Simulation Based Performance Evaluation of Routing Protocols and TCP Variants in Mobile Ad-hoc Networks.

MD NAZMUL ISLAM KHAN
RASHED AHMED
This thesis is submitted to the School of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:
Author(s):
Md. Nazmul Islam khan
E-mail: sajib333@yahoo.com

Rashed Ahmed
E-mail: rashed_babu2003@yahoo.com

Advisor:
Mr. Alexandru Popescu
Blekinge Institute of Technology
School of Computing
371 79, Karlskrona, Sweden
E-mail: alexandru.popescu@bth.se

Examiner:
Dr. Patrik Arlos
Blekinge Institute of Technology
School of Computing
371 79, Karlskrona, Sweden
E-mail: patrik.arlos@bth.se
ABSTRACT

A Mobile Ad-hoc Network (MANET) is a collection of mobile devices dynamically forming a communication network without any centralized control and pre-existing network infrastructure. Due to the presence of mobility in the MANET, the interconnections between stations are likely to change on a continual basis, resulting in frequent changes of network topology. Consequently, routing becomes a vital factor and a major challenge in such a network. This research aims to study the impact of four IETF (Internet Engineering Task Force) standardized routing protocols on MANETs and thereby comprehensively analyzes their performance under varying network sizes and node mobility rates. The four routing protocols that are considered in the analysis are Optimized Link State Routing (OLSR), Ad-hoc On-demand Distance Vector (AODV), Dynamic Source Routing (DSR) and Temporary Ordered Routing Algorithm (TORA). In addition, from a transport layer’s perspective, it is necessary to consider Transmission Control Protocol (TCP) as well for MANETs because of its wide application, which enjoys the advantage of reliable data transmission in the Internet. However, the factors such as scalability and mobility cause TCP to suffer from a number of severe performance problems in an ad-hoc environment. Hence, it is of utmost importance to identify the most suitable and efficient TCP variants that can robustly perform under these specific conditions. Therefore, this dissertation also makes an attempt to evaluate the performance of the three TCP variants (Reno, New Reno and SACK) under a variety of network conditions. The simulations results reveal that out of the three, the SACK variant can adapt relatively well to the changing network sizes and node speeds. As the study demonstrates, its use is therefore highly recommended as one of the most robust variants in a majority of environments. On the other hand, the research asserts the fact of superiority of proactive protocol, over reactive and hybrid ones when routing the same traffic in the network. Nonetheless, among the reactive protocols AODV performance (in the presence of a high mobility) has been found to be remarkable.

Keywords: MANETs, OPNET, Routing Protocol, TCP.
Acknowledgement

First of all, we would like to thanks Almighty Allah for blessing us with the ability, patience and necessary strength to complete this thesis.

We would like to thank our supervisor Mr. Alexandru Popescu for his excellent support, kind commitment, encouragement, guidance, comments and making OPNET available for simulations to be carried out during this research process. Without his contributions this research would not have been possible.

We are grateful to our examiner, Dr. Patrik Arlos for useful suggestions, comments and helping with office discipline during parts of the thesis process and for the overall cordial supervision to improve the quality of thesis work.

We are also grateful to our all of my friends those who give us suggestions, discussions and comments led to a viable contribution to the simulation experiment and dissertation write-up. Thank you for your support.

Finally, we would like to thank our parents and family for their support overseas. Without their help, we cannot complete the dissertation on time.

Md. Nazmul Islam
Rashed Ahmed
# Contents

Abstract ......................................................................................................................................... i
Acknowledgement .................................................................................................................. iii
Contents ......................................................................................................................................... iv
List of Figures .............................................................................................................................. vii
List of Table ................................................................................................................................ ix
Acronyms ...................................................................................................................................... x

1 Introduction and Methodology ................................................................................................. 1
   1.1 Background ......................................................................................................................... 1
   1.2 Problem Statement .............................................................................................................. 2
   1.3 Motivation and Main Contribution ...................................................................................... 3
   1.4 Aims and Objectives ........................................................................................................... 5
   1.5 Research Questions ............................................................................................................. 6
   1.6 Research Methodology ........................................................................................................ 6
      1.6.1 Justification of Research Methodology ......................................................................... 6
      1.6.2 Investigation Framework .............................................................................................. 7
   1.7 Related Work ....................................................................................................................... 9
   1.8 Thesis Structure .................................................................................................................. 10

2 Transport Control Protocol (TCP) ............................................................................................. 11
   2.1 Congestion Control Algorithms ......................................................................................... 11
      2.1.1 Slow start and Congestion Avoidance ........................................................................... 11
      2.1.2 Fast Retransmission and Fast recovery ......................................................................... 12
   2.2 TCP Variants ....................................................................................................................... 13
      2.2.1 TCP Tahoe ..................................................................................................................... 13
      2.2.2 TCP Reno ...................................................................................................................... 13
      2.2.3 TCP New Reno .............................................................................................................. 14
      2.2.4 TCP SACK .................................................................................................................... 15

3 TCP Performances in MANETS ............................................................................................... 16
   3.1 High BER .............................................................................................................................. 16
   3.2 Route failures ....................................................................................................................... 16
   3.3 Path Asymmetry Impact ....................................................................................................... 16
3.4 Network Partitioning ................................................................. 17
3.5 Power Scarcity ........................................................................... 18
3.6 Multipath routing ....................................................................... 18
3.7 Interaction between MAC protocol and TCP .............................. 18
3.8 Hidden and Exposed Node Impact ................................................ 18
4 The Routing Protocols .................................................................. 20
  4.1 OLSR ....................................................................................... 20
    4.1.1 OLSR Protocol Functioning ............................................... 21
      4.1.1.1 Core Functioning .......................................................... 21
      4.1.1.2 Auxiliary Functioning .................................................... 22
    4.2 AODV .................................................................................... 22
  4.3 DSR .......................................................................................... 23
    4.3.1 Route Discovery ................................................................. 24
    4.3.2 Route Maintenance ............................................................. 24
  4.4 TORA ....................................................................................... 25
    4.4.1 Route Creation ................................................................... 25
    4.4.2 Route Maintenance ............................................................. 25
    4.4.3 Route Erasing ................................................................... 26
    4.4.4 Route Optimization ............................................................ 26
  4.5 Comparison of routing protocols ................................................. 26
5 Experimental Design and Implementation ....................................... 28
  5.1 Evaluation Platform ................................................................... 28
  5.2 Performance Metrics ................................................................. 29
    5.2.1 Throughput ......................................................................... 29
    5.2.2 End-to-end delay ................................................................. 29
    5.2.3 Upload Response Time ....................................................... 30
    5.2.4 Download Response Time .................................................... 30
    5.2.5 Retransmission Attempts ...................................................... 30
  5.3 Network modeling ...................................................................... 30
  5.4 Network Configuration Parameter ............................................. 31
  5.5 Network Scenarios .................................................................... 32
6 Results and Analysis ..................................................................... 35
  6.1 Varying Network Size ............................................................... 35
6.1.1 Impact on Throughput ................................................................. 35
6.1.2 Impact on End-to-End Delay ....................................................... 38
6.1.3 Impact on Download Response Time .......................................... 40
6.1.4 Impact on Upload Response Time .............................................. 42
6.1.5 Impact on Retransmission Attempt .......................................... 44
6.2 Varying Node Mobility ................................................................. 46
   6.2.1 Impact on Throughput ........................................................... 46
   6.2.2 Impact on End-to-End Delay .................................................. 49
   6.2.3 Impact on Download Response Time ....................................... 52
   6.2.4 Impact on Upload Response Time .......................................... 53
   6.2.5 Impact on Retransmission Attempt ....................................... 54
6.3 Summary of Routing protocols Performance .................................. 57
   6.3.1 Performance evaluation with varying network density ................ 57
   6.3.2 Performance evaluation with varying mobility rate .................. 58
6.4 Summary of TCP Performance .................................................... 59
   6.4.1 Performance evaluation with varying network density ............... 60
   6.4.2 Performance evaluation with varying mobility rate .................. 61
7 CONCLUSION......................................................................................... 63
   7.1 Results ...................................................................................... 63
   7.2 Future Research ........................................................................ 64
Appendix A Model Configuration ......................................................... 65
Appendix B Network Configuration Parameters .................................... 67
References......................................................................................... 71
List of Figures

1.1: Flowchart for performance investigation ................................................................. 8
2.1: Slow Start and Congestion Avoidance Mechanism ..................................................... 12
3.1: Partition Impact: (a) before movement (b) after movement .................................... 17
3.2: Hidden Node Impact ................................................................................................. 19
3.3: Exposed Node Impact .............................................................................................. 19
4.1: Multipoint Relays of the OLSR network .................................................................. 21
4.2: Route Discovery mechanism for DSR ..................................................................... 24
6.1: Average throughput for different routing protocols; (a) Small network size, (b) Medium network size and (c) Large network size ................................................................. 36
6.2: Average end to end delay for different routing protocols; (a) Small network size, (b) Medium network size and (c) Large network size ................................................................. 39
6.3: Average download response time for different TCP variants; (a) Small network size, (b) Medium network size and (c) Large network size ................................................................. 41
6.4: Average upload response time for different TCP variants; (a) Small network size, (b) Medium network size and (c) Large network size ................................................................. 43
6.5: Average retransmission attempts for different TCP variants; (a) Small network size, (b) Medium network size and (c) Large network size ................................................................. 45
6.6: Average throughput for different node speeds 10 m/s, 20 m/s and 30 m/s; (a) AODV, (b) DSR, (c) OLSR and (d) TORA routing protocol ................................................................. 47
6.7: Average end to end delay for different node speeds 10 m/s, 20 m/s and 30 m/s; (a) AODV, (b) DSR, (c) OLSR and (d) TORA routing protocol ................................................................. 50
6.8: Average download response time for different TCP variants; (a) node speeds 10 m/s, (b) node speeds 20 m/s and (c) node speeds 30 m/s ................................................................. 52
6.9: Average upload response time for different TCP variants; (a) node mobility 10 m/s, (b) node mobility 20 m/s and (c) node mobility 30 m/s ................................................................. 54
6.10: Average retransmission attempts for different TCP variants; (a) node speeds 10 m/s, (b) node speeds 20 m/s and (c) node speeds 30 m/s ................................................................. 55
6.11: Performance of routing protocols for different network sizes; (a) In terms of throughput (b) In terms of end to end delay ................................................................. 57
6.12: Performance of routing protocol for different node speeds; (a) In terms of throughput (b) In terms of end to end delay ................................................................. 59
6.13: Performance of TCP variants for different network sizes; (a) In terms of download response time (b) In terms of upload response time (c) In terms of retransmission attempts ................................................................. 60
6.14: Performance of TCP variants for different node speeds; (a) In terms of download response time (b) In terms of upload response time (c) In terms of retransmission attempts. .......................... 62

A.1: Create empty scenario ............................................................................................................................................ 65

A.2: Create a MANET scenario with campus scale ........................................................................................................ 65

A.3: Specify the size and bounded area of a MANET network model .................................................................................. 65

A.4: An Example of 100 Nodes Simulated Network Model in MANET ................................................................. 66
List of Tables

4.1: Comparison of Routing Protocols ................................................................. 27
5.1: Description of the Experimental Scenarios for Network Scalability .................. 33
5.2: Description of the Experimental Scenarios for Node Mobility .......................... 34
B.1: General Parameters ................................................................................. 67
B.2: Wireless LAN Parameters ....................................................................... 67
B.3: TCP Parameter ....................................................................................... 68
B.4: HTTP Application Parameters ............................................................... 68
B.5: FTP Application Parameters ................................................................... 68
B.6: Profile Configuration ................................................................................ 68
B.7: Application Configuration ....................................................................... 69
B.8: AODV Parameters .................................................................................. 69
B.9: DSR Parameters ..................................................................................... 70
B.10: OLSR Parameters ............................................................................... 70
B.11: TORA Parameters ............................................................................... 70
List of Acronyms

AODV  Ad-hoc On Demand Distance Vector Routing
BER   Bit Error Rate
DAG   Directed Acyclic Graph
DSR   Dynamic Source Routing Protocol
FTP   File Transfer Protocol
GB    Gafni-Bertsekas
HTTP  Hypertext Transfer Protocol
IP    Internet Protocol
IMEP  Internet MANET Encapsulation Protocol
IETF  Internet Engineering Task Force
LAN   Local Area Network
LMR   Lightweight Mobile Routing
MANET Mobile ad-hoc Network
MAC   Media Access Control
MPR   Multipoint Relay
MID   Multiple Interface Declaration
NS    Network Simulator
OPNET Optimized Network Evaluation Tool
OMNET Operation and Maintenance New Equipment Training
OLSR  Optimized Link State Routing Protocol
RREQ  Route Request
RREP  Route Reply
RERR  Route Error
RTT   Round Trip Time
RTS   Request to Send
RTO   Retransmission Timeout
SACK  Selective Acknowledgment
TCP   Transmission Control Protocol
TC    Topology Control
TORA  Temporally Ordered Routing Algorithm
TTL   Time to Live
WG    Working Group
WRP   Wireless Routing Protocol
WLAN  Wireless Local Area Network
CHAPTER 1
INTRODUCTION AND METHODOLOGY

1.1 BACKGROUND

The use of wireless technology has become a ubiquitous method to access the Internet or making connection to the local network due to its easier and inexpensive deployment with a possibility of adding new devices to the network at no or lower cost. Devices equipped with wireless adapters together with a wireless access point constitute wireless local area networks (WLANs). Wireless access points, representing a fixed infrastructure, allow devices equipped with wireless adapters to be linked together in a local area network (LAN) and to get access to the Internet. However, the reliance upon an existing infrastructure and its potential limitations on mobility can be a major drawback [10]. Therefore, wireless-capable devices may operate as autonomous entities, communicating via multiple wireless hops without a pre-established fixed infrastructure. In the discussion that follows, such wireless-equipped devices are referred to as nodes and function as both clients and servers in the network to forward the data packets. Such a network is called a mobile ad-hoc network (MANET), where the nodes employed in the network can change their location from time to time. Nodes can also join or leave the network freely and arbitrarily without any restriction.

The idea of mobile ad-hoc networking is sometimes also known as infrastructure-less networking as it does not require any servers, routers, access-points or cables. Instead, a MANET is comprised of a set of autonomous mobile nodes where the nodes must work together in a distributed manner to enable routing among them. Because of the lack of centralized control and frequent changes of network topology, routing becomes a vital issue and a major challenge in these types of networks.

A routing protocol is mainly used to discover the shortest, most efficient and correct path(s) while providing the data transmissions between different wireless devices in ad-hoc network. In recent time, MANETs are found to be able to insert the routing functionality into the mobile nodes, which economize energy for other nodes by bringing down the routing overhead in the network. Moreover, this routing algorithm establishes the communications and formalizes agreement among nodes, which is essential to the overall performance of a MANET [54]. Routing protocols for ad-hoc networks have been of great interest for many years as the underlying Internet routing protocols are mainly intended to support the permanent infrastructure network; eventually, the properties of those protocols are found to be inappropriate for MANETs. Consequently, a variety of MANET routing protocols has evolved over recent time. Examples of such routing protocols are, among others [57], the Optimized Link State Routing Protocol (OLSR) [38], the Wireless Routing Protocol (WRP) [56], the Ad-hoc On Demand
Distance Vector Routing (AODV) [39], the Dynamic Source Routing Protocol (DSR) [41] and the Temporally Ordered Routing Algorithm (TORA) [48].

Apart from the above-mentioned network layer protocols, a transport layer protocol like Transmission Control Protocol (TCP) [46] is also needed to establish a reliable end-to-end connection in the network. TCP dominates the connection-oriented communications and ensures a reliable data transmission over the unreliable Internet Protocol (IP). Now-a-days, most of the internet traffic is carried out as well as the majority of widely used applications are provided by TCP. Applications like File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP) make use of TCP of TCP/IP suite for their operation. Hence, TCP is preferred to be implemented at the transport layer of an ad-hoc network which, eventually, facilitate in connecting to the Internet, thereby providing a large extent of applications. Hence, it is highly likely for the TCP to have a trust-worthy and stable performance in MANET environment.

Along with the reliability feature, the TCP facilitates in managing the flow and congestion control in the data communication mechanism. Due to the congestion problem, the network performance can go down by several orders of magnitude. As a consequence of that, the TCP executes four intertwined algorithms, which prevent senders from overwhelming the TCP receiver. The algorithms are defined as slow-start, congestion-avoidance, fast-retransmit and fast-recovery [3]. By implementing these mechanisms, the TCP can realize the throughput maximization so as to maintain a high performance of the network.

The congestion-control algorithms introduced in the TCP Tahoe version are: a) Slow start b) Congestion Avoidance and c) Fast Retransmission. Apart from these three mechanisms, TCP Reno, TCP New Reno and TCP Selective Acknowledgement (SACK) support fast recovery algorithm.

1.2 PROBLEM STATEMENT

Today, the TCP is responsible for providing reliable data transport in the Internet. As a matter of fact, it is extensively tuned to providing high-quality performance in the conventional wired network. However, it cannot offer reliable service while using e-mail, internet search, several application file transmission in a mobile Ad-hoc network.

Several studies reveal that TCP does not perform as well in a mobile environment as it does in other networks [23] [24]. There are several factors that affect the TCP performance in MANETs, such as dynamic topology, shared medium, signal fading and high bit errors [60]. For dynamic topology, the packet losses occur due to the broken routes between the nodes whereas TCP assumes that the losses are due to the network congestion. Therefore, the network experiences the counterproductive invocation of congestion control mechanisms employed by
the TCP. Additionally, the nodes experience hidden and exposed node-problems due to the share medium, thereby resulting in significant performance degradation in the network. Similarly, there are other types of constraints that have to be encountered when the TCP is analyzed in the MANET environment. Thus, the study of TCP performance along with the investigation of the main factors affecting the TCP performance in a MANET environment becomes an important area of research.

Since MANETs are gaining immense popularity day by day it is important to address the issue of constructing and developing an efficient MANET routing protocol. However, this is a formidable task as all routing protocols developed for MANET may not lead to adequate performance. Hence, it is now widely recognized that determining the specific routing protocols that can perform better in a given MANET scenario would be an important contribution to contemporary research.

In addition, due to the dynamic nature of MANETs, the routing mechanism experiences a number of problems by being more susceptible to errors than the wired networks. In particular, member nodes can be affected by churn leading to routes disappearing and re-appearing, which in turn leads to sudden packet losses and higher message delays in the network. Hence, the routing in MANET is becoming more complex compared to in a typical wired LAN or ad-hoc network. Similarly, there are other factors like network size, network load, bandwidth and signal strength that affect the performance of the MANET routing protocols. Therefore, a detailed analysis is required in order to gain an insight of these factors that determine the performance of the routing protocol. More specifically, it would be important to study how the different network parameters and protocols interact, and to what extent each of the individual factors affects the routing performance observed from the transport layer, i.e., the TCP.

1.3 MOTIVATION AND MAIN CONTRIBUTION

Since their inception within the past decade, MANETs have received significant attention in the world of computer research. MANETs are an evolving technology which offers a cost-effective and scalable method to connect wireless devices. Lately, this technology has become increasingly popular due to its potential application in many domains. For instance, such a network can be helpful in rescue operations where there is not sufficient time or resource to configure a wired network [4]. MANETs are also very useful in military operations where the units are moving around the battlefield in a random way and a central unit cannot be used for synchronization. [34].

Although MANET has been considered as a convincing candidate for better wireless services, research to enhancing its functionality is still in its infancy [7]. Currently, research has been undertaken with regard to the task of identifying more suitable routing protocols and TCP
variants for MANET environment. This dissertation has subjected three TCP variants as well as four routing protocols (of different categories) in order to assess their performance in a few realistic MANET scenarios, which will eventually help to better understand their comparative merits and suitability for deployment under different scenarios. Among several TCP variants, three types are considered important for our investigation, namely TCP Reno, TCP New Reno and TCP SACK. These three variants are reckoned as the most prominent transport layer mechanisms, which offer standard window-based congestion control algorithms. Apart from that these variants have readily available implementations in most of the network simulators. Again, for the routing protocols, we select two reactive routing protocols [5], such as AODV and DSR, one proactive routing protocol [1] such as OLSR, and one hybrid routing protocol [2] such as TORA. We choose these as our candidate protocols since they cover a range of design choices, including source routing, hop-by-hop routing, periodic advertisement, and on-demand route discovery. The choice of these four protocols is also motivated by the fact that they have been proposed in the IETF MANET Working Group and provide loop free operations and responsive routing information.

Even though many MANET routing protocols have been proposed in recent years, current literature reports only a limited amount of performance study between them. More specifically, no research activity has previously been attempted to contrast their performance in a realistic manner. This research therefore provides a realistic and quantitative performance analysis of several key routing protocols in the same framework within the MANETs. One of the other purposes of existing research is to make improvement to the overall TCP performance for MANET scenario. However, prior to making such improvement, it would be worthwhile to investigate as to what extent the TCP performance is degraded in MANET environment when subjected to different network stresses and topology changes. To the best of our knowledge, this study would be first of its kind, in undertaking experiment through analyzing the performance of three TCP variants (Reno, New Reno and SACK) and four routing algorithms (DSR, TORA, OLSR, AODV) in a MANET environment.

In this thesis, we begin by addressing the main challenges affecting the performance of TCP in MANET environment. Subsequently, the research investigates how well the mentioned TCP variants respond to various performance differentials, such as download response time, upload response time and retransmission attempt, aside from identifying the most suitable TCP version for a specific routing protocol in different network scenarios. Such analysis is important since it facilitates in determining the most suitable and robust TCP variant in a bid to optimizing the traffic goals in respective networks. The research also examines the routing performance with respect to TCP under a variety of network conditions. In order to evaluate such performance, end-to-end delay and throughput are considered as performance metrics. In our dissertation, a number of important system parameters such as network size and node mobility speed are taken into consideration. In order to evaluate such performance, end-to-end delay and throughput are considered as performance metrics. In our dissertation, a number of important system parameters
such as network size and node mobility speed are taken into consideration. The changes of such parameters are made (i.e. small, medium and large size network and low, medium and high node speed) to realize different realistic MANET scenarios as well as to gauge the extent of their impact on network and transport layer protocols performance.

During our experiment, discrete event simulation software, known as Optimized Network Evaluation Tool (OPNET) [49] with version 16, has been successfully used as it has proved as a well accredited simulation package that has been used by many researchers worldwide. Apart from that, the present modeler offers easy graphical interface, additionally by providing an extensive library of network components for designing simulation model more reliably and efficiently. These features makes it preferable to other simulation tools, such as Network Simulator (NS2) [18] and Operation and Maintenance New Equipment Training (OMNET) [58].

1.4 AIMS AND OBJECTIVES

Following the above background and problem statement, one of the major aims of the dissertation is to gain a thorough understanding of MANET routing protocols and TCP variants, followed by uncovering the pros and cons of the MANET existing routing protocols in terms of TCP performance. The study is also aimed at making use of computer simulation tools and discoursing different aspects of the network design, so as to explore the performance behavior with various problematic features such as scalability and mobility. The other aims included providing appropriate methodologies and guidelines that can be followed in future research of similar kind.

Following the mentioned guidelines, the major objectives of the study are summarized as follows:

- Apply both qualitative and quantitative research methods that will guide the study in proper direction.
- Set up a platform for performing the simulation in OPNET and becoming familiar with different tools of OPNET software.
- Employ the TCP congestion control algorithms during the implementation of the proposed existing TCP variants in a MANET simulation environment.
- Discuss different constraints that affect the TCP performance in wireless network and critically examine various approaches that are suggested in the literature for improving the TCP performance.
- Perform a simulation study of TCP’s behavior when many active flows compete for bandwidth over the same link.
- Simulate different routing protocols in different network scenarios against several performance metrics, and analyzing the results.
• Compare and analyze the protocols and the TCP versions based on their performance in the simulated environment.
• Draw conclusions by presenting and interpreting the outcomes.

1.5 RESEARCH QUESTIONS

A few major research questions are formulated as follows:

Q1. What MANET existing routing protocol(s) scale well (in terms of throughput and end to end delay) when the number of node is increased in the network?

Q2. What MANET existing routing protocol(s) scale well (in terms of throughput and end to end delay) when the mobility rate is increased in the network?

Q3. How well do current state-of-the-art transport protocols (specifically, three mentioned TCP variants) perform on MANET? Can they cope well with increasing the node numbers? Does their performance always degrade when the speed of the node movement is increased in a MANET?

Q4. Which existing TCP variant(s) ensures the best performance (in terms of upload and download response time and retransmission attempts) when the number of node is increased in a MANET?

Q5. Which existing TCP variant(s) can robustly perform (in terms of upload and download response time and retransmission attempts) when the mobility rate is increased in a MANET?

1.6 RESEARCH METHODOLOGY

This section presents the proposed methodology and provides a guideline for the assessment performance in a MANET environment. The key methodology consists of analyzing data that are collected through experimental measurement [4], analytical modeling [32], and computer simulation [30] in order to undertake network evaluation and performance.

1.6.1 Justification of research methodology

Experimental measurement is considered as an effective way of collecting quantitative data of real system and thereby obtaining fairly accurate results out of the research. Besides, as noted in [62], the simple levels of simulation abstraction do not present a solid base validation of
routing protocol behavior as compared to that of real experiment. However, it is a formidable
task to carry out the experimental measurement of MANET in real life scenario, mainly due to
high cost and complex nature of mobile ad-hoc networks, which eventually require numerous
efforts and resources to carry out the experiments and performance evaluations.

Analytical modeling, on the other hand, is based on mathematical computation and analysis,
which is used to forecast the performance of an emerging application. Analytical analysis is
usually found to be an idle method of formulating new routing protocols, apart from its inability
in reflecting the dynamics of data communication networks such as MANET [31].

Unlike the analytical approach, computer simulation can be conducted with fewer
assumptions, to behave as good as a real world system. As most practitioners and engineers
advocate, this methodology has been widely used as an effective method to tune, debug and
optimize the network infrastructures. With the wide variety of simulation software, flexibility is
highly influenced in the course of model development while hardware cost is minimized [15].

With regard to a dynamic ad-hoc network, computer simulation seems to be a prominent
solution over analytical modeling and experimental measurement. Page and Canova [33] stressed
that mastering complex simulation software requires pertinent expertise to achieve the validation
and verification results. As specified in [55], methods of validating and verifying a simulation
model include operational graphics, parameter validation, comparison, historical and event data
validation to compare with simulation findings in order to prove its feasibility.

In addition, simulation methodology performs significantly well in eliminating the
complexity issues of routing protocols, apart from providing more flexibility in model
development, validation and performance evaluation. Accordingly, a methodology that realizes
the computer simulation has been adopted to carry out our proposed research. As a matter of fact,
the impact of routing protocols on MANET performance has been examined using OPNET
Modeler in order to address the formulated research questions.

1.6.2 Investigation framework

This section outlines step-by-step activities needed to accomplish the performance evaluation
as shown in Figure 1.1. At the outset, the key factors influencing MANET performance (Step 1)
are identified by taking help from exiting research and knowledge. Step 2 is related to
determination of the protocols utilized at transport layer as well as at application layer while Step
3 specifies the justification of performance thresholds. From Figure 1.1, it can be observed that
the first three steps are closely inter-related, which, at a preliminary stage, can also be performed
through realizing the state-of-art technology. Following this, a detailed survey of the existing
literature related to the current area of research is conducted. This is expected to also help
ascertain the data requirement for the study; the required data were then collected for the assessment.

Upon completion of literature review, certain experiments and simulations are performed in order to produce the type of statistical data that has to be analyzed. Consequently, a recursive process like network modeling should be in place in the next step (Step 4), which requires careful “deduction and validation” [54]. For the purpose of the present study, the MANET network models are designed on the workspace of the OPNET simulator with the help of different network entities, where different network entities are configured carefully to support the offered application services and subsequently to control the mobile nodes in the network. Step 5 focuses on performance evaluation, in which multiple experiments are deployed with different routing protocols and TCP variants to investigate the network performance.

By varying network-size and node-speed independently, quantitative data such as throughput, end-to-end delay, upload response time, download response time and retransmission attempts are collected for analysis, as shown in Step 6. Finally, the step 7 includes validation of our experiment; in this context, the simulation results are shown in different statistical plots and tables, as delineated in the simulation section in Chapter 6.

![Flowchart for performance investigation](image)

**Figure 1.1:** Flowchart for performance investigation
1.7 RELATED WORK

As previously mentioned, there are a number of routing protocols developed for Mobile Ad-hoc Networks, the popular ones being OLSR, AODV, DSR and TORA. In fact, these protocols do not have similar properties; on the other hand, their behaviors differ from one network environment to another. Hence, it becomes necessary to simulate these protocols in an ideal environment to examine how they perform in a particular network. A good number of research has been carried out on individual protocols; however, not much research has been carried out on aspects relating to any comparative analysis of these routing protocols with respect to TCP.

Recent research in this context can be found in [2, 3, 9, 11 and 12], which is based on the comparison of MANET routing protocols, although the simulation parameters used by the authors are substantially different from the parameters used in our simulations. As a matter of fact, there has not been any inclusive evaluation study conducted to compare the routing performance with proactive, reactive and hybrid categories of routing protocols. On top of that, we considered total simulation time as 600 seconds over which the performance statistics are collected in our current research.

The routing performance among AODV, OLSR and TORA are investigated with fixed number of nodes and with lower network congestion in [16] and according to the authors, OLSR is argued to be the most favorite proactive protocol while AODV has been defined as the most effective on-demand protocol for MANET scenarios. Similarly, DSR and TORA routing protocols are compared in [44] where DSR has been chosen as a preferable protocol to TORA.

Besides, the research in [6] involves an analysis among DSDV, TORA SPF and EXBF with different network size that investigated the scalability of the protocols. However, the study does not employ the congestion control mechanisms. Aside from that, the study utilizes low load traffic in the network.

Our study presents the scalability of the protocols by employing heavy congestion with high load traffic for both FTP and HTTP. The performance of AODV and TORA is found to be similar in most of the cases with works, such as [8] and [42].
1.8 THESIS STRUCTURE

The thesis has been structured as follows:

Chapter 1 presents the introduction and background to the research.

Chapter 2 covers the preliminaries and basic concepts of different congestion avoidance algorithms employed by TCP, followed by a description of different TCP variants.

Chapter 3 deals with identifying constraints affecting the TCP performance in a MANET environment.

Chapter 4 describes the existing protocols for routing in MANETs. A comparison among different routing protocols is also presented at the end of this chapter.

Chapter 5 includes a discussion on aspects relating to the performance metrics used to analyze the performance of the routing protocol and the TCP variants. The experimental setups, network scenarios and the parameters required to configure them are presented in some detail in this chapter.

Chapter 6 presents a discussion on the results obtained upon running the experiments, as delineated in Chapter 5. An attempt is also made to analyze the significance of the results in this chapter.

Chapter 7 draws conclusions, built on the analysis, along with exploring avenues for future research.
CHAPTER 2
TRANSPORT CONTROL PROTOCOL (TCP)

This chapter is divided into two sections. Section 2.1 outlines the congestion control algorithms used by the TCP while a detailed description of different TCP variants is presented in section 2.2.

2.1 Congestion Control Algorithms

TCP is known as a full duplex protocol meaning each TCP connection provides a pair of byte streams in both directions. TCP implements the congestion control mechanism with each of these byte streams so that the receiver can limit the sender from transmitting more data in the network [47].

This section discusses about four intertwined congestion control mechanisms: slow start, congestion avoidance, fast retransmit and fast recovery. A TCP must not be more aggressive in sending data than these four algorithms allow.

2.1.1 Slow start and Congestion Avoidance

The TCP sender employs the slow start and congestion avoidance algorithms to avoid more data to be sent in the network than it is capable of. For implementing these algorithms, two flow control variables, namely, the congestion window and the advertised window are included in each TCP connection state. The TCP sender imposes the congestion window while the receiver imposes the advertised window. The minimum of the congestion window and the advertised window regulates the data transmission. Besides, The slow start threshold (ssthresh), known as a state variable, is used to decide which one is to be used among the slow start or congestion avoidance algorithms for controlling the data transmission. During the beginning of the transmission, there are many unfamiliar conditions present in the network; therefore TCP needs to gradually discover the network by assessing the bandwidth and determining the available capacity [35]. This will eventually prevent the network from being congested with large bursts of data.

Figure 2.1 shows the slow start and congestion avoidance mechanisms executed by the TCP. Upon establishing a new connection, TCP starts the slow start mechanisms and sets the congestion window size to one segment. The congestion window size is incremented by one for each ACK received by the TCP sender. Thus, 1 packet is sent in the first round trip time (RTT), 2 packets are for the second RTT, 4 packets are for the third RTT and continue incrementing
exponentially. This is why slow start phase is also known as the exponential growth phase where slow start increases the window size by the number of segments acknowledged. This process will be continuing until either of the following situations occurs: 1) an acknowledgment is not received for some segments 2) a predetermined slow start threshold value is reached 3) the congestion window size becomes equal to the receiver’s advertised window size.

If either of these events takes place, TCP enters the congestion avoidance (linear growth) phase. Each time an ACK is received, congestion avoidance suggests that the congestion window size should be increased by \((\text{segment size} \times \text{segment size}) / \text{congestion window}\) [46]. Here, segment size and congestion window is maintained in bytes.

![Figure 2.1: Slow Start and Congestion Avoidance Mechanism][53]

### 2.1.2 Fast Retransmission and Fast recovery

Whenever a packet segment is transmitted, TCP sets a timer each time and thus ensures the reliability. TCP retransmits the packet, if it does not obtain any acknowledgement within the fixed time-out interval. The reason for not getting any acknowledgement is due to a packet loss or the network congestion. Therefore the TCP sender implements the fast retransmit algorithm for identifying and repairing the loss. This fast retransmit phase is applied mainly based on the incoming duplicate ACKs. As TCP is not able to understand whether a packet loss or an out-of-order segment causes the generation of the duplicate ACK, it waits for more duplicate ACKs to be received [14]. Because in case of out-of-order segment, one or two duplicate ACKs will be received before the reordered segment is processed. On the other hand, if there are at least three duplicate ACKs in a row, it can be assumed that a segment has been lost. In that case, the sender will retransmit the missing data packets without waiting for a retransmission timer to expire.
After the missing segment is retransmitted, the TCP will initiate the fast recovery mechanism until a non-duplicate ACK arrives. The fast recovery algorithm is an improvement of congestion control mechanism that ensures higher throughput even during moderate congestion [35]. The receiver yields the duplicate ACK only when another segment is reached to it; therefore this segment is kept in the receiver's buffer and does not consume any network resources. This means, data flow is still running in the network, and TCP is reluctant to reduce the flow immediately by moving into the slow start phase. Thus, in fast recovery algorithm, congestion avoidance phase is again invoked instead of slow start phase as soon as the fast retransmission mechanism is completed.

2.2 TCP VARIANTS

The original design of the Transmission Control Protocol (TCP) worked reliably, but was unable to provide acceptable performance in a large and congested network. The development of the TCP has therefore been made progressively since its original incarnation in 1988. This section presents several TCP versions which have been proposed with different mechanism in order to control and avoid the network congestion.

2.2.1 TCP Tahoe

The earlier versions of TCP offered a go-back-n model which used to implement the cumulative positive acknowledgment [15]. For this purpose, retransmit timer expiration was needed in order to re-transmit the lost data. However, these former versions were unable to reduce the network congestion. Hence, for modification to earlier TCP implementations, the TCP Tahoe variant was implemented with slow-start, congestion avoidance, and fast retransmits algorithms [13]. This version modified the round-trip time (RTT) estimator which is needed for adjusting the values of retransmission timeout (RTO). In Tahoe version, when the sender accepts three duplicate acknowledgments for a single TCP segment, it assumes that a data packet is lost and hence resends the data packet regardless of the expiration of the retransmission time.

However, to identify a packet loss, the TCP Tahoe version needs a complete timeout interval or even longer sometimes due to the coarse grain timeout. In addition, upon detection of a packet loss, every time it waits until the pipeline is emptied which eventually establish a high cost in the band-width delay product links.

2.2.2 TCP Reno

Along with the implementation of the basic principles of Tahoe, the TCP Reno version adds more mechanisms so as to detect the lost packets in shorter time and also prevent the pipeline
from being empty every time a packet is lost. The packet segment is assumed to be lost as soon as the duplicate acknowledgements are reached to its threshold level. Then the TCP enters the Fast Re-transmit phase through which the lost segment is retransmitted. When the Fast Retransmit phase is completed, TCP Reno employs the Fast Recovery algorithm which does not let the pipeline to be empty and also provides extra incoming duplicate ACKs to clock subsequent outgoing packets. Moreover, Fast Recovery assumes whenever a duplicate ACK is attained, each time there is a single packet left in the pipe. As a result, the TCP Reno sender is capable of making sharp estimation over the amount of outstanding data in the network. Meanwhile, after entering the Fast Recovery phase, the TCP sender waits until half a window of dup ACKs are achieved, and then transmits a new data packet for each additional dup ACK [19]. Finally, the sender leaves the Fast Recovery phase when it receives a new ACK for the new data.

The variant TCP Reno can smoothly detect the single packet drop; however this version experiences difficulty in case of multiple packets dropped from the window and the performance becomes almost as like as Tahoe version. When multiple packets are dropped, the loss information of the initial packet is arrived after the reception of the duplicate ACK. On the other hand, the information about the second packet is obtained after the acknowledgement of the retransmitted initial packet is reached to the sender. Furthermore, this ACK of the retransmitted initial packet is arrived after one RTT and hence it takes longer time to process the second packet loss.

2.2.3 TCP New Reno

In case of multiple packet loss, the TCP New-Reno does not wait for the retransmission timer to be expired and hence this variant provides a dominating performance over the Reno version. In New Reno, the performance concerns about the behavior of the partial ACKs, which do not take TCP out of Fast Recovery phase while it takes TCP out from the Fast Recovery phase in Reno version [25]. Moreover, in New-Reno, receiving partial ACKs often indicates the loss of the packets which instantly follows the acknowledged packet in the sequence space. Thus for the multiple packet loss, the New-Reno becomes able to retransmit all the packets lost from a particular window. The New-Reno does not leave the Fast Recovery phase unless the acknowledgement for all outstanding data in the network is completed.

However, New-Reno may experience poor performance as it takes one RTT for identifying the packet loss and therefore it is possible to infer about the information of other lost packet only when the ACK for the first retransmitted segment is received [24].
2.2.4 TCP SACK

TCP uses a cumulative acknowledgment scheme through which only a single lost segment can be detected per round trip time. Moreover, this scheme does not allow the received packets that are not at the left edge of the receiver window to be acknowledged. Hence in order to discover the lost packet, the sender has to either wait for a roundtrip time or retransmit the received packet unnecessarily. Consequently, TCP loses its ACK-based clock and thus decreases the overall throughput.

In order to overcome these limitations, A Selective Acknowledgment (SACK) mechanism, combined with a selective repeat retransmission policy is arrived. TCP SACK is basically an upgraded version of TCP New Reno which takes steps to solve the major problems experienced by the New Reno version. Such problems include the detection of multiple lost packets and retransmission of more than one lost packet per RTT [27]. With selective acknowledgments, the information about the arrived data segments can be reached successfully to the sender. As a result the sender only needs to retransmit the actual lost packet. The TCP SACK offers a significant feature so that the segments are acknowledged selectively instead of being acknowledged cumulatively. In addition, there is a block present in each ACK which monitors the acknowledgments and reports the sender of which segments have been acknowledged. For increasing and decreasing the congestion window size, the congestion control algorithms of SACK version are found almost same as Reno. The TCP SACK retains the basic properties and services of Tahoe and Reno, for instance, ensures high robustness even in the existence of the out-of-order packets. However, when multiple packets are lost from the data window, the properties between SACK and other variants can be differentiated.

In the Fast Recovery stage of SACK version, a variable is maintained by the sender in order to measure the number of outstanding data in the network. This variable is called a pipe and it is not maintained in any of the earlier TCP versions. As long as the estimated number of outstanding packets is found below than the congestion window value, a data is transmitted or retransmitted by the sender [37]. Moreover, when the sender sends a new data or retransmits an old packet, the variable pipe is incremented by one while it is decremented by the same value upon receiving a duplicate ACK with a selective acknowledgment option.

Though TCP SACK provides many advantages, it is not an easy task to implement selective acknowledgment options in TCP SACK version. Hence, currently the TCP receivers are found to be reluctant for providing the selective acknowledgment option.
CHAPTER 3
TCP PERFORMANCE IN MANETS

Even though TCP ensures reliable end-to-end message transmission over wired networks, a number of existing researches have showed that TCP performance can be substantially degraded in mobile ad-hoc network [3, 45]. Along with the traditional difficulties of wireless environment, the mobile ad-hoc network includes further challenges to TCP. In particular, challenges like route failures and network partitioning are to be taken into consideration. Furthermore, MANET experiences several types of delays and losses which may not be related to congestions, though TCP considers these losses as a congestion signal. These non-congestion losses or delays mostly occur due to the inability of TCP’s adaptation to such mobile network. Appropriate cares have to be taken for assessing such losses and also to distinguish them from congestion losses so that TCP can be sensitive while invoking the congestion control mechanism.

The next subsections present an analysis of different types of constraints influencing the TCP performance in MANET environment.

3.1 High BER

High bit error rate is caused due to multipath fading, Doppler shift and signal attenuation. This causes TCP data segments to be lost and thereby the congestion control mechanisms are triggered unnecessarily by the TCP sender.

3.2 Route Failures

In MANET, the mobility of the node is considered as the major reason for the route failure and the route reestablishment is needed instantly in case of route failure. However, it is likely that a new route establishment may experience longer duration than the RTO of the sender. In consequence of that, the TCP sender will unnecessary deploy congestion control mechanism.

3.3 Path Asymmetry Impact

The network topology is changed very frequently and arbitrarily within MANETs, which leads to the creation of an asymmetric path. This path formation negatively influences the TCP performance since TCP is highly dependent on time responsive feedback information. The sender starts transmitting data in a burst when a number of ACKs are gathered together, which
causes the packet to be lost. In MANETs, path asymmetry can be grouped into different forms such as loss rate asymmetry, bandwidth asymmetry and route asymmetry.

3.4 Network Partitioning

A network partition takes place when a node departs from the network, resulting in an isolation of some parts of a mobile ad-hoc network. These fragmented portions are defined as partitions. In a MANET environment, TCP considers network partitioning as one of the most imperative challenges which is mainly caused due to mobility or energy-constrained operation of nodes. When the source and the destination of a TCP connection lie in different parts of the network, all transmitting packets are found to be dropped by the network. As a result, the congestion control algorithm will be invoked instantly by the TCP sender [23].

Again, the serial timeouts at the TCP sender can be generated in case of frequent disconnections in the network. This may trigger a longer idle period for the network through which the connection can be re-established. However, the TCP does not found to move from the back off state. An ideal example is illustrated in Figure 3.1:

![Figure 3.1: Partition Impact (a) before movement (b) after movement](image-url)
In Figure 3.1a, when node 5 goes away from node 4, this causes a fragmentation of the network into two parts. The network fragmentation is depicted in Figure 3.1b. If the disconnection continues for longer period than the RTO, the exponential back-off algorithm will be activated automatically by the TCP [13]. This mechanism doubles the RTO value as soon as the timeout expires and continues doubling the RTO until the maximum timeout value of 64 sec is reached [24].

3.5 Power Scarcity

Each mobile node carries batteries which have limited power supply; hence the network suffers the node lifetime problem. Each node in MANET works as a router and an end system, therefore needless retransmissions of the packet cause the consumption of this limited energy resource. As a result, an inefficiency of the available power is utilized.

3.6 Multipath routing

In order to reduce the frequency of route re-computation, some routing protocols preserve multiple routes between the sender and the receiver. However, this may result in the arrival of a huge number of out-of-sequence packets to the receiver. Consequently, it causes the receiver to generate duplicate ACKs and the sender to employ the congestion control mechanisms [26].

3.7 Interaction between MAC protocol and TCP

In a MANET environment, the intercommunication between the TCP mechanisms and 802.11 MAC protocol may lead to unexpected severe challenges such as link capture effect, instability, and one-hop unfairness. The causes of these problems include the hidden station and exposed station problems of the 802.11 MAC protocols [28].

3.8 Hidden and Exposed Node Impact

Figure 3.2 presents a typical hidden node condition where packet transmission starts from node A to node E. Since, node B cannot sense node D, node B assumes the channel as an idle channel and therefore initiates its transmission by dispatching a Request to Send (RTS) to node C. However, transmitting RTS unexpectedly introduces collisions because node C is found in the interference range of node D. This problem is termed as "Hidden Node" impact where node D is called the hidden node with respect to node B.
Figure 3.2: Hidden Node Impact

Figure 3.3 depicts a condition through which the exposed node problem can be realized. When node D intends to transmitting data toward node E, node C will not be able to send any data frame to node B. Node C must wait until node D finishes its current transmission to node E. This is because node D is within the sensing range of node C. This problem is known as "Exposed Node" impact where node D is called the exposed node with respect to node C.

Figure 3.3: Exposed Node Impact
CHAPTER 4
THE ROUTING PROTOCOLS

In the latest years, research has been conducted on improving the performance of the MANET routing protocols. To deal with the complexity of the routing protocols, MANET has become a vital issue for The Internet Engineering Task Force (IETF) and therefore a MANET working group (WG) is established by IETF. The role of this group is to be involved in the development of different routing protocols such as OLSR, DSR, AODV, TORA and so on.

These protocols are categorized into two groups as Reactive and Proactive based on the updated time of the routing information. In addition, the WG also offers a converged approach, for instance, a hybrid routing protocol. There are two other classes of routing protocol present based on the content of the routing table which are defined as distance vector class and link state class [38] [40]. The distance vector protocols disseminate the distance lists to the destination while the link state protocols involves in maintaining the network topology. Generally, the link state protocols exhibits more stability and robustness than the distance vector protocols though they are found much more complex to use in MANETs.

This chapter continues with a description of different MANET routing protocols and presents a comparison among them.

4.1 OLSR

The Optimized Link State Routing (OLSR) is operated as a proactive (table-driven) routing protocol i.e. frequently exchanges topology information with other nodes of the network [22]. This protocol is basically an optimization of traditional link state protocol developed for mobile ad-hoc network. The responsibilities of OLSR protocol are to minimize the required number of control packets transmission and also to shorten the size of control packets. On top of that, OLSR trims down the control traffic overhead in the network with the help of Multipoint Relays (MPRs) [51]. The MPR concept is the key idea behind OLSR protocol which is basically a node's one-hop neighbors in the network as shown in Figure 4.1. For route calculation, the MPR technique is employed in order to form the route between the source and the destination in the network. In addition, the MPRs yield an efficient mechanism for flooding control traffic by significantly minimizing the number of packet transmissions. Yet, the MPRs are to be involved in another task during the time the link state information is declared in the network. The task includes declaring the link-state information for their MPR selectors and hence providing the shortest paths to all destinations. In MANET, the MPRs are assigned from the one-hop adjacent nodes with "symmetric" (bi-directional) linkages. Thus, by determining the path through the
multipoint relays, it is possible to keep away the difficulties experienced during the packet transmission over a uni-directional link.

![Multipoint Relays of the OLSR network.](image)

**Figure 4.1**: Multipoint Relays of the OLSR network.

OLSR employs three different types of control messages [22], namely 1) HELLO, 2) Topology Control (TC), and 3) Multiple Interface Declaration (MID). OLSR minimizes the maximum time interval while periodically transmitting these control messages and thus preserves the routes incessantly to all destinations in MANETs. This feature eventually makes the OLSR protocol to be more favorable for large and dense networks. In terms of OLSR protocol, the larger and denser a network, the more optimization can be obtained as compared to the pure link state algorithm [38]. OLSR is independent of the central entities and designed to operate in such a way where a complete distribution algorithm can be achieved.

### 4.1.1 OLSR Protocol Functioning

OLSR is categorized into core functionality and a set of auxiliary functionalities [50]. The core functionality specifies a protocol which can provide routing in a stand-alone MANET whereas each auxiliary function offers further functionality, which can be implemented in particular scenarios, i.e. a scenario where a node establishes connectivity between the MANET and another routing domain.

#### 4.1.1.1 Core Functioning

The core functionality describes about the OLSR interfaces and the mobile nodes present in the MANET. Moreover, this comprises a universal specification of OLSR protocol messages and
their transmission, topology diffusion, route calculation and link sensing. Generally, the core function includes the following elements:

- Packet Format and Forwarding
- Link Sensing
- Neighbor detection
- MPR Selection and MPR Signaling
- Topology Control Message Diffusion
- Route Calculation

### 4.1.1.2 Auxiliary Functioning

Apart from the core functioning, it is also required to have additional functionalities for OLSR in some situations. These situations include scenarios where a node consists of several interfaces and a few of them participate in another routing domain. Likewise, the situation could be a scenario, where it is required to offer additional topological information to the network.

### 4.2 AODV

The Ad-hoc On Demand Distance Vector (AODV) is considered an efficient MANET routing protocol and supports both unicast and multicast routing mechanisms. The AODV routing protocol utilizes an on-demand technique in order to discover the routes. This means that the route between two endpoints (nodes) is formed as per requirement for the source node and maintained as long as the routes are needed. Moreover, the protocol uses a destination sequence number to recognize the most recent path and to guarantee the freshness of the routes. Reactive protocols like AODV shrinks the control traffic overhead at the cost of higher latency in discovering new routes [39]. Although AODV is a reactive protocol, some characteristics of a proactive protocol are often followed by this. For instance, the protocol broadcasts the periodic HELLO messages to notify the neighbor nodes that the link is still active. AODV does not have any function until there is a valid route between the source and the destination in MANET. Upon requiring the formation of a new route, the source node transmits a Route Request (RREQ) packet.

After flooding the RREQ packet, the source node waits until a Route Reply (RREP) packet is received as an acknowledgement. However, within a specific time, a RREP may not be received and in that case a new RREQ is to be sent again by the source node. And for this additional transmission of RREQ, the predefined waiting interval needs to provide a binary exponential back-off and therefore it is multiplied by two (2) each time. The binary exponential back-off must be utilized in order to reduce the network congestion. After receiving a RREQ, the
neighbor node either generates a RREP message to the sender or rebroadcasts the RREQ depending on the availability of a valid route to the destination. The validity of the route is confirmed after making a comparison between the sequence number of the intermediate node and the destination sequence number of the Route Request packet. Once the RREP is received by the source node, it stores the information of this particular route and starts transmitting data toward that destination. However, in case of the reception of the multiple RREPs, the route with the shortest hop count will be selected.

In case a link failure is experienced, a Route Error (RERR) message is created and returned to the originator of the data in a hop-by-hop fashion and the process replicates. The purpose of generating the RERR message is to inform other nodes about the current broken link. The source node disables the route as soon as it receives the Route Error message and invokes the route discovery mechanism again if it is necessary.

4.3 DSR

Dynamic Source Routing (DSR) is a widely used reactive (on-demand) routing protocol which is designed particularly for the mobile ad-hoc networks. DSR permits the network to run without any existing network infrastructure and thus the network becomes as a self-organized and self-configured network. This protocol maintains an on-demand approach and hence extinguishes the periodic table-update messages needed in the table-driven approach [41]. Consequently, it is able to prevent the control packets from consuming much bandwidth. Like other on-demand routing protocols, DSR does not provide the transmission of any periodic hello packet (beacon), which is essential for informing its presence to other nodes. Instead, during the route construction phase, it establishes the route by flooding a Route Request packet in the network. Each Route Request packet holds a sequence number which is generated by all the nodes the packet is flooded through. By using this sequence number, loop formation and multiple transmission of the same Route Request is possible to be evaded. When a Route Request packet is reached to its final destination, the destination node sends a Route Reply packet to the source node through the opposite way the Route Request is travelled. Since, it cannot be an efficient mechanism for the nodes to provide continuous flooding; DSR utilizes the route caches to store the routing information [9].

In MANETs, the DSR protocol generates two mechanisms namely Route Discovery and Route Maintenance for the purpose of discovering and maintaining the route between the endpoints. Both mechanisms are utilized to support the unidirectional (asymmetric routes) links in wireless ad-hoc network.
4.3.1 Route Discovery

Figure 4.2 shows that to commence the Route Discovery mechanism, node M floods a Route Request to all nodes which are in the wireless transmission range of M. In the network, the initiator (source node) and target (destination node) of the Route Discovery is identified by each Route Request packet. The source node also provides a unique request identification number in its Route Request packet and in Figure 4.2 this is given as ID= 3.

![Diagram of Route Discovery mechanism for DSR]

For responding to the Route Request, the target node Q generally scans its own Route Cache for a route before sending the Route Reply toward the initiator node M. However, if no suitable route is found, node Q will execute its own Route Discovery mechanism in order to reach toward the initiator.

4.3.2 Route Maintenance

The Route Maintenance mechanism is used when the source node is unable to use its current route to the destination due to changes in the network topology. In such case, the source has to use any other route to the destination. However, it may invoke the Route Discovery mechanism again to discover a new route. Each node while using a source route has to ensure that data can be transmitted properly from that particular node to the subsequent nodes. Consequently, an acknowledgement is made for confirming that a link is able to transmit the data. In wireless networks, acknowledgements are often provided either as an existing standard part of the MAC protocol in use (such as the link-layer acknowledgement frame defined by IEEE 802.11), or by a "passive acknowledgement" [17]. However, a specific software acknowledgement can be implemented by the DSR itself in case of no built-in acknowledgement present in the system.
4.4 TORA

The Temporally-Ordered Routing Algorithm (TORA) is a highly efficient distributed routing protocol and known as a hybrid protocol which can simultaneously support both table-driven and on-demand approach in multi-hop wireless networks. This protocol belongs to the family of the link reversal routing mechanism based on the Gafni-Bertsekas (GB) and The Lightweight Mobile Routing (LMR) algorithms [48]. The TORA protocol’s reaction to link failure is structured as a temporally-ordered sequence of diffusing computations, where all computations comprising of a sequence of directed link reversals.

TORA is implemented on the top of the Internet MANET Encapsulation Protocol (IMEP) and ensures link status sensing, reliability, loop-free routes, multiple routes and many others essential services. TORA implements four mechanisms in the network, which are known as creating routes, maintaining routes, erasing routes, and optimizing routes [20].

4.4.1 Route Creation

The route formation toward the destination needs the establishment of a sequence of directed links if there are undirected links present between the source and the destination. Therefore, in a network, formation of routes significantly involves the implementation of direction assignment mechanism to all undirected links. A direction to the link can be upstream or downstream and is assigned by two routers based on the relative values of the router’s height. The heights of the routers and the link directional assignments conjointly establish a multipath, loop-free routing system where all the paths are directed downstream to the destination. The route is built with the help of the query (QRY) and update (UPD) packets. These packets are used to form a directed acyclic graph (DAG) which is often termed as a “destination oriented” DAG [44].

4.4.2 Route Maintenance

In the network, the DAG is broken due to node mobility and thereby the route maintenance unit becomes essential for adapting the routing structure. For instance, some directed links may be broken down when any loss occurred in a router’s last downstream link. To cope with this situation, the link reversal algorithm is implemented in order to re-establish the valid routes quickly. There are two types of link reversal algorithms, a partial reversal and a full reversal approach. For maintaining the routes, TORA applies an approach which is similar to the partial reversals method [50].
4.4.3 Route Erasing

Though the GB algorithms are very useful in connected networks, they exhibit instability in case a network is partitioned from the destination. The network will experience an inefficient use of the available bandwidth until it becomes connected. As a consequence of that, TORA implements an algorithm, which is capable of detecting the network partitions and erasing the invalid routes. Upon detection of the network partition, the route erasure phase is invoked and the clear packet (CLR) is broadcasted throughout the network in order to wipe out the undirected invalid routes.

4.4.4 Route Optimization

Finally, TORA incorporates a method for optimizing routes, where the routers re-select their heights for meliorating the routing structure. For route optimization, TORA employs a different control message called Optimization (OPT).

4.5 Comparison of routing protocols

Table 4.1 [44] depicts the differences between four MANET routing protocols. The parameters used for the comparison, are route computation, routing updates, loop freedom, advantages, disadvantages etc.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>OLSR</th>
<th>DSR</th>
<th>AODV</th>
<th>TORA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routing mechanism</td>
<td>Table-driven</td>
<td>On-demand</td>
<td>On-demand</td>
<td>On demand or Table-driven</td>
</tr>
<tr>
<td>Multiple route mechanism</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Source routing mechanism</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Structure of the routing mechanisms</td>
<td>Flat</td>
<td>Flat</td>
<td>Flat</td>
<td>Flat</td>
</tr>
<tr>
<td>Network information maintainance</td>
<td>Route table</td>
<td>Route cache</td>
<td>Route table</td>
<td>Route table</td>
</tr>
<tr>
<td>Routing method</td>
<td>Flooding</td>
<td>Broadcast</td>
<td>Broadcast or Flooding</td>
<td>Broadcast</td>
</tr>
<tr>
<td>Update of routing information</td>
<td>Periodically</td>
<td>As required</td>
<td>As required</td>
<td>As required</td>
</tr>
<tr>
<td>Multicasting possibilities</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>----------------------------</td>
<td>----</td>
<td>----</td>
<td>-----</td>
<td>----</td>
</tr>
<tr>
<td>Depth of information</td>
<td>The whole topology</td>
<td>Path information towards the destination node</td>
<td>Up to neighbor nodes</td>
<td>The height of the neighbor nodes</td>
</tr>
<tr>
<td>Control messages</td>
<td>Hello message, Topology Control and Multiple Interface Declaration</td>
<td>No beacon or hello message</td>
<td>Only hello messages used for neighbor detection</td>
<td>LMR messages</td>
</tr>
<tr>
<td>Loop free Routing</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Drawbacks</td>
<td>The MPR sets could be overlapped</td>
<td>Large end-to-end delays, scalability problems caused by flooding and source routing mechanisms.</td>
<td>Scalability and large delay problem.</td>
<td>Temporary routing loops results in larger delays in the network.</td>
</tr>
<tr>
<td>Advantages</td>
<td>Trim down the number of broadcasts.</td>
<td>Provide multiple routes and avoid loop formation.</td>
<td>Much more efficient to dynamic topologies.</td>
<td>Multiple, loop free, reliable routes.</td>
</tr>
</tbody>
</table>

**Table 4.1:** Comparison of routing protocols
CHAPTER 5
EXPERIMENTAL DESIGN AND IMPLEMENTATION

This chapter describes how the study is carried out. More specifically, it deals with the analytical framework, including the methodological issues, such as evaluation procedure, methods of assessments, scenarios and parameters, implied limitations and scope of the study.

The chapter begins with a brief outline of the OPNET simulator, followed by an overview of the performance metrics upon which the evaluation of different routing protocols and TCP variants are established. More specifically, the chapter comprises a description of the designed network model and the necessary parameters, which are realized in configuring the network model. Finally, various simulation scenarios with different network conditions are presented at the end of this chapter.

5.1 Evaluation Platform

The design of an efficient network model and its performance evaluation is of immense importance in a real-world network scenario. However, it is a challenging task to evaluate the performance of the proposed network in a real situation. Therefore, a number of network simulators have been introduced in order to design and simulate the network models in several perspectives; for instance, NS-2 and OPNET are the two very well-known simulators. NS-2 is open source software while OPNET is a commercial simulator and the kernel source code of OPNET modeler is not open for all. However, OPNET has a comprehensive built-in development environment to design and simulate network models [36].

As explained in the preceding chapter, the research is conducted using discrete event simulation software known as OPNET Modeler, which is just one of several tools provided from the OPNET Technologies suite. In order to undertake the experimental evaluation, the most recently available version, namely the OPNET Modeler 16 has been adopted in our study. The OPNET is one of the most extensively used commercial simulators based on Microsoft Windows platform, which incorporates most of the MANET routing parameters compared to other commercial simulators available [49]. Aside from this, the modeler incorporates a number of features to support an increase in stability and mobility in the mobile ad-hoc network.
5.2 Performance Metrics

In OPNET simulator, a number of parameters are present for MANET environment in order to study the overall network performance. These parameters are known as performance metrics. Specific application and transport layer protocols demand on its own set of performance metrics to evaluate the network efficiency. For instance, with the introduction of a variety of network parameters, end-to-end delay and average throughput are substantially affected by the routing algorithm; hence, such parameters play an important role in the selection of an efficient routing protocol in any communication network. Similarly, the performance of different TCP variants appears to be sensitive to upload response time, download response time and retransmission attempts. Further elaboration of performance metrics used in this dissertation can be described as follows:

5.2.1 Throughput

The average rate at which the data packet is delivered successfully from one node to another over a communication network is known as throughput. The throughput is usually measured in bits per second (bits/sec). A throughput with a higher value is more often an absolute choice in every network. Mathematically, throughput can be defined by the following formula [44].

\[
\text{Throughput} = \frac{\text{Number of delivered packet} \times \text{Packet size (Bytes)} \times 8}{\text{Total duration of simulation (sec)}}
\]

(1)

5.2.2 End-to-End Delay

The end-to-end delay is the time needed to traverse from the source node to the destination node in a network. End-to-end delay assesses the ability of the routing protocols in terms of use-efficiency of the network resources. Generally, the end-to-end-delay is measured as per the following equation [61].

\[
D_{\text{end_to_end}} = N \left[ D_{\text{trans}} + D_{\text{prop}} + D_{\text{proc}} \right]
\]

(2)

Where
\[
D_{\text{end_to_end}} = \text{end to end delay.}
D_{\text{trans}} = \text{transmission delay.}
D_{\text{prop}} = \text{propagation delay.}
D_{\text{proc}} = \text{processing delay.}
N = \text{a scalar number.}
\]
5.2.3 Upload Response Time

The time duration elapsed between sending a file and receiving the response is known as upload response time.

5.2.4 Download Response Time

Download response time is defined as the time elapsed between sending a request and receiving back the response packet, which is measured between the time a client sends a request to the server and the time it receives back a response packet.

5.2.5 Retransmission Attempts

Retransmission attempts can be defined as the total number of retransmission attempts of packets by all WLAN MACs in the network that have been lost or damaged due to a link failure. It also shows the number of packets failed in the process, which, in effect, requires retransmission.

5.3 Network modeling

The network models of the current study are designed, in the OPNET simulator, by taking help of different network entities. An example of such network models is presented in Figure A.4 (In Appendix A) where a network size of 100 nodes is confined in a (1000×1000) square meter area. The network entities used during the design of the network model are wireless server, application configuration, profile configuration, mobility configuration and workstations (nodes). These model objects are basically a series of network components that allow attribute definition and tuning.

Application configuration is an essential object that defines the transmitted data, file size and traffic load. More often, it supports common applications, namely, HTTP, FTP, Database, Email, Print and so on. We have chosen FTP and HTTP applications for data traffic analysis where each application is considered with heavy traffic load (individually), in line with the requirement for bandwidth utilization.

On the other hand, profile configuration determines where the data is received by specifying the interaction between servers and clients [32]. This is employed to create the user profiles whereas these profiles are specified on different nodes in the network for generating the application traffic. For instance, an FTP profile is created in a profile configuration entity in order to support the FTP traffic, which is generated by an application configuration entity.
One of the other important entities is the mobility configuration, which is used for the purpose of determining the mobility model of the nodes. Moreover, it has to select several appropriate parameters such as speed start time, stop time, pause time and the like, to properly control the movement of the nodes in the network. The reason for configuring the mobility object is to allow the nodes to move within the specific allocated network area, which is chosen as 1000 square meters in our simulation network model. In other words, the traffic generated from outside this specific range, if any, will not be taken into account. Nevertheless, in order to configure the nodes with a mobility option, a widely used mobility model known as the default random waypoint mobility is used for all simulation purposes in the present study. As described in [59], Random Waypoint model allows the mobile nodes to keep moving until they arrive at a random destination defined by such algorithm. Upon arrival at this destination, the nodes get stop at this place for a period of time, which is called the pause interval. A new movement is further made with a random direction and speed as soon as the pause time has expired. The combination of pause time and velocity sets up relative degrees of mobility between mobile nodes in the simulated network. In order to symbolize the mobile behavior of the nodes, the speed of the node is initially set to 10 m/s with a pause time of 50 sec to observe the network behavior with low mobility. At some later stage, the speed is increased to 20 and 30 m/s with the same pause time so that the nodes can travel with greater speed in the network. The reason for increasing the node speed is to observe the impact of mobility on MANET performance.

In our thesis, the server module is configured to support and control the application services (i.e. FTP and HTTP) based on the user profile. This is basically a WLAN server through which a particular routing protocol and a TCP variant can be selected. The nodes are defined as workstations with client server applications running over TCP/IP, support the underlying WLAN connection at 1 Mbps, 2 Mbps, 5.5 Mbps and 11 Mbps. The connection speed is set at 5.5 Mbps in our study.

Finally, all mobile nodes are configured to generate FTP and HTTP traffic randomly, with the ability to route the data packets to the desired destinations.

5.4 Network Configuration Parameter

This section presents several design attributes along with their values in tabular formats, which are configured during the implementation of the proposed network model. All these tables are provided in Appendix B. Table B.1 demonstrates the general parameters used in the process of all simulation experiments of the study. Meanwhile, the parameters used for wireless LAN configuration are portrayed in Table B.2, where parameter values are similar to those provided in the research work of [37], with the exception of the buffer size, which is set to 256,000 bits. As a matter of fact, a medium flow of application has been intended to be generated in our simulation experiment. Likewise, in order to avoid the potential problem related to manual error, the
channel setting is fixed at that which is auto-assigned. On the other hand, the parameters such as slow start initial count, initial RTO, minimum RTO and maximum RTO are used as TCP simulation parameters, which are set to the default values as shown in Table B.3. Aside from that, the configuration parameters, as defined in Tables B.4 and B.5 are used during the configuration of FTP (heavy load) and HTTP (high browsing) applications. Since these two applications transfer files at a fixed interval, exponential (360) is set to generate the FTP heavy traffic load while exponential (60) is assigned to create the HTTP heavy browsing load. Again, as explained in the OPNET product specifications, the start time for a file transfer session is computed by adding the inter request time to the time that the previous file transfer started. Following that, we specified the profile generation start time as uniform (100, 110), while the start time for the application is set as constant (5). Further, the profile and application repeatability are occurred only once at a start time as can be shown in Table B.6 and B.7. Finally, Tables B.8 through to B.11 are presented to show the parameters utilized to configure the proposed routing protocols where the configuration parameters for all the protocols have been set to default values, as are also set by OPNET Modeler 16. This is because of the fact that the results with these default values often show more significant performance than those configured values used in [22], [42], [50] and [52].

5.5 Network Scenarios

In this section, three scenarios are described under two specific categories, presented in tabular form. Table 5.1 presents different scenarios for the network scalability while various scenarios for the node mobility are shown in Table 5.2.
### Node size Investigations (Scalability)

<table>
<thead>
<tr>
<th>Types of Scenario</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scenario 1 (Small Size Network)</strong></td>
<td>Scenario 1 is similar to what is shown in Figure 5.1; this is a network environment designed with different entities, configured for a network size of 30 nodes, the file size of 50,000 bytes (for FTP) and 1000 bytes (for HTTP), a node speed of 10 m/s with a pause time of 100 sec. Thereafter, different MANET routing protocols and TCP algorithms are employed in the network and their performance is evaluated for the small-sized network (i.e. node size = 30), based on the analysis of the performance metrics.</td>
</tr>
<tr>
<td><strong>Scenario 2 (Medium Size Network)</strong></td>
<td>Scenario 2 represents a medium-sized network where the network model is designed with 60 nodes. However, the value of node speed and the file size have not been subject to changes but set at, as in Scenario 1. The intention is to observe the performance of the routing protocols and the TCP variants through varying the node sizes from 30 to 60.</td>
</tr>
<tr>
<td><strong>Scenario 3 (Large Size Network)</strong></td>
<td>This network scenario (Scenario 3) is similar to that of Scenario 1 and Scenario 2, except that the network size is increased to 100 nodes, so as to observe the impact of scalability in MANET.</td>
</tr>
</tbody>
</table>

**Table 5.1:** Description of the Experimental Scenarios for Network Scalability
<table>
<thead>
<tr>
<th>Type of Scenario</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 4 (Low Mobility Network)</td>
<td>The network scenario is configured with different network objects, for a node speed of 10 m/s, a pause time of 50 sec, a network size of 60 nodes and the file size of 50,000 bytes (for FTP) and 1000 bytes (for HTTP). The justification of designing such a scenario includes evaluating the network performance with lower mobility rate within a medium size MANET.</td>
</tr>
<tr>
<td>Scenario 5 (Medium Mobility Network)</td>
<td>The scenario focuses on analyzing the effects of routing protocols and TCP variants whilst the mobility rate is varied from 10 m/s to 20 m/s in a medium size network. The pause time value is kept same as scenario 4.</td>
</tr>
<tr>
<td>Scenario 6 (High Mobility Network)</td>
<td>Similar to the scenario 1 and 2, a network environment is designed with different network entities and configured with a network size of 60 nodes; however the node speed is increased to 30 m/s with a pause time of 50 sec. The purpose of designing such scenario is to evaluate the impact of high mobility in a medium size network. Particularly, this scenario aims to investigate the behavior of the routing protocols and TCP variants when the node speed changes from 20 m/s to 30 m/s.</td>
</tr>
</tbody>
</table>

**Table 5.2:** Description of the Experimental Scenarios for Node Mobility
CHAPTER 6
RESULTS AND ANALYSIS

In chapter 5, the aspects related to the task of MANET network modeling and experimental designs were presented. This chapter (Chapter 6) presents experimental results for two different network scenarios in a MANET environment. Section 6.1 outlines the impact of network size extension on the performance of routing protocols and TCP versions while section 6.2 deals with mobility issue and its impact on the network performance.

6.1 Varying Network Size

This section elaborates the results based on the experimental scenarios 1, 2 and 3 as outlined in chapter 5. The performance analysis of different routing protocols and TCP variants within MANETs has been carried out through these scenarios where each scenario is presented against node sizes representing small, medium and large network.

The routing performance is evaluated using TCP SACK variant since this is considered as a newer and widely deployed version now-a-days [53]. On the other hand, the performances of different TCP variants are assessed with DSR routing protocol as the DSR interacts with TCP more efficiently than the other protocols under different realistic MANET scenarios [45]. To observe the impact of node variation on routing and TCP performance, the target applications are run with various networks sizes (30, 60 and 100 nodes). Though this section deals with network size issue; it is much more realistic for a MANET environment to generate at least a low mobility rate instead of keeping it fully static. Accordingly, a moving speed of 10 m/s with average pause time of 100 sec is set as to allow the mobile nodes to move slowly in the network.

6.1.1 Impact on Throughput

Throughput refers to the amount of traffic successfully received by the destination node. The routing efficiency can be predicted by observing the overall throughput received by the network.
As mentioned in previous chapter, the start time of profile and application generation is set to 100 sec and 5 sec, respectively. No traffic is generated therefore up to 105 sec of the simulation time. This period is often known as the warm up time. A warm up time allows the queues and other aspects in the simulation to get into conditions which are typical of normal running conditions in the system [29].

Figure 6.1: Average throughput for different routing protocols; (a) Small network size (node=30), (b) Medium network size (node=60) and (c) Large network size (node=100).
Figure 6.1 demonstrates the average throughput of OLSR, DSR, AODV and TORA under various network scenarios. The X axis shows the simulation time in seconds while the Y axis shows the throughput in bits/sec. In a small network (Fig 6.1.a), when transmitting a FTP and HTTP traffic in the network, OLSR exhibits quite satisfactory performance compared to the other three routing protocols, receiving an average throughput of about 735,422 bits/sec. Considering the reactive protocols, AODV provides better performance than DSR and TORA, achieving up to 278,145 bit/sec throughput on average. Meanwhile, the average throughput for TORA and DSR are found to be 82,205 bits/sec and 78,259 bits/sec, respectively. The packet received for TORA is found to be slightly better than DSR due to the presence of mobility in the network. However, the performance of TORA tends to fall at 420 seconds whereas DSR is found to experience some improvement at the same time.

With the network size shifting to a medium one (Figure 6.1.b), the overall throughput tends to increase since more nodes are available to route the packets to the destination. It is apparent that OLSR keeps outperforming other routing protocols through achieving a higher throughput of 4,670,035 bits/sec on average. On the other hand, DSR and TORA achieve the lowest amount of traffic in the network, approximately 170,950 bits/sec and 141,743 bits/sec, respectively. Meanwhile, AODV receives an average throughput of 1,063,200 bits/sec and is favored over DSR and TORA thereby.

In a large network (Figure 6.1.c), the average throughput of OLSR is about 9,800,697 bits/sec, which is approximately 2 and 13 percent higher than that of a medium and a small network, respectively. In such a network, OLSR protocol continues to be dominating over AODV, DSR and TORA. On the other hand, AODV has been found to perform better than those with TORA and DSR. In a large network, a consistent throughput of 1,455,589 bits/sec (on average) is maintained by AODV, which is 27 and 80 percent higher than that of a medium and a small network, respectively. Over and above, the average throughput for TORA and DSR are found to be 328,978 bits/sec and 160,546bits/sec, respectively. However, initially DSR receives slightly higher throughput than with the TORA. TORA starts outperforming over DSR at 140 seconds, which is maintained until the end of the simulation time.

Out of the three sizes of the network, OLSR can be reckoned as the most effective one among the four proposed existing routing protocols. The significant performance achieved by OLSR can be considered due to the proactive characteristics, which are often followed by this protocol. OLSR continuously maintains and updates the routing information with the help of Multipoint Relays (MPR) in the network, resulting in the reduction of routing overhead in the network. In addition, the independency of network size and network traffic also causes OLSR protocol to receive more data packets. For OLSR protocol, the larger the network size, the more optimization that can be achieved, as compared to the other routing algorithms.
Similarly, AODV protocol is also desirable when the goal is to achieve more throughputs regardless of the incremental network size. This protocol follows hop-by-hop routing mechanism and eliminates the source routing overhead in the network. Apart from that, the availability of multiple route information in AODV facilitates in producing the higher amount of throughput in the network.

On the other hand, both TORA and DSR receive the least amount of throughput despite that the performance tends to be better when the network becomes denser. Since DSR follows a source routing mechanism, the byte overhead in each packet drastically affects the total byte overhead when the size of the network increases [41]. As a result, DSR tends to achieve lower amount of data packets in more stressful network. On the other hand, TORA is found to increase unnecessary overhead due to its route adaptation feature (i.e. updating path information and route establishment) in response to topological changes. This feature eventually decreases the throughput in the TORA based network. Moreover, because of updating the routing information TORA has to transmit a large number of control packets as it indirectly maintains a table driven approach (flooding method).

### 6.1.2 Impact on End-to-End Delay

End-to-end delay for a data packet is measured from the time it is created to the time it is received. High end-to-end delay indicates more broken links and frequent re-routing during the transmission of the data packet.
Figure 6.2: Average End to End Delay for different routing protocols; (a) Small network size (node=30), (b) Medium network size (node=60) and (c) Large network size (node=100).

As can be seen in Figure 6.2a, the OLSR has the lowest (but steady) end-to-end delay of about 0.71 milliseconds (ms) on average, while the end-to-end delay for the TORA is about 5.57 ms, displaying a maximum delay among all the routing algorithms. However, at the initial stage of the simulation time, the delays of DSR and AODV are found approximately 8.5 ms and 6.5 ms, respectively, which are even higher than TORA. With increasing simulation time, they settle around 5 and 2 ms, respectively, and remain there for the rest of the time.

As the network size shifts to medium (Fig 6.2b), the OLSR and the AODV still achieve the lower end-to-end delay, approximately 1.08 ms and 4.76 ms, respectively. Despite that the delay is increased by about 34 and 56 percent respectively, as opposed to the case of a small network scenario. On the other hand, the maximum packet delay is found in DSR and TORA based network, approximately 11.81 and 13.30 ms, respectively. Initially the delay value of DSR was even larger than TORA, approx 55 ms. However, it is surpassed by the TORA at about 180 seconds of simulation time.

In the case of a large size network (Fig 6.2c), the end-to-end delay for both the DSR and the TORA initially rises dramatically, and then start dropping at 100 sec and 220 sec of the simulation time, respectively. Both protocols end up with less end-to-end delays although the TORA maintains higher delays than that with the DSR, on average. On the other hand, both the AODV and the OLSR require the lowest time to transfer the data packet and their performances are found to be quite stable throughout the simulation.
When analyzing the packet end-to-end delay variations against different sizes of network, the results using the OLSR protocol are of particular importance as it establishes quick connections between nodes without making significant delays. On the other hand, the delays experienced in the TORA and the DSR based network are much higher. Unlike other routing protocols, the OLSR does not use much time in Route Discovery mechanism since the routes are available beforehand in the OLSR when the data transmission is needed, thereby resulting in the lowest end-to-end delay. Even with a higher density of the network, the performance is not found to be degraded and a constant lower delay is noticed for the OLSR. This is because it has the advantage of utilizing the MPR nodes to enable forwarding of the control messages to other nodes. Thus it eventually helps to minimize the network overhead.

One of the factors responsible for the relatively poor performance of the TORA is related to the formation of temporary loops within the network, where the collisions of the MAC layer are held by the transmitted routing packets and. Consequently, the links to neighbor nodes could have been broken by IMEP. In response to link failures, TORA sends more updated packets, whereas an acknowledgement of the re-transmitted update packet might not be received, resulting in a serious congestion of the network. As a result, an extremely high delay is introduced in the network, which is further enhanced with an increase in the network size.

Likewise, DSR is also not able to establish the node connection quickly and, thus, perform unreliably in the network. Since the DSR adopts a reactive approach, it is very likely that data packets keep on waiting in buffers until a route is discovered enroute to the destination. Besides, when a route request packet is sent in order to discover the route, the destination node replies to all route request packet it receives. As a matter of fact, the DSR needs significant time to determine the least congested route. Apart from that, the DSR follows a source routing mechanism where the complete route information is included in the packet header, causing an increase in the packet length, and thereby an increase in the delay experienced by the packets in the network. Thus, it can be inferred that the denser the network, the higher the delays that are likely to be experienced in the network while utilizing the DSR protocol.

**6.1.3 Impact on Download Response Time**

The efficiency and effectiveness of upload and download activities are evaluated by the extent of upload response time and download response time. Hence, in data traffic measurements these two quantified parameters play a vital role where the lower the value achieved, the faster the task proceeded.
Figure 6.3: Average download response time for different TCP variants; (a) Small network size (node=30), (b) Medium network size (node=60) and (c) Large network size (node=100).
Figure 6.3 demonstrates the download response time for transmitting a FTP file (50,000 bytes) where X axis and Y axis in each figure represent simulation time and download response time, respectively. For all the scenarios, the highest download response time takes place at the beginning of the simulation time, approximately 0.42 sec (with TCP SACK) in a small network, 5.5 sec (with TCP SACK) in a medium network and 17.5 sec (with TCP New Reno) in a large network. Subsequently, the download response time for all the TCP variants reduces sharply within the instance of first few minutes, which is then maintained to stabilize in the long run, although a smoother drop occurs for all the TCP variants in 30 nodes scenario, as evident in Figure 6.3a.

In a small network (Fig 6.3a), the average download response time for Reno is approximately 0.286 sec, which estimates as 0.96 and 0.88 times shorter than that of SACK and New Reno, respectively. In a medium density network, Reno again achieves the shortest response time in acquiring a FTP file as shown in Fig 6.3b. However, in terms of large size network, Reno cannot ensure the best performance. In a large network (Fig 6.3c), TCP SACK version reduces its response time dramatically and outperforms thereby.

The three algorithms have almost identical file download response time in the case of small size network. Although, the performance of FTP file download response time degrades with large number of users. Again, because of the fact that more links are established in a higher density scenario the network becomes more prone to signal attenuation and multipath fading, thereby causing performance fluctuations and degradation among different TCP variants. In effect, TCP unnecessarily invokes counterproductive and time consuming congestion control mechanisms. As a result, more time is taken to finish the data recovery activities. Similarly, more time is to be spent to download a file in the presence of high number of nodes in a network.

As noted in [36], in the presence of a high signal attenuation, TCP SACK and Reno maintain a larger congestion window size, compared to the other TCP variants while it is observed in [37], that the larger the congestion window size, the shorter the file response time would be. Thus, a validated simulation is accomplished since our study demonstrates that both TCP SACK and Reno achieve a shorter file response time in a high-density network where the signal attenuation is more frequently induced.

6.1.4 Impact on Upload Response Time

The findings obtained from the preceding section demonstrate the performance of different TCP variants in terms of their download response times. In this section, a comparative analysis is carried out to determine the effects of upload response time.
Figure 6.4: Average upload response time for different TCP variants; (a) Small network size (node=30), (b) Medium network size (node=60) and (c) Large network size (node=100).
Looking at the Figure 6.4, one can make the observation that the upload response time varies greatly with the increase in network density. It is evident that a data file (50000 bytes) via FTP application results in a maximum of 0.28 sec (TCP SACK) response time in a small network, 1.9 sec (TCP SACK) in a medium network, and 5.9 sec (TCP New Reno) in a large network.

In a small network (Fig 6.4a), the upload response time for all the TCP variants is initially found to be higher. However, as the simulation progresses, the response time drops and settles for the remaining time. The lowest upload response time in such a network is observed for New Reno, approximately 0.19 sec (on average), followed by 0.20 and 0.22 sec for SACK and Reno, respectively. Meanwhile, with the medium (Fig 6.4b) and the large network (Fig 6.4c), the response time increases dramatically (for all the TCP variants) at 120 sec of simulation time, which then reaches their peak, followed by a decrease until the end of the simulation. The New Reno ensures the lowest response time in a medium network, approximately 0.46 sec on average, while the SACK outperforms other TCP versions in a large network by taking the lowest response time of 1.12 sec.

It appears that the small and medium networks accommodate such FTP file very quickly and thus ensure smooth transmission since in such networks. The average response time takes less than one second for all the variants regardless of the increase in the network size (up to 60 nodes). However, due to uplink limitation a much higher response is required in a situation where a large number of users are present (e.g. Network with 100 nodes). In addition, since all 100 nodes start uploading concurrently, extra load is gained from the large network size, eventually resulting in higher response times in order to complete the uploading task. Apart from that, due to the presence of a high packet error rate in a large mobile ad-hoc network, the TCP retransmission mechanism is generated more frequently, thereby consuming more network bandwidth. In consequence, this leads to huge delay to upload a FTP file in a high-density network.

### 6.1.5 Impact on Retransmission Attempt

The quantitative parameter, called Retransmission Attempt, not only determines the rate of retransmission attempt, but can also figure out the number of packet drops per second, which has to be retransmitted. So, the lower is the retransmission attempt, the more reliable is the TCP variant.
Figure 6.5: Average retransmission attempts for different TCP variants; (a) Small network size (node=30), (b) Medium network size (node=60) and (c) Large network size (node=100).

In all the scenarios of Figure 6.5, the highest packet drops are noticed at the beginning of transmission. Hence, the maximum retransmissions are attempted at that duration. Looking at the figures, it can be seen that in all the cases, the curves drop abruptly but tend to get settled in the long run. However, a mild exception appeared with Reno in the 30 and 100 nodes scenario while it keeps on with a slight increase at 420 sec of the simulation time.
In the case of wire connection, the TCP retransmissions are caused usually due the network congestion. As compared with the wired media, the wireless medium provides much more noisy physical links for data transmissions where signals propagated through these links can suffer from degradation, interference, and noise. Accordingly, more packet losses are experienced, which cause more retransmissions.

With the increase in node number, the numbers of retransmissions are increased as well for all three window-based congestion control protocols. This is due to the physical layer disconnection as well as the increase in packet error rates in the high-density network. In addition, the channel contention is also found to be increased as more routing loads are experienced in larger networks.

The TCP Reno variant dominates in both small and medium density networks, maintaining the lowest average retransmission rate of 0.05 packets / sec (medium network) and 0.24 packets / sec (small network). On the other hand, the lowest retransmission in a large network is attempted with SACK variant, which is approximately 1.09 and 1.26 times lower than Reno and New Reno, respectively. In a large size network, aggressive employment of window mechanisms is considered one of the main factors responsible for causing more retransmission in TCP New Reno. During the slow start phase, the aggressive and inappropriate window growth of New Reno causes the network to be overloaded, which induces periodic packet losses on the link layer and more frequent timeouts in the transport layer. Thus, frequent link contentions and more link failures are occurred in the MAC layers and resulted in a great number of retransmission in the network [29].

6.2 Varying Node Mobility

This section presents details of the experiments carried out to evaluating the routing and TCP performance whilst the mobility rate is varied in a MANET environment. The analysis is elaborated based on three experimental scenarios 4, 5 and 6, as presented in the preceding chapter. The scenarios considered in this analysis consist of 60 nodes moving with node speeds of 10, 20 and 30 m/s. The pause time, as already mentioned, is set to 50 sec for all three speeds.

6.2.1 Impact on Throughput

In this sub-section, the performance of the routing protocols in terms of throughput is examined with respect to mobility of the nodes.
Figure 6.6: Average throughput for different node speeds (10 m/s, 20 m/s and 30 m/s); (a) AODV routing Protocol, (b) DSR routing protocol, (c) TORA routing protocol and (d) OLSR routing protocol.
Figures 6.6 displays a graphical representation of a comparative analysis on the throughputs derived from various mobility scenarios. The X axis shows the simulation time in seconds while the Y axis shows the throughput in bits/sec. In Figure 6.6a, the topmost curve represents the AODV throughput, generated when the mobility rate is of 10 m/s. As can be seen, at the very beginning the throughput rises gradually and starts surpassing 1,500,000 bit/sec at some later stage. When the node mobility is shifted to a medium rate (20 m/s), lower throughput is achieved, amounting to approximately 977,152 bit/sec, on average. Similar to the medium mobility network, the throughput in a high mobility network keeps on rising gradually, however, with a lower rate than that of the medium rate network. The average throughput received in a 30 m/s network is about 957,896 bit/sec, although the performance tends to show improvement towards the end of the simulation period.

Meanwhile, in the case of DSR protocol (Figure 6.6b), the decrease of the throughput is somewhat noticeable but not dramatic in high mobility scenarios. Among the three scenarios, it appears that the low mobility results in the highest average throughput of 170,950 bit/sec, which is approximately 1.03 and 1.06 times as much as that of a medium and a high mobility rate. On the other hand, as depicted in Figure 6.6c, the throughput of TORA initially increases for all mobility speeds and then reaches a peak, followed by, a gradual reduction until approaching the end of the simulation task. When the mobility rate varies in TORA, a slightly lower throughput is observed in a high mobility scenario, compared to that in a low and a medium mobility.

Now turning to Figure 6.6d, it can be observed that OLSR protocol attains a higher throughput, followed by those with AODV, DSR and TORA. Throughout the entire simulation time, OLSR is found to maintain a consistent throughput. Even with higher mobility rates in the network, OLSR keeps its performance at a steady level. The highest average throughput of OLSR is attained in a 10 m/s speed, which is approximately 1,063,200 bit/sec. Subsequently the throughput reduces to 977,152 bit/sec and 957,896 bit/sec when the mobility rate is increased to 20 m/s and 30 m/s respectively.

With the incidence of increased mobility rates, frequent changes of the nodes and their neighbors occur, subsequently causing frequent changes in the link state and further more packet losses. With the lower mobility rate, however, the performance of AODV is found to be considerably enhanced as the network topology remains almost constant for a low speed network. Even when the speed increases, AODV is slightly affected as the routing tables are more frequently updated in response to topology changes in the network, resulting in fewer packet drops and less performance degradation.

Similarly, the route stored in DSR cache can be used effectively with a lower node speed prevailed in the network. Nonetheless, in the presence of a high mobility rate, one can observe a larger drop in DSR throughput because of it’s yet dependence on the cache routes, which are more likely to become stale at higher speeds.
It is also apparent that the performance of TORA deteriorates with the increase in mobility, although it provides the multipath routing mechanisms. In responding to topological changes (due to high mobility), TORA follows an adaptive method of updating the path information apart from re-establishing the route. In effect, this route adaptation feature increases the network overhead and causes fewer amounts of throughputs to be received by the network.

In contrast, OLSR outperforms the reactive and hybrid routing protocols due to its ability to maintaining the constant information of the network topology. It is apparent that even with a high mobility condition, the OLSR performance is not degraded. In our views, the superiority of OLSR is due to its ability of promptly detecting the route failure and carrying out continuous searches for the routes to all possible destinations, thereby updating the routing information quickly. In such event, a fewer number of packets are likely to have been dropped, resulting in more data packets successfully received in the network.

6.2.2 Impact on End to End Delay

In this sub-section, the routing performance is analyzed in terms of end-to-end delay with the variations of node speeds.
Figure 6.7: Average End to End Delay for different node speeds (10 m/s, 20 m/s and 30 m/s); (a) AODV routing Protocol, (b) DSR routing protocol, (c) TORA routing protocol and (d) OLSR routing protocol.

Figures 6.7 demonstrate the end-to-end delay of AODV, DSR, TORA and OLSR respectively, under various mobility rates. The X axis represents the simulation time and the Y axis represents the end-to-end delay.

The AODV, with a node speed of 10 m/s, maintains a lower delay level of about 4.76 ms on average; a delay of 4.87 ms is achieved when the node speed is 20 m/s. It increases further to 4.95 ms when the speed is changed to 30 m/s (Figure 6.7a). On the other hand, the average end-to-end delay of DSR is found to be 11.81, 12.24 and 15.25 m/s with corresponding node speeds of 10, 20, and 30 m/s respectively (Figure 6.7b). It can be observed that DSR follows a trend similar to AODV in increasing delays with the increase in mobility. However, their average end to end delays for all the mobility rates are much higher than those of AODV. Further, the delay results in Figure 6.7c show that with the increase in mobility rates, the performance of TORA drastically deteriorates in respect to the network. The average end-to-end delay for TORA in a high mobility network is observed as about 15.55 ms, which is 1.09 and 1.17 times as much as that of a medium and a low mobility network, respectively.

In contrast, as the Figure 6.7d depicts, OLSR consistently maintains a lower rate of end-to-end delay, as opposed to those of other routing protocols. Still, one can observe the impact of mobility from the results. In effect, with a node speed of 30 m/s, OLSR has a little higher delay.
value than that of 20 and 10 m/s. Following this, with the node speeds of 10, 20, and 30 m/s, the
lowest delays of 1.08, 1.13 and 1.16 ms respectively can be achieved.

Being a reactive protocol, AODV utilizes the on-demand routing strategy, which is not able to
preserve the unused routes in the network. Instead, AODV carries out searches for the new
routes when they are needed. This strategy usually generates less control traffic. However, it
increases the overall end-to-end delay in the network as packets remain waiting at buffers until
they are transmitted through the new routes. In addition, AODV maintains only one route per
destination in its routing table. Therefore, whenever a route breakage occurs in the network (due
to high mobility), an additional route discovery is needed each time to establish the new route
[39]. This implies that the number of route discovery in AODV is directly proportional to the
number of link failures. Again, when the route discovery mechanism is generated owning to
node mobility, it takes a significant time in each occasion. As a result, more delays are likely to
be induced in the network.

Unlike AODV protocol, DSR does not trigger the route discovery mechanism so often due to
the presence of the abundant route caches at each node. Consequently, a route discovery is not
initiated unless all cached routes are broken. However, it has a high probability for these caches
to become stale in high mobility scenario [7]. In addition, the interference to data traffic is
increased in DSR network due to the generation of a high MAC overhead during the route
discovery mechanism. This MAC overhead, together with the cache staleness, causes significant
performance degradation in the network.

On the other hand, the advantages used by TORA are due to the fact it can maintain multipath
capability nature. However, as noted in [48], TORA takes longer time to complete its initial route
discovery mechanism. This might affect the performance in the event of occurrence of a network
partition due to the high mobility. Thus, the overhead of finding and maintaining multiple paths
appears to have outweighed the potential benefits. In addition, the loss of distance information
due to the link failure in a mobility network also causes TORA to have a poor delay performance
in the network.

By contrast, OLSR does not explicitly show its reaction to link failure since it is a link state
protocol and the associated MPR nodes periodically transmit topology information to other
nodes across the network. As a matter of fact, it exhibits the lowest end-to-end delay among the
four routing protocols, the delay even being found almost insensitive to changes in speed.
Furthermore, OLSR maintains the route before it is demanded, followed by a lower delay
introduced in the network.
6.2.3 Impact on Download Response Time

In this sub-section, the performance of Reno, New Reno and SACK variants are evaluated in terms of download response time.

![Figure 6.8: Average download response time for different TCP variants; (a) node speeds 10 m/s, (b) node speeds 20 m/s and (c) node speeds 30 m/s.](image)

Figure 6.8 demonstrates the download response time of three TCP variants whilst the mobility rate is varied in the network. In this case, X-axis and Y-axis in each figure stand for simulation time and download response time, respectively. The download performance is evaluated in terms of FTP traffic with a file size of 50,000 bytes.
Figure 6.8a depicts a scenario against a node speed of 10 m/s where the lowest average
download response time is observed in Reno version, amounting to approximately 0.96 sec,
followed by 0.99 sec and 1.74 sec in New Reno and SACK versions respectively.

Similarly, in a 20 m/s network (Figure 6.8b), Reno continues in accounting for the lowest
download response time of about 1.18 sec, on average, while New Reno and SACK versions
require nearly 1.36 sec and 1.52 sec respectively, to download the above mentioned FTP file. At
the beginning, the required response time is found to be quite high for all the three variants,
about 5 sec in SACK, 3.8 sec in Reno and 3.7 sec in New Reno. However, the response time
gradually falls with even a little fluctuation, which tends to be stabilized towards the end of
simulation.

As can be observed from Figure 6.8c, the average download response time for all the variants
in a 30 m/s network is slightly less than that in a 20 m/s network. More specifically, the average
response time for Reno is approximately 0.89 sec, which is 1.32 times lower than that of a 20
m/s network. On the other hand, SACK and New Reno achieves an average response time of
1.08 and 1.28 sec respectively, which estimates as 1.42 and 1.07 times shorter than that of a
medium mobility network.

### 6.2.4 Impact on Upload Response Time

The performance of different TCP versions is analyzed in this sub-section in terms of their
response time to uploading a FTP file of 50,000 bytes.
Figure 6.9: Average upload response time for different TCP variants; (a) node speeds 10 m/s, (b) node speeds 20 m/s and (c) node speeds 30 m/s.

The X-axis shows the simulation time while the Y-axis shows the upload response time (Figure 6.9). Looking at the figures, it becomes apparent that the upload response time does not vary significantly with the increase in node speeds. In order to carry out uploading of a FTP file of 50,000 bytes, the maximum required time is estimated as 0.571 sec for Reno in a low mobility network, followed by 0.572 sec for SACK and 0.72 sec for New Reno in a medium and a high mobility network, respectively.

For 20 m/s network, (Figure 6.9b), TCP Reno and New Reno require shorter response time, approximately 0.51 and 0.52 sec respectively. When the node speed is changed to 30 m/s (Figure 6.9c), TCP New Reno is no longer able to perform superiorly; the performance is rather deteriorated abruptly. TCP Reno version, on the other hand, continues to be dominating in a high mobility network by ensuring the lowest response time of about 0.44 sec, on average. Meanwhile, SACK variant maintains a moderate response time in a high node speed, although the performance tends to be outperforming in the long run.

6.2.5 Impact on Retransmission Attempt

In this sub-section, the performance of different TCP versions in terms of retransmission attempt is examined with respect to mobility of the nodes.
Figure 6.10: Average retransmission attempts for different TCP variants; (a) node speeds 10 m/s, (b) node speeds 20 m/s and (c) node speeds 30 m/s.

Figure 6.10 demonstrates the retransmission attempts for Reno, New Reno and SACK variants in the presence of different mobility rates within the MANETs. As one can observe, the increase in the node speeds in a high mobility network (30 m/s) results in an increase in retransmissions, whereas the increase in node speeds in a medium mobility network (20 m/s) results in a decrease in retransmission.
As can be observed from Figure 6.10b, Reno variant achieves the lowest average retransmission rate throughout the simulation period, which amounts to about 0.039 packets/sec. On the other hand, the second lowest rate of approximately 0.045 packets/sec is achieved by TCP SACK version, followed by approximately 0.049 packets/sec by New Reno.

The impact of the mobility can be observed from Figure 6.10c, where a higher rate of retransmissions is attempted by all the TCP variants due to the increase in node speed to 30 m/s. With the New Reno variant, the highest retransmission is attempted to the extent of 0.067 packets/sec, which accounts for 27.22 and 24.01 percent higher, compared to those in a medium and a low network respectively. Again, the SACK version accounts for the lowest retransmission, approximately 0.046 packets/sec, which is about 1.07 and 1.46 times as less as that of Reno and New Reno variants respectively.

Unlike in the case of wired links, wireless links, which use air as a transmission medium, suffer from wireless channel error and link failure within the network. Since the communication path in a MANET is associated with multiple wireless links, the link failures (either due to mobility of nodes or high bit error rate) can cause a significant amount of packet losses in such a network. In response to a packet loss in traditional network, TCP retransmits the lost packet from its own source. However, in a MANET associated with a high error rate, TCP may have to take several retransmissions to deliver a packet to its destination successfully.

In our observation, when low mobility (10 m/s) is present in the network, the communication path can be considered relatively stable and hence fewer packets are dropped. On the other hand, in the case of a high mobility (30 m/s), all the three TCP variants retransmit higher amount of packets as a reaction to route breakages in the network. This is attributed to the fact that all of these versions are not capable of adjusting the congestion window size dynamically according to the status of the bottleneck, resulting in getting more susceptible to packet losses in a wireless medium.

When link failure takes place due to the high mobility in a MANET, all of three TCP versions mostly differentiate the data packet loss through observing the TCP RTO timer. As none of them are designed to cope with such situations (i.e., link losses), they are all found to react similarly in a mobile ad-hoc network. However, the TCP SACK is found to be relatively more robust to the dynamics of the wireless channels. Since this version allows a receiver to only indicate segments that are received, the sender usually retransmits only the lost segments, resulting in the least number of retransmission attempts compared to in the other two versions.
6.3 Summary of Routing protocols Performance

This section presents the summary results on the routing protocols performance in terms of throughput and end-to-end delay. Amongst the four routing protocols, Figures 6.11 and 6.12 represent the performance graphs for different network densities and node speeds, where it can be seen that the different properties of each protocol have led to a variety of differences in their performances.

6.3.1 Performance evaluation with varying network density

Today, the creation of large scale ad-hoc networks has become the utmost choice in the field of tactical military networks, natural disaster recovery services, vehicular networks and consumer networks. Following this, there is a pressing need for a scalable ad-hoc routing protocol to support the networks that are larger by one or several orders of magnitude. In this section, we present what MANET existing routing protocol can scale reasonably well when node size in the network is increased from 30, 60 and 100.

Figure 6.11: Performance of routing protocols for different network sizes; (a) In terms of throughput (b) In terms of end to end delay

The throughput outcomes of the routing protocols can be summarized in Figure 6.11a. In a low-size network, the four protocols have close levels of throughput, however, with a slightly higher performance in the case of OLSR. The values are close due to the small number of nodes
that leads to a less routing overhead in the network. The OLSR throughput is found to be increasing for any further increases in number of nodes in the network. For instance, in a network consisting of 100 nodes, the received throughput of OLSR is 2 percent greater than that of 60 nodes and 13 percent greater than that of 30 nodes. Similarly, the AODV also adapts to large networks as its performance increases almost linearly with the increase in network sizes. However, the difference between performance levels of OLSR and AODV are found to be greater in a large size network. On the other hand, TORA and DSR protocols are not found to perform adequately in our simulation. Both of the protocols keep receiving the lowest amount of data packets regardless of the incremental network sizes, because of which this cannot be considered as scalable protocols.

The packet end-to-end delay variation is shown in Figure 6.11b. In a 30 node scenario, one can observe that the delay value is almost the same for each protocol. Nevertheless, when the number of nodes increases, the delay of DSR and TORA increases drastically and maintain a growth delay in the network which is almost exponential in nature. On the other hand, in the case of AODV, the delay growth is nearly linear while for OLSR, it is found to experience hardly any increase. Our results demonstrate that there is a very small increase in proactive based network as they have routes before they are demanded. On the other hand, reactive protocols experience a higher delay as they do not preserve the unused routes, only to search when they are needed. Eventually, this leads to an increase in the delay, as it is highly likely that data packets keep on waiting in buffers until a route is discovered to reach to its destination.

6.3.2 Performance evaluation with varying mobility rate

The presence of node mobility in a mobile ad-hoc network results in a stress to routing protocols due to the frequent link failures and subsequent route discovery cost. In the case of link breakages in the network, some routing protocols are not likely to perform well with different traffic patterns and result in a significant degradation in the network performance. Hence, the mobility factor often plays a vital role in determining the routing performance. The impact of node mobility on routing performance is shown in Figure 6.12. We fix the pause time at 50 sec and gradually increase the node speed from 10, 20 and 30 m/s. The ultimate goal of such experiments is to explore how the protocols scale as the rate of topology changes in the network.

As can be seen in Figure 6.12a, there is a slight decrease in OLSR and AODV throughput with the changes of mobility rates to higher values. This implies that both of the protocols are able to perform well at higher mobility speed in a MANET scenario. On the other hand, TORA and DSR can attain a lower amount of throughput even when the node speed is fixed at 10 m/s. The performance continues to deteriorate with the further increases in the mobility rates (e.g., 20 and 30 m/s). The gradual declines in the throughput suggest that TORA and DSR protocols are not suitable to adapt the dynamics of the wireless channels.
The average end-to-end delays for various protocols are shown in Figure 6.12b. Being a link state protocol, the OLSR provides the shortest path routes, consequently showing the minimum delay characteristics. The OLSR keeps its delay performance almost at a steady level even when the node mobility is increased. On the other hand, the shortest paths are not maintained by reactive protocols and hence more delays are likely to be generated in AODV and DSR based networks. However, the delay value of AODV increases slightly with the increase in the mobility and exhibits a relatively better performance than the DSR. This is attributed to the fact that a fresh route is initiated by ADOV as soon as the old routes are expired. In effect, this could be an ideal protocol in the network that can be implemented when the goal is to maintain a certain performance regardless of the behavior of its nodes. Finally, an extremely high delay is introduced in the TORA network, which is further worsened with higher mobility rates.

6.4 Summary of TCP Performance

The simulation results reveal some important MANET characteristics which have impacted on the TCP performance. The changes in network size and mobility rate to higher values result in a variety of reactions to different TCP versions. In this section we have summarized the performance results of Reno, New Reno and SACK variants within the MANET environment. The performance is studied in terms of download response time, upload response time and retransmission attempts.
6.4.1 Performance evaluation with varying network density

With a large number of nodes, the network is supposed to experience extra high loads and thereby the TCP performance is expected to be dramatically affected. In what follows, the summary results are displayed, followed by a brief discussion for each case.

![Figure 6.13](image)

**Figure 6.13:** Performance of TCP variants for different network sizes; (a) In terms of download response time (b) In terms of upload response time (c) In terms of retransmission attempts.
In Figure 6.13a, different TCP versions under study are analysed in accordance with the download response time. For all the TCP variants, the download response time is found to experience increases when more nodes are added in the network. In a small network (e.g., 30 nodes), all the three versions achieve almost an equal response time. When the 60 nodes are used, however, Reno performs relatively better while the SACK version outperforms the other TCP versions in a large network (e.g., 100 nodes).

Similar to download response time, Figure 6.13b shows that the response time for uploading a file increases when the number of users increases in a MANET environment. However, one can observe that the upload response time of a FTP (50000 bytes) file is always shorter than that of the download response time. In all the scenarios (30, 60 and 100 nodes), the SACK version ensures the lowest upload response time, thereby maintaining its superiority.

From Figure 6.13c, one can notice the highest packet drops in the case of large networks (100 nodes). Hence, all the TCP versions experience the maximum retransmission attempts in such a network. However, the performance of the SACK is found to be quite impressive in a large network as it attains a relatively lower retransmission rate, compared to those of the other variants. On the other hand, in a small and a medium network, the lowest retransmission attempt is experienced by the Reno version.

### 6.4.2 Performance evaluation with varying mobility rate

![Graphs showing the performance evaluation with varying mobility rate](image1)

(a) Download Response Time (sec) vs. Node Speeds (meters/sec)

(b) Upload Response Time (sec) vs. Node Speeds (meters/sec)
Figure 6.14: Performance of TCP variants for different node speeds; (a) In terms of download response time (b) In terms of upload response time (c) In terms of retransmission attempts.

From Figure 6.14a, one can find that the Reno variant is the best choice to be implemented in MANETs, as it achieves the lowest download response time, compared to others. When the node speed is changed from 10 to 20 m/s, the average response time of all the TCP variants, except for the SACK, is found to increase. Surprisingly though, the response time of all the TCP versions decreases when the speed is changed to 30 m/s. This leads one to conclude that the increasing node velocity does not always work as a degradation factor for the TCP performance in a wireless environment. It is true that setting the node speed at 30 m/s increases the probability of frequent changes in the network topology and frequent link breakages. However, it is also likely that it enhances the possibility for the ad-hoc routing protocol to reestablish the link faster [8]. As a result, the download response time at a particular node speed (e.g., 30 m/s) is found to decrease in our results. However, it is not obvious that furthering the node velocity will keep reducing the response time. Instead, it might increase the response time to a higher extent. Therefore, the choice of a right mobility rate within MANET can be considered as an important area of research. When the upload response time against different mobility rates is analyzed (Figure 6.14b), again, the Reno variants are found to be of particular importance as they achieve the lowest response time in all the scenarios. On the other hand, the New Reno is found to have taken relatively more time to upload a file.

Figure 6.14c demonstrates the retransmission attempt of TCP variants under various mobility rates. The Reno version shows an outstanding performance in 10 and 20 m/s scenarios through achieving the lowest retransmission rate. As the node speed is changed to 30 m/s, the performance of the Reno version is found to be slightly degraded. On the other hand, the SACK outperforms the other variants in a 30 m/s scenario since it ensures the lowest retransmission attempt in such a network.
CHAPTER 7
CONCLUSION

7.1 Results

Lately, the technology of mobile ad-hoc networking has received a lot of attention since the wireless networking and the mobile computing devices are now capable of supporting the requirements of this technology. In recent years, a variety of new routing protocols have been developed for MANETs. However, little research on performance evaluation, including any comparative analyses on such protocols is available. On the other hand, TCP optimization in such a network has become a challenging issue because of some unique characteristics of MANETs. Hence, in this dissertation, a thorough understanding of the MANET routing protocols and TCP versions has been gained through conducting several simulation experiments.

This research makes contribution in three areas. Firstly, the study undertakes an analysis towards a comprehensive performance evaluation of four IETF standardized routing protocols in a MANET environment. The considered routing protocols are DSR, AODV, OLSR and TORA, covering a range of design choices, including source routing, hop-by-hop routing, periodic advertisement, and on-demand route discovery. Secondly, the study analyzes the performance of the three most widely used TCP variants (Reno, New Reno and SACK) in an ad-hoc environment. In this respect, an investigation is made into aspects as to how well these variants respond to different network conditions in a MANET environment, with respect to extension of network size and variation of mobility rate. Finally, using the simulation environment, an analysis is carried out on the results of throughput, end-to-end delay, upload response time, download response time and retransmission attempts. These results have facilitated in determining the most suitable routing protocols and TCP variants that can perform more efficiently and robustly in a mobile ad-hoc network.

Each of the protocols and TCP variants studied in this research are found to have performed well in most cases. However, some are associated with certain drawbacks. The key observations of the research are as follows.

The OLSR performs quite well in our simulation. It achieves the highest amount of data packets and the lowest amount of end-to-end delay. It can be highlighted that OLSR performance is not even degraded in the presence of high mobility and larger number of nodes in the network. On the other hand, AODV performs well in a medium and a high node density, with particular reference to a case where end-to-end delays are very critical. However, it is not able to outperform OLSR, either in terms of delay or throughput. The performance of AODV degrades as the node speeds are increased in the network; however, it is not as much extreme as it is found
in other reactive protocols such as DSR. An extremely higher delay is induced in a DSR-based network, which further increases as the number of nodes and mobility rates get higher. In addition, the DSR suffers from achieving a significant throughput as a means of dropping more data packets in such a network. The use of DSR, however, can be restricted to small size and low mobility network. Last but not least in importance, the simulation results reveal that the higher the mobility rates and node sizes, the worse is the performance of TORA in a mobile ad-hoc network. The generation of enormous control traffics as well as the dependence of an underlying protocol such as IMEP makes TORA’s use not very encouraging. In conclusion, OLSR protocol is proved to be a very effective and efficient route discovery protocol for MANETs, which ensures its particular suitability, irrespective of network size and mobility.

The study also reveals some interesting findings on TCP variants when their performances are evaluated over dynamic topologies in MANETs. In a high density network when congestions are very likely, TCP SACK outperforms other variants in terms of retransmission attempts, upload and download response time. The performance of Reno is also noteworthy, which is, however, limited to a small and a medium size network. Meanwhile, with the variations of mobility rates, TCP Reno dominates other congestion control algorithms in most of the cases. However, the performance of SACK is also remarkable in a high mobility scenario. Particularly in terms of retransmission attempts, the SACK variant demonstrates its superiority over other versions. On the other hand, New Reno is found to be the worst performer under higher network stresses and mobility conditions.

7.2 Future research

In this dissertation, a comparative analysis on four MANET routing protocols (viz. OLSR, AODV, DSR and TORA) have been carried out to evaluate their performance, the outcomes of which would be useful in many other situations. However, there are other protocols such as DSDV, ZRP and SSR that can be pursued in any future research. Aside from this, an investigation as to how ad-hoc network performance can be improved, using the cross-layer interactions can also be an important area of future research. The pursuit of future research may also include aspects relating to evaluation of the MANET performance under the higher mobility scenarios that were not included in our present study. This may include a 50 m/s node speed with a pause time of 10 sec, for example. Furthermore, since a MANET is formed without centralized controls, it is posing vulnerable to security attacks now-a-days. Hence, in any future study, such security issues in an ad-hoc network can be pursued.
Appendix A
Model Configuration

**Figure A.1:** Create empty scenario.

**Figure A.2:** Create a MANET scenario with campus scale.

**Figure A.3:** Specify the size and bounded area of a MANET network model.
Figure A.4: An Example of 100 Nodes Simulated Network Model in MANET
Appendix B

Network Configuration Parameters

<table>
<thead>
<tr>
<th>General Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>1000x1000 Square meters</td>
</tr>
<tr>
<td>Network size (No of Nodes)</td>
<td>30, 60, 100</td>
</tr>
<tr>
<td>Data rate</td>
<td>5.5 Mbps</td>
</tr>
<tr>
<td>Mobility Model</td>
<td>Random Way point</td>
</tr>
<tr>
<td>File Size</td>
<td>High Load</td>
</tr>
<tr>
<td>Data type</td>
<td>FTP, HTTP</td>
</tr>
<tr>
<td>Mobility Speed</td>
<td>10 meters/second (m/s)</td>
</tr>
<tr>
<td>Simulation Time</td>
<td>600 Seconds</td>
</tr>
<tr>
<td>Address Mode</td>
<td>IPv4</td>
</tr>
</tbody>
</table>

**Table B.1: General Parameters**

<table>
<thead>
<tr>
<th>Wireless LAN Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wireless LAN Mac Address</td>
<td>Auto Assigned</td>
</tr>
<tr>
<td>BSS Identifier</td>
<td>Auto Assigned</td>
</tr>
<tr>
<td>Physical Characteristics</td>
<td>Direct Sequence</td>
</tr>
<tr>
<td>Data Rate (bps)</td>
<td>5.5 Mbps</td>
</tr>
<tr>
<td>Channel Settings</td>
<td>Auto Assigned</td>
</tr>
<tr>
<td>Transmit Power(W)</td>
<td>0.005</td>
</tr>
<tr>
<td>Packet Reception-Power Threshold (dBm)</td>
<td>-95</td>
</tr>
<tr>
<td>Rts Threshold (bytes)</td>
<td>None</td>
</tr>
<tr>
<td>Fragmentation Threshold (bytes)</td>
<td>1024</td>
</tr>
<tr>
<td>CTS-to-self Option</td>
<td>Enabled</td>
</tr>
<tr>
<td>Short Retry Limit</td>
<td>7</td>
</tr>
<tr>
<td>Long Retry Limit</td>
<td>4</td>
</tr>
<tr>
<td>AP Beacon Interval (seconds)</td>
<td>0.02</td>
</tr>
<tr>
<td>Max Receive Lifetime (seconds)</td>
<td>0.5</td>
</tr>
<tr>
<td>Buffer Size (bits)</td>
<td>256000</td>
</tr>
<tr>
<td>Large Packet Processing</td>
<td>Fragment</td>
</tr>
<tr>
<td>PCF Parameters</td>
<td>Disabled</td>
</tr>
<tr>
<td>HCF Parameters</td>
<td>Not Support</td>
</tr>
</tbody>
</table>

**Table B.2: Wireless LAN Parameters**
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slow start initial count (MSS)</td>
<td>1</td>
</tr>
<tr>
<td>Receive buffer size (bytes)</td>
<td>8,760</td>
</tr>
<tr>
<td>Maximum ACK segment</td>
<td>2</td>
</tr>
<tr>
<td>Duplicate ACK threshold</td>
<td>3</td>
</tr>
<tr>
<td>Initial RTO (seconds)</td>
<td>1.0</td>
</tr>
<tr>
<td>Minimum RTO (seconds)</td>
<td>0.5</td>
</tr>
<tr>
<td>Maximum RTO (seconds)</td>
<td>64</td>
</tr>
<tr>
<td>RTT gain</td>
<td>0.125</td>
</tr>
<tr>
<td>Deviation gain</td>
<td>0.25</td>
</tr>
<tr>
<td>RTT deviation coefficient</td>
<td>4.0</td>
</tr>
</tbody>
</table>

**Table B.3: TCP Parameter**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command Mix (Get/Total)</td>
<td>50%</td>
</tr>
<tr>
<td>Inter-Request Time (Seconds)</td>
<td>Exponential (360)</td>
</tr>
<tr>
<td>File Size (Bytes)</td>
<td>Constant (50000)</td>
</tr>
<tr>
<td>Symbolic Server Name</td>
<td>FTP Server</td>
</tr>
<tr>
<td>Type of Service</td>
<td>Best Effort (0)</td>
</tr>
<tr>
<td>RSVP Parameters</td>
<td>None</td>
</tr>
<tr>
<td>Back0End Custom Application</td>
<td>Not Used</td>
</tr>
</tbody>
</table>

**Table B.5: FTP Application Parameters**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP Specification</td>
<td>HTTP 1.1</td>
</tr>
<tr>
<td>Page Interarrival Time (Seconds)</td>
<td>Exponential (60)</td>
</tr>
<tr>
<td>Page Properties (bytes)</td>
<td>Constant (500), Small Image</td>
</tr>
<tr>
<td>Server Selection</td>
<td>Browse</td>
</tr>
<tr>
<td>RSVP Parameters</td>
<td>None</td>
</tr>
<tr>
<td>Type of Service</td>
<td>Best Effort (0)</td>
</tr>
</tbody>
</table>

**Table B.4: HTTP Application Parameters**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Profile</td>
<td>2 (FTP and HTTP)</td>
</tr>
<tr>
<td>Operation mode</td>
<td>Simultaneous</td>
</tr>
<tr>
<td>Start time (seconds)</td>
<td>Uniform (100,110)</td>
</tr>
<tr>
<td>Duration (seconds)</td>
<td>End of Simulation</td>
</tr>
<tr>
<td>Profile Repeatability</td>
<td>Once at Start Time</td>
</tr>
<tr>
<td>Inter-repetition time (seconds)</td>
<td>Constant (300)</td>
</tr>
<tr>
<td>Number of repetitions</td>
<td>Constant (0)</td>
</tr>
<tr>
<td>Repetition pattern</td>
<td>Serial</td>
</tr>
</tbody>
</table>

**Table B.6: Profile Configuration**
### Application Configuration

<table>
<thead>
<tr>
<th>Application Configuration</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Application</td>
<td>2 (FTP and HTTP)</td>
</tr>
<tr>
<td>Start time offset (seconds)</td>
<td>Constant (5)</td>
</tr>
<tr>
<td>Duration (seconds)</td>
<td>End of Profile</td>
</tr>
<tr>
<td>Application Repeatability</td>
<td>Once at Start Time</td>
</tr>
<tr>
<td>Inter-repetition time (seconds)</td>
<td>Constant (300)</td>
</tr>
<tr>
<td>Number of repetitions</td>
<td>Constant (0)</td>
</tr>
<tr>
<td>Repetition pattern</td>
<td>Serial</td>
</tr>
</tbody>
</table>

**Table B.7: Application Configuration**

### Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Route Discovery Parameters</td>
<td>Default</td>
</tr>
<tr>
<td>Active Route Timeout (seconds)</td>
<td>3</td>
</tr>
<tr>
<td>Hello Interval (seconds)</td>
<td>Uniform (1, 1.1)</td>
</tr>
<tr>
<td>Allowed Hello Loss</td>
<td>2</td>
</tr>
<tr>
<td>Net Diameter</td>
<td>35</td>
</tr>
<tr>
<td>Node Traversal Time (seconds)</td>
<td>0.04</td>
</tr>
<tr>
<td>Route Error Rate Limit (pkts/sec)</td>
<td>10</td>
</tr>
<tr>
<td>Timeout Buffer</td>
<td>2</td>
</tr>
<tr>
<td>Packet Queue Size (packets)</td>
<td>Infinity</td>
</tr>
<tr>
<td>Local Repair</td>
<td>Enabled</td>
</tr>
<tr>
<td>Addressing Mode</td>
<td>IPv4</td>
</tr>
</tbody>
</table>

**Table B.8: AODV Parameters**
### Table B.9: DSR Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Route expiry time (seconds) in Route cache</td>
<td>300</td>
</tr>
<tr>
<td>Expiry Timer (seconds)</td>
<td>30</td>
</tr>
<tr>
<td>Request Table Size (Nodes) in Route Discovery</td>
<td>64</td>
</tr>
<tr>
<td>Max Request Table identifiers in Route Discovery</td>
<td>16</td>
</tr>
<tr>
<td>Max Request Retransmissions in Route Discovery</td>
<td>16</td>
</tr>
<tr>
<td>Max Request Period (seconds)</td>
<td>10</td>
</tr>
<tr>
<td>Initial Request Period (seconds)</td>
<td>0.5</td>
</tr>
<tr>
<td>Non Propagating Request Time (seconds)</td>
<td>0.03</td>
</tr>
<tr>
<td>Gratuitous Route Reply Time (seconds)</td>
<td>1</td>
</tr>
<tr>
<td>Max Buffer Size (packets)</td>
<td>50</td>
</tr>
<tr>
<td>Maintenance Handoff Time (seconds)</td>
<td>0.25</td>
</tr>
<tr>
<td>Max Maintenance Retransmissions (retransmissions)</td>
<td>2</td>
</tr>
<tr>
<td>Maintenance Acknowledgement Time (seconds)</td>
<td>0.5</td>
</tr>
<tr>
<td>Route Replies Using Cached Route</td>
<td>Enabled</td>
</tr>
<tr>
<td>Packet Salvaging</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

### Table B.10: OLSR Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hello Interval (seconds)</td>
<td>2.0</td>
</tr>
<tr>
<td>TC Interval (seconds)</td>
<td>5.0</td>
</tr>
<tr>
<td>Neighbor Hold Time (seconds)</td>
<td>6.0</td>
</tr>
<tr>
<td>Topology Hold Time (seconds)</td>
<td>15.0</td>
</tr>
<tr>
<td>Duplicate Message Hold Time (seconds)</td>
<td>30.0</td>
</tr>
<tr>
<td>Address Mode</td>
<td>IPv4</td>
</tr>
</tbody>
</table>

### Table B.11: TORA Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode of Operation</td>
<td>On-Demand</td>
</tr>
<tr>
<td>OPT Transmit Interval (seconds)</td>
<td>300</td>
</tr>
<tr>
<td>IP Packet Discard Timeout (seconds)</td>
<td>10</td>
</tr>
<tr>
<td>Beacon Period (seconds)</td>
<td>20</td>
</tr>
<tr>
<td>Max Beacon Timer (seconds)</td>
<td>60</td>
</tr>
<tr>
<td>Max Retries (number of attempts)</td>
<td>3</td>
</tr>
<tr>
<td>Max IMEP Packet Length (bytes)</td>
<td>1,500</td>
</tr>
</tbody>
</table>
Reference


