B < \alpha_R < \alpha_A < \alpha_C$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$		
$\alpha_B < \alpha_R < \alpha_C < \alpha_A$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$		
$\alpha_B < \alpha_A < \alpha_R < \alpha_C$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$		
$\alpha_B < \alpha_A < \alpha_C < \alpha_R$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$		
$\alpha_B < \alpha_C < \alpha_R < \alpha_A$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$		
$\alpha_B < \alpha_C < \alpha_A < \alpha_R$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$		
$\alpha_C < \alpha_R < \alpha_A < \alpha_B$	n/a	$2Ti + \alpha_C$	$Ti + \alpha_B$	$2Ti + \alpha_C$	$2Ti + \beta_B$	$3Ti + \beta_B$		
$\alpha_C < \alpha_R < \alpha_B < \alpha_A$	n/a	$2Ti + \alpha_C$	$Ti + \alpha_B$	$2Ti + \alpha_C$	$2Ti + \beta_B$	$3Ti + \beta_B$		
$\alpha_C < \alpha_A < \alpha_R < \alpha_B$	n/a	$2Ti + \alpha_C$	$Ti + \alpha_B$	$2Ti + \alpha_C$	$2Ti + \beta_B$	$3Ti + \beta_B$		
$\alpha_C < \alpha_A < \alpha_B < \alpha_R$	n/a	$2Ti + \alpha_C$	$Ti + \alpha_B$	$2Ti + \alpha_C$	$2Ti + \beta_B$	$3Ti + \beta_B$		
$\alpha_C < \alpha_B < \alpha_R < \alpha_A$	n/a	$2Ti + \alpha_C$	$Ti + \alpha_B$	$2Ti + \alpha_C$	$2Ti + \beta_B$	$3Ti + \beta_B$		
$\alpha_C < \alpha_B < \alpha_A < \alpha_R$	n/a	$2Ti + \alpha_C$	$Ti + \alpha_B$	$2Ti + \alpha_C$	$2Ti + \beta_B$	$3Ti + \beta_B$		
y is $\beta_B < \alpha_A \mid \beta_B > \alpha_C$, z is $\alpha_A < \beta_B < \alpha_C$								

$$f_{OLSR}^{Sc.3.BC}(\xi_{3_R}^{in}) = \begin{cases} \frac{1}{Ti} P_{3A}(\xi_{3_R}^{in} - Ti), & Ti \le \xi_{3_R}^{in} \le 2Ti, \\ \frac{1}{Ti}\left(1 - P_{3A}(\xi_{3_R}^{in} - 2Ti) - 3P_{4D}(\xi_{3_R}^{in} - 2Ti)\right), & 2Ti < \xi_{3_R}^{in} \le 3Ti, \\ \frac{3}{Ti} P_{4D}(\xi_{3_R}^{in} - 3Ti), & 3Ti < \xi_{3_R}^{in} \le 4Ti, \\ 0, & \text{otherwise}, \end{cases} \tag{5.68}$$

TABLE 5.16: Simplifying $\xi^{in}_{3_R}$ in Scenario Sc.3.BC for OLSR

$\xi^{in}_{3_R}$	Condition
	Simplification Round 1
$Ti + \beta_B$	$(\alpha_A < \alpha_C < \alpha_B \,\&\, \alpha_B < \beta_B \,\&\, \alpha_A < \beta_B < \alpha_C)\|(\alpha_A < \alpha_B < \alpha_C \,\&\, \alpha_C < \beta_B)\|(\alpha_B < \alpha_A < \alpha_C \,\&\, \alpha_C < \beta_B)\|(\alpha_B < \alpha_C < \alpha_A \,\&\, \alpha_C < \beta_B)$
$3Ti + \beta_B$	$(\alpha_A < \alpha_C < \alpha_B \,\&\, \alpha_C > \beta_B \,\&\, (\beta_B < \alpha_A\|\beta_B > \alpha_C))\|(\alpha_C < \alpha_A < \alpha_B \,\&\, \alpha_C > \beta_B)\|(\alpha_C < \alpha_B < \alpha_A \,\&\, \alpha_C > \beta_B)$
$2Ti + \beta_B$	otherwise
	Simplification Round 2
$Ti + \beta_B$	$\alpha_B < \alpha_C < \beta_B$
$3Ti + \beta_B$	$\beta_B < \alpha_A < \alpha_C < \alpha_B\|\beta_B < \alpha_C < \alpha_A < \alpha_B\|\beta_B < \alpha_C < \alpha_B < \alpha_A$
$2Ti + \beta_B$	otherwise

$f^{Sc.3.BC}_{OLSR}(\xi^{in}_{3_R})$ is shown in Figure 5.54 and compared to simulation results with $Ti = 2s$.

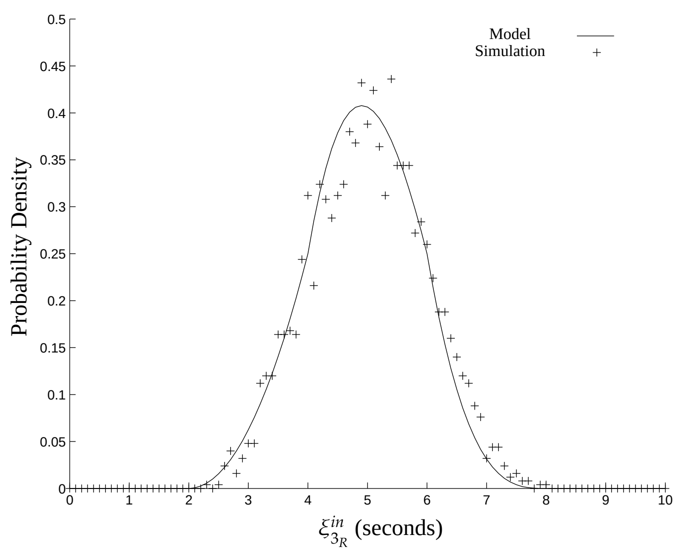

FIGURE 5.54: $f^{Sc.3.BC}_{OLSR}(\xi^{in}_{3_R})$ with $Ti = 2s$

5.1.8.4 Scenario Sc.3.AB

The scenario in Figure 5.55 can be reconstructed by combining two scenarios. the first scenario is obtained by renaming the nodes B, A and R in Figure 5.13 as C, B and A which their behavior is presented in Table 5.6. The second scenario is obtained renaming nodes B and R in Figure 5.13 as R and B, respectively, which their behavior is summarized in Table 5.17 leading to the formation of combination Table 5.18.

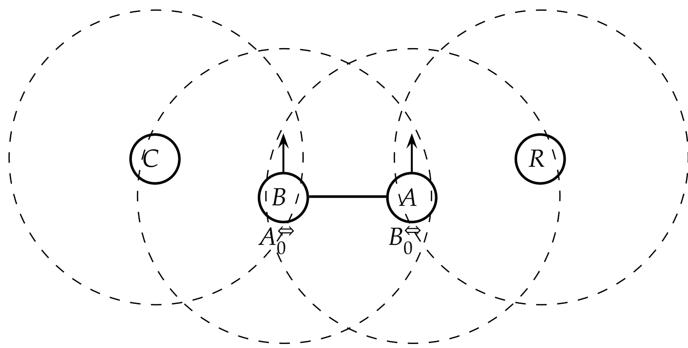

FIGURE 5.55: Scenario Sc.3.AB

TABLE 5.17: Renaming Instance F

Condition	&	$\lambda_{B\rightarrow A}$	$\lambda_{R\rightarrow A}$
$\alpha_B < \alpha_C < \alpha_A$	n/a	$Ti + \alpha_B$	$Ti + \alpha_R$
$\alpha_B < \alpha_A < \alpha_C$	n/a	$Ti + \alpha_B$	$Ti + \alpha_R$
$\alpha_C < \alpha_B < \alpha_A$	n/a	$2Ti + \alpha_B$	$Ti + \alpha_R$
$\alpha_A < \alpha_B < \alpha_C$	n/a	$Ti + \alpha_B$	$Ti + \alpha_R$
$\alpha_C < \alpha_A < \alpha_B$	n/a	$2Ti + \alpha_B$	$Ti + \alpha_R$
$\alpha_A < \alpha_C < \alpha_B$	y	$2Ti + \alpha_B$	$Ti + \alpha_R$
	z	$Ti + \alpha_C$	
y is $\beta_A < \alpha_R \vert \beta_A > \alpha_B$, z is $\alpha_R < \beta_A < \alpha_B$			

TABLE 5.18: Deriving $\xi_{3_R}^{in}$ in Scenario Sc.3.AB for OLSR

Condition	&	$\lambda_{C\to B}$	$\lambda_{B\to A}$	biggest	$\xi_{3_R}^{in}$			
					$\alpha_C < \beta_B$	$\alpha_C > \beta_B$	$\alpha_B < \beta_B$	$\alpha_B > \beta_B$
$\alpha_R < \alpha_A < \alpha_B < \alpha_C$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$		
$\alpha_R < \alpha_A < \alpha_C < \alpha_B$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_B$			$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_R < \alpha_B < \alpha_A < \alpha_C$	y	$Ti + \alpha_C$	$2Ti + \alpha_B$	$2Ti + \alpha_B$			$2Ti + \beta_B$	$3Ti + \beta_B$
	z	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$		
$\alpha_R < \alpha_B < \alpha_C < \alpha_A$	y	$Ti + \alpha_C$	$2Ti + \alpha_B$	$2Ti + \alpha_B$			$2Ti + \beta_B$	$3Ti + \beta_B$
	z	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$		
$\alpha_R < \alpha_C < \alpha_A < \alpha_B$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_B$			$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_R < \alpha_C < \alpha_B < \alpha_A$	y	$Ti + \alpha_C$	$2Ti + \alpha_B$	$2Ti + \alpha_B$			$2Ti + \beta_B$	$3Ti + \beta_B$
	z	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_B$			$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_A < \alpha_R < \alpha_B < \alpha_C$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$		
$\alpha_A < \alpha_R < \alpha_C < \alpha_B$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_B$			$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_A < \alpha_B < \alpha_R < \alpha_C$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$		
$\alpha_A < \alpha_B < \alpha_C < \alpha_R$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$		
$\alpha_A < \alpha_C < \alpha_R < \alpha_B$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_B$			$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_A < \alpha_C < \alpha_B < \alpha_R$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_B$			$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_B < \alpha_R < \alpha_A < \alpha_C$	n/a	$Ti + \alpha_C$	$2Ti + \alpha_B$	$2Ti + \alpha_B$			$2Ti + \beta_B$	$3Ti + \beta_B$
$\alpha_B < \alpha_R < \alpha_C < \alpha_A$	n/a	$Ti + \alpha_C$	$2Ti + \alpha_B$	$2Ti + \alpha_B$			$2Ti + \beta_B$	$3Ti + \beta_B$
$\alpha_B < \alpha_A < \alpha_R < \alpha_C$	n/a	$Ti + \alpha_C$	$2Ti + \alpha_B$	$2Ti + \alpha_B$			$2Ti + \beta_B$	$3Ti + \beta_B$
$\alpha_B < \alpha_A < \alpha_C < \alpha_R$	n/a	$Ti + \alpha_C$	$2Ti + \alpha_B$	$2Ti + \alpha_B$			$2Ti + \beta_B$	$3Ti + \beta_B$
$\alpha_B < \alpha_C < \alpha_R < \alpha_A$	n/a	$Ti + \alpha_C$	$2Ti + \alpha_B$	$2Ti + \alpha_B$			$2Ti + \beta_B$	$3Ti + \beta_B$
$\alpha_B < \alpha_C < \alpha_A < \alpha_R$	n/a	$Ti + \alpha_C$	$2Ti + \alpha_B$	$2Ti + \alpha_B$			$2Ti + \beta_B$	$3Ti + \beta_B$
$\alpha_C < \alpha_R < \alpha_A < \alpha_B$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_B$			$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_C < \alpha_R < \alpha_B < \alpha_A$	y	$Ti + \alpha_C$	$2Ti + \alpha_B$	$2Ti + \alpha_B$			$2Ti + \beta_B$	$3Ti + \beta_B$
	z	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_B$			$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_C < \alpha_A < \alpha_R < \alpha_B$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_B$			$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_C < \alpha_A < \alpha_B < \alpha_R$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_B$			$Ti + \beta_B$	$2Ti + \beta_B$
$\alpha_C < \alpha_B < \alpha_R < \alpha_A$	n/a	$Ti + \alpha_C$	$2Ti + \alpha_B$	$2Ti + \alpha_B$			$2Ti + \beta_B$	$3Ti + \beta_B$
$\alpha_C < \alpha_B < \alpha_A < \alpha_R$	n/a	$Ti + \alpha_C$	$2Ti + \alpha_B$	$2Ti + \alpha_B$			$2Ti + \beta_B$	$3Ti + \beta_B$
y is $\beta_A < \alpha_R \vert \beta_A > \alpha_B$, z is $\alpha_R < \beta_A < \alpha_B$								

TABLE 5.19: Simplifying $\xi^{in}_{3_R}$ in Scenario Sc.3.AB for OLSR

$\xi^{in}_{3_R}$	Condition
	Simplification Round 1
$Ti+\beta_B$	$(\alpha_A<\alpha_B<\alpha_C \& \alpha_C<\beta_B)\|(\alpha_A<\alpha_C<\alpha_B \& \alpha_B<\beta_B)\|(\alpha_C<\alpha_A<\alpha_B \& \alpha_B<\beta_B)\|(\alpha_R<\alpha_B<\alpha_A<\alpha_C \ \& \ \alpha_C<\beta_B \ \& \ \alpha_R<\beta_A<\alpha_B)\|(\alpha_R<\alpha_B<\alpha_C<\alpha_A \ \& \ \alpha_C<\beta_B \ \& \ \alpha_R<\beta_A<\alpha_B)\|(\alpha_R<\alpha_C<\alpha_B<\alpha_A \ \& \ \alpha_B<\beta_B \ \& \ \alpha_R<\beta_A<\alpha_B)\|(\alpha_C<\alpha_R<\alpha_B<\alpha_A \ \& \ \alpha_B<\beta_B \ \& \ \alpha_R<\beta_A<\alpha_B)$
$3Ti+\beta_B$	$(\alpha_R<\alpha_B<\alpha_A \ \& \ \alpha_B>\beta_B \ \& \ (\beta_A<\alpha_R\|\beta_A>\alpha_B))\|(\alpha_B<\alpha_R<\alpha_A \ \& \ \alpha_B>\beta_B)\|(\alpha_B<\alpha_A<\alpha_R \ \& \ \alpha_B>\beta_B)$
$2Ti+\beta_B$	otherwise
	Simplification Round 2
$Ti+\beta_B$	$\alpha_A<\alpha_B<\alpha_C<\beta_B\|\alpha_A<\alpha_C<\alpha_B<\beta_B\|\alpha_C<\alpha_A<\alpha_B<\beta_B\|\alpha_R<\beta_A<\alpha_B<\alpha_A<\alpha_C<\beta_B\|\alpha_R<\beta_A<\alpha_B<\alpha_C<\alpha_A<\beta_B\|\alpha_R<\beta_A<\alpha_B<\alpha_C<\beta_B<\alpha_B\|\alpha_R<\beta_A<\alpha_C<\alpha_B<\alpha_A<\beta_B\|\alpha_R<\alpha_C<\beta_A<\alpha_B<\alpha_A<\beta_B\|\alpha_R<\beta_A<\alpha_C<\alpha_B<\beta_B<\alpha_A\|\alpha_R<\alpha_C<\beta_A<\alpha_B<\beta_B<\alpha_A\|\alpha_C<\alpha_R<\beta_A<\alpha_B<\alpha_A<\beta_B\|\alpha_C<\alpha_R<\beta_A<\alpha_B<\beta_B<\alpha_A$
$3Ti+\beta_B$	$\beta_A<\beta_B<\alpha_R<\alpha_B<\alpha_A\|\beta_B<\beta_A<\alpha_R<\alpha_B<\alpha_A\|\beta_A<\alpha_R<\beta_B<\alpha_B<\alpha_A\|\beta_B<\alpha_R<\alpha_B<\beta_A<\alpha_A\|\beta_B<\alpha_R<\alpha_B<\alpha_A<\beta_A\|\alpha_R<\beta_B<\alpha_B<\beta_A<\alpha_A\|\alpha_R<\beta_B<\alpha_B<\alpha_A<\beta_A\|\beta_B<\alpha_B<\alpha_R<\alpha_A\|\beta_B<\alpha_B<\alpha_A<\alpha_R$
$2Ti+\beta_B$	otherwise

The values of $\xi^{in}_{3_R}$ are simplified in Table 5.19 which shows that the support of $\xi^{in}_{3_R}$ can be divided into three different ranges $[Ti, 2Ti]$, $[2Ti, 3Ti]$ and $[3Ti, 4Ti]$. When $\xi^{in}_{3_R} \in [Ti, 2Ti]$, we can simplify the derivation problem by introducing a new random variable $\xi^{in'}_{3_R} \in [0, Ti]$ and equals β_B. Then, we use (5.6), (5.15) and (5.16) to get:

$$
\begin{aligned}
f^{Sc.3.AB}_{OLSR}(\xi^{in'}_{3_R}) = & \ P[\xi^{in'}_{3_R} = \beta_{B_0}] \times (P[\alpha_A < \alpha_B < \alpha_C < \beta_{B_0}] \\
& + P[\alpha_A < \alpha_C < \alpha_B < \beta_{B_0}] + P[\alpha_C < \alpha_A < \alpha_B < \beta_{B_0}] \\
& + P[\alpha_R < \beta_A < \alpha_B < \alpha_A < \alpha_C < \beta_{B_0}] + P[\alpha_R < \beta_A < \alpha_B < \alpha_C < \alpha_A < \beta_{B_0}] \\
& + P[\alpha_R < \beta_A < \alpha_B < \alpha_C < \beta_{B_0} < \alpha_B] + P[\alpha_R < \beta_A < \alpha_C < \alpha_B < \alpha_A < \beta_{B_0}] \\
& + P[\alpha_R < \alpha_C < \beta_A < \alpha_B < \alpha_A < \beta_{B_0}] + P[\alpha_R < \beta_A < \alpha_C < \alpha_B < \beta_{B_0} < \alpha_A] \\
& + P[\alpha_R < \alpha_C < \beta_A < \alpha_B < \beta_{B_0} < \alpha_A] + P[\alpha_C < \alpha_R < \beta_A < \alpha_B < \alpha_A < \beta_{B_0}] \\
& + P[\alpha_C < \alpha_R < \beta_A < \alpha_B < \beta_{B_0} < \alpha_A]) \\
= & \ \tfrac{1}{Ti}\left(3P_{4A}(\xi^{in'}_{3_R}) + 5P_{6A}(\xi^{in'}_{3_R}) + 4P_{6B}(\xi^{in'}_{3_R})\right), \quad 0 \le \xi^{in'}_{3_R} \le Ti. \qquad (5.69)
\end{aligned}
$$

Considering the case when $\xi_{3_R}^{in} \in [3Ti, 4Ti]$, we simplify the derivation problem by introducing a new random variable $\xi_{3_R}^{in'''} = \beta_B$. Using (5.9), (5.12), (5.13), (5.14), we get:

$$\begin{aligned}
f_{OLSR}^{Sc.3.AB}(\xi_{3_R}^{in'''}) = P[\xi_{3_R}^{in'''} &= \beta_{B_0}] \times (P[\beta_A < \beta_{B_0} < \alpha_R < \alpha_B < \alpha_A] \\
&+ P[\beta_{B_0} < \beta_A < \alpha_R < \alpha_B < \alpha_A] + P[\beta_A < \alpha_R < \beta_{B_0} < \alpha_B < \alpha_A] \\
&+ P[\beta_{B_0} < \alpha_R < \alpha_B < \beta_A < \alpha_A] + P[\beta_{B_0} < \alpha_R < \alpha_B < \alpha_A < \beta_A] \\
&+ P[\alpha_R < \beta_{B_0} < \alpha_B < \beta_A < \alpha_A] + P[\alpha_R < \beta_{B_0} < \alpha_B < \alpha_A < \beta_A] \\
&+ P[\beta_{B_0} < \alpha_B < \alpha_R < \alpha_A] + P[\beta_{B_0} < \alpha_B < \alpha_A < \alpha_R]) \\
&= \frac{1}{Ti}\Big(2P_{4D}(\xi_{3_R}^{in'''}) + P_{5C}(\xi_{3_R}^{in'''}) + 3P_{5D}(\xi_{3_R}^{in'''}) \\
&+3P_{5E}(\xi_{3_R}^{in'''})\Big), \quad 0 \le \xi_{3_R}^{in'''} \le Ti.
\end{aligned} \tag{5.70}$$

The final case is when $\xi_{3_R}^{in} \in [2Ti, 3Ti]$, which takes the complements of probabilities presented in (5.69) and (5.70). Introducing a new random variable $\xi_{3_R}^{in''} = \beta_B$, we get:

$$\begin{aligned}
f_{OLSR}^{Sc.3.AB}(\xi_{3_R}^{in''}) = P[\xi_{3_R}^{in''} &= \beta_{B_0}] \times (1 - P[\alpha_A < \alpha_B < \alpha_C < \beta_{B_0}] \\
&- P[\alpha_A < \alpha_C < \alpha_B < \beta_{B_0}] - P[\alpha_C < \alpha_A < \alpha_B < \beta_{B_0}] \\
&- P[\alpha_R < \beta_A < \alpha_B < \alpha_A < \alpha_C < \beta_{B_0}] - P[\alpha_R < \beta_A < \alpha_B < \alpha_C < \alpha_A < \beta_{B_0}] \\
&- P[\alpha_R < \beta_A < \alpha_B < \alpha_C < \beta_{B_0} < \alpha_B] - P[\alpha_R < \beta_A < \alpha_C < \alpha_B < \alpha_A < \beta_{B_0}] \\
&- P[\alpha_R < \alpha_C < \beta_A < \alpha_B < \alpha_A < \beta_{B_0}] - P[\alpha_R < \beta_A < \alpha_C < \alpha_B < \beta_{B_0} < \alpha_A] \\
&- P[\alpha_R < \alpha_C < \beta_A < \alpha_B < \beta_{B_0} < \alpha_A] - P[\alpha_C < \alpha_R < \beta_A < \alpha_B < \alpha_A < \beta_{B_0}] \\
&- P[\alpha_C < \alpha_R < \beta_A < \alpha_B < \beta_{B_0} < \alpha_A] - P[\beta_A < \beta_{B_0} < \alpha_R < \alpha_B < \alpha_A] \\
&- P[\beta_{B_0} < \beta_A < \alpha_R < \alpha_B < \alpha_A] - P[\beta_A < \alpha_R < \beta_{B_0} < \alpha_B < \alpha_A] \\
&- P[\beta_{B_0} < \alpha_R < \alpha_B < \beta_A < \alpha_A] - P[\beta_{B_0} < \alpha_R < \alpha_B < \alpha_A < \beta_A] \\
&- P[\alpha_R < \beta_{B_0} < \alpha_B < \beta_A < \alpha_A] - P[\alpha_R < \beta_{B_0} < \alpha_B < \alpha_A < \beta_A] \\
&- P[\beta_{B_0} < \alpha_B < \alpha_R < \alpha_A] - P[\beta_{B_0} < \alpha_B < \alpha_A < \alpha_R]) \\
&= \frac{1}{Ti}\Big(1 - 3P_{4A}(\xi_{3_R}^{in''}) - 5P_{6A}(\xi_{3_R}^{in''}) - 4P_{6B}(\xi_{3_R}^{in''}) - 2P_{4D}(\xi_{3_R}^{in''}) - P_{5C}(\xi_{3_R}^{in''}) \\
&-3P_{5D}(\xi_{3_R}^{in''}) - 3P_{5E}(\xi_{3_R}^{in''})\Big), \quad 0 \le \xi_{3_R}^{in''} \le Ti.
\end{aligned} \tag{5.71}$$

The assumptions made in (5.69), (5.70) and (5.71) are relaxed by replacing $\xi_{3_R}^{in\prime}$, $\xi_{3_R}^{in\prime\prime}$ and $\xi_{3_R}^{in\prime\prime\prime}$ by $\xi_{3_R}^{in} - Ti$, $\xi_{3_R}^{in} - 2Ti$ and $\xi_{3_R}^{in} - 3Ti$ respectively. Then, by combining them, we obtain the *pdf* of $\xi_{3_R}^{in}$ in Scenario Sc.3.AB running OLSR as:

$$f_{OLSR}^{Sc.3.AB}(\xi_{3_R}^{in}) = \begin{cases} \frac{1}{Ti}U(\xi_3^{in}), & Ti < \xi_3^{in} \leq 2Ti, \\ \frac{1}{Ti}\left(1 - U(\xi_3^{in} - Ti) - V(\xi_3^{in} + Ti)\right), & 2Ti < \xi_3^{in} \leq 3Ti, \\ \frac{1}{Ti}V(\xi_3^{in}), & 3Ti < \xi_3^{in} \leq 4Ti, \\ 0, & \text{otherwise}, \end{cases} \tag{5.72}$$

where $U(\xi_3^{in}) = 3P_{4A}(\xi_{3_R}^{in} - Ti) + 5P_{6A}(\xi_{3_R}^{in} - Ti) + 4P_{6B}(\xi_{3_R}^{in} - Ti)$ and $V(\xi_3^{in}) = 2P_{4D}(\xi_{3_R}^{in} - 3Ti) + P_{5C}(\xi_{3_R}^{in} - 3Ti) + 3P_{5D}(\xi_{3_R}^{in} - 3Ti) + 3P_{5E}(\xi_{3_R}^{in} - 3Ti)$. $f_{OLSR}^{Sc.3.AB}(\xi_{3_R}^{in})$ is shown in Figure 5.56 when compared to simulation results with $Ti = 2s$.

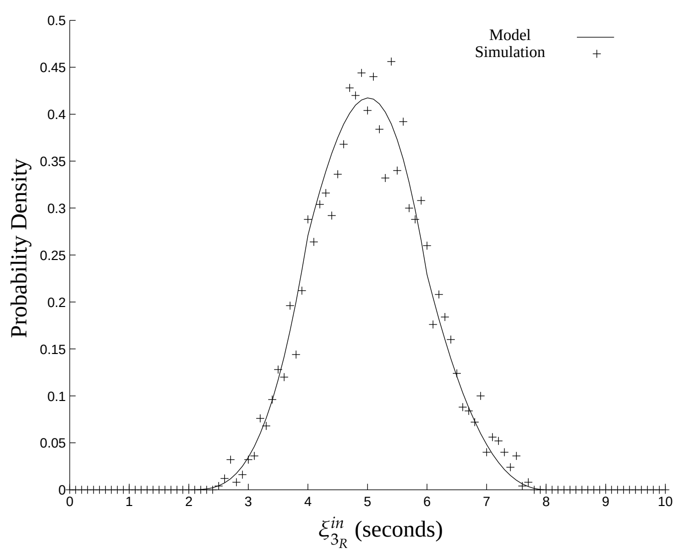

FIGURE 5.56: $f_{OLSR}^{Sc.3.AB}(\xi_{3_R}^{in})$ with $Ti = 2s$

5.1.8.5 Scenario Sc.3.AC

Scenario Sc.3.AC, shown in Figure 5.41, is the last possible scenario for forming a three hops *LPath*. In this scenario, nodes R, A and B form the same topology as in Figure 5.7 with behavior recorded in (5.28) and (5.29). On the other hand, nodes A, B and C have also the same topology as in Figure 5.7 after nodes R, A and B are renamed as A, B and C, respectively, which we have the behavior recorded by (5.55) and (5.56). Using (5.28), (5.29), (5.55) and (5.56) we can fill the combination Table 5.20. The values of $\xi_{3_R}^{in}$ are simplified in Table 5.21.

We notice that the support of $\xi_{3_R}^{in}$ can be divided into three different ranges $[Ti, 2Ti]$, $[2Ti, 3Ti]$ and $[3Ti, 4Ti]$. When $\xi_{3_R}^{in} \in [Ti, 2Ti]$, introducing a new random variable $\xi_{3_R}^{in'} = \beta_B$ simplifies the derivation problem. Then, we use (5.6) to get:

$$
\begin{aligned}
f_{OLSR}^{Sc.3.AC}(\xi_{3_R}^{in'}) &= P[\xi_{3_R}^{in'} = \beta_{B_0}] \times P[\alpha_A < \alpha_B < \alpha_C < \beta_{B_0}] \\
&= \tfrac{1}{Ti} P_{4A}(\xi_{3_R}^{in'}), \quad 0 \le \xi_{3_R}^{in'} \le Ti.
\end{aligned} \tag{5.73}
$$

TABLE 5.20: Deriving $\xi_{3_R}^{in}$ in Scenario Sc.3.AC for OLSR

Condition	&	$\lambda_{C \to B}$	$\lambda_{B \to A}$	biggest	$\xi_{3_R}^{in}$			
					$\alpha_C < \beta_B$	$\alpha_C > \beta_B$	$\alpha_B < \beta_B$	$\alpha_B > \beta_B$
$\alpha_R < \alpha_A < \alpha_B < \alpha_C$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$		
$\alpha_R < \alpha_A < \alpha_C < \alpha_B$	n/a	$2Ti + \alpha_C$	$Ti + \alpha_B$	$2Ti + \alpha_C$	$2Ti + \beta_B$	$3Ti + \beta_B$		
$\alpha_R < \alpha_B < \alpha_A < \alpha_C$	n/a	$Ti + \alpha_C$	$2Ti + \alpha_B$	$2Ti + \alpha_B$			$2Ti + \beta_B$	$3Ti + \beta_B$
$\alpha_R < \alpha_B < \alpha_C < \alpha_A$	n/a	$Ti + \alpha_C$	$2Ti + \alpha_B$	$2Ti + \alpha_B$			$2Ti + \beta_B$	$3Ti + \beta_B$
$\alpha_R < \alpha_C < \alpha_A < \alpha_B$	n/a	$2Ti + \alpha_C$	$Ti + \alpha_B$	$2Ti + \alpha_C$	$2Ti + \beta_B$	$3Ti + \beta_B$		
$\alpha_R < \alpha_C < \alpha_B < \alpha_A$	n/a	$2Ti + \alpha_C$	$2Ti + \alpha_B$	$2Ti + \alpha_B$			$2Ti + \beta_B$	$3Ti + \beta_B$
$\alpha_A < \alpha_R < \alpha_B < \alpha_C$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$		
$\alpha_A < \alpha_R < \alpha_C < \alpha_B$	n/a	$2Ti + \alpha_C$	$Ti + \alpha_B$	$2Ti + \alpha_C$	$2Ti + \beta_B$	$3Ti + \beta_B$		
$\alpha_A < \alpha_B < \alpha_R < \alpha_C$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$		
$\alpha_A < \alpha_B < \alpha_C < \alpha_R$	n/a	$Ti + \alpha_C$	$Ti + \alpha_B$	$Ti + \alpha_C$	$Ti + \beta_B$	$2Ti + \beta_B$		
$\alpha_A < \alpha_C < \alpha_R < \alpha_B$	n/a	$2Ti + \alpha_C$	$Ti + \alpha_B$	$2Ti + \alpha_C$	$2Ti + \beta_B$	$3Ti + \beta_B$		
$\alpha_A < \alpha_C < \alpha_B < \alpha_R$	n/a	$2Ti + \alpha_C$	$Ti + \alpha_B$	$2Ti + \alpha_C$	$2Ti + \beta_B$	$3Ti + \beta_B$		
$\alpha_B < \alpha_R < \alpha_A < \alpha_C$	n/a	$Ti + \alpha_C$	$2Ti + \alpha_B$	$2Ti + \alpha_B$			$2Ti + \beta_B$	$3Ti + \beta_B$
$\alpha_B < \alpha_R < \alpha_C < \alpha_A$	n/a	$Ti + \alpha_C$	$2Ti + \alpha_B$	$2Ti + \alpha_B$			$2Ti + \beta_B$	$3Ti + \beta_B$
$\alpha_B < \alpha_A < \alpha_R < \alpha_C$	n/a	$Ti + \alpha_C$	$2Ti + \alpha_B$	$2Ti + \alpha_B$			$2Ti + \beta_B$	$3Ti + \beta_B$
$\alpha_B < \alpha_A < \alpha_C < \alpha_R$	n/a	$Ti + \alpha_C$	$2Ti + \alpha_B$	$2Ti + \alpha_B$			$2Ti + \beta_B$	$3Ti + \beta_B$
$\alpha_B < \alpha_C < \alpha_R < \alpha_A$	n/a	$Ti + \alpha_C$	$2Ti + \alpha_B$	$2Ti + \alpha_B$			$2Ti + \beta_B$	$3Ti + \beta_B$
$\alpha_B < \alpha_C < \alpha_A < \alpha_R$	n/a	$Ti + \alpha_C$	$2Ti + \alpha_B$	$2Ti + \alpha_B$			$2Ti + \beta_B$	$3Ti + \beta_B$
$\alpha_C < \alpha_R < \alpha_A < \alpha_B$	n/a	$2Ti + \alpha_C$	$Ti + \alpha_B$	$2Ti + \alpha_C$	$2Ti + \beta_B$	$3Ti + \beta_B$		
$\alpha_C < \alpha_R < \alpha_B < \alpha_A$	n/a	$2Ti + \alpha_C$	$2Ti + \alpha_B$	$2Ti + \alpha_C$	$2Ti + \beta_B$	$3Ti + \beta_B$		
$\alpha_C < \alpha_A < \alpha_R < \alpha_B$	n/a	$2Ti + \alpha_C$	$Ti + \alpha_B$	$2Ti + \alpha_C$	$2Ti + \beta_B$	$3Ti + \beta_B$		
$\alpha_C < \alpha_A < \alpha_B < \alpha_R$	n/a	$2Ti + \alpha_C$	$Ti + \alpha_B$	$2Ti + \alpha_C$	$2Ti + \beta_B$	$3Ti + \beta_B$		
$\alpha_C < \alpha_B < \alpha_R < \alpha_A$	n/a	$2Ti + \alpha_C$	$2Ti + \alpha_B$	$2Ti + \alpha_B$			$2Ti + \beta_B$	$3Ti + \beta_B$
$\alpha_C < \alpha_B < \alpha_A < \alpha_R$	n/a	$2Ti + \alpha_C$	$2Ti + \alpha_B$	$2Ti + \alpha_B$			$2Ti + \beta_B$	$3Ti + \beta_B$

TABLE 5.21: Simplifying $\xi_{3_R}^{in}$ in Scenario Sc.3.AC for OLSR

$\xi_{3_R}^{in}$	Condition
	Simplification Round 1
$Ti+\beta_B$	$(\alpha_R < \alpha_A < \alpha_{B<\alpha_C}$ & $\alpha_C < \beta_B)\|(\alpha_A < \alpha_R < \alpha_B < \alpha_C$ & $\alpha_C < \beta_B)\|(\alpha_A < \alpha_B < \alpha_R < \alpha_C$ & $\alpha_C < \beta_B)\|(\alpha_A < \alpha_B < \alpha_C < \alpha_R$ & $\alpha_C < \beta_B)$
$3Ti+\beta_B$	$(\alpha_C < \alpha_A < \alpha_B$ & $\alpha_C > \beta_B)\|(\alpha_A < \alpha_C < \alpha_B$ & $\alpha_C > \beta_B)\|(\alpha_C < \alpha_R < \alpha_B < \alpha_A$ & $\alpha_C > \beta_B)\|(\alpha_B < \alpha_A < \alpha_C$ & $\alpha_B > \beta_B)\|(\alpha_B < \alpha_C < \alpha_A$ & $\alpha_B > \beta_B)\|(\alpha_R < \alpha_C < \alpha_B < \alpha_A$ & $\alpha_B > \beta_B)\|(\alpha_C < \alpha_B < \alpha_R < \alpha_A$ & $\alpha_B > \beta_B)\|(\alpha_C < \alpha_B < \alpha_A < \alpha_R$ & $\alpha_B > \beta_B)$
$2Ti+\beta_B$	otherwise
	Simplification Round 2
$Ti+\beta_B$	$\alpha_A < \alpha_B < \alpha_C < \beta_B$
$3Ti+\beta_B$	$\beta_B < \alpha_C < \alpha_A < \alpha_B\|\beta_B < \alpha_A < \alpha_C < \alpha_B\|\alpha_A < \beta_B < \alpha_C < \alpha_B\|\beta_B < \alpha_C < \alpha_R < \alpha_B < \alpha_A\|\beta_B < \alpha_B < \alpha_A < \alpha_C\|\beta_B < \alpha_B < \alpha_C < \alpha_A\|\beta_B < \alpha_R < \alpha_C < \alpha_B < \alpha_A\|\alpha_R < \beta_B < \alpha_C < \alpha_B < \alpha_A\|\alpha_R < \alpha_C < \beta_B < \alpha_B < \alpha_A\|\beta_B < \alpha_C < \alpha_B < \alpha_R < \alpha_A\|\alpha_C < \beta_B < \alpha_B < \alpha_R < \alpha_A\|\beta_B < \alpha_C < \alpha_B < \alpha_A < \alpha_R\|\alpha_C < \beta_B < \alpha_B < \alpha_A < \alpha_R$
$2Ti+\beta_B$	otherwise

When $\xi_{3_R}^{in} \in [3Ti, 4Ti]$, we can simplify the derivation problem by introducing another new random variable $\xi_{3_R}^{in'''} = \beta_B$; then using (5.8), (5.9), (5.12), (5.13) and (5.14), we get:

$$
\begin{aligned}
f_{OLSR}^{Sc.3.AC}(\xi_{3_R}^{in'''}) = P[\xi_{3_R}^{in'''} = \beta_{B_0}] \times \big(&P[\beta_{B_0} < \alpha_C < \alpha_A < \alpha_B] \\
&+ P[\beta_{B_0} < \alpha_A < \alpha_C < \alpha_B] + P[\alpha_A < \beta_{B_0} < \alpha_C < \alpha_B] \\
&+ P[\beta_{B_0} < \alpha_C < \alpha_R < \alpha_B < \alpha_A] + P[\beta_{B_0} < \alpha_B < \alpha_A < \alpha_C] \\
&+ P[\beta_{B_0} < \alpha_B < \alpha_C < \alpha_A] + P[\beta_{B_0} < \alpha_R < \alpha_C < \alpha_B < \alpha_A] \\
&+ P[\alpha_R < \beta_{B_0} < \alpha_C < \alpha_B < \alpha_A] + P[\alpha_R < \alpha_C < \beta_{B_0} < \alpha_B < \alpha_A] \\
&+ P[\beta_{B_0} < \alpha_C < \alpha_B < \alpha_R < \alpha_A] + P[\alpha_C < \beta_{B_0} < \alpha_B < \alpha_R < \alpha_A] \\
&+ P[\beta_{B_0} < \alpha_C < \alpha_B < \alpha_A < \alpha_R] + P[\alpha_C < \beta_{B_0} < \alpha_B < \alpha_A < \alpha_R]\big) \\
&= \frac{1}{Ti}\Big(P_{4C}(\xi_{3_R}^{in'''}) + 4P_{4D}(\xi_{3_R}^{in'''}) + P_{5C}(\xi_{3_R}^{in'''}) \\
&+3P_{5D}(\xi_{3_R}^{in'''}) + 4P_{5E}(\xi_{3_R}^{in'''})\Big), \quad 0 \le \xi_{3_R}^{in'''} \le Ti.
\end{aligned} \tag{5.74}
$$

Lastly when $\xi_{3_R}^{in} \in [2Ti, 3Ti]$, we use the complements of the probabilities shown in (5.73) and (5.74); then introducing a third random variable $\xi_{3_R}^{in''} = \beta_B$ in order to simplify the derivation problem and obtain the following:

$$\begin{aligned}
f_{OLSR}^{Sc.3.AC}(\xi_{3_R}^{in''}) = P[\xi_{3_R}^{in''} &= \beta_{B_0}] \times (1 - P[\alpha_A < \alpha_B < \alpha_C < \beta_{B_0}] \\
&- P[\beta_{B_0} < \alpha_C < \alpha_A < \alpha_B] - P[\beta_{B_0} < \alpha_A < \alpha_C < \alpha_B] \\
&- P[\alpha_A < \beta_{B_0} < \alpha_C < \alpha_B] - P[\beta_{B_0} < \alpha_C < \alpha_R < \alpha_B < \alpha_A] \\
&- P[\beta_{B_0} < \alpha_B < \alpha_A < \alpha_C] - P[\beta_{B_0} < \alpha_B < \alpha_C < \alpha_A] \\
&- P[\beta_{B_0} < \alpha_R < \alpha_C < \alpha_B < \alpha_A] - P[\alpha_R < \beta_{B_0} < \alpha_C < \alpha_B < \alpha_A] \\
&- P[\alpha_R < \alpha_C < \beta_{B_0} < \alpha_B < \alpha_A] - P[\beta_{B_0} < \alpha_C < \alpha_B < \alpha_R < \alpha_A] \\
&- P[\alpha_C < \beta_{B_0} < \alpha_B < \alpha_R < \alpha_A] - P[\beta_{B_0} < \alpha_C < \alpha_B < \alpha_A < \alpha_R] \\
&- P[\alpha_C < \beta_{B_0} < \alpha_B < \alpha_A < \alpha_R] - P[]) \\
&= \frac{1}{Ti}\Big(1 - P_{4A}(\xi_{3_R}^{in''}) - P_{4C}(\xi_{3_R}^{in''}) - 4P_{4D}(\xi_{3_R}^{in''}) - P_{5C}(\xi_{3_R}^{in''}) \\
&-3P_{5D}(\xi_{3_R}^{in''}) - 4P_{5E}(\xi_{3_R}^{in''})\Big), \quad 0 \le \xi_{3_R}^{in''} \le Ti. \qquad (5.75)
\end{aligned}$$

The assumptions made in (5.73), (5.74) and (5.75) are relaxed by replacing $\xi_{3_R}^{in'}$, $\xi_{3_R}^{in''}$ and $\xi_{3_R}^{in'''}$ by $\xi_{3_R}^{in} - Ti$, $\xi_{3_R}^{in} - 2Ti$ and $\xi_{3_R}^{in} - 3Ti$ respectively. Then, by combining them, we write (5.76) for the *pdf* of $\xi_{3_R}^{in}$ in Scenario Sc.3.AB running OLSR:

$$f_{OLSR}^{Sc.3.AC}(\xi_{3_R}^{in}) = \begin{cases} \frac{1}{Ti} P_{4A}(\xi_{3_R}^{in} - Ti), & Ti < \xi_3^{in} \le 2Ti, \\ \frac{1}{Ti}\Big(1 - P_{4A}(\xi_{3_R}^{in} - 2Ti) - W(\xi_3^{in} + Ti)\Big), & 2Ti < \xi_3^{in} \le 3Ti, \\ \frac{1}{Ti} W(\xi_3^{in}), & 3Ti < \xi_3^{in} \le 4Ti, \\ 0, & \text{otherwise,} \end{cases} \qquad (5.76)$$

where $W(\xi_3^{in}) = P_{4C}(\xi_{3_R}^{in} - 3Ti) + 4P_{4D}(\xi_{3_R}^{in} - 3Ti) + P_{5C}(\xi_{3_R}^{in} - 3Ti) + 3P_{5D}(\xi_{3_R}^{in} - 3Ti) + 4P_{5E}(\xi_{3_R}^{in} - 3Ti)$. The *pdf* $f_{OLSR}^{Sc.3.AC}(\xi_{3_R}^{in})$ is shown in Figure 5.57 and compared to simulation results with $Ti = 2s$.

In the mobility model in section 3.3, a 3-hops *TPath* might form according to either Scenario Sc.3.C or Sc.3.BC with equal probabilities while remaining scenarios have probabilities close to zero. As mentioned before, this is due to the fact that the remaining scenarios require the formation of two topological links, *TLinks*, at the same exact instant

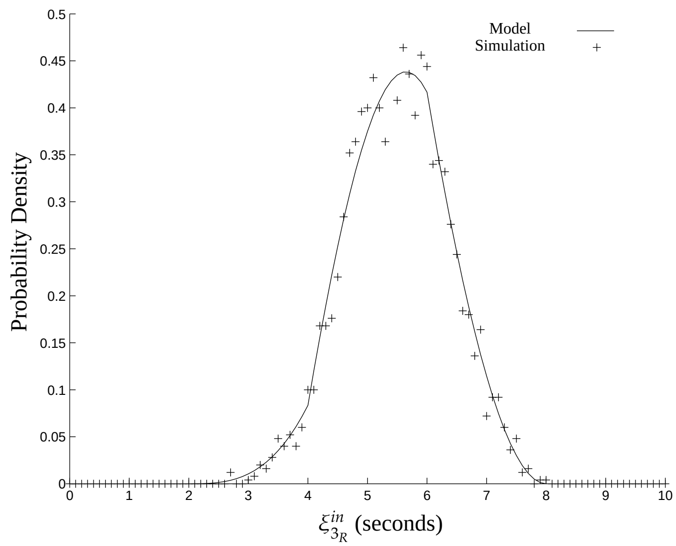

FIGURE 5.57: $f^{Sc.3.AC}_{OLSR}(\xi^{in}_{3_R})$ with $Ti = 2s$

which is very unlikely. As a result, we write $f_{OLSR}(\xi^{in}_3)$ in (5.77) using (5.50), (5.54) and (5.68). In Figure 5.58, we show the model of $f_{OLSR}(\xi^{in}_3)$ against simulation results when simulating with mobility model described in section 3.3 and using simulation parameters in Table 5.2.

$$f_{OLSR}(\xi^{in}_3) = \frac{f^{Sc.3.C}_{OLSR}(\xi^{in}_{3_R}) + f^{Sc.3.C}_{OLSR}(\xi^{in}_{3_C}) + f^{Sc.3.BC}_{OLSR}(\xi^{in}_{3_R})}{3} \tag{5.77}$$

5.1.9 Modeling ξ^{in}_k in MMT

In previous MMT sections, we observed that only one new *pdf* is derived for a particular number of hops. For instance, hops 1, 2, and 3 introduced the new *pdf*s of $f^{Sc.1.R}_{MMT}(\xi^{in}_1)$, $f^{Sc.2.R}_{MMT}(\xi^{in}_2)$ and $f^{Sc.3.R}_{MMT}(\xi^{in}_3)$ in (5.17), (5.24) and (5.39) respectively. As a matter of fact, $f^{Sc.2.R}_{MMT}(\xi^{in}_2)$ and $f^{Sc.3.R}_{MMT}(\xi^{in}_3)$ can be derived from $f^{Sc.1.R}_{MMT}(\xi^{in}_1)$ by performing as many convolutions as the number of hops; hence the new *pdf* for a scenario of k hops and when R is moving is shown below:

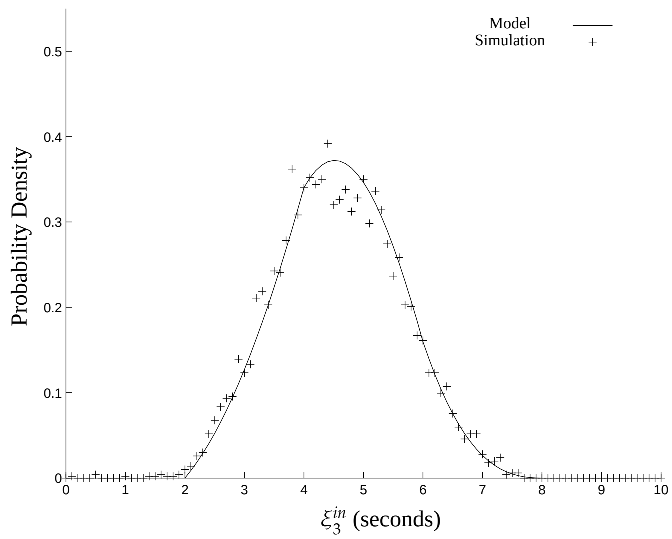

FIGURE 5.58: $f_{OLSR}(\xi_3^{in})$ with $Ti = 2s$

$$f_{MMT}^{Sc.k.R}(\xi_k^{in}) = \underbrace{f_{MMT}^{Sc.1.R}(\xi_1^{in}) * f_{MMT}^{Sc.1.R}(\xi_1^{in}) * \cdots * f_{MMT}^{Sc.1.R}(\xi_1^{in})}_{\text{k times}} \tag{5.78}$$

The *pdf*s involving node A moving, $f_{MMT}^{Sc.k.A}(\xi_k^{in})$, are similar to those when node R is moving, $f_{MMT}^{Sc.k.R}(\xi_k^{in})$, as in (5.23), (5.40), (5.41) and (5.42). Moreover, for a particular hop count, some of the *pdf*s are similar to those of fewer hops as in (5.25), (5.43), (5.44) and (5.45). To generalize this pattern, we number nodes based on their position starting by 1 for node R; we call this number *pID*. Then, we identify the moving nodes and note the smallest *pID*, *spID*. To find the *pdf* for a scenario of k hops with nodes X, Y and Z $\neq R$ and they are moving, we use the formula:

$$f_{MMT}^{Sc.k.XYZ}(\xi_k^{in}) = f_{MMT}^{Sc.(k-spID+2).R}(\xi_{(k-spID+2)}^{in}) \tag{5.79}$$

Then to find $f_{MMT}(\xi_k^{in})$, we can use:

$$f_{MMT}(\xi_k^{in}) = \frac{\sum_{i=1}^{k} f_{MMT}^{Sc.i.R}(\xi_i^{in})}{k} \tag{5.80}$$

5.1.10 Modeling ξ_k^{in} in OLSR

To model *AdaptationDelay*s for k-hops in OLSR, we break it down to as many as (k-1) of 2-hops scenarios. These 2-hops scenarios are a collection of either Sc.2.R and/or Sc.2.A after some node rearranging/renaming when necessary. A similar methodology described in section 5.1.8 can be followed starting by defining the random variables involved in the *AdaptationDelay* of interest which are α_x where x is the ID of every node in the k-hops scenario in addition to β_y where y is the ID of the node sending *TC* packet we are interested in. In a 4-hops scenario, for example, $x \in \{R, A, B, C, D\}$; if we want to model $\xi_{4_D}^{in}$, we are looking for the *TC* packet sent by A containing *Selector* ID R making $y = A$. In the case we want to model $\xi_{4_R}^{in}$, our interest is the *TC* packet sent by C containing *Selector* ID D making $y = C$.

After defining the random variables, we construct a combination table, simplify it and use similar formulations as in section 5.1.1. In the combination table, the first column would list all possible orders of α_x instances making the number of rows $(k+1)!$. Then, we fill the (k-1) columns of $\lambda_{s\to r}$ where s and r are the IDs of the two adjacent neighbors in a line topology. Considering the example of 4-hops scenario and assuming we are modeling for $\xi_{4_D}^{in}$, then the existing pairs are $(s,r) \in \{(R,A),(A,B),(B,C)\}$ while if we are modeling $\xi_{4_R}^{in}$ then the pairs are $(s,r) \in \{(D,C),(C,B),(B,A)\}$. Finally, we find the biggest of $\lambda_{s\to r}$ and fill the remaining columns as explained in section 5.1.8.

Chapter 6

Performance Analysis

This chapter provides performance models of MANETs with mobility based on combining the findings in *Topological* and *Adaptability* modeling in chapters 4 and 5, respectively. The objective provides a clear insight why protocols, in general, have lower performance with mobility and why some perform better than others. The protocol stack used in this chapter are OLSRI and MMTI.

6.1 Modeling Usable Duration $f(\varpi_k)$

Usable duration, ϖ_k, of a topological path, *TPath*, is the fraction of its total duration, φ_k, which can be used for successfully sending and receiving packets. Modeling ϖ_k is the key to understand how mobility and *AdaptationDelays* have an impact on the performance of MANETs protocols. ϖ_k was defined in (3.3) as the difference between the two independent random variables φ_k and ξ_k^{in} with both their probability density functions, *pdf*, later modeled in Chapters 4 and 5, respectively. Notice that ϖ_k can take in principle a negative value when $\xi_k^{in} > \varphi_k$. In such cases, the *TPath* exists in the topology but the routing protocol does not have the chance to use it before it disappears.

To find the *pdf* of the usable duration, $f(\varpi_k)$, we start by assuming that z is a variable that is the sum of two independent random variables x and y, $z = x + y$, with *pdf* $g_x(x)$ and $h_y(y)$. Then we can find the probability of $z = z_0$ as the following:

$$
\begin{aligned}
P[z = z_0] &= P[x = w] \times P[y = z_0 - w] \\
&= g_x(w) \times h_y(z_0 - w)
\end{aligned} \tag{6.1}
$$

However, w can have any value from $-\infty$ to ∞, while $P[z = z_0]$ is the *pdf* value of z, $f(z)$ at z_0. As a result, we write:

$$
f(z_0) = \int_{-\infty}^{\infty} g_x(w) \times h_y(z_0 - w)\mathrm{d}w \tag{6.2}
$$

The previous (6.2) is the definition of convolution for which we can rewrite as:

$$
f(z) = g_x(z) * h_y(z) \tag{6.3}
$$

Now let us assume that $t = x - y$; hence, t is the addition of two independent random variables x and $-y$. Therefor:

$$
f(t) = g_x(t) * h_y(-t) \tag{6.4}
$$

From previous discussion we can find $f(\varpi_k)$ as the following[1]:

$$
f_{\varpi_k}(x) = f_{\varphi_k}(x) * f_{\xi_k^{in}}(-x) \tag{6.5}
$$

Simulation results for this section were collected using the three scenarios specified in Table 6.1. Each of these scenarios was run with mobility model is section 3.3 and simulation parameters shown in Table 3.9. Figures 6.1 through 6.6 show a subset [2] of

[1]Note that $f_{\varphi_k}(x)$ and $f_{\xi_k^{in}}(x)$ are the same *pdf* derived in Chapters 4 and 5 as $f(\varphi_k)$ and $f(\xi_k^{in})$

[2]Only selected values of Sp_{avg} and Ti are shown to provide better readability

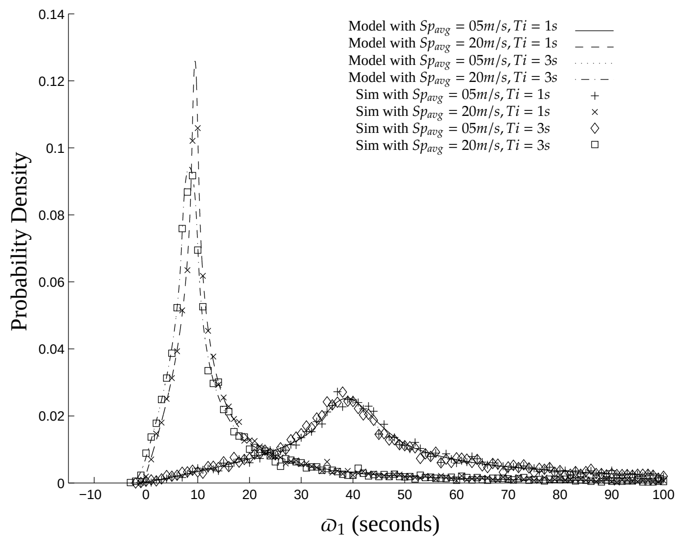

FIGURE 6.1: Model vs Simulation of $f(\varpi_1)$ in MMT with $D_{TX} = 200m$

the comparison between the model and simulation results of $f(\varpi_k)$, $k \in \{1, 2, 3\}$, which shows a tight agreement between the two.

TABLE 6.1: Summary of Performance Modeling in Random Mobility Scenarios

Scenario	Number of hops	Nodes
C.1	1	2
C.2	2	3
C.3	3	4

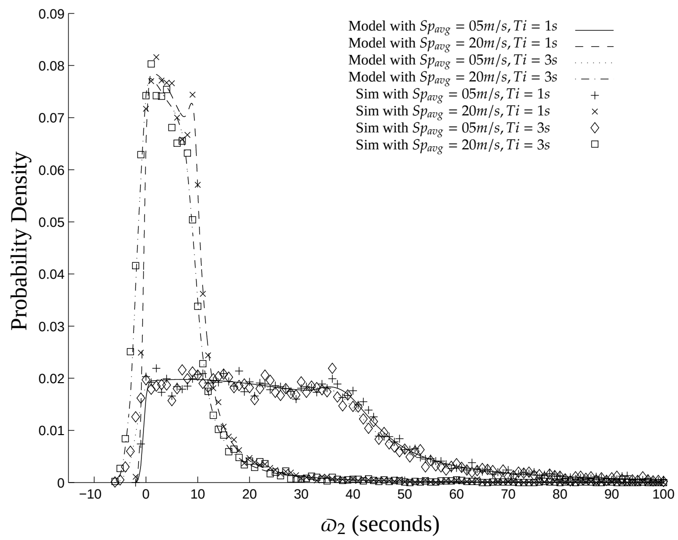

FIGURE 6.2: Model vs Simulation of $f(\varpi_2)$ in MMT with $D_{TX} = 200m$

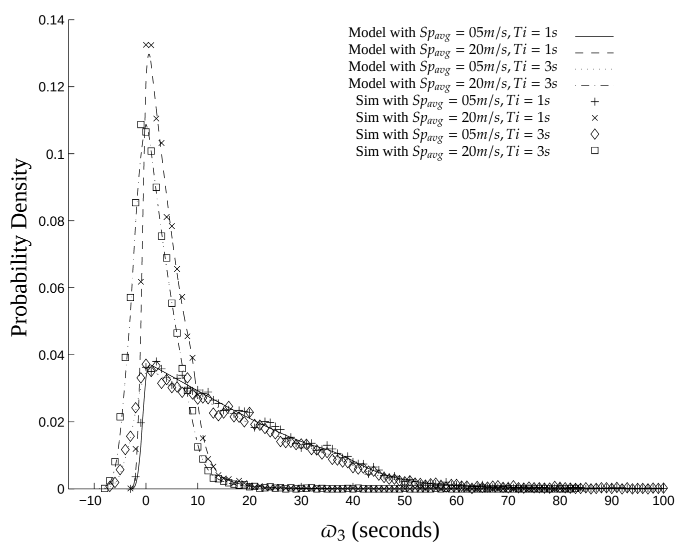

FIGURE 6.3: Model vs Simulation of $f(\varpi_3)$ in MMT with $D_{TX} = 200m$

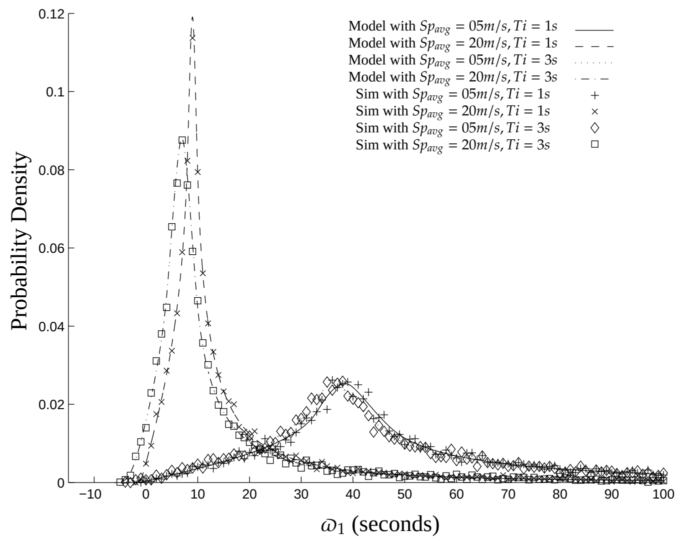

FIGURE 6.4: Model vs Simulation of $f(\varpi_1)$ in OLSR with $D_{TX} = 200m$

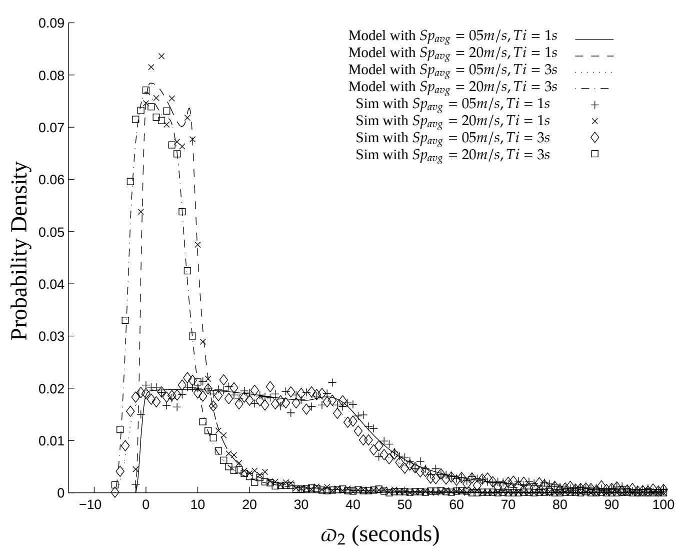

FIGURE 6.5: Model vs Simulation of $f(\varpi_2)$ in OLSR with $D_{TX} = 200m$

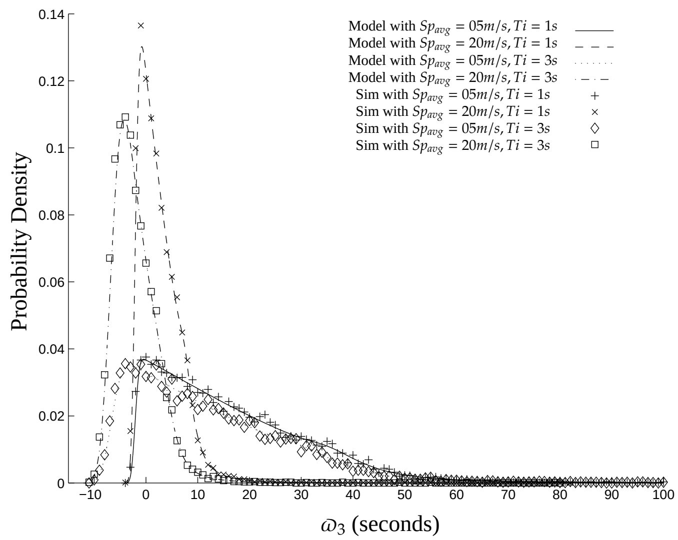

FIGURE 6.6: Model vs Simulation of $f(\varpi_3)$ in OLSR with $D_{TX} = 200m$

6.2 Modeling Utilization Ratio $\Im_k$

Referring to (3.7), we notice that $\Im_k$ is dependant on the usable duration ϖ_k and φ_k for a specific number of hops k. However, we are only interested in the range when $\varpi_k \geq 0$ as $\Im_k$ can not be a negative value. As a result, we define a new random variable ϖ_k^+ with its *pdf* as follows:

$$
\begin{aligned}
f(\varpi_k^+) &= \frac{f(\varpi_k) \times u(0)}{\int_0^\infty f(\varpi_k) \mathrm{d}\varpi_k} \\
&= \frac{f(\varpi_k) \times u(0)}{1 - F(\varpi_k = 0)}
\end{aligned}
\tag{6.6}
$$

Where $u(0)$ is a unit step function having its rising edge at $x = 0$ and the denominator is simply scaling the result so that $F(\varpi_k^+ = \infty) = 1$. To find $\Im_k$ analytically, we can use the following equation which calculates the expected value of the non-negative usable duration, $E[\varpi_k^+]$, multiplied by the probability that $\varpi_k \geq 0$, then divided by the expected value of *TPath* duration, $E[\varphi_k]$. Using (6.6), we get:

$$\begin{aligned}\Im_k &= \frac{E[\varpi_k^+] \times P[\varpi_k \geq 0]}{E[\varphi_k]} \\ &= \frac{E[f(\varpi_k) \times u(0)] \times (1 - F(\varpi_k = 0))}{(1 - F(\varpi_k = 0)) \times E[\varphi_k]} \\ &= \frac{\int_0^\infty \varpi_k f(\varpi_k) \mathrm{d}\varpi_k}{\int_{-\infty}^\infty \varphi_k f(\varphi_k) \mathrm{d}\varphi_k} \end{aligned} \tag{6.7}$$

Simulation results are collected using the scenarios detailed in Table 6.1. Each of these scenarios was run with mobility model is section 3.3 and simulation parameters shown in Table 3.9. In the meanwhile, nodes are generating data using Constant Bit Rate, CBR, packet generator. CBR was chosen to simulate the operation of a streaming application. It also provides a predictable relationship with time, in more details, the number of packets generated in duration T can be found by multiplying T by the rate of packet generation. As a result, we can compare the results collected from models using (6.7) and simulation using (3.9) as we show in Figures 6.7 through 6.15. Analyzing these figures we conclude the following observations:

1. $\Im_k$ for MMT is higher than OLSR. The reason is that MMT has lower ξ_k^{in}, regardless of Ti, than OLSR as shown in Figures 3.12 through 3.14 and derived in Chapter 5. Lower ξ_k^{in} means higher ϖ_k according to (3.3) resulting in higher $\Im_k$ as shown in (3.7).

2. Increasing Sp_{avg} decreases $\Im_k$. The reason is that increasing Sp_{avg} results in shorter φ_k as indicated in plotting $\varphi_{k_{avg}}$ in Figure 3.8 and in the derivations in Chapter 4. As a result of lower φ_k, we notice that $\Im_k$ also decreases according to (3.8).

3. OLSR is more impacted by increasing Sp_{avg} than MMT which is evident in the steeper decrease in $\Im_k$ for OLSR than MMT. This can be explained by referring to (3.8) where increasing Sp_{avg} decreases φ_k in the denominator resulting in higher rate of increase for the ratio $\frac{\xi_k^{in}}{\varphi_k}$ in OLSR than MMT as OLSR has higher ξ_k^{in} in the nominator which its impact on $\Im_k$ will be magnified with decreased denominator.

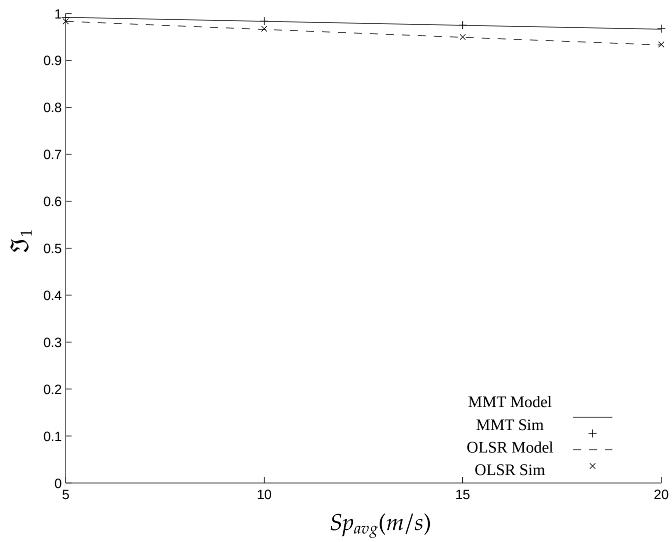

FIGURE 6.7: Model vs Simulation of $\mathfrak{I}_1$ in MMT and OLSR with Sp_{avg} and $Ti = 1s$

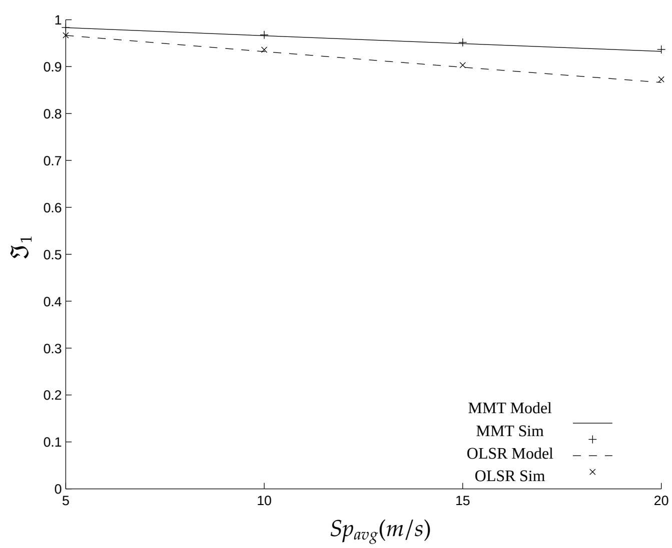

FIGURE 6.8: Model vs Simulation of $\mathfrak{I}_1$ in MMT and OLSR with Sp_{avg} and $Ti = 2s$

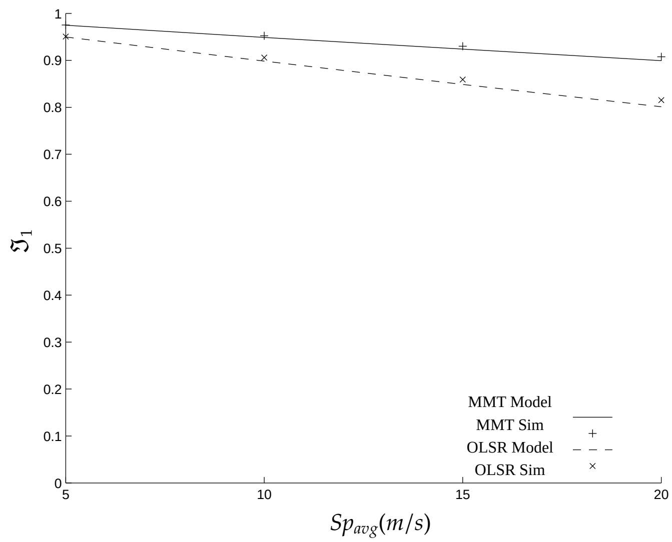

FIGURE 6.9: Model vs Simulation of $\mathfrak{I}_1$ in MMT and OLSR with Sp_{avg} and $Ti = 3s$

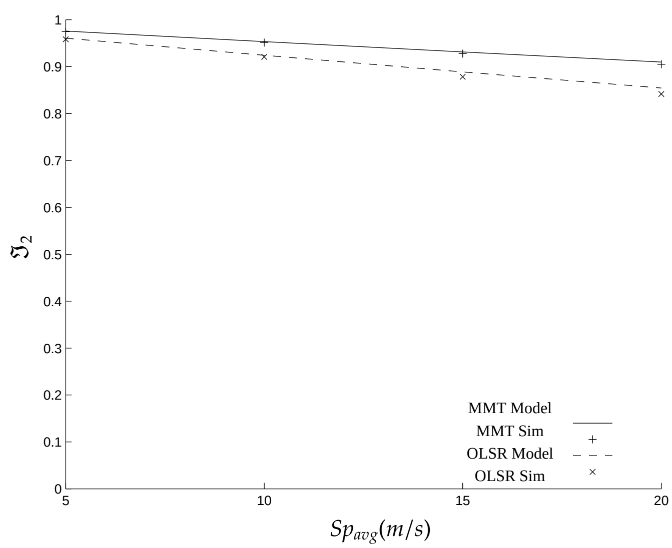

FIGURE 6.10: Model vs Simulation of $\mathfrak{I}_2$ in MMT and OLSR with Sp_{avg} and $Ti = 1s$

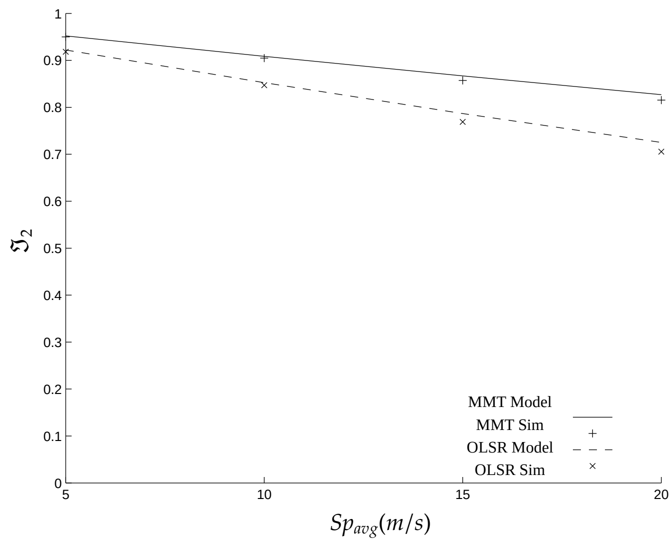

FIGURE 6.11: Model vs Simulation of $\mathfrak{I}_2$ in MMT and OLSR with Sp_{avg} and $Ti = 2s$

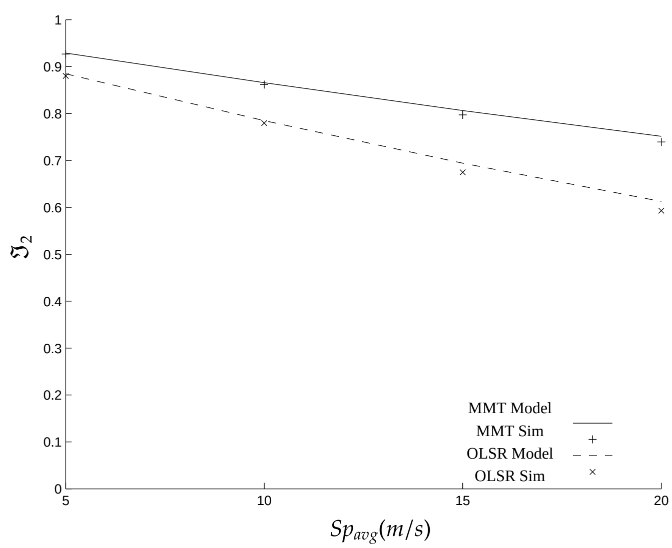

FIGURE 6.12: Model vs Simulation of $\mathfrak{I}_2$ in MMT and OLSR with Sp_{avg} and $Ti = 3s$

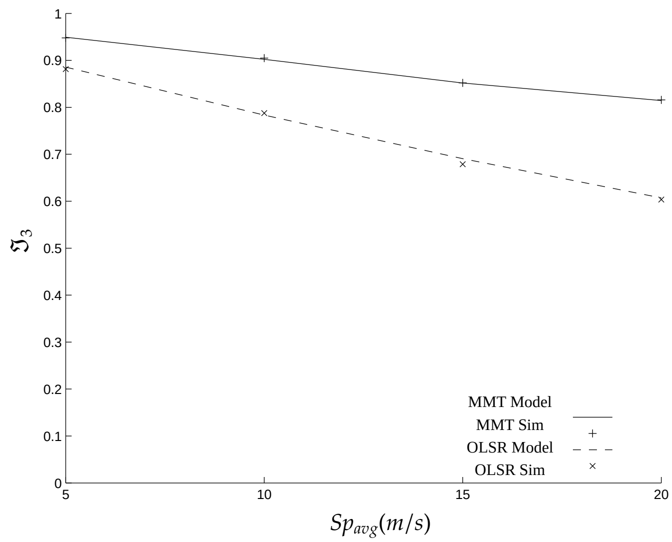

FIGURE 6.13: Model vs Simulation of $\mathfrak{I}_3$ in MMT and OLSR with Sp_{avg} and $Ti = 1s$

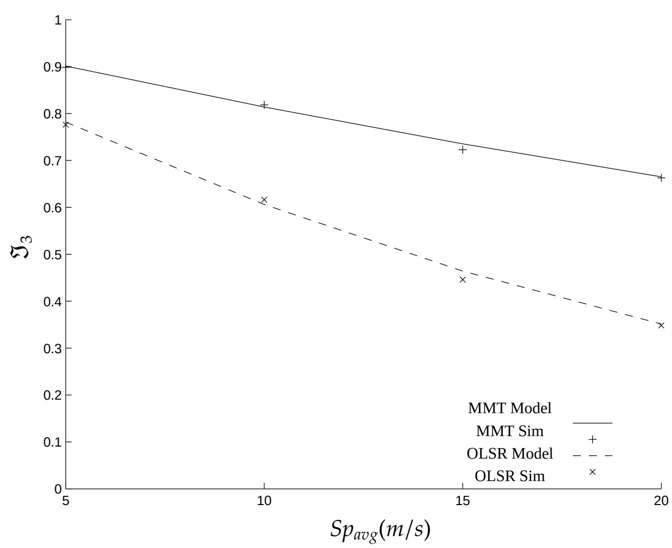

FIGURE 6.14: Model vs Simulation of $\mathfrak{I}_3$ in MMT and OLSR with Sp_{avg} and $Ti = 2s$

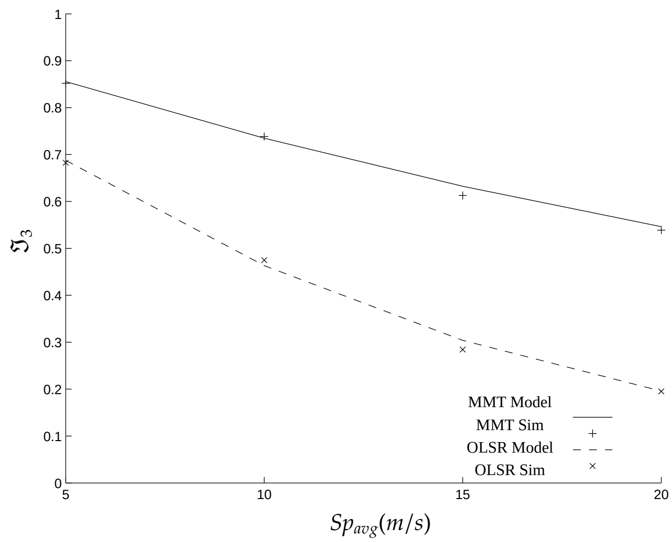

FIGURE 6.15: Model vs Simulation of $\Im_3$ in MMT and OLSR with Sp_{avg} and $Ti = 3s$

To find the overall utilization ratio $\Im$ of a routing protocol, we can calculate the weighted sum of $\Im_k$ for each particular number of hops k. The weight of each $\Im_k$ is obtained from the contribution of a *TPaths* of k hops to overall packets sent in the network. To calculate this contribution two things are needed, first, the likelihood of forming a *TPath* of k hops, the probability mass function of k ($m(k)$), secondly the expected duration of a *TPath* of k hops. In Table 6.2, we show the values of $m(k)$ as $k \in \{1, 2, 3\}$ collected from simulation[3]. Note that these values do not change with changing speed, Sp_{avg}[4]. Table 3.7 shows the expected duration of a *TPath* of k hops while Table 6.3 shows the calculated weights of $\Im_k$ in overall $\Im$. Finally, we can calculate $\Im$ using (6.8) and compare it with simulation results using (3.10) as depicted in Figures 6.16 through 6.18.

[3]Finding the model of $m(k)$ is only possible when considering the past of nodes' spacial locations and velocities which is outside the scope of this work

[4]Consider the case of running a scenario for t seconds and nodes are moving at speed of Sp_{avg}. Increasing Sp_{avg} by rate r, is basically having the scenario run at original speed for rt seconds then squeezing the time line in just t seconds, hence not impacting $m(k)$

$$\Im = \sum_{k=1}^{k_{max}} \Im_k \times \underbrace{\left(\frac{m(k) \times \varphi_{k_{avg}}}{\sum_{h=1}^{h=k_{max}} m(h) \times \varphi_{h_{avg}}} \right)}_{\Im_k \text{ weight}} \tag{6.8}$$

TABLE 6.2: Values of $m(k)$ and $k_{max} = 3$

Number of hops	$k = 1$	$k = 2$	$k = 3$
$m(k)$	0.5687	0.3081	0.1232

TABLE 6.3: Calculating $\Im_k$ weight in $\Im$ and $k_{max} = 3$

Number of hops	$k = 1$	$k = 2$	$k = 3$
$\Im_k$ weight	0.7438	0.2018	0.0544

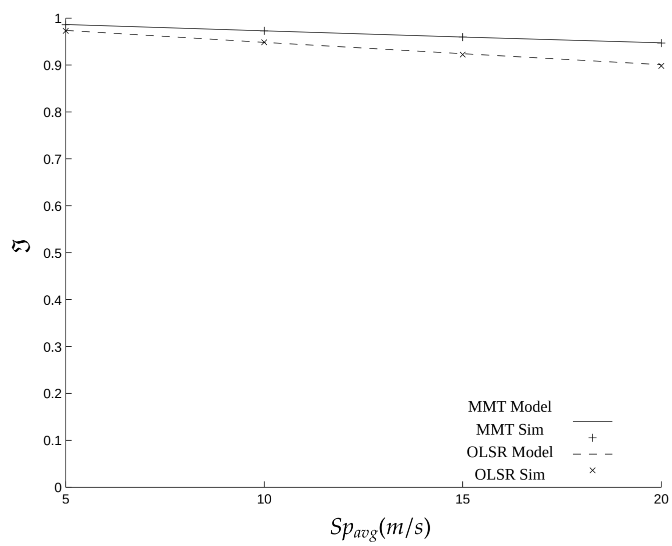

FIGURE 6.16: Model vs Simulation of $\Im$ in MMT and OLSR with Sp_{avg} and $Ti = 1s$

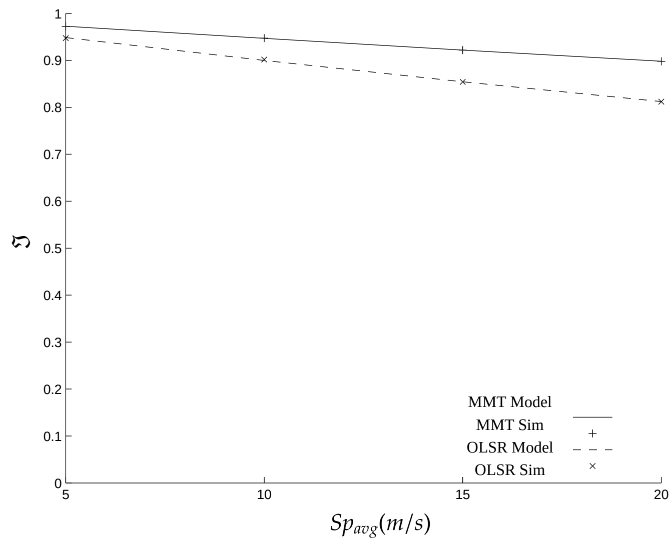

FIGURE 6.17: Model vs Simulation of $\Im$ in MMT and OLSR with Sp_{avg} and $Ti = 2s$

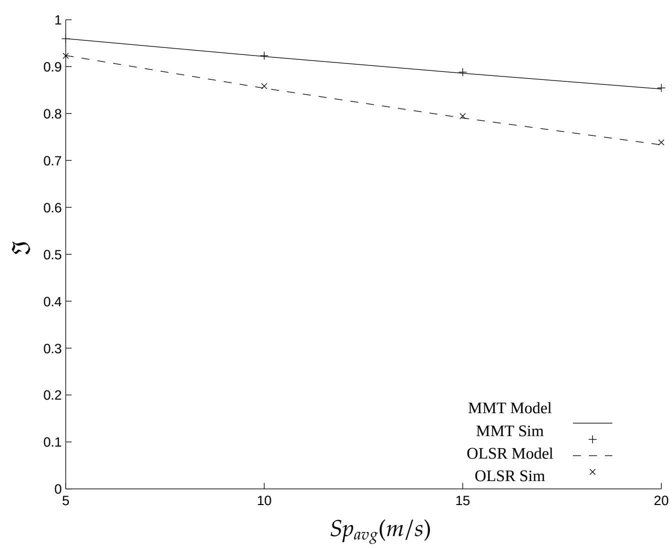

FIGURE 6.18: Model vs Simulation of $\Im$ in MMT and OLSR with Sp_{avg} and $Ti = 3s$

Chapter 7

Performance Enhancement

The purpose of this chapter is to show the significance of proposed models in understanding the factors impacting a protocol stack for MANET and how we can use this understanding to enhance the performance. Two studies are presented in following sections; the first is to use previous findings to enhance logical paths *LPaths* selection. The second is to study the impact of building and using longer *LPaths*, in terms of number of hops k, on networks's performance. The two studies are applied on MMT protocol; However, the methodology can be applied to other protocols as well.

7.1 Improving MMTs VID Selection

The purpose of a routing protocol is to find the best logical path, *LPath*, between two communicating entities based on a selection criterion. In many cases, this selection criterion is limited to selecting the *LPath* with the least number of hops. This is a valid selection criterion since the average duration of topological paths *TPaths*, $\varphi_{k_{avg}}$ decreases with increasing the number of hops k as shown in Table 3.7 and plotted in Figure 3.8. Indeed, selecting *LPaths* with longer duration not only minimizes packet retransmissions and failures, but also reduces the overhead associated with establishing and restarting transmission on alternative *LPaths*. In addition, selecting an *LPath* with lower number of hops for packet transmission minimizes the end to end packet delay as the packet is forwarded and queued fewer times.

In MMT, the selection criterion has another purpose which is the selection of the *VIDs* that are the best to grow and extend the MMT tree. In section 3.1, we presented the *VID* as a tuple consisting of three pieces of information, $(RID, LID, hops)$. *hops* was considered as a cost metric used in the function selecting the best *VID* with the minimum *hops* from a neighboring node as shown in line 10 and 23 in Algorithm 1. The problem arises when the MMT algorithm has to choose among two or more *VIDs* with the same *hops* value. Originally, the solution was a random selection. We will call this the *Legacy* selection criterion. A better selection criterion is needed replacing the *Legacy* selection criterion with a new metric to do the tiebreaker between *VIDs* of the same *hops* value.

It is key to remember that a *VID* is a mere representation of a logical path *LPath* based on a topological path *TPath*, the ground truth. In many cases, the time elapsed between the formation of the *TPath* and the acquisition of the corresponding *VID* is the *AdaptationDelay*[1]. When node B acquires a new *VID* with *hops* value of k derived from a neighboring node $A's$ VID, it means that $\xi_k^{in} > \varphi_k$ and $\varpi_k > 0$ as defined in (3.3). As a result, we can use the random variable and *pdf* in (6.6) to represent the usable time duration of a *VID* at the time of its acquisition. As time T_p passes on an acquired *VID*, then the usable time duration given T_p, $(\varpi_k \mid_{Tp})$, follows a *pdf* which can be deduced from (6.6) as:

$$f(\varpi_k \mid_{Tp}) = \frac{f(\varpi_k^+) \times u(T_p)}{1 - F(\varpi_k^+ = T_p)} \tag{7.1}$$

It is possible to use (7.1) and calculate the expected value of $(\varpi_k \mid_{Tp})$, $E[\varpi_k \mid_{Tp}]$ using (7.2). Hence, for a group of *VIDs* with equal values of *hops*, we can use the expected remaining usable duration given T_p, $E[\varpi_k^r \mid_{Tp}]$, shown in (7.3) as a tiebreaker by selecting the *VID* with the maximum $E[\varpi_k^r \mid_{Tp}]$ in a new selection criterion called the *Enhanced*.

$$E[\varpi_k \mid_{Tp}] = \int_{-\infty}^{\infty} \varpi_k \times f(\varpi_k \mid_{Tp}) \mathrm{d}\varpi_k \tag{7.2}$$

[1]In MMT, this is true only when the acquisition of the *VID* happens immediately after receiving the parental *VID* in a *hello* packet for the first time. This note will be discussed in more details later in this section

$$E[\varpi_k^r \,|_{Tp}] = E[\varpi_k \,|_{Tp}] - T_p \tag{7.3}$$

Up till now, we assumed the case depicted in Figure 7.1 where node B will immediately derive and acquire a new VID, $newVID_B$, from a parent node A the first time it sees the prospective parental VID_A announced in A's *hello* packet. Remember that ξ_k^{in} was defined in (3.1) as the delay between the time when the topological path, *TPath*, of k hops is formed, T_T^{in}, and the time when node B logs the logical path, *LPath*, information in the form of acquiring $newVID_B$, T_L^{in}. In this case, the usable duration, ϖ_k^+, of $newVID_B$ starts at T_L^{in} and it is the instant when the parental VID_A was announced for the first time, T_{1Ann}. Note that at time T_{1Ann}, the usable time passed T_p equals to zero.

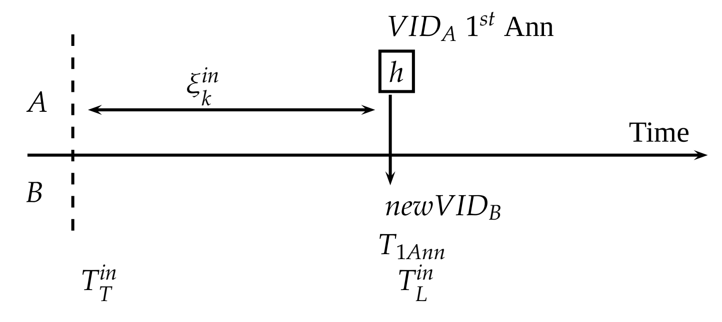

FIGURE 7.1: Acquiring *VID* immediately after the first announcement of parental *VID*

On the other hand, it is possible for node B to receive the first announcement of parental VID_A but it decides to derive a $newVID_B$ after some delay. A good example on this case is when B has a full *VIDList* and VID_A does not qualify yet as a better parental *VID* to derive from, or node B simply is busy processing some data packets that it decided to wait some time till it becomes free. A depiction of this case is shown in Figure 7.2 where we see that the $newVID_B$ was acquired when parental VID_A was announced for the second time resulting in $T_L^{in} - T_T^{in} = \xi_k^{in} + DecisionDelay$. In addition, the instant when $newVID_B$ could have been usable started some time before its acquisition which is the time when the parental VID_A was announced for the first time at T_{1Ann}. Note that at time T_{1Ann}, the usable time passed $T_p = 0$.

In both cases in Figures 7.1 and 7.2, the beginning of the usable duration of a newly acquired *VID*, ϖ_k^+, is when the parental *VID* is announced for the first time at T_{1Ann}. Hence, we can calculate T_p as follow:

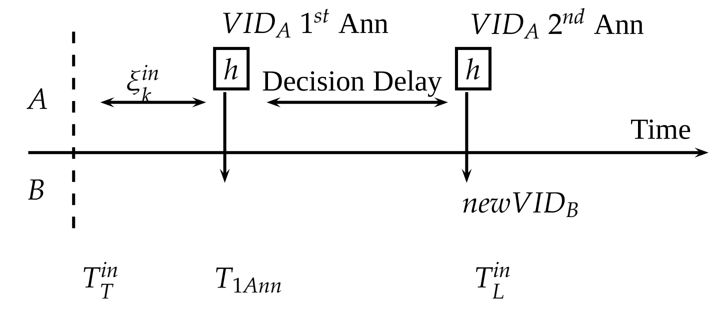

FIGURE 7.2: Acquiring *VID* with delay after the first announcement of parental *VID*

$$T_p = currentTime - T_{1Ann} \tag{7.4}$$

Clearly from (7.4), to keep track of the usable duration passed on a *VID*, T_p, we need to record the associated time for announcing the parental *VID* for the first time, T_{1Ann}. Hence, we can modify the original implementation of the MMT algorithm and protocol presented in sections 3.1 and 3.2 in order to accommodate the *Enhanced* selection criterion as follow:

- The addition of a new announcement list, *AnnList*, which stores pairs of *VIDs* announced by neighbors in *hello* packets and their T_{1Ann}.
- The addition of a new piece of information to the presentation of *VID* so it becomes $(RID, LID, hops, T_{1Ann})$. T_{1Ann} is set when a *VID* is acquired with the value associated with the parental *VID* recorded in *AnnList*.
- The use of (7.4) to get T_p; then using (7.2) we can calculate the other cost metric of $E[\varpi_k^{r|T_p}]$ to implement the *Enhanced* selection criterion and use it in lines 10 and 23 in Algorithm 1.

To gauge the benefits of using the *Enhanced* over the *Legacy* selection criteria, we will use simulation results collected from running scenarios in Table 3.4 and using mobility model in section 3.3 and simulation parameters in Table 3.9. We record the selections of *Enhanced* and *Legacy*; then compare their remaining usable duration from the instant of *VID* selection to the time it is no longer valid in topology. In Figure 7.3, we show the probability that *Enhanced* selection criterion is selecting 1-hop *VIDs* that have longer remaining usable duration compared to *Legacy* when $Ti \in \{1s, 2s, 3s\}$ and $Sp_{avg} \in \{5m/s, 10m/s, 15m/s, 20m/s\}$. In addition, we plot a solid line to represent the average probability as Sp_{avg} changes. Similar plots are shown in Figures 7.4 and 7.5 for selecting 2-hops and 3-hops *VIDs*, respectively.

Analyzing the results, we observe that the probability of *Enahnced* is better than *Legacy* has relatively the same value as we increase *Ti*. Recall the *Enhanced* selection decisions are based on calculating the $E[\varpi_k^{r|T_p}]$ in (7.2) which eventually uses the findings in section 6.1, namely the *pdf* $f(\varpi_k)$ in (6.5). For MMT, representative plots of $f(\varpi_k)$ are shown in Figures 6.1 through 6.3 which clearly shows that $f(\varpi_k)$ is not greatly impacted by changing *Ti* as Sp_{avg} is kept the same. This resulted in relatively the same probability of *Enahnced* is better than *Legacy* regardless of *Ti*.

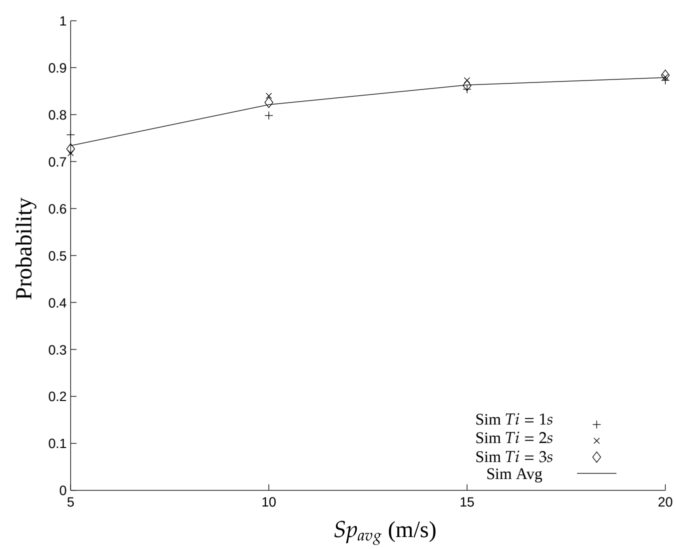

FIGURE 7.3: Probability that *Enahnced* is better than *Legacy* for 1-hop *VIDs*

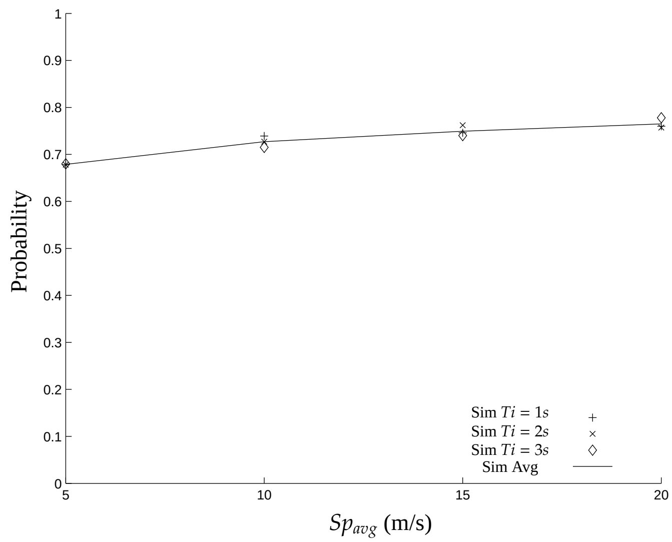

FIGURE 7.4: Probability that *Enahnced* is better than *Legacy* for 2-hops *VIDs*

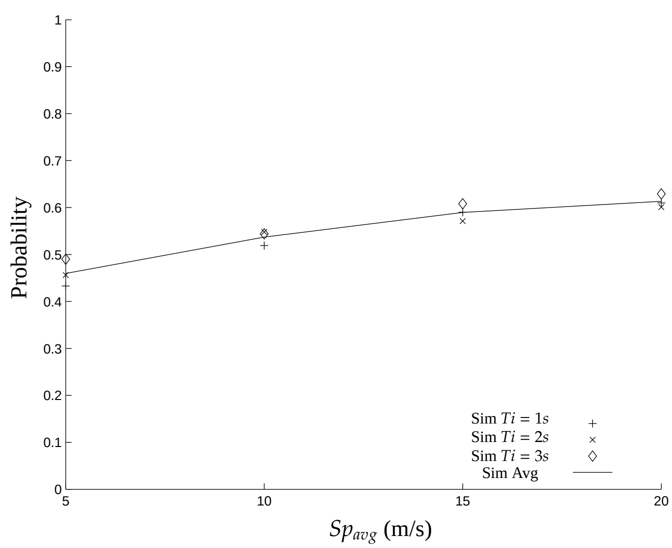

FIGURE 7.5: Probability that *Enahnced* is better than *Legacy* for 3-hops *VIDs*

We gathered the rate of acquiring new *VIDs* when using the *Enhanced* and *Legacy* in addition to the *Ideal* selection criterion which is hypothetical selection mechanism that is able to predict the future and select the *VID* that has longest remaining usable duration. The *Ideal* selection criterion serves as absolute minimum of any other selection criterion. Note that the lower the rate of acquiring new *VIDs* means more stable *VIDs* as they persist longer and do are not required to be replaced frequently. It also means stable communication and less overhead associated with acquiring new *VIDs* in the form of *RegistrationRequest* and *RegistrationAccept* packets as explained in section 3.2.

Figure 7.6, depicts the rate of acquiring new 1-hop *VIDs* using the *Legacy*, *Enhanced* and *Ideal* selection criterion. Note the clustering of results collected using different values of Ti. Also, we plot the average rate of each of the selection criterion which proves the benefits of using the *Enhanced* over the *Legacy* selection criterion. For each of the selection criteria, it is evident that the rate of acquiring new 1-hop *VIDs* is relatively the same regardless of Ti due to the fact that $f(\varpi_k)$ is not greatly impacted by changing Ti as Sp_{avg} is kept the same. Similar presentations are found in Figures 7.7 and 7.8 were the comparison of selection criterion was applied for 2-hops and 3-hops *VIDs*.

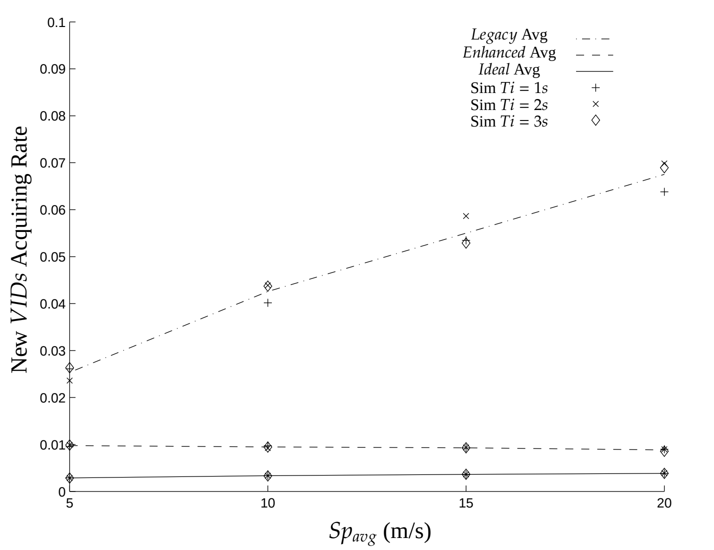

FIGURE 7.6: Rate of Acquiring new 1-hop *VIDs*

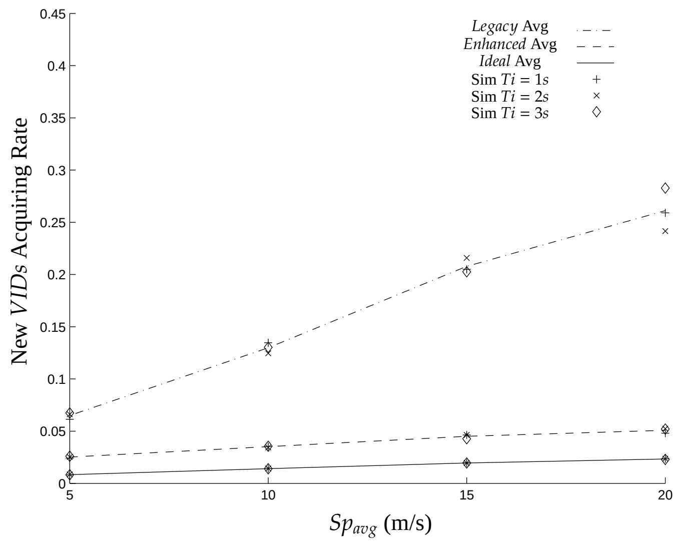

FIGURE 7.7: Rate of Acquiring new 2-hops $VIDs$

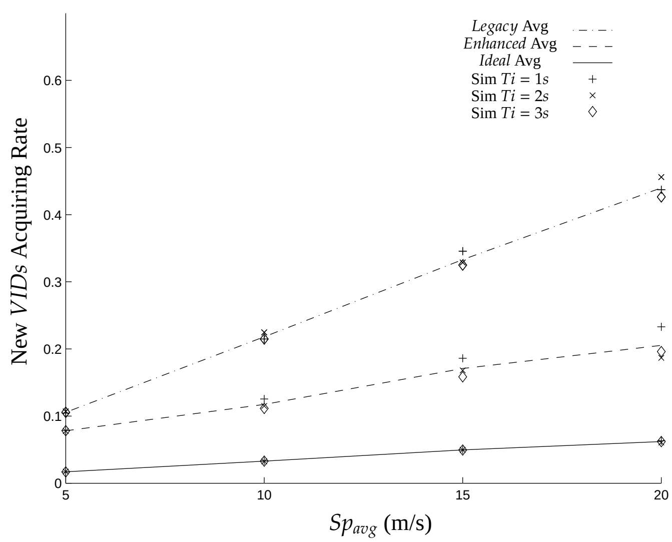

FIGURE 7.8: Rate of Acquiring new 3-hops $VIDs$

Despite previous benefits, employing the *Enahnced* selection criterion in MMT has memory, processing and communication challenges for real life adoption. Referring to sections 3.1 and 3.2 and assuming the following limitation of MMT creation: $maxHop = 3$, $maxVID = 5$, $maxChild = 9$ and $maxClient = 20$; then, we can summarize the following disadvantages:

- The *Legacy VID* representation is $(RID, LID, hops)$ where the first component, RID, can be any uniquely identifiable ID such as the Medium Access Control (MAC) address which is 6 byte in length. Since $maxHops = 3$, it can be represented in 3 bits; Also since, $maxChild = 9$, then the maximum value of LID can be 999, it can be represented in 10 bits (if we use $maxHops = 5$, then the maximum value of LID can be 99999). As a result, the *Legacy* size of VID, $VIDsize$, is the ceil of $6 + \frac{3+10}{8}$, which is 8 bytes. Employing the *Enahanced* selection criterion, the addition detail of T_{1Ann} is added to the VID representation. T_{1Ann} is of type "double" (8 bytes) raising the total VID size, $VIDsize$, to 16 bytes.

- Every node in MMT maintains two lists, *VIDList* and *ChildList*. With the value of $maxVID = 5$, the *Legacy* size of $VIDList$ can be calculated as $maxVID \times VIDsize$ which equals to 40 bytes. Similarly using a value of $maxVID = 5$ and $maxChild = 9$, we calculate the size of the *ChildList* in the worst case scenario as $maxVID \times VIDsize \times maxChild$ which equals to 360 bytes. However, employing the *Enhanced* selection criteria doubles $VIDsize$ resulting in double the sizes of $VIDList$ and *ChildList* to become 80 and 720 bytes, respectively. In addition, root nodes in MMT should maintain *ClientList* which its size is $maxVID \times VIDsize \times maxClient$ which means 800 bytes in the *Legacy* selection criteria and 1600 bytes in the *Enhanced* one.

- To use the *Enhanced* selection criterion, every node running MMT should implement and maintain the *AnnList* which is considered both memory and processing overhead. Considering memory overhead, the length of the list is subject to two factors: the number of neighbors a node has and $maxVID = 5$. In scenarios shown in Table 3.4, node density was 25×10^{-6} per m^2 and using transmission range of $D_{TX} = 200m$ we calculate the expected number of nodes in a transmission area as $\pi \times D_{TX}^2 \times 25 \times 10^{-6}$ which is 3.14 nodes which we round up to 4 nodes. Not counting the node itself, each node will have on average about 3 neighbors each of which has 5 *VIDs*. As a result, the size of *AnnList* is about 3 neighbors $\times$ 5 *VIDs* $\times$ *VIDsize*; which results in a total list size of about 120 bytes.

- Finally, the *Enhanced* selection criterion uses the finding in (7.3) which requires implementing a mechanism for looking up stored array of $E[\varpi_k\ |_{Tp}]$ which is eventually calculated from an array representing $f(\varpi_k)$ with predefined Ti and Sp_{avg} values. The calculation link between $E[\varpi_k\ |_{Tp}]$ and $f(\varpi_k)$ is clarified in (6.6) through (7.3). The need for the look up mechanism is due to the fact that $f(\varpi_k)$ in (6.6) is dependent on $f(\varphi_k)$, as shown in (6.5), which is modeled as a non-closed form model for every value of Sp_{avg} in chapter 4.

7.2 The Impact of *maxHop* on $\mathfrak{I}$ for MMT

Referring to sections 3.1 and 3.2, it is evident that constructing and maintaining *VIDs* is resource consuming which is justifiable if those *VIDs* are of great use in the over all network's performance, specifically the utilization ratio $\mathfrak{I}$. In section 6.2, we plotted models and simulation results of utilization ratio of 3-hops topological paths, $\mathfrak{I}_3$, for MMT and OLSR in Figures 6.13 through 6.15 while $Ti \in \{1s, 2s, 3s\}$ which shows a steep decline as we increase Sp_{avg}. Note that for MMT, the value of k_{max} in $\mathfrak{I}_k$ is taken from the variable of limiting MMT creation $maxHop = k$. In addition, we notice that the contribution of $\mathfrak{I}_3$ on the over all utilization $\mathfrak{I}$, using 6.8 and computed in Table 6.3, is 0.0544. This raises the question of what is the gain of increasing *maxHop* on the overall utilization ratio $\mathfrak{I}$ and wether it is worth the associated overhead to build and maintain *VIDs* with more hops.

To answer the previous question, we choose to compare the findings of utilization ratio $\mathfrak{I}$ in section 6.2 where $maxHops = 3$ against similar study with $maxHops = 5$ which is presented in this section. To find $\mathfrak{I}$ using (6.8), values of $\mathfrak{I}_k$ and their weights are needed which requires: finding $f(\varphi_k)$ and $f(\varpi_k)$ as $k \in \{1, 2, 3, 4, 5\}$ to calculate $\mathfrak{I}_k$ as in (6.7); In addition to, $m(k)$ and $\varphi_{k_{avg}}$ to calculate the weights of $\mathfrak{I}_k$. Models of $f(\varphi_k)$ and $f(\varpi_k)$ as $k \in \{1, 2, 3\}$ were presented in chapter 4 and section 6.1, respectively. However, for the remaining number of hops, i.e $k \in \{4, 5\}$, we need to find $f(\varphi_4)$, $f(\varphi_5)$, $f(\varpi_4)$ and $f(\varpi_5)$.

$f(\varphi_4)$ and $f(\varphi_5)$ are obtained by applying 4.17 which are plotted in Figures 7.9 and 7.10, respectively, at different values of Sp_{avg}. On the other hand, $f(\varpi_4)$ and $f(\varpi_5)$ can be found with the aid of (6.5) which also requires the derivation of $f(\xi_4^{in})$ and $f(\xi_5^{in})$. To derive $f(\xi_k^{in})$ in MMT, $f_{MMT}(\xi_k^{in})$, we use (5.80) while $f_{MMT}^{Sc.i.R}(\xi_i^{in})$ when $i \in \{1,2,3,4,5\}$ is shown in (5.78). In Figure 7.11, we present the models of $f(\xi_4^{in})$ and $f(\xi_5^{in})$ with $Ti = 2s$ while models of $f(\varpi_4)$ and $f(\varpi_5)$ for representative values of Ti and Sp_{avg} are plotted in Figures 7.12 and 7.13, respectively.

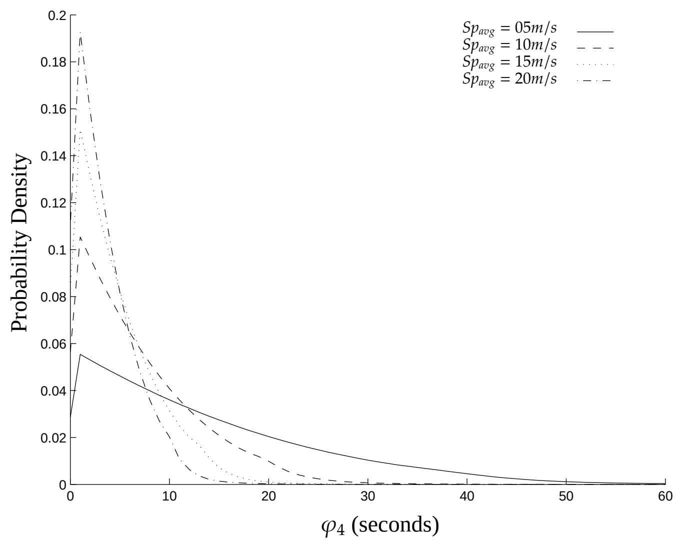

FIGURE 7.9: Model of $f(\varphi_4)$ with Sp_{avg} and $D_{TX} = 200m$

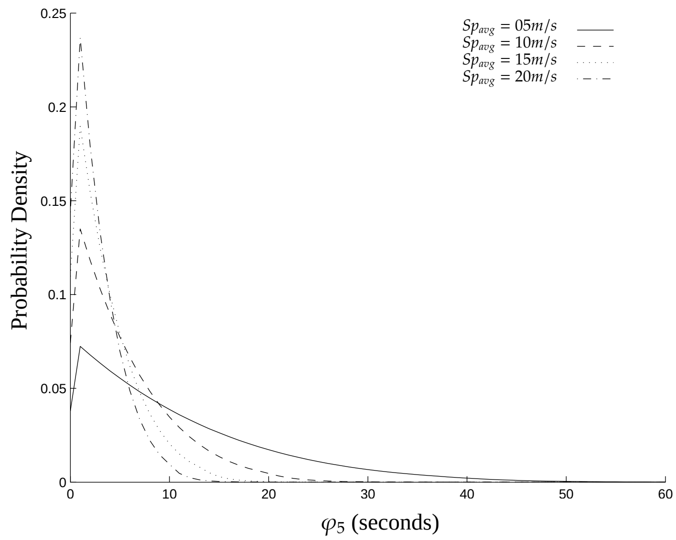

FIGURE 7.10: Model of $f(\varphi_5)$ with Sp_{avg} and $D_{TX} = 200m$

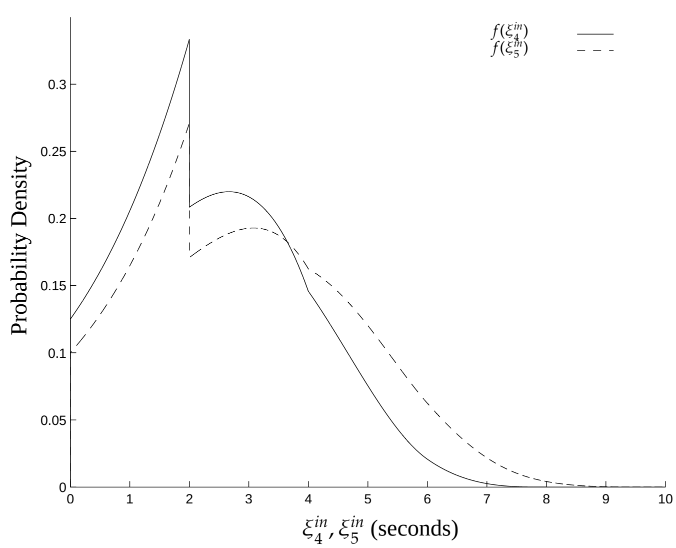

FIGURE 7.11: Model of $f(\xi_4^{in})$ and $f(\xi_5^{in})$ with $Ti = 2s$

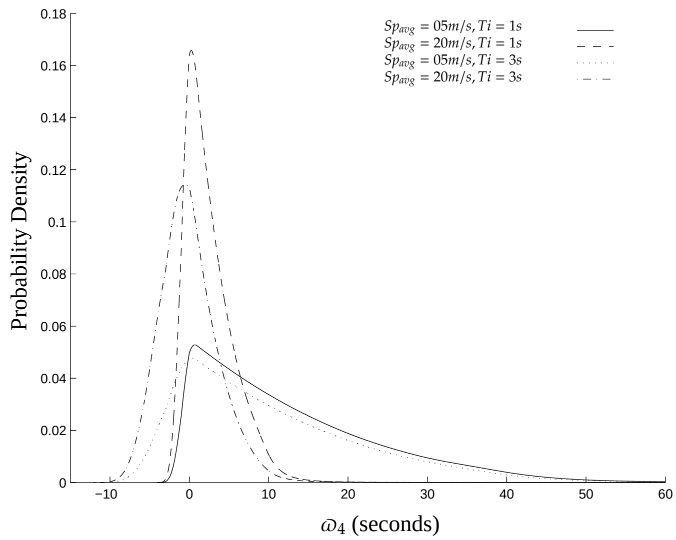

FIGURE 7.12: Model of $f(\varpi_4)$ in MMT with $D_{TX} = 200m$

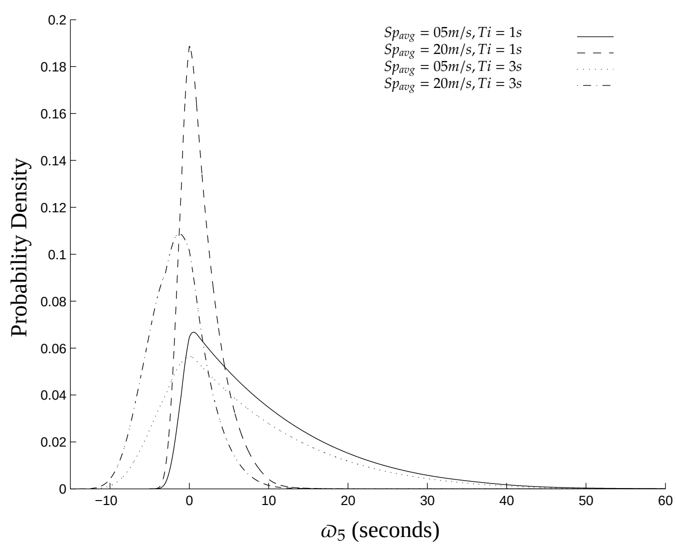

FIGURE 7.13: Model of $f(\varpi_5)$ in MMT with $D_{TX} = 200m$

At this point, we can calculate the values of $\mathfrak{I}_4$ and $\mathfrak{I}_5$ using (6.7) which are depicted in Figures 7.14 and 7.15, respectively, as $Ti \in \{1s, 2s, 3s\}$ and $Sp_{avg} \in \{5m/s, 10m/s, 15m/s, 20m/s\}$. As mentioned before, to compute the weights of $\mathfrak{I}_k$ in the total utilization ratio, $\mathfrak{I}$, we need to find $\varphi_{k_{avg}}$ and $m(k)$. Table 7.1 shows the values of $\varphi_{k_{avg}}$ as $k_{max} = 5$ which is an expansion of Table 3.7. Similarly, we collected the values of $m(k)$ from simulation as shown in Table 7.2. Using (6.8), we calculated the weights of $\mathfrak{I}_k$ as shown in Table 7.3.

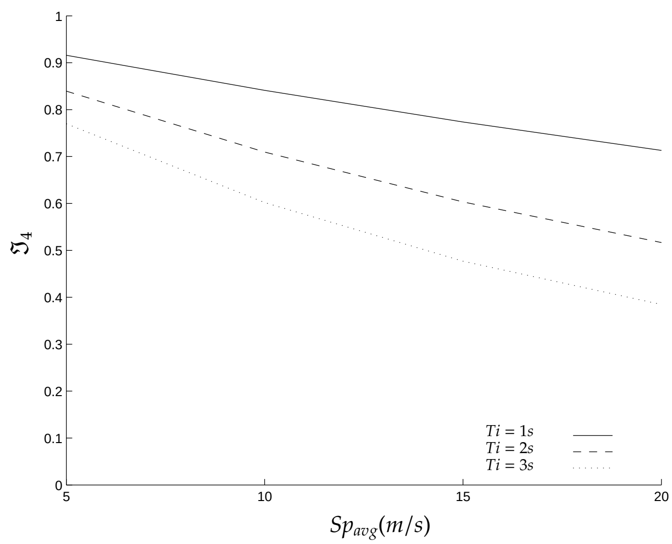

FIGURE 7.14: Model of $\mathfrak{I}_4$ in MMT with Sp_{avg}

TABLE 7.1: Values of $\varphi_{k_{avg}}$ with Sp_{avg} and $k_{max} = 5$

Number of hops k	Sp_{avg} $5m/s$	$10m/s$	$15m/s$	$20m/s$
$k = 1$	59.53s	29.25s	19.57s	14.81s
$k = 2$	29.81s	14.63s	09.75s	07.36s
$k = 3$	20.10s	09.80s	06.49s	04.94s
$k = 4$	14.11s	07.12s	04.74s	03.56s
$k = 5$	11.78s	05.61s	03.75s	02.81s

TABLE 7.2: Values of $m(k)$ and $k_{max} = 5$

Number of hops	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
$m(k)$	0.5216	0.2890	0.1232	0.0485	0.0177

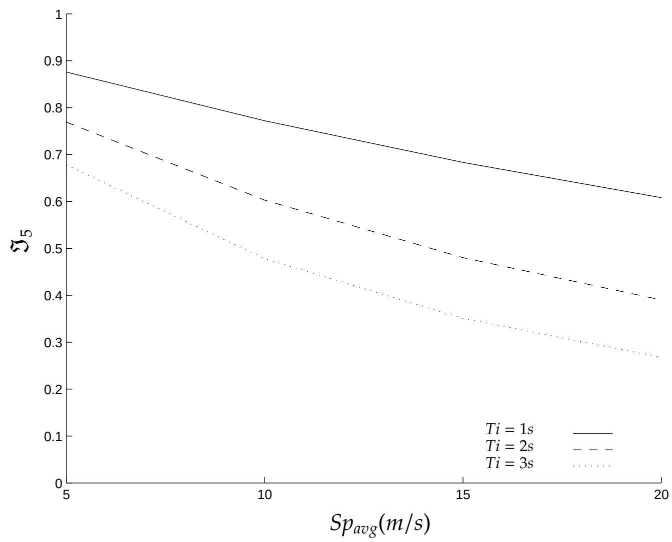

FIGURE 7.15: Model of $\Im_5$ in MMT with Sp_{avg}

TABLE 7.3: Calculating $\Im_k$ weight in $\Im$ and $k_{max} = 5$

Number of hops	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
$\Im_k$ weight	0.7215	0.2000	0.0575	0.0159	0.0049

We computed the total utilization ratio, $\Im$, when $k_{max} = 5$ and compare it with the findings in section 6.2 when $k_{max} = 3$. Figure 7.16 depicts the difference in $\Im$, $\Delta\Im = \Im^{@k_{max}=5} - \Im^{@k_{max}=3}$. It is evident that increasing k_{max} decreased the over all utilization ratio, $\Im$, as $\Delta\Im$ is negative. Considering the decrease magnitude we observe that it is almost negligible. To explain this decrease, we refer to the values of $\Im_1$ and $\Im_2$ for MMT in Figures 6.7 through 6.12 and compare it with the values of $\Im_4$ and $\Im_5$ in Figures 7.14 and 7.15 which clearly shows that $\Im_1$ and $\Im_2$ are higher than $\Im_4$ and $\Im_5$. Also, the data in Table 7.3 shows that the combined weights of $\Im_4$ and $\Im_5$ is approximately 0.02 which is the same reduction in the original weights of $\Im_1$ and $\Im_2$ in Table 6.3, thus reducing the utilization ratio $\Im$. The reduction in $\Im$ increases as we increase Sp_{avg} and Ti due to the steep decrease in $\Im_4$ and $\Im_5$ in Figures 7.14 and 7.15.

As a result, allowing MMT to build *VIDs* with high number of hops is not necessarily beneficial. *VIDs* of high number of *hops* are proven to have lower utilization ratio and hence degrading the overall utilization ratio in the network $\Im$. However, in some cases when node's density is low and the network is fragmented, building *VIDs* with higher number of *hops* is the only solution to achieve connectivity in the network.

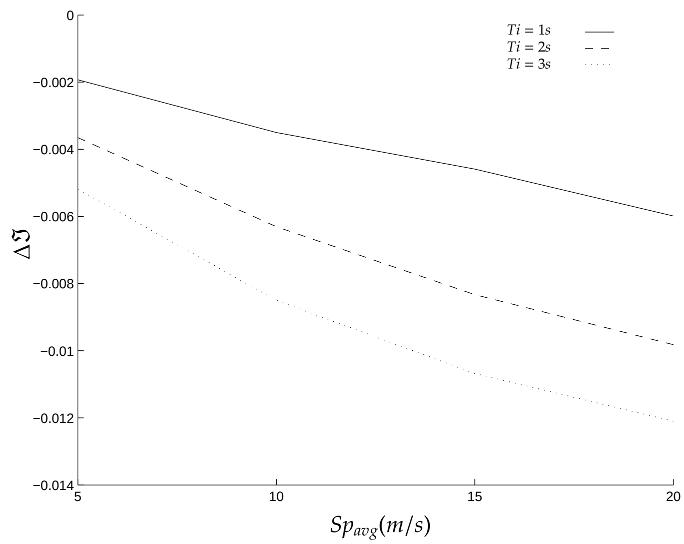

FIGURE 7.16: $\Delta\mathfrak{I}$ with Ti and Sp_{avg}

Chapter 8

Conclusions and Future Work

In this dissertation, novel analytical models were presented which are the key to study the impact of mobility on network performance. These models focused on studying the interactions among node mobility, changing topology and routing protocol performance. Firstly, we derived *Topological* models which provides the *pdf* of *TLinks* and *TPaths* time durations with different speeds. Secondly, we presented *Adaptability* models which deeply analyzes OLSR and MMT routing protocols targeting their behavior and the time elapsed in adapting to topology changes and translating them to logical information at the routing layer. *Adaptability* study models the *pdf* of *AdaptationDelays* in regards to different number of hops.

Then, we provided performance models of routing protocols through modeling usable duration of a *TLink* or *TPath*. The performance models were obtained by combining the *Topological* and *Adaptability* models. The usable duration models are the key in understanding the impact of mobility on network's performance and essential in identifying the benefits or the shortcomings of using a routing protocol over the other. Finally, we improved the performance of MMT in the light of the previous models by introducing an *Enhanced VID* selection criterion which is able to reduce communication overhead by reducing the rate of acquiring new *VIDs* regardless of their number of hops. In some cases, communication overhead was found to be very close to the ideal situation.

As this dissertation is unique in completing the impact chain of mobility on network's performance, we identified directions where further work is suggested. In *Topological* models, we used (4.17) to model $f(\varphi_k)$ when $k = 2$ or more, which is dependent on the random variable δ modeled using empirical results. $f(\delta)$ was approximated to a uniform distribution on $[0, 1.3]$; however analytical models of δ are still needed to decouple *Topological* models from simulation.

In *Adaptability* study we noticed that the average of *AdaptationDelays*, $\xi^{in}_{k_{avg}}$, increases with the number of hops k. This increase is expected to continue for MMT as evident in (5.78), (5.79) and (5.80). In fact, the maximum value of $\xi^{in}_{k_{max}}$ in MMT is found to be $k \times Ti$ seconds. On the other hand, $\xi^{in}_{k_{max}}$ in OLSR can be calculated by referring to the discussion in section 5.1.10 which details that *LPath* is built when all nodes involved have selected the needed *MPR*s then a *TC* packet containing required logical information is sent. Selecting all *MPR*s can take up to $3Ti$ seconds while the waiting to send a *TC* packet is Ti seconds resulting in $\xi^{in}_{k_{max}}$ in OLSR equals to $4Ti$ seconds. As a result, it is expected that $\xi^{in}_{k_{avg}}$ in OLSR will stop increasing while increasing the number of hops k, unlike MMT. This makes OLSR a more desirable routing protocol when the number of hops on *LPath* is large. However, the probability mass function of the number of hops k, $m(k)$, decreases dramatically when increasing k as evident in Table 7.2 which means that *LPath*s with more k hops have lower contribution to the overall network performance.

An important question can be raised. What is the impact of allowing the build and maintaining *LPath*s with larger k? A glimpse of this impact was discussed in section 7.2 which showed a decrease in network performance. The impact can be extended to include the increase on communication overhead. At the same time, an accompanying study on the need of *LPaths* with more k hops to prevent network segmentation and ensure connectivity is needed. Obviously, the study can investigate, as well, the need to balance the objectives of reducing communication overhead, maintaining network performance and connectivity.

Finally, similar *LPath* section criterion as the *Enhanced* in MMT, presented in section 7.1, can be adopted by other routing protocols. Moreover, incorporating the readings of received signal strengths from neighbors such as signal to noise ration *SNR* or received signal strength indication *RSSI* can be used to predict and avoid the use of failing *LPath*s. Similarly, using and sharing sensor readings among neighbors, such as the gyroscope which is readily available on many of mobile devices these days, can be beneficial in detecting movement and provide more intelligent *LPath* selection criterion.

Appendix A

MATLAB Modeling Code

A.1 Modeling $f(\ell)$

Arguments:

- res: The computational resolution, default value is 0.0005
- TX: The transmission range TX

Returns:

- ell: ℓ
- fell: $f(\ell)$

```
function [ell fell] = getfell(res,TX)
    ell=res:res:2*TX-res;
    [mell nell]=size(ell);
    onesell=ones(mell,nell);
    ell2 = ell.^2;
    TXM = onesell*TX;
    TX2 = TXM.^2;
    undersqrt = 4*TX2 - ell2;
    sqroot = undersqrt.^0.5;
    fell = (1/(2*TX)) * (ell./sqroot);
end
```

A.2 Modeling $f(v_r \mid_{v_R, v_A})$

Arguments:

- res: The computational resolution, default value is 0.0005
- vR: The speed of node R, v_R.
- vA: The speed of node A, v_A.

Returns:

- vrgiven: $v_r \mid_{v_R, v_A}$
- fvrgiven: $f(v_r \mid_{v_R, v_A})$

```
function [vrgiven fvrgiven] = getfvrgiven(res,vR,vA)
    vrgiven=abs(vR-vA)+res:res:vR+vA-res;
    [mvr nvr]=size(vrgiven);
    onesvrgiven=ones(mvr,nvr);
    vrgiven2 = vrgiven.^2;
    vR2 = (vR*onesvr).^2;
    vA2 = (vA*onesvr).^2;
    undersqrt = onesvrgiven - ((1/(2*vR*vA))*(vR2+vA2-vrgiven2)).^2;
    sqroot = undersqrt.^0.5;
    fvrgiven = (1/(pi*vR*vA)) * (vrgiven./sqroot);
end
```

A.3 Modeling $f(v_r)$

Arguments:

- res: The computational resolution, default value is 0.001
- Spmin: The minimum allowed speed, Sp_{min}.
- Spmax: The maximum allowed speed, Sp_{max}.

Returns:

- vr: v_r
- fvr: $f(v_r)$

```
function [vr fvr] = getfvr(res,Spmin,Spmax)
    vA=Spmin+res:res:Spmax-res;
    fvA=1/(Spmax-Spmin);
    vR=Spmin+res:res:Spmax-res;
    fvR=1/(Spmax-Spmin);
    [mv nv]=size(vA);
    vA=repmat(vA,nv,mv);
    vR=repmat(vR,nv,mv);
    vR=vR';
    [mv nv]=size(vA);
    vr=0+res:res:2*Spmax-res;
    [mvr nvr]=size(vr);
    fvr=zeros(mvr,nvr);
    multiplier = res*res * fvR * fvA * (1/mv*nv);
    for i=1:mv*nv
        [vrgiven fvrgiven]=getfvrgiven(res,vR(i),vA(i));
        fvrgiven = fvrgiven * multiplier;
        mask = (vr >= vrgiven(1)-res/100) & ~(vr > vrgiven(end)+res/100);
        fvr(mask) = fvr(mask) + fvrgiven;
    end
end
```

A.4 Modeling $F(\varphi_1)$

Arguments:

- res: The computational resolution, suggested value is 0.0005
- TX: The transmission range TX
- Spmin: The minimum allowed speed, Sp_{min}.
- Spmax: The maximum allowed speed, Sp_{max}.
- phimax: The maximum modeled φ_1, suggested value is 300s.

Returns:

- phi: φ_1
- Fphi: $F(\varphi_1)$

```
function [phi Fphi] = getTLinkDuration (res, TX, Spmin, Spmax, phimax)
    [ell fell]=getfell(res,TX);
    [vr fvr]=getfvr(res,Spmin,Spmax);
    [mvr nvr]=size(vr);
    phi=res:res:phimax;
    [mphi nphi]=size(phi);
    Fphi=zeros(mphi,nphi);
    for j=1:nphi
        currsum=0;
        for i=1:nvr
            if((vr(i)*phi(j))<2*TX-res)
                temp = 1 - sqrt( 1 - ( ( vr(i)*phi(j) ) / ( 2*TX ) )^2 );
            else
                temp = 1 - sqrt( 1 - ( ( 2*TX-res ) / ( 2*TX ) )^2 );
            end
            temp = temp*fvr(i)*res;
            currsum = currsum +temp;
        end
        Fphi(j)=currsum;
    end
end
```

A.5 Generating an array of random values following a known CDF

Arguments:

- m: Number of rows in the returned random array
- n: Number of rows in the returned random array
- X: X values of the passed CDF function for which the random values are generated
- Y: Y values of the passed CDF function for which the random values are generated

Returns:

- result: The array of random values

```
function [result] = getRandomArray(m, n, X, Y)
    [limit1 limit2]=size(Y);
    dump = diff(X);
    res=dump(1);
    Random = rand(1,m*n)*Y(end);
    for i=1:(m*n)
        temp=(Random(i)>=Y);
        temp=temp(temp);
        [a b]=size(temp);
        if(b==limit2)
            Random(i)=-inf;
        else
            Random(i)=X(b+1)-rand(1)*res;
        end
    end
    result=[];
    for i=1:m
        result=[result;Random(((i-1)*n)+1:i*n)];
    end
end
```

A.6 Generating $f(\xi_k^{in})$ for MMT

Arguments:

- res: The computational resolution, suggested value is 0.0005
- k: Number of hops
- Ti: The duration of sending topological information

Returns:

- scale: The x-axis of $f(\xi_k^{in})$
- pdf: The y-axis of $f(\xi_k^{in})$

```
function [scale pdf]=getAdaptationMMT(hops,Ti,res)
    baseScale=0:res:Ti;
    [baseM baseN]=size(baseScale);
    basepdf=ones(baseM,baseN)*(1/Ti);
    basepdf(end)=0;
    if(hops==1)
        scale=baseScale;
        pdf=basepdf;
    else
        currentScale=baseScale;
        currentpdf=basepdf;
        for i=2:hops
            currentScale=[currentScale(1:end-1),currentScale(end)*ones(baseM,
   baseN)+baseScale];
            currentpdf=conv(currentpdf,basepdf)*res;
        end
        scale=currentScale;
        pdf=currentpdf;
    end
end
```

A.7 Implementing Core Probabilities for Adaptability Study

Arguments:

- Vars: The number of variables in the formulation, in P_{2A}, Vars = 2
- ID: The ID of the formulation, in P_{2A}, ID = 'A'
- Ti: The duration of sending topological information
- axisIN: the x-axis to be used in computing the returned formulation
- ShiftFactor: default is 0.0, can be used to accommodate shifting the resulting formulation ShiftFactor*Ti to the right

Returns:

- returnModel: The resulting formulation, for example, P_{2A}

```matlab
function returnModel = PModel( Vars, ID, Ti, axisIN, ShiftFactor)
    [m n]=size(axisIN);
    ONES = ones(m,n);
    axis=axisIN-Ti*ShiftFactor*ONES;
    switch Vars
        case 1
            returnModel = ((1/(1*(Ti^1)))*(axis).^0);
        case 2
            switch ID
                case 'A'
                    returnModel = ((1/(1*(Ti^2)))*(axis).^1);
                case 'B'
                    returnModel = ((1/(1*(Ti^1)))*(axis).^0) -((1/(1*(Ti^2)))*(axis).^1);
            end
        case 3
            switch ID
                case 'A'
                    returnModel = ((1/(2*(Ti^3)))*(axis).^2);
                case 'B'
                    returnModel = ((1/(1*(Ti^2)))*(axis).^1) -((1/(1*(Ti^3)))*(axis).^2);
```

```
                case 'C'
                    returnModel = ((1/(2*(Ti^1)))*(axis).^0) -((1/(1*(Ti^2)))*(axis).^1) +((1/(2*(Ti^3)))*(axis).^2);
            end
        case 4
            switch ID
                case 'A'
                    returnModel = ((1/(6*(Ti^4)))*(axis).^3);
                case 'B'
                    returnModel = ((1/(2*(Ti^3)))*(axis).^2) -((1/(2*(Ti^4)))*(axis).^3);
                case 'C'
                    returnModel = ((1/(2*(Ti^2)))*(axis).^1) -((1/(1*(Ti^3)))*(axis).^2) +((1/(2*(Ti^4)))*(axis).^3);
                case 'D'
                    returnModel = ((1/(6*(Ti^1)))*(axis).^0) -((1/(2*(Ti^2)))*(axis).^1) +((1/(2*(Ti^3)))*(axis).^2) -((1/(6*(Ti^4)))*(axis).^3);
            end
        case 5
            switch ID
                case 'A'
                    returnModel = ((1/(24*(Ti^5)))*(axis).^4);
                case 'B'
                    returnModel = ((1/(6*(Ti^4)))*(axis).^3) -((1/(6*(Ti^5)))*(axis).^4);
                case 'C'
                    returnModel = ((1/(4*(Ti^3)))*(axis).^2) -((1/(2*(Ti^4)))*(axis).^3) +((1/(4*(Ti^5)))*(axis).^4);
                case 'D'
                    returnModel = ((1/(6*(Ti^2)))*(axis).^1) -((1/(2*(Ti^3)))*(axis).^2) +((1/(2*(Ti^4)))*(axis).^3) -((1/(6*(Ti^5)))*(axis).^4);
                case 'E'
                    returnModel = ((1/(24*(Ti^1)))*(axis).^0) -((1/(6*(Ti^2)))*(axis).^1) +((1/(4*(Ti^3)))*(axis).^2) -((1/(6*(Ti^4)))*(axis).^3) +((1/(24*(Ti^5)))*(axis).^4);
            end
        case 6
            switch ID
                case 'A'
                    returnModel = ((1/(120*(Ti^6)))*(axis).^5);
                case 'B'
                    returnModel = ((1/(24*(Ti^5)))*(axis).^4) -((1/(24*(Ti^6)))*(axis).^5);
```

```
                case 'C'
                    returnModel = ((1/(12*(Ti^4)))*(axis).^3) -((1/(6*(Ti^5)))*(axis).^4) +((1/(12*(Ti^6)))*(axis).^5);
                case 'D'
                    returnModel = ((1/(12*(Ti^3)))*(axis).^2) -((1/(4*(Ti^4)))*(axis).^3) +((1/(4*(Ti^5)))*(axis).^4) -((1/(12*(Ti^6)))*(axis).^5);
                case 'E'
                    returnModel = ((1/(24*(Ti^2)))*(axis).^1) -((1/(6*(Ti^3)))*(axis).^2) +((1/(4*(Ti^4)))*(axis).^3) -((1/(6*(Ti^5)))*(axis).^4) +((1/(24*(Ti^6)))*(axis).^5);
                case 'F'
                    returnModel = ((1/(120*(Ti^1)))*(axis).^0) -((1/(24*(Ti^2)))*(axis).^1) +((1/(12*(Ti^3)))*(axis).^2) -((1/(12*(Ti^4)))*(axis).^3) +((1/(24*(Ti^5)))*(axis).^4) -((1/(120*(Ti^6)))*(axis).^5);
            end
    end
end
```

A.8 Modeling $f(\varpi_k)$

Arguments:

- res: The computational resolution, suggested value is 0.0005
- scaleAdaptation: The x-axis of $f(\xi_k^{in})$
- pdfAdaptation: The y-axis of $f(\xi_k^{in})$
- scaleDuration: The x-axis of $f(\varphi_k)$
- pdfDuration: The y-axis of $f(\varphi_k)$

Returns:

- scalewkModel: The x-axis of $f(\varpi_k)$
- pdfwkModel: The y-axis of $f(\varpi_k)$

```
function [scalewkModel pdfwkModel] = modelwkAndPoswk(scaleAdaptation,
    pdfAdaptation,scaleDuration,pdfDuration,res)
    temp = scaleAdaptation(1):res:scaleAdaptation(end);
    pdfAdaptation = interp1(scaleAdaptation,pdfAdaptation,temp);
    scaleAdaptation = temp;
    temp = scaleDuration(1):res:scaleDuration(end);
    pdfDuration = interp1(scaleDuration,pdfDuration,temp);
    scaleDuration = temp;
    scalewkModel=[-1*scaleAdaptation(end):res:-1*res,scaleDuration];
    pdfwkModel=conv(fliplr(pdfAdaptation),pdfDuration)*res;
end
```

Bibliography

[1] Richard J. La and Yijie Han. Distribution of path durations in mobile ad hoc networks and path selection. *IEEE/ACM Trans. Netw.*, 15(5):993–1006, October 2007. ISSN 1063-6692. doi: 10.1109/TNET.2007.896241. URL http://dx.doi.org/10.1109/TNET.2007.896241.

[2] Narayanan Sadagopan, Fan Bai, Bhaskar Krishnamachari, and Ahmed Helmy. Paths: analysis of path duration statistics and their impact on reactive manet routing protocols. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '03, pages 245–256, New York, NY, USA, 2003. ACM. ISBN 1-58113-684-6. doi: 10.1145/778415.778444. URL http://doi.acm.org/10.1145/778415.778444.

[3] G. Carofiglio, C.-F. Chiasserini, M. Garetto, and E. Leonardi. Route stability in manets under the random direction mobility model. *Mobile Computing, IEEE Transactions on*, 8(9):1167 –1179, sept. 2009. ISSN 1536-1233. doi: 10.1109/TMC.2009.20.

[4] F. Bai, N. Sadagopan, B. Krishnamachari, and A. Helmy. Modeling path duration distributions in manets and their impact on reactive routing protocols. *Selected Areas in Communications, IEEE Journal on*, 22(7):1357 – 1373, sept. 2004. ISSN 0733-8716. doi: 10.1109/JSAC.2004.829353.

[5] Le The Dung and Beongku An. A modeling framework for supporting and evaluating performance of multi-hop paths in mobile ad-hoc wireless networks. *Comput. Math. Appl.*, 64(5):1197–1205, September 2012. ISSN 0898-1221. doi: 10.1016/j.camwa.2012.03.063. URL http://dx.doi.org/10.1016/j.camwa.2012.03.063.

[6] Le The Dung, Beongku An, Nam-Soo Kim, and Do-Hyeon Kim. An analytical model for performance evaluation of multi-hop paths in mobile ad-hoc wireless

networks. In *Ubiquitous and Future Networks (ICUFN), 2012 Fourth International Conference on*, pages 58 –62, july 2012. doi: 10.1109/ICUFN.2012.6261664.

[7] P. Samar and S.B. Wicker. Link dynamics and protocol design in a multihop mobile environment. *Mobile Computing, IEEE Transactions on*, 5(9):1156 –1172, sept. 2006. ISSN 1536-1233. doi: 10.1109/TMC.2006.131.

[8] Prince Samar and Stephen B. Wicker. On the behavior of communication links of a node in a multi-hop mobile environment. In *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '04, pages 145–156, New York, NY, USA, 2004. ACM. ISBN 1-58113-849-0. doi: 10.1145/989459.989478. URL http://doi.acm.org/10.1145/989459.989478.

[9] Abbas Nayebi, Gunnar Karlsson, and Hamid Sarbazi-Azad. Evaluation and design of beaconing in mobile wireless networks. *Ad Hoc Netw.*, 9(3):368–386, May 2011. ISSN 1570-8705. doi: 10.1016/j.adhoc.2010.08.014. URL http://dx.doi.org/10.1016/j.adhoc.2010.08.014.

[10] Abbas Nayebi and Hamid Sarbazi-Azad. Analysis of link lifetime in wireless mobile networks. *Ad Hoc Networks*, 10(7):1221 – 1237, 2012. ISSN 1570-8705. doi: 10.1016/j.adhoc.2012.03.007. URL http://www.sciencedirect.com/science/article/pii/S1570870512000480.

[11] N. Meghanathan and A. Farago. Survey and taxonomy of unicast routing protocols for mobile ad hoc networks. *Technical Report UTDCS-40-04, University of Texas at Dallas*, nov. 2004.

[12] N. Qadri and A. Liotta. Analysis of pervasive mobile ad hoc routing protocols. *Pervasive Computing: Innovations in Intelligent Multimedia and Applications, Computer Communications and Networks, Springer-Verlag London*, nov. 2009. doi: 10.1007/978-1-84882-599-419.

[13] Tridib Mukherjee, Sandeep K. S. Gupta, and Georgios Varsamopoulos. Analytical model for optimizing periodic route maintenance in proactive routing for manets. In *Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*, MSWiM '07, pages 201–208, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-851-0. doi: 10.1145/1298126.1298163. URL http://doi.acm.org/10.1145/1298126.1298163.

[14] J. Jubin and J.D. Tornow. The darpa packet radio network protocols. *Proceedings of the IEEE*, 75(1):21 – 32, jan. 1987. ISSN 0018-9219. doi: 10.1109/PROC.1987.13702.

[15] Zygmunt J. Haas and Marc R. Pearlman. The performance of query control schemes for the zone routing protocol. *IEEE/ACM Trans. Netw.*, 9(4): 427–438, August 2001. ISSN 1063-6692. doi: 10.1109/90.944341. URL http://dx.doi.org/10.1109/90.944341.

[16] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. In *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International*, pages 62 – 68, 2001. doi: 10.1109/INMIC.2001.995315.

[17] Guangyu Pei, M. Gerla, and Tsu-Wei Chen. Fisheye state routing: a routing scheme for ad hoc wireless networks. In *Communications, 2000. ICC 2000. 2000 IEEE International Conference on*, volume 1, pages 70 –74 vol.1, 2000. doi: 10.1109/ICC.2000.853066.

[18] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. *SIGCOMM Comput. Commun. Rev.*, 24(4):234–244, October 1994. ISSN 0146-4833. doi: 10.1145/190809.190336. URL http://doi.acm.org/10.1145/190809.190336.

[19] Shree Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *Mob. Netw. Appl.*, 1(2):183–197, October 1996. ISSN 1383-469X. doi: 10.1007/BF01193336. URL http://dx.doi.org/10.1007/BF01193336.

[20] J.J. Garcia-Luna-Aceves and M. Spohn. Efficient routing in packet-radio networks using link-state information. In *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, pages 1308 –1312 vol.3, 1999. doi: 10.1109/WCNC.1999.796949.

[21] B. Bellur and R.G. Ogier. A reliable, efficient topology broadcast protocol for dynamic networks. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 178 –186 vol.1, mar 1999. doi: 10.1109/INFCOM.1999.749266.

[22] David B. Johnson, David A. Maltz, and Josh Broch. Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks. In *In Ad Hoc Networking, edited by Charles E. Perkins, Chapter 5*, pages 139–172, Boston, MA, USA, 2001. Addison-Wesley Longman Publishing Co., Inc. ISBN 0-201-30976-9. URL http://dl.acm.org/citation.cfm?id=374547.374552.

[23] Vincent D. Park and M. Scott Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of the INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution*, INFOCOM '97, pages 1405–, Washington, DC, USA, 1997. IEEE Computer Society. ISBN 0-8186-7780-5. URL http://dl.acm.org/citation.cfm?id=839292.843010.

[24] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. In *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, pages 90 –100, feb 1999. doi: 10.1109/MCSA.1999.749281.

[25] Zygmunt J. Haas and Marc R. Pearlman. The performance of query control schemes for the zone routing protocol. *IEEE/ACM Trans. Netw.*, 9(4): 427–438, August 2001. ISSN 1063-6692. doi: 10.1109/90.944341. URL http://dx.doi.org/10.1109/90.944341.

[26] Nicholas Martin, Yamin Al-Mousa, and Nirmala Shenoy. An integrated routing and medium access control framework for surveillance networks of mobile devices. In *Proceedings of the 12th international conference on Distributed computing and networking*, ICDCN'11, pages 315–327, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 3-642-17678-X, 978-3-642-17678-4. URL http://dl.acm.org/citation.cfm?id=1946143.1946171.

[27] Michael Iannacone, Yamin Al-Mousa, Nicholas Martin, Nirmala Shenoy, and John Fischer. A low-latency tdma scheduler for multi-hop cluster based manets with directional antennas. In Jun Zheng, Shiwen Mao, Scott F. Midkiff, and Hua Zhu, editors, *ADHOCNETS*, volume 28 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 896–912. Springer, 2009. ISBN 978-3-642-11722-0. URL http://dblp.uni-trier.de/db/conf/adhocnets/adhocnets2009.html#IannaconeAMSF09.

[28] N. Shenoy, Yin Pan, D. Narayan, D. Ross, and C. Lutzer. Route robustness of a multi-meshed tree routing scheme for internet manets. In *Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*, volume 6, pages 6 pp. –3351, dec. 2005. doi: 10.1109/GLOCOM.2005.1578394.

[29] Y. Al-Mousa, W. Huba, and N. Shenoy. An integrated tdma-based mac and routing solution for airborne backbone networks using directional antennas. In *ICNS 2011, The Seventh International Conference on Networking and Services*, pages 234–239, Venice/Mestre, Italy, may 2011. ISBN 978-1-61208-133-5.

[30] Geunhwi Lim, Kwangwook Shin, Seunghak Lee, H. Yoon, and Joong Soo Ma. Link stability and route lifetime in ad-hoc wireless networks. In *Parallel Processing Workshops, 2002. Proceedings. International Conference on*, pages 116 – 123, 2002. doi: 10.1109/ICPPW.2002.1039720.

[31] Yangcheng Huang, G. Kannan, S. Bhatti, S.N. Merchant, and U.B. Desai. Route dynamics for shortest path first routing in mobile ad hoc networks. In *Wireless Telecommunications Symposium, 2008. WTS 2008*, pages 236 –242, april 2008. doi: 10.1109/WTS.2008.4547570.

[32] Chai-Keong Toh. Associativity-based routing for ad hoc mobile networks. *Wirel. Pers. Commun.*, 4(2):103–139, March 1997. ISSN 0929-6212. doi: 10.1023/A:1008812928561. URL http://dx.doi.org/10.1023/A:1008812928561.

[33] R. Dube, C.D. Rais, Kuang-Yeh Wang, and S.K. Tripathi. Signal stability-based adaptive routing (ssa) for ad hoc mobile networks. *Personal Communications, IEEE*, 4(1):36 –45, feb 1997. ISSN 1070-9916. doi: 10.1109/98.575990.

[34] S. Agarwal, A. Ahuja, J.P. Singh, and R. Shorey. Route-lifetime assessment based routing (rabr) protocol for mobile ad-hoc networks. In *Communications, 2000. ICC 2000. 2000 IEEE International Conference on*, volume 3, pages 1697 –1701 vol.3, 2000. doi: 10.1109/ICC.2000.853783.

[35] William Su, Sung-Ju Lee, and Mario Gerla. Mobility prediction and routing in ad hoc wireless networks. *Int. J. Netw. Manag.*, 11(1):3–30, January 2001. ISSN 1099-1190. doi: 10.1002/nem.386. URL http://dx.doi.org/10.1002/nem.386.

[36] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing (WCMC):*

Special Issue on Mobile Ad Hoc Networking: Research, Trends and Applications, 2:483–502, 2002.

[37] J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 1312 – 1321 vol.2, march-3 april 2003. doi: 10.1109/INFCOM.2003.1208967.

[38] E. Hyytia, P. Lassila, and J. Virtamo. Spatial node distribution of the random waypoint mobility model with applications. *Mobile Computing, IEEE Transactions on*, 5(6):680 – 694, june 2006. ISSN 1536-1233. doi: 10.1109/TMC.2006.86.

[39] Vincent Lenders, Jörg Wagner, and Martin May. Analyzing the impact of mobility in ad hoc networks. In *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, REALMAN '06, pages 39–46, New York, NY, USA, 2006. ACM. ISBN 1-59593-360-3. doi: 10.1145/1132983.1132991. URL http://doi.acm.org/10.1145/1132983.1132991.

[40] D. Turgut, S.K. Das, and M. Chatterjee. Longevity of routes in mobile ad hoc networks. In *Vehicular Technology Conference, 2001. VTC 2001 Spring. IEEE VTS 53rd*, volume 4, pages 2833 –2837 vol.4, 2001. doi: 10.1109/VETECS.2001.944118.

[41] Yangcheng Huang, G. Kannan, S. Bhatti, S.N. Merchant, and U.B. Desai. Route dynamics for shortest path first routing in mobile ad hoc networks. In *Wireless Telecommunications Symposium, 2008. WTS 2008*, pages 236 –242, april 2008. doi: 10.1109/WTS.2008.4547570.

[42] Ming Zhao, Yujin Li, and Wenye Wang. Modeling and analytical study of link properties in multihop wireless networks. *Communications, IEEE Transactions on*, 60(2):445 –455, february 2012. ISSN 0090-6778. doi: 10.1109/TCOMM.2012.010512.090739.

[43] Yijie Han, Richard J. La, Armand M. Makowski, and Seungjoon Lee. Distribution of path durations in mobile ad-hoc networks: Palm's theorem to the rescue. *Comput. Netw.*, 50(12):1887–1900, August 2006. ISSN 1389-1286. doi: 10.1016/j.comnet.2005.10.005. URL http://dx.doi.org/10.1016/j.comnet.2005.10.005.

[44] Yijie Han, Richard J. La, and M. Makowski. Distribution of path durations in mobile ad-hoc networks palms theorem at work. In *in Proceedings of the ITC Specialist Seminar on Performance Evaluation of Wireless and Mobile Systems*, 2004.

[45] Y. Han, R. J. La, and H. Zhang. Path selection in mobile ad-hoc networks and distribution of path duration. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1 –12, april 2006. doi: 10.1109/INFOCOM.2006.11.

[46] A. Triviño Cabrera, J. Garcia-de-la Nava, E. Casilari, and F. J. González-Cañete. An analytical model to estimate path duration in manets. In *Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems*, MSWiM '06, pages 183–186, New York, NY, USA, 2006. ACM. ISBN 1-59593-477-4. doi: 10.1145/1164717.1164749. URL http://doi.acm.org/10.1145/1164717.1164749.

[47] Yu-Chee Tseng, Yueh-Feng Li, and Yu-Chia Chang. On route lifetime in multihop mobile ad hoc networks. *Mobile Computing, IEEE Transactions on*, 2(4):366 – 376, oct.-dec. 2003. ISSN 1536-1233. doi: 10.1109/TMC.2003.1255651.

[48] K. Namuduri and R. Pendse. Analytical estimation of path duration in mobile ad hoc networks. *Sensors Journal, IEEE*, 12(6):1828 –1835, june 2012. ISSN 1530-437X. doi: 10.1109/JSEN.2011.2176927.

[49] I. Gruber and Hui Li. Link expiration times in mobile ad hoc networks. In *Local Computer Networks, 2002. Proceedings. LCN 2002. 27th Annual IEEE Conference on*, pages 743 – 750, nov. 2002. doi: 10.1109/LCN.2002.1181857.

[50] M. Pascoe, J. Gomez, V. Rangel, and M. Lopez-Guerrero. Modeling route duration in mobile ad-hoc networks. In *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on*, pages 1 –9, oct. 2007. doi: 10.1109/MOBHOC.2007.4428615.

[51] Michael Pascoe-Chalke, Javier Gomez, Victor Rangel, and Miguel Lopez-Guerrero. Route duration modeling for mobile ad-hoc networks. *Wirel. Netw.*, 16(3):743–757, April 2010. ISSN 1022-0038. doi: 10.1007/s11276-009-0166-1. URL http://dx.doi.org/10.1007/s11276-009-0166-1.

[52] Abbas Nayebi. A comment on link dynamics and protocol design in a multi-hop mobile environment. In *Wireless Sensor Network*, volume 3, pages 114–116, 2011. doi: 10.4236/wsn.2011.33012.

[53] Xianren Wu, Hamid R. Sadjadpour, and J. J. Garcia-Luna-Aceves. From link dynamics to path lifetime and packet-length optimization in manets. *Wirel. Netw.*, 15(5):637–650, July 2009. ISSN 1022-0038. doi: 10.1007/s11276-007-0086-x. URL http://dx.doi.org/10.1007/s11276-007-0086-x.

[54] Xianren Wu, H.R. Sadjadpour, and J.J. Garcia-Luna-Aceves. Link lifetime as a function of node mobility in manets with restricted mobility: Modeling and applications. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops, 2007. WiOpt 2007. 5th International Symposium on*, pages 1 –10, april 2007. doi: 10.1109/WIOPT.2007.4480023.

[55] Jian-Kai Chen, Chien Chen, Rong-Hong Jan, and Hsia-Hsin Li. Expected link life time analysis in manet under manhattan grid mobility model. In *Proceedings of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems*, MSWiM '08, pages 162–168, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-235-1. doi: 10.1145/1454503.1454534. URL http://doi.acm.org/10.1145/1454503.1454534.

[56] Dan Yu, Hui Li, and I. Gruber. Path availability in ad hoc network. In *Telecommunications, 2003. ICT 2003. 10th International Conference on*, volume 1, pages 383 – 387 vol.1, feb.-1 march 2003. doi: 10.1109/ICTEL.2003.1191260.

[57] A.B. McDonald and T. Znati. A path availability model for wireless ad-hoc networks. In *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, pages 35 –40 vol.1, 1999. doi: 10.1109/WCNC.1999.797781.

[58] M. Gerharz, C. de Waal, M. Frank, and P. Martini. Link stability in mobile wireless ad hoc networks. In *Local Computer Networks, 2002. Proceedings. LCN 2002. 27th Annual IEEE Conference on*, pages 30 – 39, nov. 2002. doi: 10.1109/LCN.2002.1181761.

[59] A. Nayebi and H. Sarbazi-Azad. Lifetime analysis of the logical topology constructed by homogeneous topology control in wireless mobile networks. In *Parallel and Distributed Systems, 2007 International Conference on*, volume 2, pages 1 –8, dec. 2007. doi: 10.1109/ICPADS.2007.4447825.

[60] S. Arbindi, K. Namuduri, and R. Pendse. Statistical estimation of route expiry times in on-demand ad hoc routing protocols. In *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on*, pages 8 pp. –23, nov. 2005. doi: 10.1109/MAHSS.2005.1542769.

[61] Dongsheng Chen, A. Babaei, and P. Agrawal. An efficient cluster update model in manet. In *System Theory (SSST), 2012 44th Southeastern Symposium on*, pages 73 –77, march 2012. doi: 10.1109/SSST.2012.6195153.

[62] Yangcheng Huang, S. Bhatti, and S.-A. Sorensen. Reducing neighbour detection latency in olsr. In *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*, pages 1 –5, sept. 2007. doi: 10.1109/PIMRC.2007.4394779.

[63] Chen Zhao and Mihail L. Sichitiu. Contact time in random walk and random waypoint: Dichotomy in tail distribution. *Ad Hoc Networks*, 9(2): 152 – 163, 2011. ISSN 1570-8705. doi: 10.1016/j.adhoc.2010.03.005. URL http://www.sciencedirect.com/science/article/pii/S1570870510000466.