Evaluation of the Impact of Midstream Route Flaps on TCP Performance

By

Emad Hussain Aljarrah

Advisor:

Dr.Mohammad Al-Jarrah

Program: Master of Science in Computer Engineering/

Industrial Systems Engineering

Evaluation of the Impact of Midstream Route Flaps on TCP Performance

By

Emad Hussain Aljarrah

COMMITTEE MEMBERS

Dr. Mohammad A. Al-Jarrah ______________________ Chairman

Associate Professor, Computer Engineering, Yarmouk University

Dr. Ahmad S. Musa ______________________ External Member

Associate Professor, Communication Engineering, Yarmouk University

Dr. Osameh M. Al-Kofahi ______________________ Member

Assistant Professor, Computer Engineering, Yarmouk University

December 2016

Declaration

I am **Emad Aljarrah**, recognize what plagiarism is and I hereby declare that this thesis, which is submitted to the department of **Computer Engineering** at **Hijjawi Faculty for Engineering Technology**, for the partial fulfillment of the requirements for the degree of Master of Science, is my own work. I have not plagiarized from any sources. All references and acknowledgments of sources are given and cited in my thesis. Each significant contribution to and quotation in this report from work of other people has been attributed and referenced.

Name: Emad Hussain Aljarrah

Signature: Emad

Dedication

This thesis is dedicated to my family, my father, my mother, my brothers, and my sisters.

Acknowledgments

First of all, I would like to thank my advisors Dr.Mohammad Al-Jarrah and Dr. Zakaria Al-Qudah, for their endless support and great patience.

I would like also to take this opportunity to express my gratitude to all who gave me the possibility to complete this thesis and present it in this form.

I would like to express my love and gratitude to my beloved family for their understanding and endless love throughout the duration of my studies. Special thanks to my loving parents, who always inspired and guided me throughout my life.

Table OF CONTENTS

List of Figures

List of Tables1

Abbreviations

RTT	Round Trip Time
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
MPTCP	Multipath Transport Control Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
NS2	Network Simulator Version 2
GUI	Graphical User Interface

ABSTRACT

Evaluation of the Impact of Midstream Route Flaps on TCP Performance

Emad Aljarrah, Evaluation of the Impact of Midstream Route Flaps on TCP Performance, the degree is Master of science in Computer Engineering/Industrial Automation Systems, Department of Computer Engineering, Yarmouk University,2016.(Supervisor : Dr. Mohammed Al-Jarrah).

In this thesis, we studied TCP's performance effect of path changes in the middle of a TCP connection. Route changes can happen due to a variety of reasons such as re-routing due to failures, load balancing, and traffic engineering. The heart of TCP is congestion control mechanism that maintains and estimates path characteristics (e.g., available bandwidth, drop ratio, and round-trip time delay) between source-destination pair. After the initialization of TCP session, TCP sender is supposed to build a reasonable estimate on these characteristics and thus transmit data accordingly. Nevertheless, if the path is changed during the lifetime of the connection for whatever reason, the characteristics of the new path may different from the currently used path. In this thesis, we study TCP's behavior in response to a sudden change in path characteristics similar to those that could happen in the Internet including path bandwidth and transmission delay. We first simulated this behavior using NS2 simulations. Later, we utilized FreeBSD-based DummyNet to conduct similar experiments that reflect the behavior of real TCP implementation to sudden changes of the end-to-end path during the lifetime of a TCP session.

We noticed that, the TCP Performance when changing path to a new one with higher bandwidth is not as expected. The TCP protocol takes a very long time to fully utilize the

available bandwidth. Similar behavior is happened when the transmission delay becomes smaller.

CHAPTER 1
INTRODUCTION

1.1 Introduction

Route changes are an inevitable reality of the Internet. Many reasons could cause the end-to-end Internet route change. For instance, devices, links, and software along the path are all subject to failure. While these failures are not expected to be happening with high frequency, Internet protocols are designed to handle failures along a path by rerouting the paths that has been affected by this failure. Miss configuration is another cause of route changes. A miss configured router could unintentionally reroute traffic thousands of miles away from its destination. While failures and miss configurations are issues that network operators attempt to avoid, network operators might introduce other mechanism that could end up of route changes. Examples of these mechanisms include traffic engineering (e.g., to off-load a congested link) [18], load balancing, and orderly decommissioning of devices/links along a network path. These mechanisms typically trigger a path change at a much higher frequency than failure and miss configuration.

Transport Control Protocol (TCP) [1,8,17] builds an estimate of the path characteristics over the lifetime of the connection. For example, TCP maintains estimates the experienced Round Trip Time (RTT) and available bandwidth. Based on these estimates, TCP decides when to transmit data and how much. Surely these decisions affect the overall throughput, jitter and delay of packets transmitted over TCP connections.

When a path change occurs, the characteristics of the old path might be very different from those of the new path. Therefore, the knowledge that a TCP sender assembled for a given connection before a path change occurs might be totally invalid for the path after the change. In fact, this knowledge might be harmful. For example, if a TCP connection is in the congestion avoidance phase[6,16] and a significant amount of extra bandwidth becomes available when the path change occur, TCP may take ramp up slowly to assume the newly available bandwidth due to the additive increase in the transmission rate of a TCP sender in the congestion avoidance phase. Therefore, the TCP sender will waste an opportunity to increase its throughput during this time.

On the other hand, if the path changed to one with smaller amount of available bandwidth or longer RTT, TCP will continue to transmit at the rate that was decided based on the knowledge from the old path. In this case, TCP will encounter a burst of packet drops which will cause the sender to cut down the transmission rate, potentially entering a period of retransmission timeouts [15] leading to slow-starting again. During this period, the TCP sender will inject packets in the networks, which will be eventually dropped, wasting network resources. Furthermore, the performance of the TCP connection itself will suffer.

Due to the prevalence of route changes in the Internet [19], we set out to investigate the performance of TCP when the end-to-end path changes. We studied this performance via NS2 [12] simulations and DummyNet-based network emulation [14].Furthermore, we investigated TCP's performance when a path changes under a wide range of parameter values (ranges of delay and bandwidth).

Changing a link in the path would change overall path properties such as delay or bandwidth. So, what are the implications of change on the performance of transmission using TCP/IP protocol? This is the question we endeavor to answer in this thesis by focusing on TCP behavior for a given path change after the establishment of a connection between source and destination.

1.2 Motivation

As was previously mentioned, in Internet route may change even for an established connection. The route change means that we may have a new path with different parameters such as bandwidth, delay. This change can affect the performance of data transmission. On principle, path changes aim to achieve better performance based on selecting the best path according to predefined metrics. These metrics could be the shortest path, or the minimum time delay between source and destination [3]. Unfortunately, the change of the path may degrade the performance of established connection for TCP protocol because TCP protocol develops its behavior based on gained experience in utilizing the established connection. This thesis studied TCP behavior for path changing after a connection establishment between source and destination.

1.3 Contribution

In this thesis, we explored the effect of path changes on TCP/IP connection with all possible scenarios. This study focuses on understanding the behavior of TCP/IP for the case when the path between sender and receiver has changed a given that the properties of the new paths are different from the old one. The expected result concluded a great impact on TCP/IP protocol performance for real connection properties that change via the Internet. The results recommend at an update to the TCP/IP protocol for the case of path changes.

1.4 The objectives of the study

The main objective of this thesis is to know the impacts of changing packets route path while the data is transmitted using the TCP protocol and to study the behavior of the TCP protocol if the path between sender and receiver is changed in a manner that not perceived by TCP/IP protocol. Specifically, the main objectives are to:

- Study the impacts of changing the link bandwidth between source and destination on transmission performance.
- Study the impacts of changing delay of path on TCP protocol performance.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

2.1 Introduction

This thesis aims to study the effect of route change in the performance of TCP protocol. In this chapter we introduce the tools used to achieve the goal of this thesis which are Network Simulator 2 (NS2) and DummyNet emulator. The NS2 simulator and DummyNet emulator are the most popular tools used for research and test network scenarios. Thus, we used them because they have many features that can help us to study and analyze the networks and protocols and its performance.

In addition, a deep understanding of this thesis is accomplished by presenting a recent related papers and thesis that analyze the performance of network and protocols. Also we summarized the results and merits of these published research.

2.2 TCP Protocol congestion

There are different congestion control and avoidance mechanisms which have been proposed for TCP/IP protocols, namely: Tahoe, Reno, New-Reno, TCP Vegas and SACK. Every mechanism has different features and behavioral in the face of congestion. In this thesis, we utilized new-Reno implementation in NS2 and DummyNet.

TCP New-Reno has been gotten from TCP Reno with slightly modification, when receives multiple duplicate packets, TCP New-Reno enters into fast-retransmit as TCP Reno. This version doesn't exit fast-recovery until all the data which was out standing at the time it entered fast recovery is acknowledged and allows for multiple re-transmissions.

There are two scenarios may occur when ACK is received, the first is If it ACK's all the segments which were outstanding when we entered fast recovery, then it exits fast recovery and sets CWD to ssthresh (Slow Start Threshold) and continues congestion avoidance. The second scenario is if the ACK is a partial ACK, then it sets the number of duplicate ACKS received to zero and it re-transmits that segment and deduces the next segment in line was lost. Figure 1 shows the TCP-Reno Sender State Diagram and Figure 2 shows the TCP-Tahoe Sender State Diagram, the figures show how TCP Reno and Tahoe works[26,27] .

.

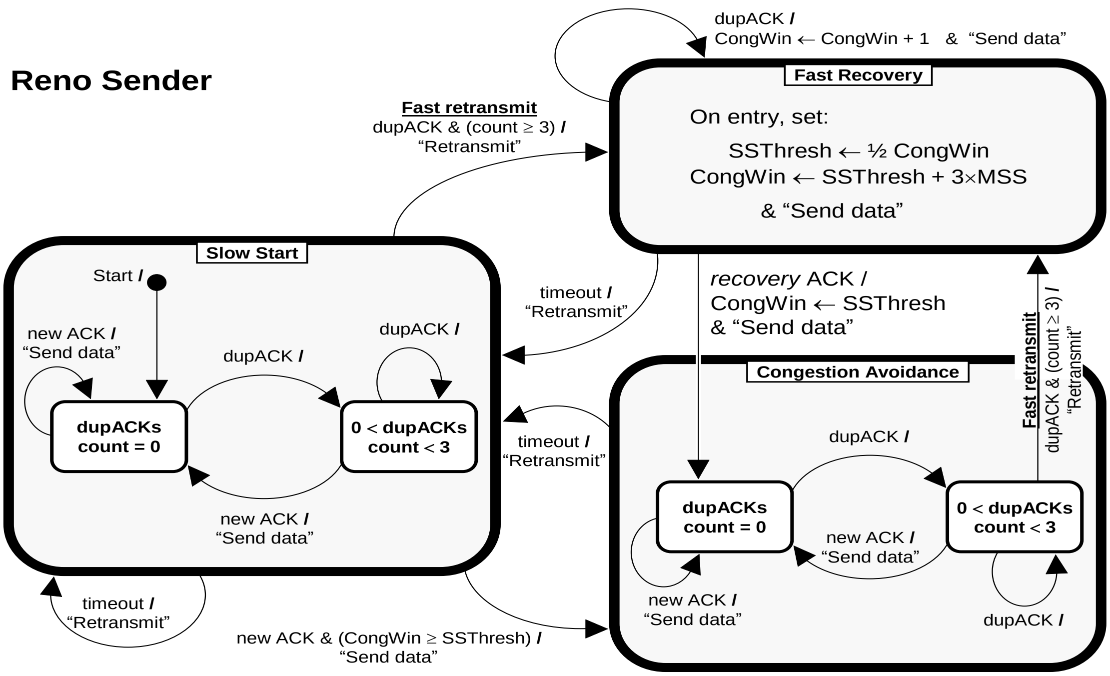


Figure 1: TCP-Reno Sender State Diagram.

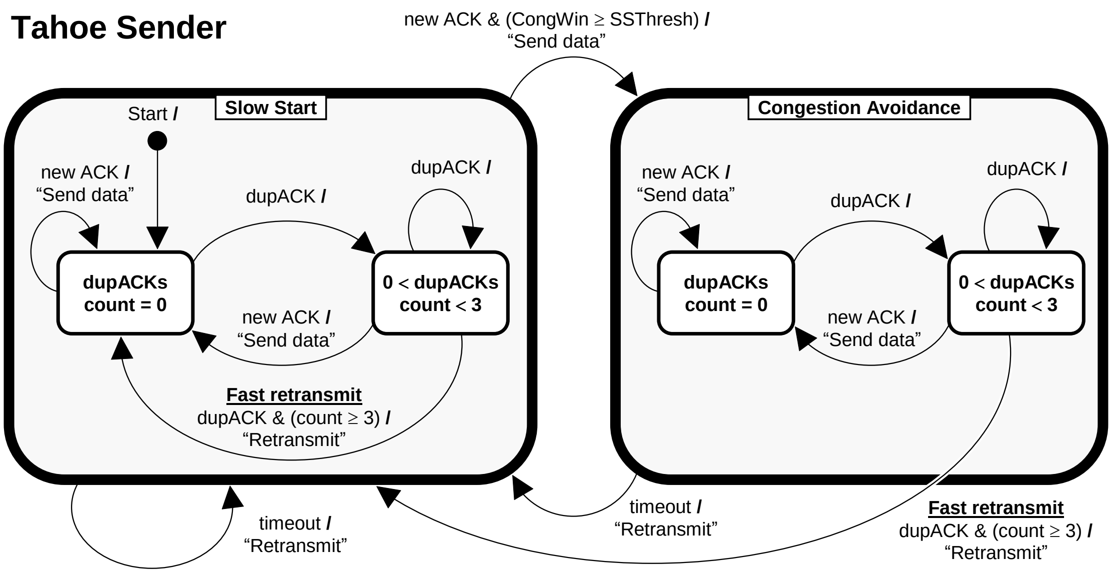


Figure 2: TCP-Tahoe Sender State Diagram.

2.3 Network Simulation 2

Simulator Version 2 (NS2) is the most valuable and utilized tool in the studies of related to networking and the Internet. NS2is amongst the best and most popular open source software for simulation. It has been developed over the years due to its research potential and support for IP-based network simulation such as multicast and unicast routing protocols, different transport protocols (TCP, UDP, RTP, etc..),and most common applications (FTP, Telnet, HTTP)[12].

Network simulator version 2 provides a simple interface and flexible features especially in time control. It provides a schedule time via the times bar that gives a user a full control on time (the time can be controlled up to milliseconds). Figure 3 shows the interface of NS2 and some of these features.

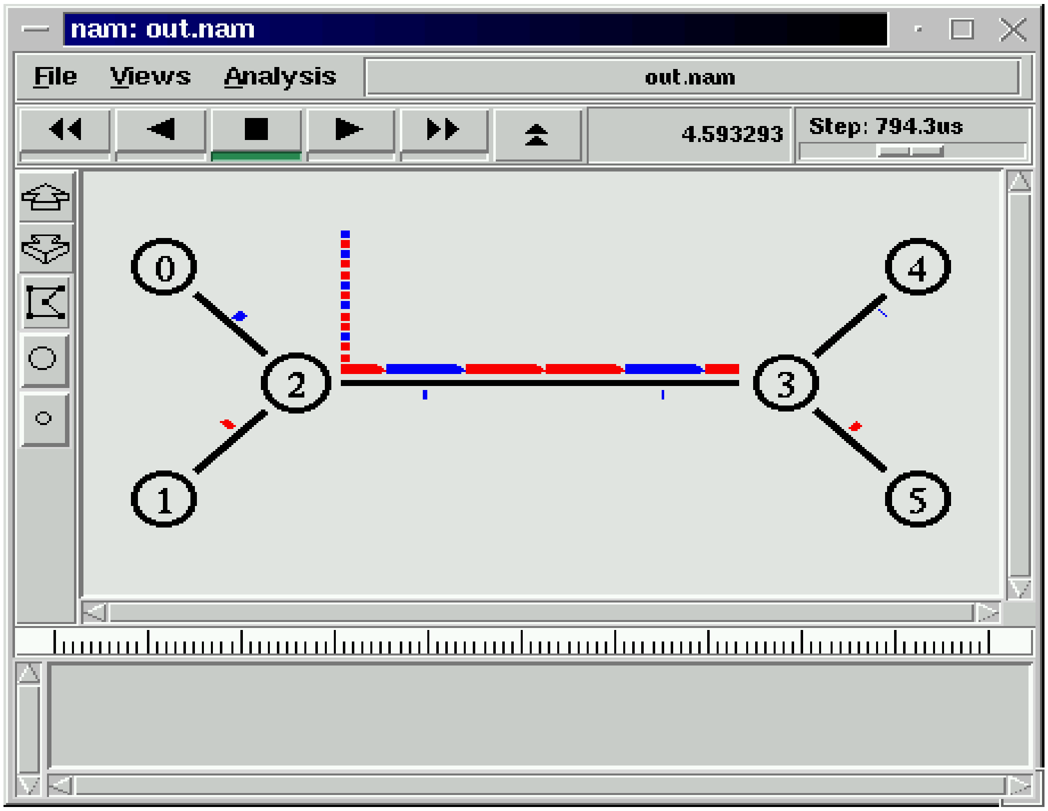


Figure 3: NS2 interface.

Programs or procedures in NS2 can be developed Linux like scripting with tcl extension. In This thesis, we utilized NS3 ver. 2 because this version is the most stable and tested one. Figure 4 shows a sample of scripts that I developed to conduct my experiments. Finally, NS2 collects logs of simulation in out.tr or trace rout file. Trace rout file contains information about the packets such as arrived time, protocol type, sending time or size. Figure 5 shows a screenshot from of trace file of our work.

```
# create nodes

set n0 [$ns node]

set n1 [$ns node]

set r1 [$ns node]

set r2 [$ns node]

#Create links between the nodes

$ns duplex-link $n0 $r1 1Gb 1ms DropTail

$ns duplex-link $r1 $r2 1Mb 100ms DropTail

$ns duplex-link $r2 $n1 1Gb 1ms DropTail

setmyLink [[$ns link $r1 $r2] link]

$ns at 30 "$myLink set delay_ 10ms"

$ns at 60 "$myLink set delay_ 200ms"
```

Figure 4: sample tcl script developed for NS2 simulator.

In this thesis, we developed a java program is used to calculate statistics of conducted experiments such as number of transmission time for each packet. Moreover, experiment scenario statistics is calculated such as throughput. More details are included in Chapter 3.

```
+ 0.5 0 2 tcp 40 ------- 1 0.0 1.0 0 0
- 0.5 0 2 tcp 40 ------- 1 0.0 1.0 0 0
r 0.501 0 2 tcp 40 ------- 1 0.0 1.0 0 0
+ 0.501 2 3 tcp 40 ------- 1 0.0 1.0 0 0
- 0.501 2 3 tcp 40 ------- 1 0.0 1.0 0 0
r 0.60132 2 3 tcp 40 ------- 1 0.0 1.0 0 0
+ 0.60132 3 1 tcp 40 ------- 1 0.0 1.0 0 0
- 0.60132 3 1 tcp 40 ------- 1 0.0 1.0 0 0
r 0.602321 3 1 tcp 40 ------- 1 0.0 1.0 0 0
+ 0.602321 1 3 ack 40 ------- 1 1.0 0.0 0 1
- 0.602321 1 3 ack 40 ------- 1 1.0 0.0 0 1
r 0.603321 1 3 ack 40 ------- 1 1.0 0.0 0 1
+ 0.603321 3 2 ack 40 ------- 1 1.0 0.0 0 1
- 0.603321 3 2 ack 40 ------- 1 1.0 0.0 0 1
r 0.703641 3 2 ack 40 ------- 1 1.0 0.0 0 1
+ 0.703641 2 0 ack 40 ------- 1 1.0 0.0 0 1
- 0.703641 2 0 ack 40 ------- 1 1.0 0.0 0 1
r 0.704641 2 0 ack 40 ------- 1 1.0 0.0 0 1
+ 0.704641 0 2 tcp 1500 ------- 1 0.0 1.0 1 2
- 0.704641 0 2 tcp 1500 ------- 1 0.0 1.0 1 2
+ 0.704641 0 2 tcp 1500 ------- 1 0.0 1.0 2 3
- 0.704653 0 2 tcp 1500 ------- 1 0.0 1.0 2 3
r 0.705653 0 2 tcp 1500 ------- 1 0.0 1.0 1 2
+ 0.705653 2 3 tcp 1500 ------- 1 0.0 1.0 1 2
- 0.705653 2 3 tcp 1500 ------- 1 0.0 1.0 1 2
r 0.705665 0 2 tcp 1500 ------- 1 0.0 1.0 2 3
+ 0.705665 2 3 tcp 1500 ------- 1 0.0 1.0 2 3
- 0.717653 2 3 tcp 1500 ------- 1 0.0 1.0 2 3
r 0.817653 2 3 tcp 1500 ------- 1 0.0 1.0 1 2
+ 0.817653 3 1 tcp 1500 ------- 1 0.0 1.0 1 2
- 0.817653 3 1 tcp 1500 ------- 1 0.0 1.0 1 2
r 0.818665 3 1 tcp 1500 ------- 1 0.0 1.0 1 2
+ 0.818665 1 3 ack 40 ------- 1 1.0 0.0 1 4
```

Figure 5: Trace route file sample.

2.4 DummyNet

DummyNet emulator, is which is an open source program, is one of the most popular emulators used by researchers in the networking filed .It is a flexible tool that can be used to test developed protocols and network architecture. Similar to Ns2, DummyNet logs all information about transmitted packets. These logs can be used to calculate many statistics such as bandwidth utilization, packet delay, packet loss rate, and throughput[14]. Results of DummyNet experiments are closer to reality than network simulation.

DummyNet emulator experiments conducted using three computers. Two computers, which are connected to third one, are considered client computers. The third one, which is the DummyNet server, represents the networks or the Internet. The client computers do not have any special software related to the DummyNet. They just establish connection to DummyNet server. DummyNet emulator is installed in the DummyNet severe. The DummyNet severe acts as Network between the two client computers. Therefore, we can adjust the property of the network through configuring the DummyNet server. To control the bandwidth or any parameters of a connection, the DummyNet creates a pipe that has specific parameters which are determined by the user. Figure 6 shows A DummyNet pipe. Moreover, DummyNet supports many protocols such as TCP, UDP, and IP. Figure 7 depicts a sample script to create a pipe and configure its properties.

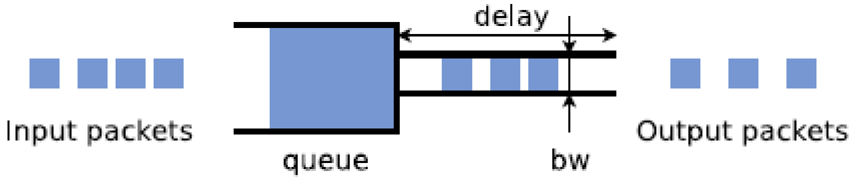


Figure 6: The structure of DummyNet pipe configurable parameters include bandwidth, delay and queue size [25].

```
ipfw add pipe 1 in proto tcp

ipfw add pipe 2 in proto tcp

ipfw pipe 1configbw 20Mbit/s delay 1ms

ipfw pipe 2configbw1Mbit/s  delay 1 ms
```

Figure 7: Sample script to configure DummyNet pipe.

2.5 Literature review

Research in recent years involved the exploration of the parameters that can affect network performance or throughput. In the following items, we summarized the most recent and noteworthy published papers on related topics:

- Mohammed Al-Sarayreh mentioned in his thesis that path ways may be changed (stability of an Internet path),and these changes may affect the prosperities of path characteristics. Also, it affects the performance of network. One of the parameters that may be affected is RTT (Round Trip Time). The RTT may increase or decrease according to new path prosperities. Al-Sarayreh noticed that approximately 50% of path changes lead to increase in the RTT and the other 50% of changes either decrease or do not cause the RTT to fluctuate. This data suggests that path changes do not always alter path properties [19].
- Jain and Kasbekar mentioned in their paper(Buffer bloat: Dark buffers in the Internet.) that TCP has certain features preventing it from becoming too conservative too quickly [2].They concluded that the bottleneck path and its properties are a major portion of the network, which affect the performance of the network and the latency of packets. Latency is a term used to describe the time needed to transmit a packet from source to destination. The connection between the source and destination must be sufficiently filled with the packets in-flight and the number of packets in flight will be increased until it reaches the maximum bottleneck rate. If a router has a large buffer, the number of packets in flight will be increased and the link will accept the extra packet, however, the delay will increase. If the buffer is full or the bottleneck cannot accept extra packets, the new packets will be dropped and must be retransmitted again. The conducted experiments reveal that if the packet in flight reaches the bottleneck bandwidth the delay increases linearly. Figure 8

shows the relationship between throughput and delay with respect to packets in-flight [2].

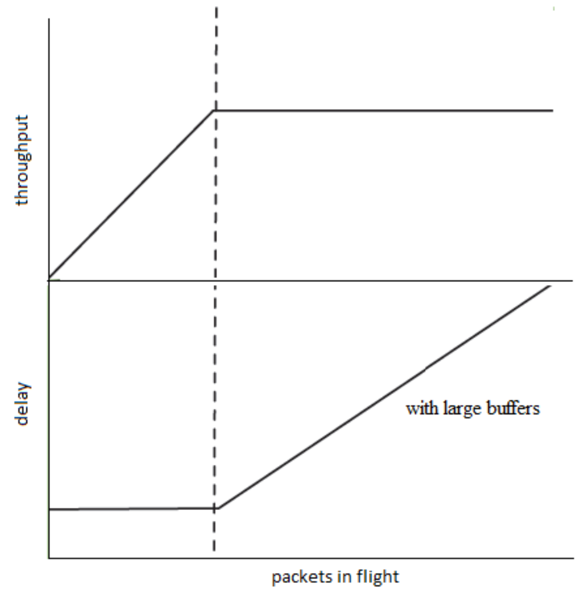


Figure 8: Relation between throughput or delay and packets on flight [2].

- Dukkipati, Refice, Cheng, Chu, Agarwal, Jain, and Sutin in their paper entitled “An Argument for increasing TCP’s initial congestion window, "discussed the performance of the network [4].They concluded that the initial congestion window (cwnd) is a very important parameter in the web because these web transactions are short-lived and the initial congestion window affects data flows and how they finish it. Packet losses over a bottleneck buffer cause an increase in latency by adding extra RTTs and retransmission time outs. A high initial congestion window will reduce the latency, accelerate recovery from losses, allow short transfers to compete fairly with bulk data traffic, and keep up with growth in Web page sizes. If the initial congestion window was increased, the average latency would improve [4]. Figure 9 illustrates the TCP latency with respect to window size as measured by Dukkipatiet. al. [4].

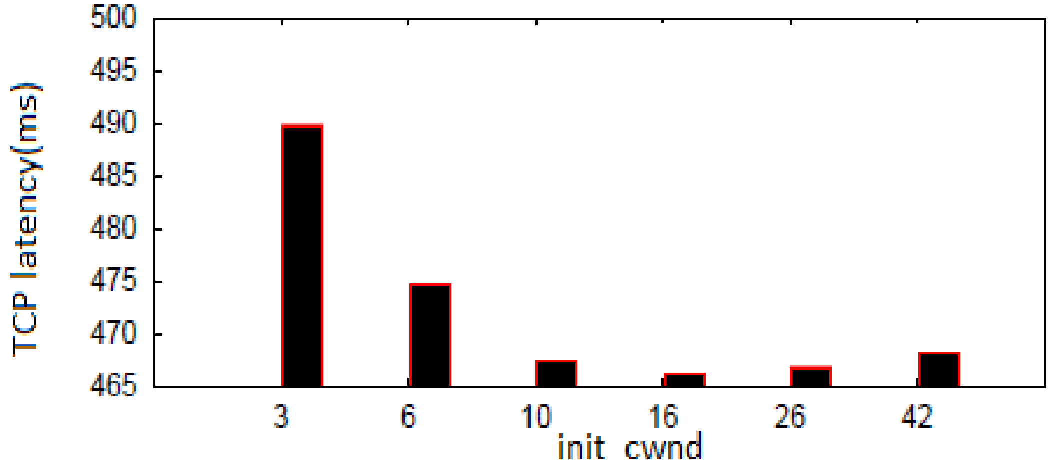


Figure 9: TCP latency for Google search with different initial cwnd values [4].

- Another parameter that may affect general network performance is packet size. Shah, Bhatt and Agarwal, in "Effect of packet-size, over network performance" indicated that the packet size may have a great effect in throughput for a given network. With all other parameters such as bandwidth and delay being fixed, they solely focused on changing the packet size in their experiments and they analyzed the effect of packet size on performance and the overall effect on throughput. Figure7 shows the results of the effect of packet size on throughput [5].They concluded that an increase in packet size will achieve an increase in throughput until specific point at which the throughput will be decreased. This conclusion is drawn from Figure 10 and Table 1.

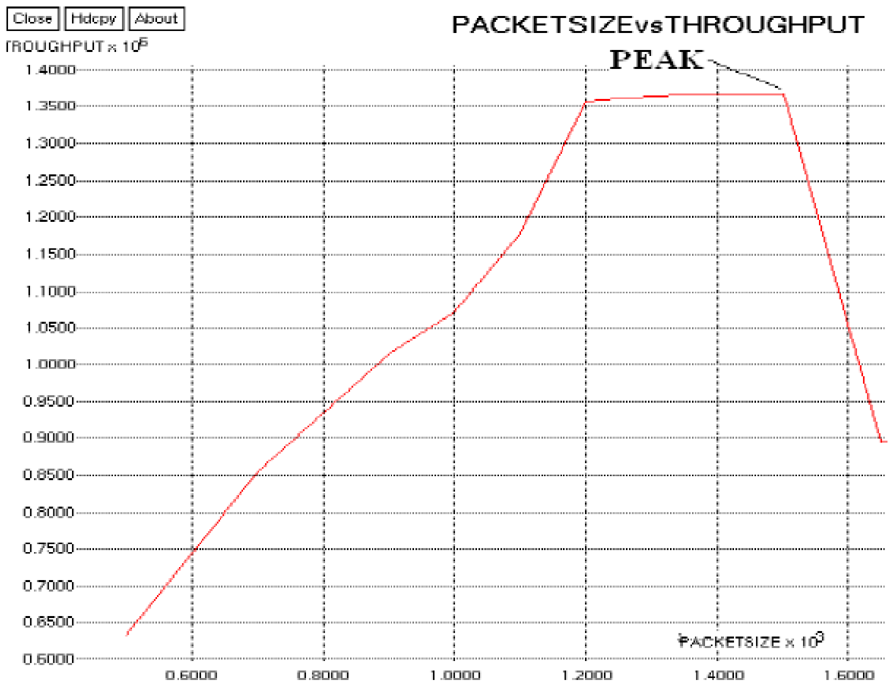


Figure 10: The effect of packet size on throughput [5].

Table 1: Throughput achieved with respect to packets size [5].

SR. NO	PACKET SIZE	AVERAGE THROUGHPUT
1	500	633477
2	700	854804
3	900	1.01499e+06
4	1000	1.07144e+06
5	1100	1.17763e+06
6	1200	1.35717e+06
7	1300	1.36528e+06
8	1400	1.36804e+06
9	1500	1.36804e+06
10	1650	894677

- Shaikh, Fiedler and Collange mentioned in their paper entitled "Quality of experience from user and network perspectives," that throughput is very important parameter in network performance and it was influenced by network properties such as the number of hops and distances between sender and receiver [7].They concluded that access link throughput depends on loss ratio. Therefore, if the loss ratio increases the throughput will decrease and vice versa. It also has an effect on the number of packets sent. Other parameters may affect the performance of the network such as the file size and access link.

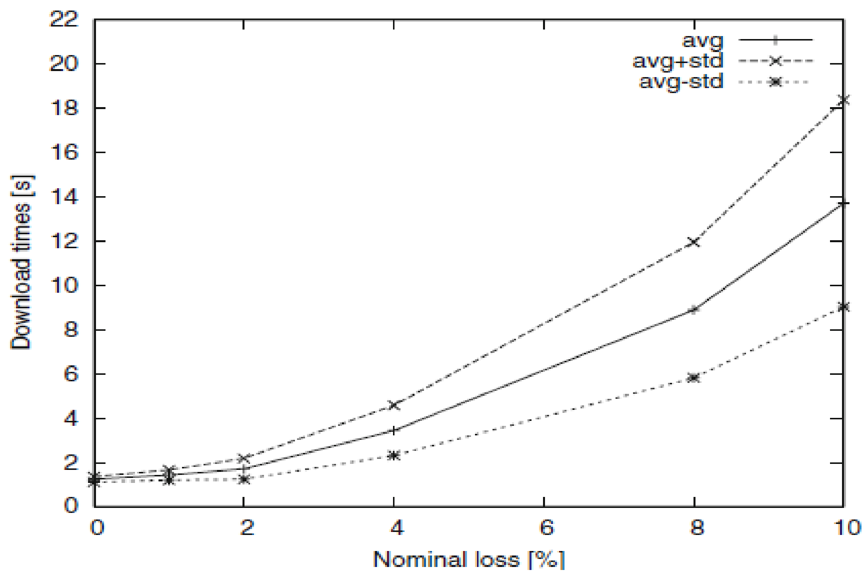


Figure 11: Download time (average±standard deviation) as a function of nominal loss [7].

The experiment conducted by Shaikh*et. al*, is represented in Figure 11, which shows the relation between loss ratio and time to download. If loss increases, the time needed to download will increase and vice versa [7].

- The paper entitled "Impacts of latency on throughput of a corporate computer network" by John, Okonigene, and Adelakun, outlined that there is a linear relationship between packet size, latency, round time trip (RTT), and the throughput. The quantity of users can affect network performance and if the number of user's increases, the throughput will decrease for every user, thus the overall data size along the link will be equal to the capacity bandwidth of the link. For a single user, if the file size increases, the throughput will increase. Hence, if all users could transfer data at equivalent data size, the packet lost rate would be minimal [9].

- Tobias Kretz studied the effect of path and distance on network performance or throughput in his paper entitled "The dynamic distance potential field in a situation with asymmetric bottleneck capacities." He concluded that the ratio of bottleneck usage comes closer to the ratio of widths and the number of hubs or routers can affect the throughput and latency of the network in case of a short bottleneck. Therefore, the shortest path technique is efficient in achieving better throughput and performance [10].

- Islam, Alam, Abdul Hamid, Hong, and Lee mention in their paper that entitled "EFT: A High Throughput Routing Metric for IEEE 802.11s Wireless Mesh Networks," defined queuing delay as the amount of time a packet spends waiting in the transmitter's queue before it is transmitted and propagation delay is the amount of time a packet spends until received on another side. Queuing delay may affect network performance and throughput. Hence, if the transmitter's queue is full, the new incoming packet must wait until one of the packets in the transmitter's queue is transmitted successfully. Therefore, to achieve high throughput and good performance, the packets must be sent through the path that has less propagation (minimum end-to-end delay) and queuing delay. That is good to achieve traffic balance nodes among network nodes [11].

- Dizhi Zhou,Minghui Shi in their paper that entitled "Goodput Improvement for Multipath TCP by Congestion Window Adaptation in Multi-Radio Devices ",that TCP protocol has various scenarios to contend with in terms of packets losses and congestion control. For congestion control, the sender that uses a TCP protocol halves its congestion window when three duplicate ACK (Acknowledgments) are received from receiver to notify the sender that a loss transmission has occurred.

Multipath Transport Control Protocol (MPTCP) has been standardized by the Internet Engineering Task Force (IETF) and, as displayed in Figure 12,in Multipath transmission, every path has its parameters that are different from others paths such as bandwidth or RTT..Etc.

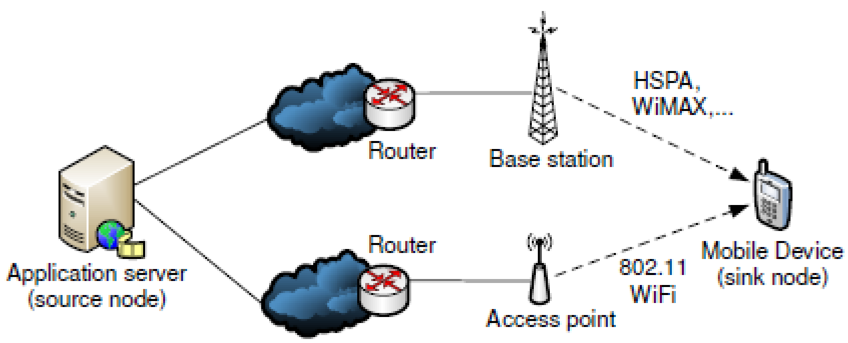
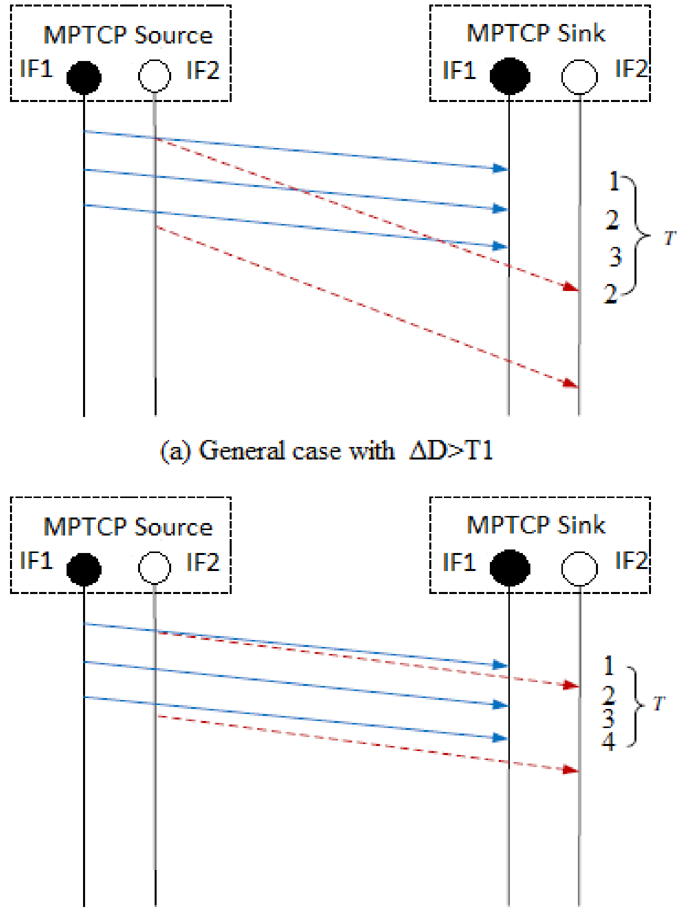


Figure 12: Multipath scenario in wireless networks [20].

For example, the RTT varies for each paths, so there is a high probability of flight for the packets that have lower sequence numbers over to paths that have lower RTT and packets that have higher sequence numbers over to pathways that have a higher RTT, see Figure 13, through this process, the receiver acquire out-of-order packets and returns to sender a duplicate packet to notify it of packet loss. The sender will in turn halve its congestion window and enters fast retransmit and recovery stage. This scenario can affect the TCP performance and the transmission speed [20].



(a) General case with ΔD>T1

(b) Near optimal case with ΔD≤T1

Figure 13: Special cases with two transmission paths [20].

- Jianyong Chen, Cunying Hu, and Zhen Ji said in their paper entitled "Self-Tuning Random Early Detection Algorithm to Improve Performance of Network Transmission" that network congestion is one of the main parameters that may affect network performance and the probability of dropping packets affect transmission data between two nodes.

Another factor affecting network performance is queue length or stability of it in a good dropping probability value. In order to improve performance for a network, we need to develop a queue management policy to keep queue length stable while making a good a tradeoff between high throughput, a third component effecting network performance is a bottleneck path, see figure 14. Bottleneck paths have less bandwidth or delay as compared to other path ways, so its parameters will limit the network performance [21].

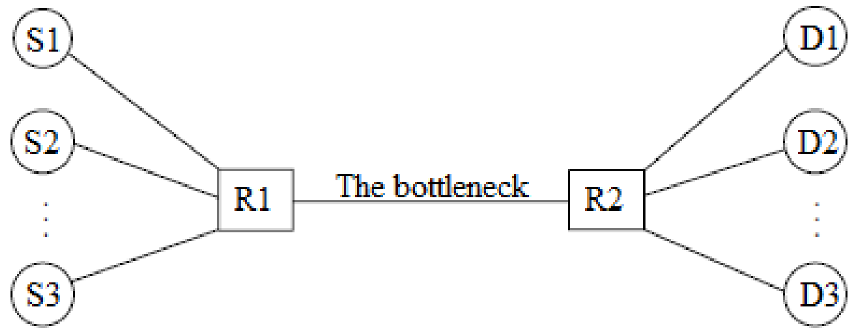


Figure 14: Bottleneck path [21].

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this thesis, we aim to study the effect of path properties change on TCP protocol, to achieve this; we developed a network model that represents the communication between two nodes via the Internet. Then we conduct experiments for transmuting data between these two nodes utilizing the TCP/IP protocol. Each conducted experiment will represent a real communication scenario over the Internet. In each experiment, we focus on one issue such as changing bottleneck bandwidth or transmission delay and monitor the behavior of the TPC/IP protocol and measure the conducted session performance and connection throughput.

The experiments described above will be conducted twice. First, we simulated these experiments using the NS2 (Network Simulator Version 2) simulator. In the second step, we developed a connection between client and server nodes via DummyNet emulation software.

3.2 Network Simulation 2

In the first experiment, we utilized NS2 simulator to analyze the change in path properties in TCP performance. Figure 15 shows an experimental sample setup which will be used. In this experiment, we have two nodes that represent the sender and receiver with two routers and links between them. The sender and receiver are connected to the routers via drop tail link with 1 GB bandwidth and 1 ms delay. The link between routers is the bottleneck link and the properties of this link will change for every experiment scenarios.

The size of the packet that will be sent will also be changed according to the conducted scenarios. But, for one scenario, the size will be the same for whole experiment. Then use log of packets in both routers. The logged file is used to calculate throughput, delay, and drop packet rate. To show the impact of changing the bandwidth or delay, we changed the properties of the bottleneck path between routers.

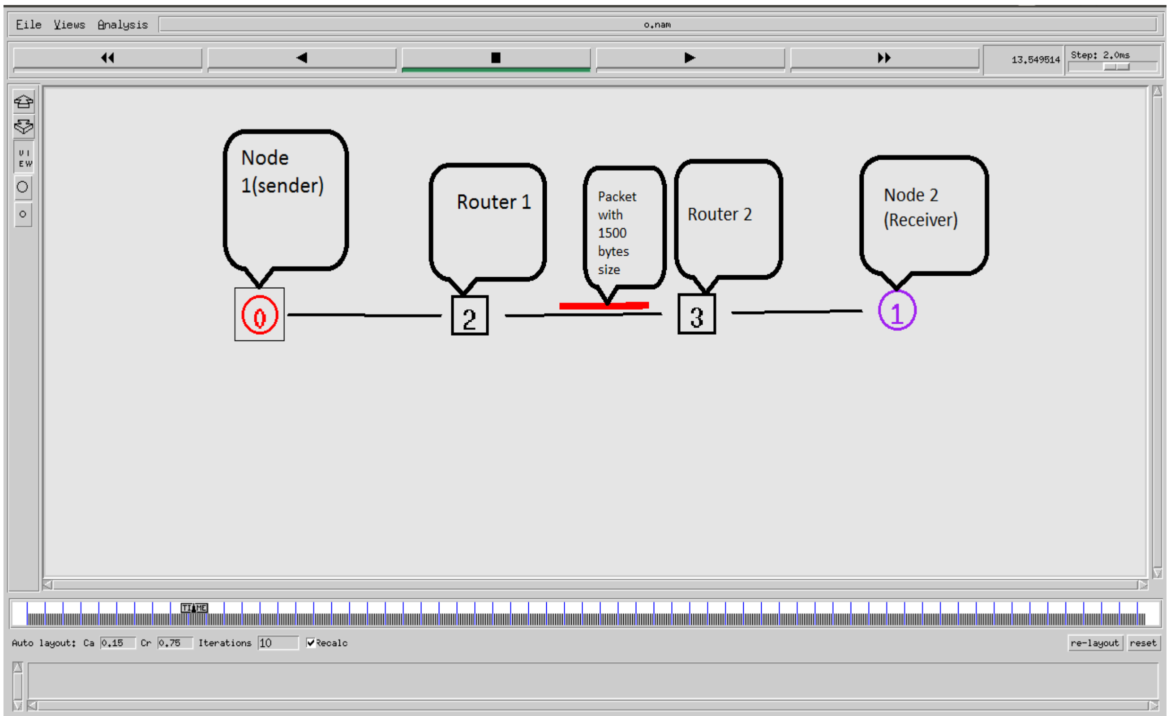


Figure 15: Proposed experiment network structure for NS2.

As an example of experimental scenarios, we changed the bottleneck (the link between the two routers) bandwidth over time period. For the first 21 seconds, the bandwidth is10Mb/s. Then, we change to 1 Mb/s for 21 seconds. Lastly, we change it to 20 Mb/s for the last 21 seconds. Therefore, the experiment period is 65 seconds and every 21 seconds the path bandwidth has been changed. For confirm that all other path properties assured to be the same for the whole experiment. In the second experiment, the property under investigation is the delay. For example, the delay is 70ms for the first 21 seconds of the experiment, and then it changed to 100 ms for 21 more seconds. Finally, in the last 21 seconds it changed to 50ms. The experiment total time is 65 seconds and every 21 seconds the path delay is changed. The bandwidth value will be the same for the whole experiment.

The experiments are configured and conducted using NS2 simulator after building the network model shown in Figure 15. The packets in receiver and the sender router will be logged in a text file. The logged file structure has the header as shown in Figure 14.The log file contains information about the packets that were sent and received with respect to time.

Event	Time	From node	To node	Pkt type	Pkt size	Flags	Fid	Src addr	Dst addr	Seq num	Pkt id

Figure 16: The header of the log file saved by the router [23,13].

As shown in Figure 16, the log file provides the send time and the receive time for transmitted packets. Thus, we can calculate the transmission time for each packet [13]. Moreover, we can calculate the number of dropped packets, the packet loss ratio, and the throughput.

To calculate the throughput over a time period, we count the number of packets and multiply it with the packet size, then; we divide the result be the transmission period. The following equation computed he throughput [14]:

$$Throughput = \frac{Pkt\ Size * Number\ of\ Pkts}{time} \quad (1)$$

Where *pkt* means packet.

3.3 DummyNet

After we completed the NS2 experiments, we repeated them using DummyNet. Initially, we developed an Internet model using the DummyNet emulator. The model includes two client computers connected to server computer running DummyNet emulator. T utilize the TCP protocol in the mentioned setup, we established an ftp session between the two computers to send a file from one to the other. The experiment configuration connection properties are controlled via DummyNet emulator.

The DummyNet emulator is entirely different from the NS2 simulator. It needs a separate PC to install the DummyNet which we named it DummyNet server. Figure 17 shows the experiment connection that used to conduct the designated experiments.

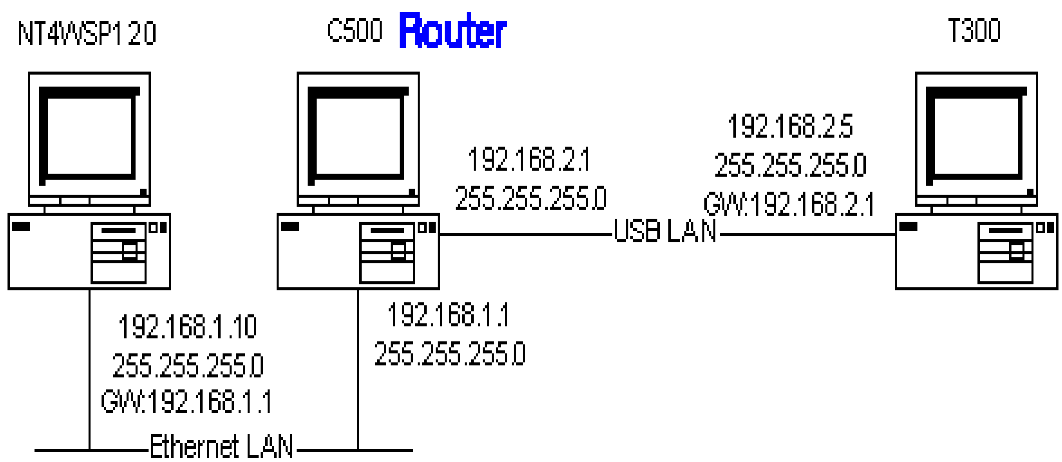


Figure 17: Proposed network structure for DummyNet experiment [22].

As demonstrated in Figure 17, three PC's are connected to carry out the DummyNet experiment scenarios. The left most computer is the sender and the right most is the receiver and the central PC is DummyNet server which has two network interface cards (Ethernet cards). The left side computer (sender) is connected to the first Ethernet interface. The receiver computer is connected to the second Ethernet interface on the DummyNet server. The sender side network is completely different from receiver side one. Therefore the sender cannot communication with the receiver directly. The communication must go through the DummyNet server which will act as a routers and a link. The connection between the PC's are established using twisted pair cable (CAT 5) [24].

DummyNet software has been installed on the DummyNet server, which controls the link properties between the sender and the receiver. Therefore, using the DummyNet emulator, we can configure the network parameters (bandwidth or delay) and change them over certain period of time similar to the NS2 experiment scenarios.

Finally, to computer the outcomes of the conducted experiments, we developed a Java program to calculate the total number of packets and to write a code for the client and server programs utilized in sending and receiving the packets between two sides (left and right). The packet size, bandwidth and delay are identical to that used in NS2 experiment (1500 bytes for every packet) and the same equation is used to calculate the throughput.

CHAPTER 4

EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Introduction

As we mentioned in Chapter 3, our experiments are divided into two stages, in the first stage, we developed a set of communication scenarios utilizing TCP protocol to be simulated using NS2. In the second stage, the same set scenarios are modified and emulated using Dummy Net emulator. In both case, every scenario focuses in one property of path. The studied properties are the bandwidth and the delay.

Our experiments setup includes changing the target property during of transmission session. Moreover, this property is change multiple times. After that, through the received packets, we analyzed the behavior of the TCP protocol with respect to the change on bandwidth or delay.

Next sections explore the results of each scenario conducted utilizing NS2 and DummyNet and the behavior of TCP according to change on the experiment setup.

4.2 Network Simulation 2 Experiments

In this stage, we simulated the target scenario on the NS2 simulator. The experiment setup starts with bandwidth equal to 10Mb/s. after 21 seconds we changed the bandwidth to 1Mb/s, and after 21 seconds the bandwidth is raised to 20 Mb/s. The delay is equal to 50 ms. Figure 18 shows the throughput with respect to time. As expected, for the TCP initialization, the bandwidth is shortly fully utilized. This means that, the windows size for TCP protocol is reached the proper values to utilize the full bandwidth. In the second part where the bandwidth is reduced to 1 Mb/s, TCP protocol is adopt itself to fully utilize the available bandwidth. Finally, during the third stage of the experiment (beginning at 41 seconds), it takes approximately 15 seconds to achieve the steady state and it's noticed that the increase of the bandwidth utilization is linear. Actually, this is the expected behavior of the TCP protocol where after reaching a steady state for the first time, the adaption of the windows to follow changing in the available bandwidth is linear specially if the available bandwidth become much higher the previous one. Thus, the first and second changes require less time to adopt in comparison to the third change.

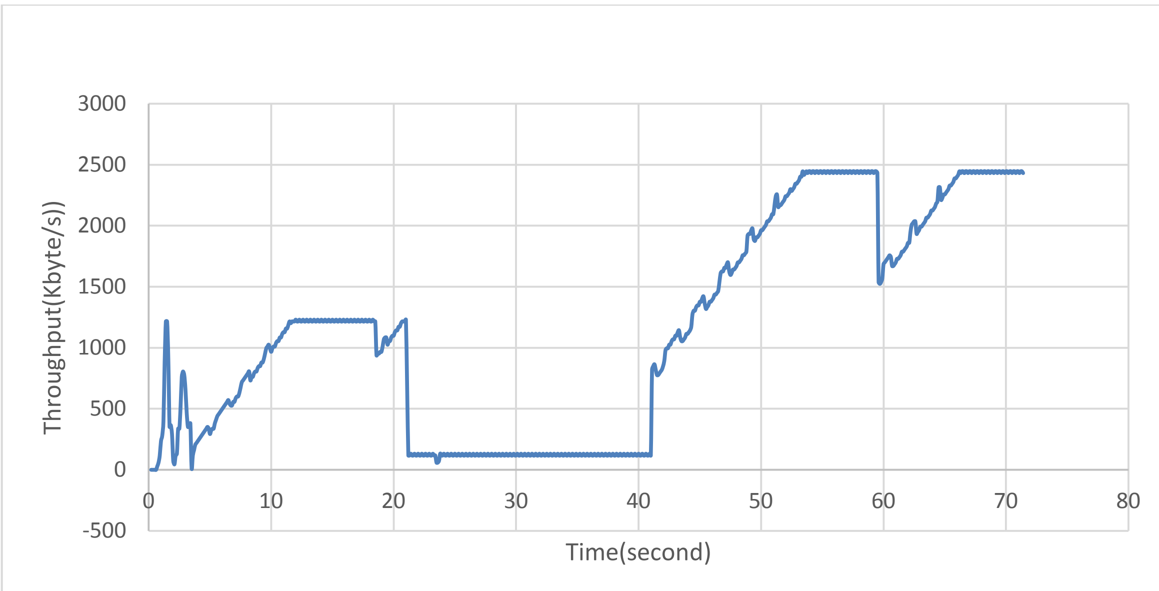


Figure 18: Obtained Throughput for TCP session where the bandwidth connection is changed over time (10-1-20Mb/s with 50ms delay) by NS2 Simulator.

To verify the linear behavior of TCP protocol, we repeated the previous experiment but we changed the bandwidth of the last 21 second to 30Mb/s. the obtained result is shown in Figure 19. Figure 19 confirms the linear behavior of TCP protocol to reach the steady state. The TCP protocol fully utilized the 30Mb/s bandwidth after more than 25 seconds.

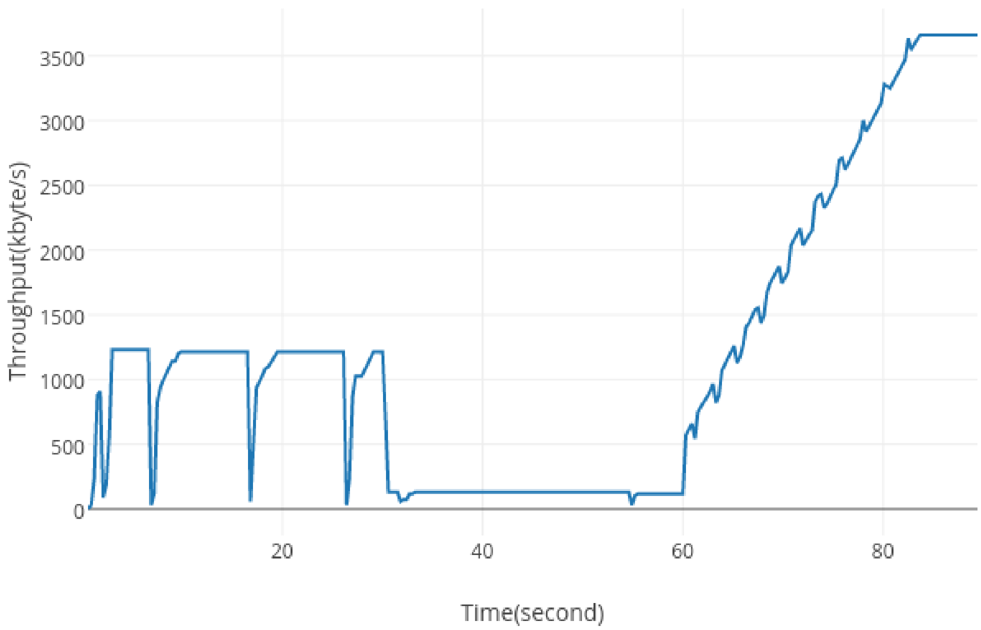


Figure 19: Obtained Throughput for TCP session where the bandwidth connection is changed over time (10-1-30Mb/s with 50ms delay) by NS2 Simulator.

In the second experiment, which is similar to the first one, we set the delay to 100 ms, while the bandwidth is changed similar to experiment one. Figure 20 displays the obtained results of the experiment. As expected, the initialization stage (from time equal 0 to 20 ms), takes longer time than the first experiment. This is due to long round-trip time, which is 200 ms. thus, the TCP window needs more time to reach the proper value for fully utilizing the available bandwidth. In the second stage (bandwidth dropped from 10 Mb/s to 1 Mb/s), the TCP protocol react properly to optimize its window for full bandwidth utilization. Finally, in the third phase, TCP protocol modifies its window linearly trying to benefit from available bandwidth. Unfortunately, TCP window update is linear and every update for

TCP windows require at least one round-trip time (RTT). Therefore, TCP protocol requires very long time to utilize the available bandwidth. Therefore, the experiment time is expanded to 120 seconds. Figure 21 shows the measure throughput with respect to time. This Figure shows that TCP takes more than 30 second to reach full utilization when the bandwidth is 10 Mb/s.

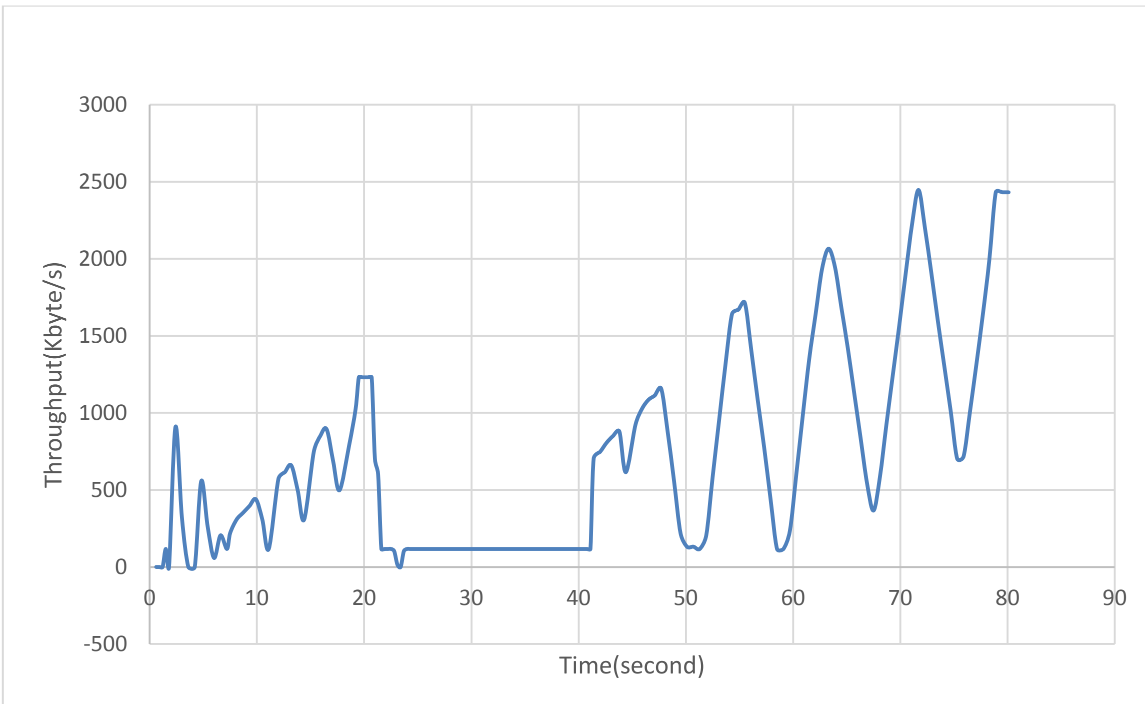


Figure 20: Obtained Throughput for TCP session where the bandwidth connection is changed over time (10-1-20Mb/s with 100ms delay) by NS2 Simulator.

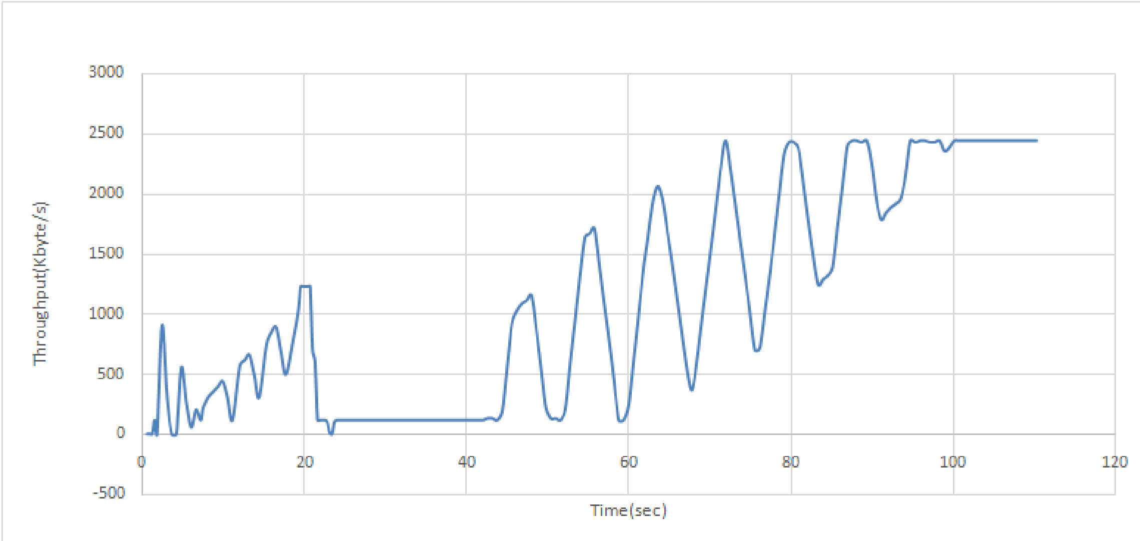


Figure 21: Obtained Throughput for TCP session where the bandwidth connection is changed over time (10-1-20Mb/s with 100ms delay) by NS2 Simulator where experiment time is extended to 120 second.

The third scenario simulated by the NS2 is similar to previous setup but we changed the link delay every 21 seconds. The delay starts with 100ms for 21 seconds, and then we changed it to 10ms. After another 21 seconds we set the delay to 200 ms. the bandwidth 10Mb/s. Figure 22 shows the obtained results. The TCP utilization for the bandwidth is very poor. This is due to the long round trip time (RTT). The increase of bandwidth utilization is linear. In the second first 20 seconds, TCP reached the full link utilization very fast. This is due to the fast acknowledgments due to the small RTT. In the third 20 seconds, the TCP reset because the RTT becomes very long and the sender thought that the sent packet has been lost. To gain the knowledge of the new link capability, TCP needs very long time due to linear change in the TCP window size. The peaks at 51 and does not mean that TCP has fully utilized the link bandwidth. It would be as a result of buffering. To identify the time needed to fully utilize the bandwidth, we extended this experiment to 120 seconds. Figure 22 shows the results for the extended experiment. This Figure shows that the TCP fully gained the knowledge of the link after more than 40 seconds.

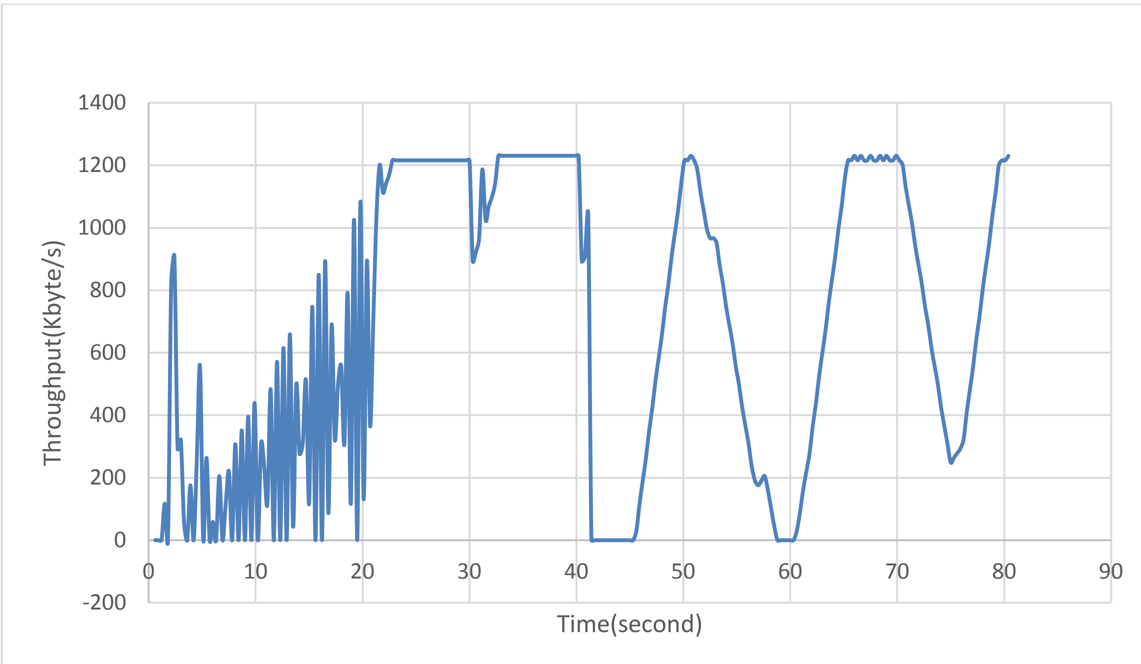


Figure 22: NS2 simulation results of changing delay (100-10-200ms for 10Mb/s bandwidth link).

The final scenario involves delay changes as we did in the previous scenarios; however we used a 20 Mb/s bandwidth instead of 10Mb/s with 100-10-200 milliseconds delay for every 21 second time period. Figure 23 shows the result, similarly to the aforementioned scenarios, TCP does not reached the steady state in the first 21. In the second part of the experiment, TCP continue increasing its window linearly but with higher slop due to the fast RTT. In the third part of the experiment, when the delay changed to 200 ms, TCP protocol reached time out multiple times. Thus the window size is reduced multiple times. In this stage, TCP needs very long time to gain the proper knowledge to fully utilize the valuable bandwidth.

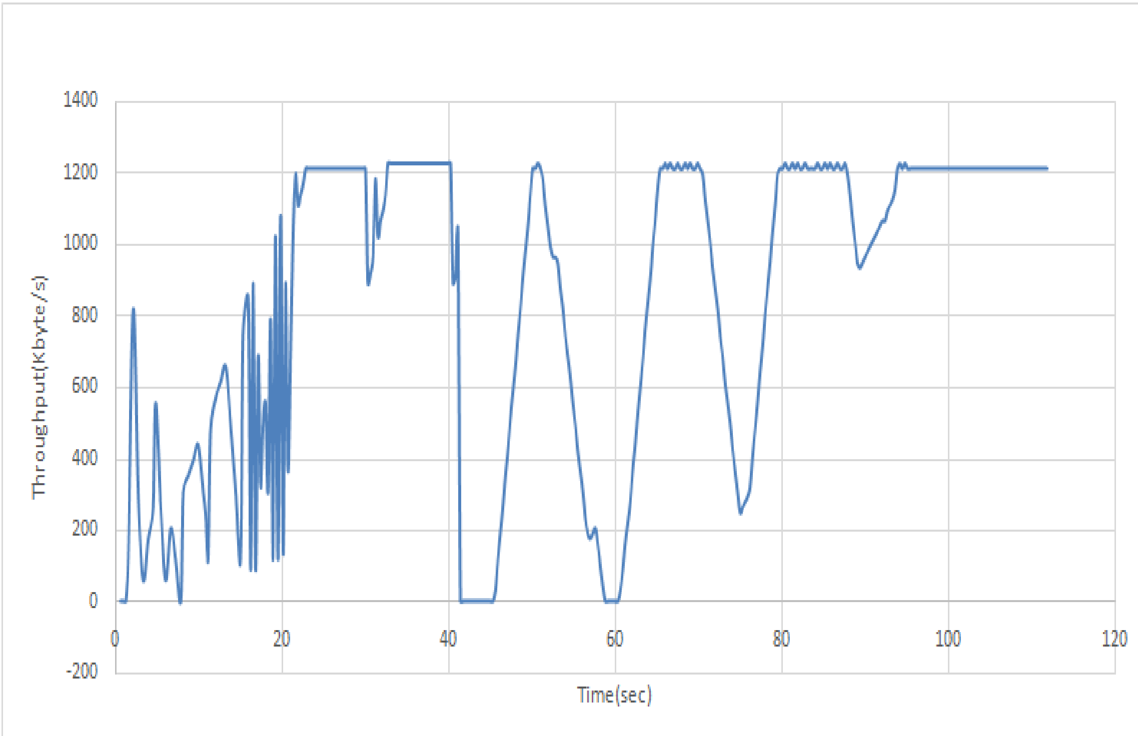


Figure 23: NS2 simulation results of changing delay (100-10-200 ms for 10Mb/s bandwidth link) when experiment time is extended to 120 second.

From the Figure one can see the time that spends to reach the steady state for TCP protocol. Hence, if the delay has been changed to up the time that takes to reach full utilization will be increased and vice versa.

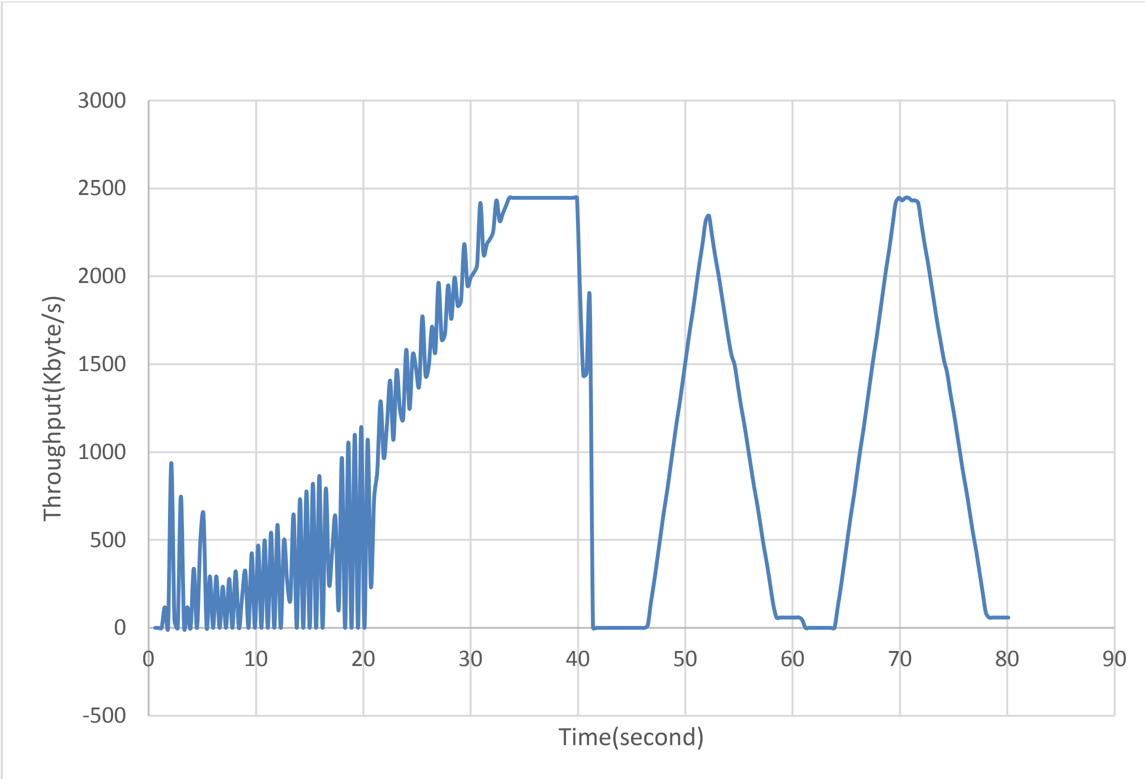


Figure 24: NS2 simulation results of changing delay (100-10-200 ms for 20Mb/s bandwidth link).

4.3 DummyNet Experiments

This section demonstrates the result of experiments conducted using DummyNet emulator. We applied the same scenarios applied to the NS2 simulator. The first of emulates the change of bandwidth during transmission session using TCP protocol. The bandwidths was changed from 10 to 1 and to 20 Mb/s every 21 seconds intervals. The delay was delay of 50 milliseconds for the whole session. The throughput of the conducted experiment is shown in Figure 25.The results from the DummyNet emulator are quite similar to those obtained by NS2 simulator.

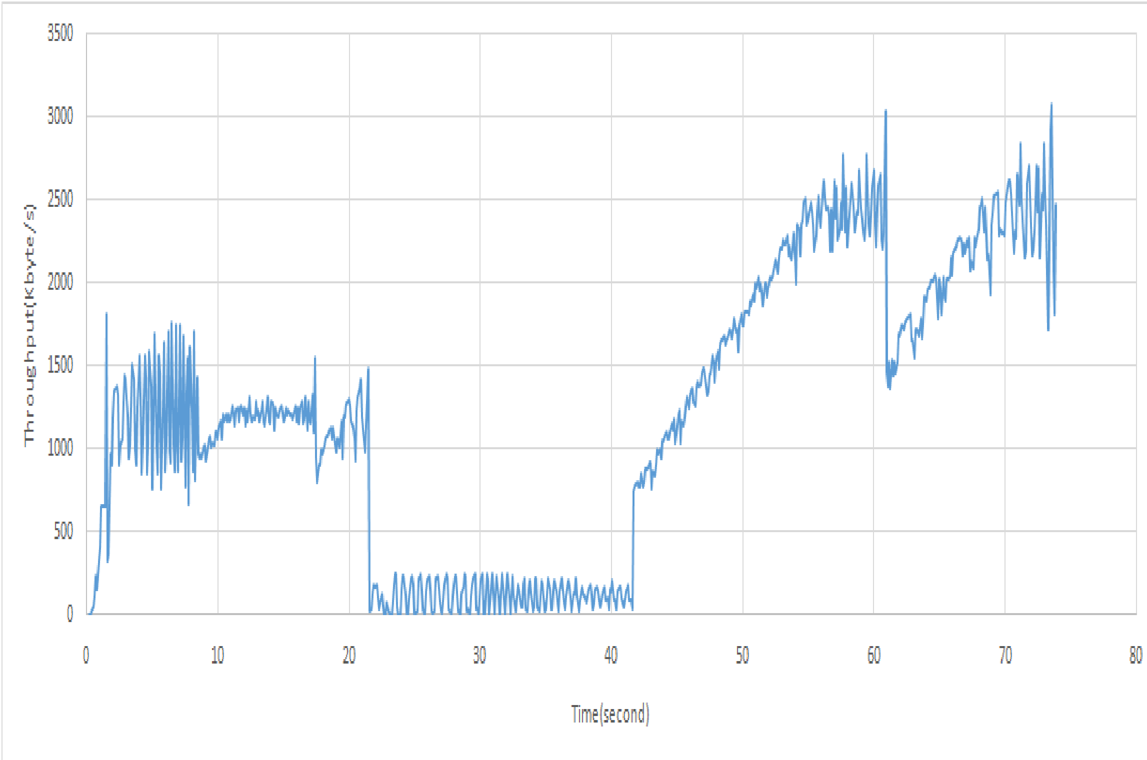


Figure 25: Obtained Throughput for TCP session where the bandwidth connection is changed over time (10-1-20Mb/s with 50ms delay) by DummyNet emulator.

The second scenario involved10-1-20 Mb/s bandwidth change every 21 seconds with a fixed 100 milliseconds delay. This experiment result is depicted in Figure 26.Similar to NS2 results, TCP protocol takes approximately 15 seconds to reach steady state when the bandwidth is changed from 1 Mb/s to 20 Mb/s.

It is very clear from both result of Ns2 and DummyNet that TCP protocol need more time to utilize available bandwidth in the case of changing link bandwidth to a higher value. This is expected because TCP protocol updates its window linearly to the increase of available bandwidth.

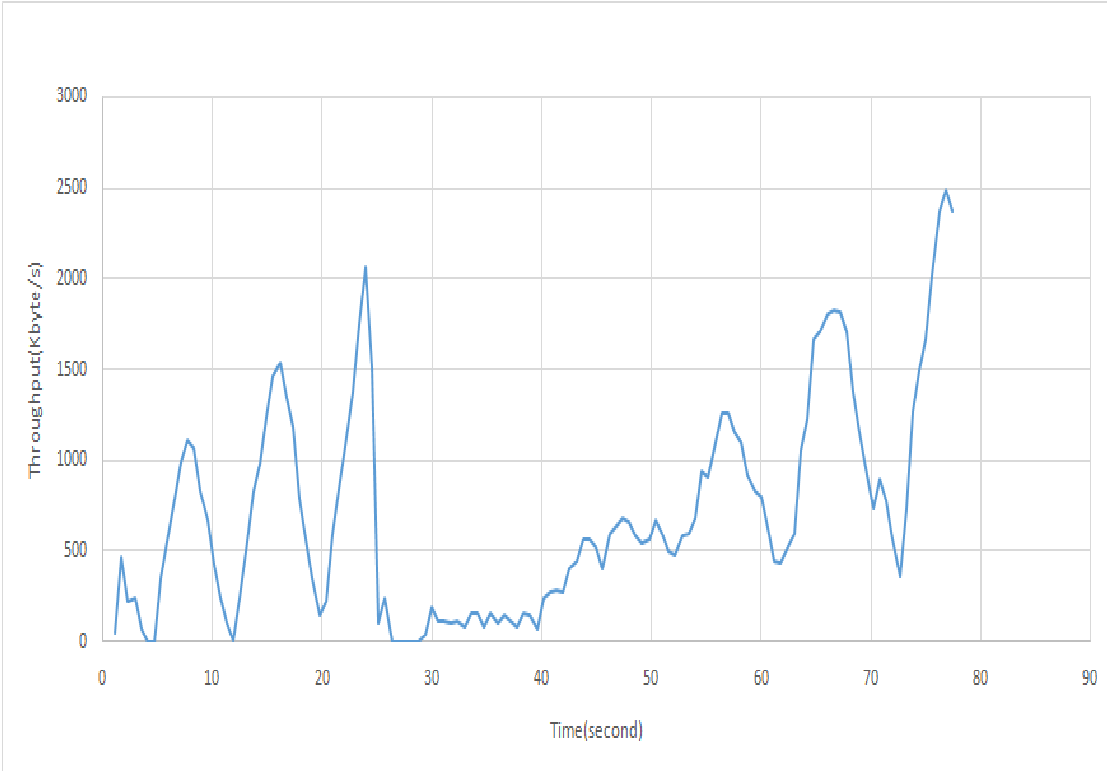


Figure 26: Obtained Throughput for TCP session where the bandwidth connection is changed over time (10-1-20Mb/s with 100ms delay) by DummyNet emulator.

In the second scenario, we conducted experiments to measures delay changing to TCP throughput. First, we configured DummyNet to emulate a change in delay from 100ms to 10ms, and then to 200ms. The change happened every 21 seconds. The bandwidth is10Mb/s. Figure 27 shows the measure throughput with respect to time. As expected, the results are very close to the ones we obtained by NS2 simulator. To reach a steady state and full utilization, the first time interval with 100 milliseconds takes longer time than the second time interval with 10 milliseconds. The third interval with 200 milliseconds took the longest amount of time.

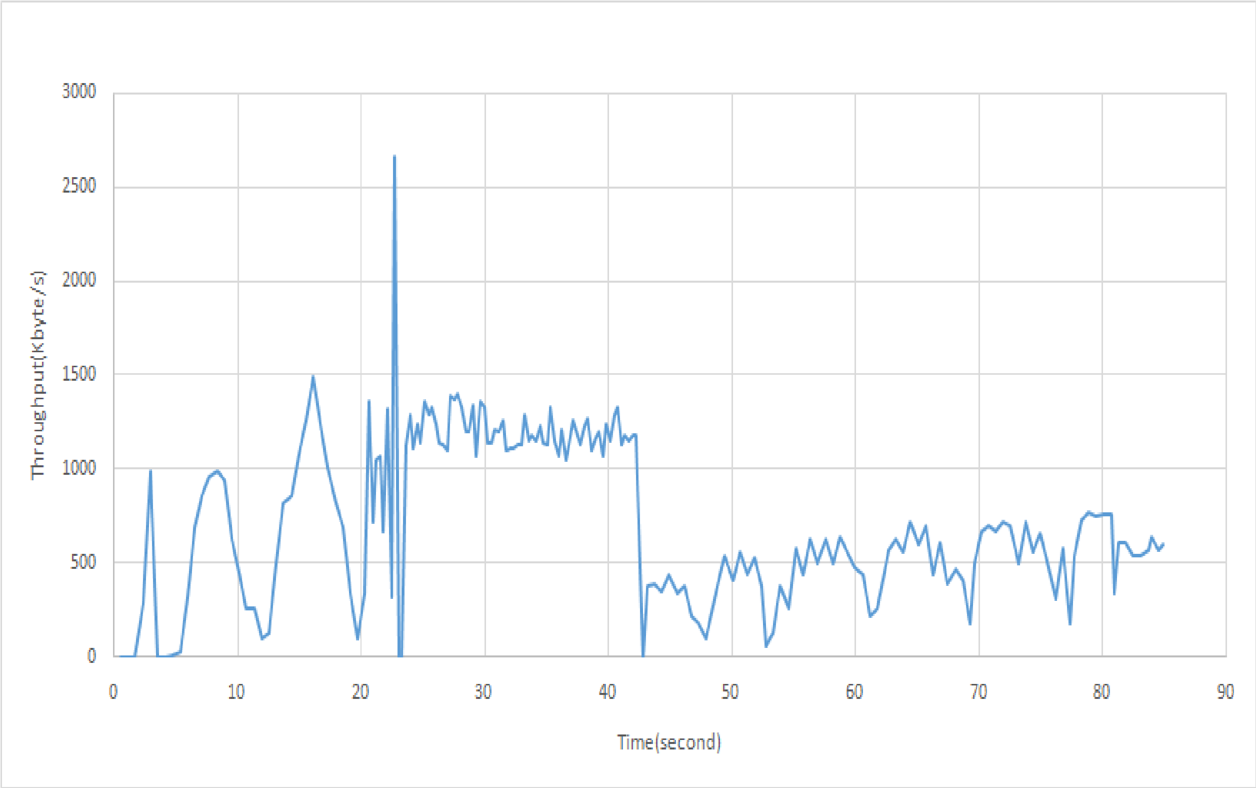


Figure 27: DummyNet emulation results of changing delay (100-10-200 ms for 10Mb/s bandwidth link).

Finally, a 100-10-200 millisecond delay with a fixed bandwidth of 20Mb/s is imposed to mail link of TCP transmission session that emulated by DummyNet. , Figure 28 shows the obtained results. During the first time interval (first 21 seconds), where the delay is 100 milliseconds, throughput increases linearly. In the next 20 second, the delay is changed to 10ms. Therefore, TCP reached steady state or full bandwidth utilization very fast due to the decrease of RTT. The third interval beginning at 40 seconds when delay has been changed to 200 milliseconds, many failures occurred and causes the TCP to reset it window to 1.in this situation, TCP will take very long time to learn path properties.

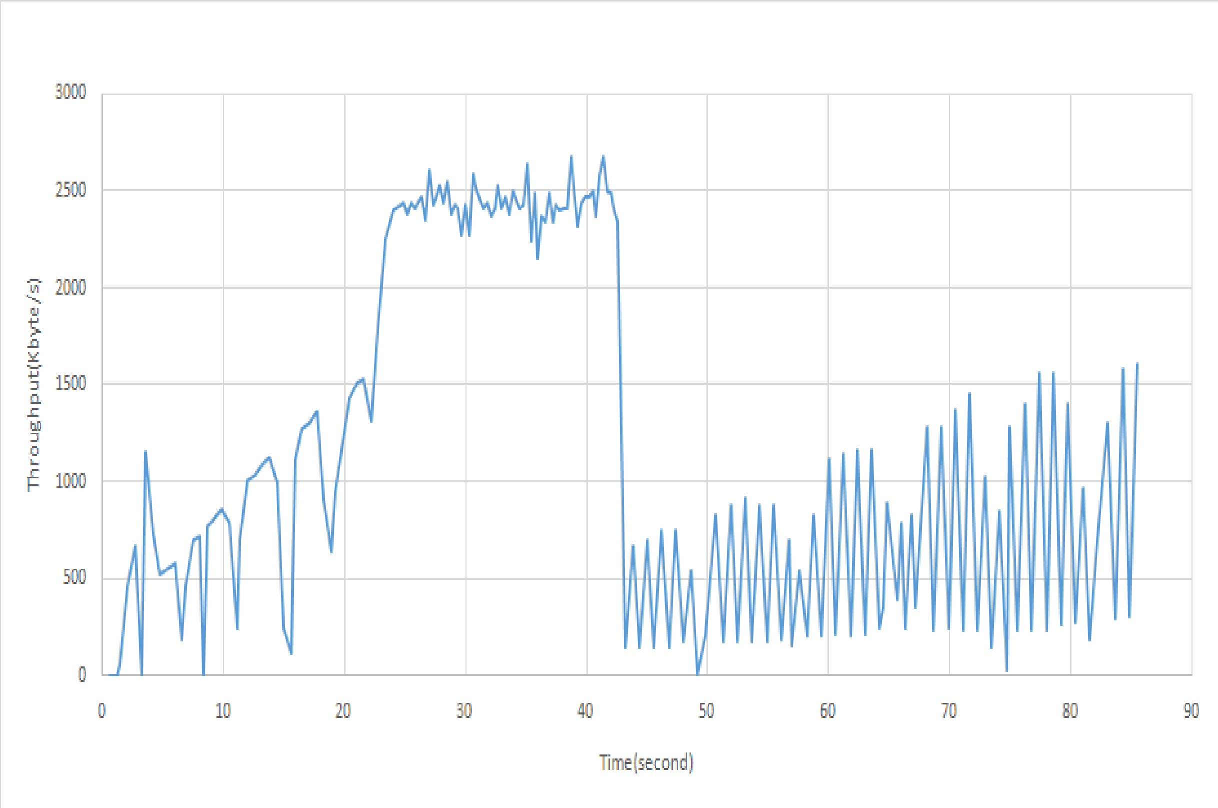


Figure 28: DummyNet emulation results of changing delay (100-10-200 ms for 10Mb/s bandwidth link).

4.4 DummyNet Vs NS2 Experiments

To compare the result obtained from Ns2 with the result obtained from DummyNet, we combined the results in plot. Figure 29 shows the result obtained from NS2 and DummyNet for changing the bandwidth scenario with 50ms delay. The blue line represents the NS2 results and the red one represents the DummyNet results. Both results are very close to each other.

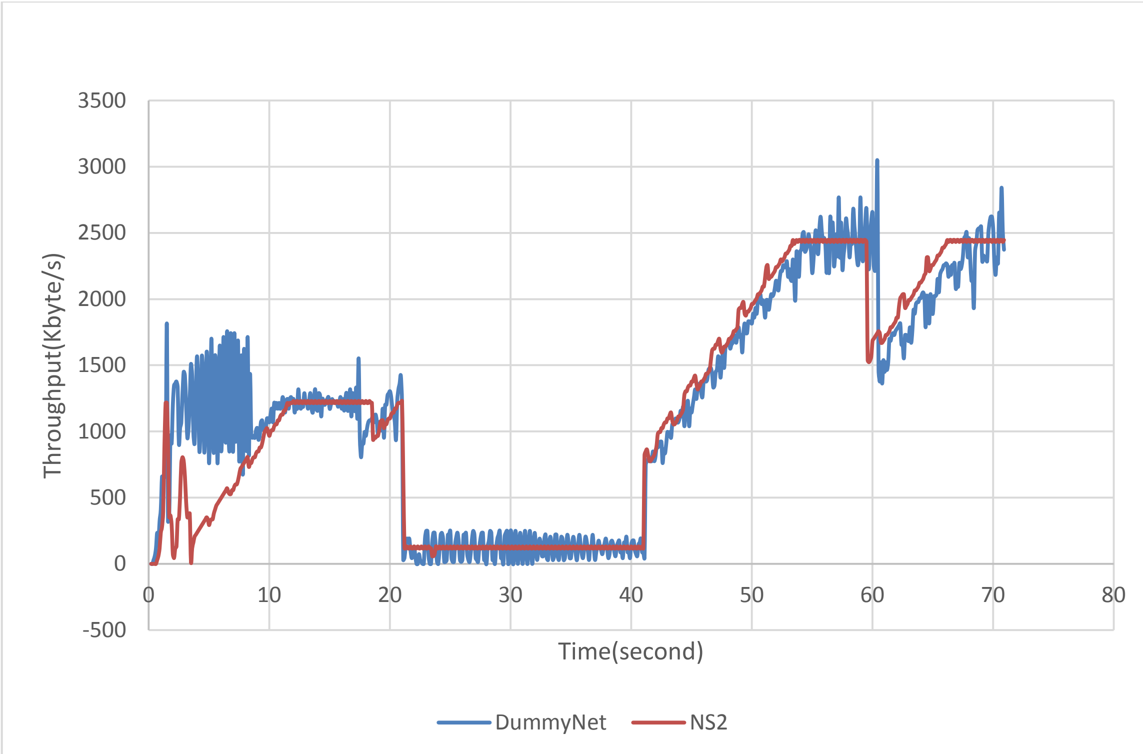


Figure 29: DummyNet Vs NS2 of changing bandwidth (10-1-20Mb/s with 50ms delay).

Figure 30 depicts result obtained by NS2 and DummyNet for changing bandwidth with 100ms delay where blue line represents the NS2 results and the red line represents the DummyNet results. Once more, the obtained result from NS2 and DummyNet are very similar.

Finally, Figures 31 and 32 shows the result obtained from NS2 and DummyNet for changing the delay with 10Mb/s and 20Mb/s respectively. Blue line represents the NS2 results and the red line represents the DummyNet results. The results showed more confirmation of agreement between the NS2 results and the DummyNet results.

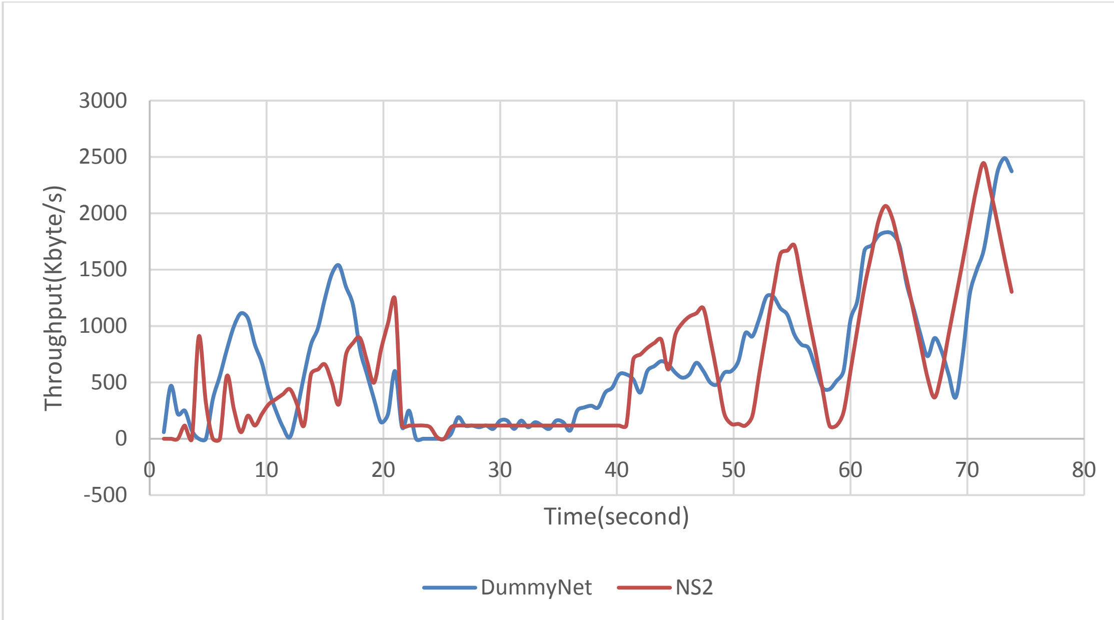


Figure 30: DummyNet Vs NS2 of changing bandwidth (10-1-20Mb/s with 100ms delay).

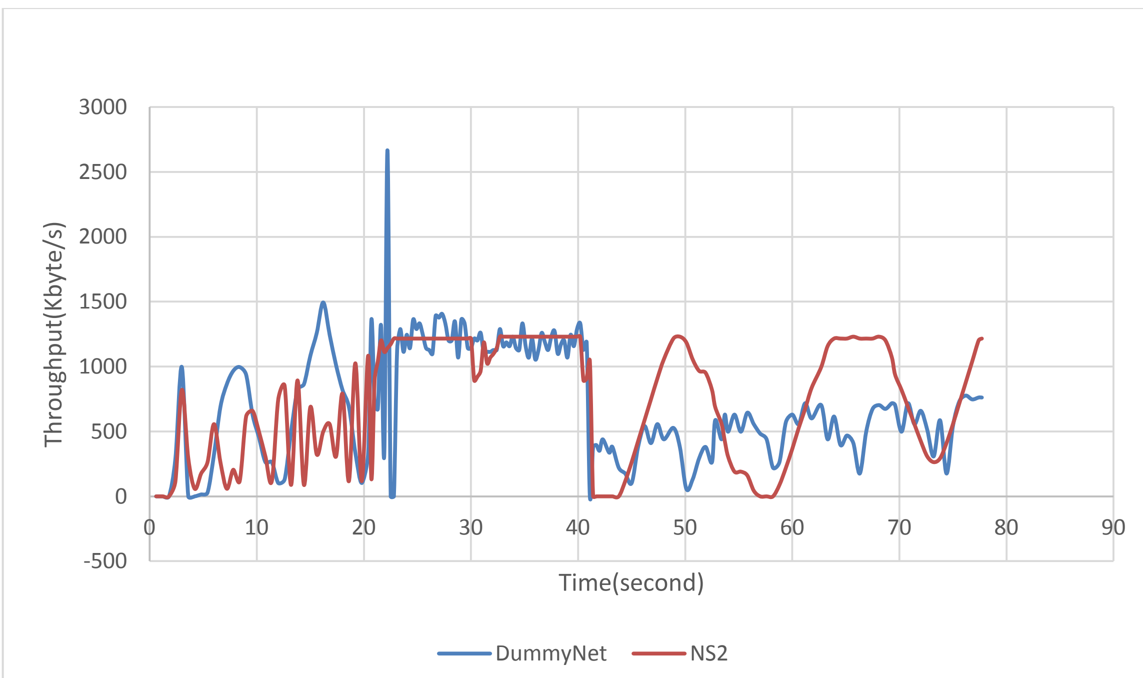


Figure 31: DummyNet Vs NS2 of changing delay (100-10-200ms with 10Mb/s).

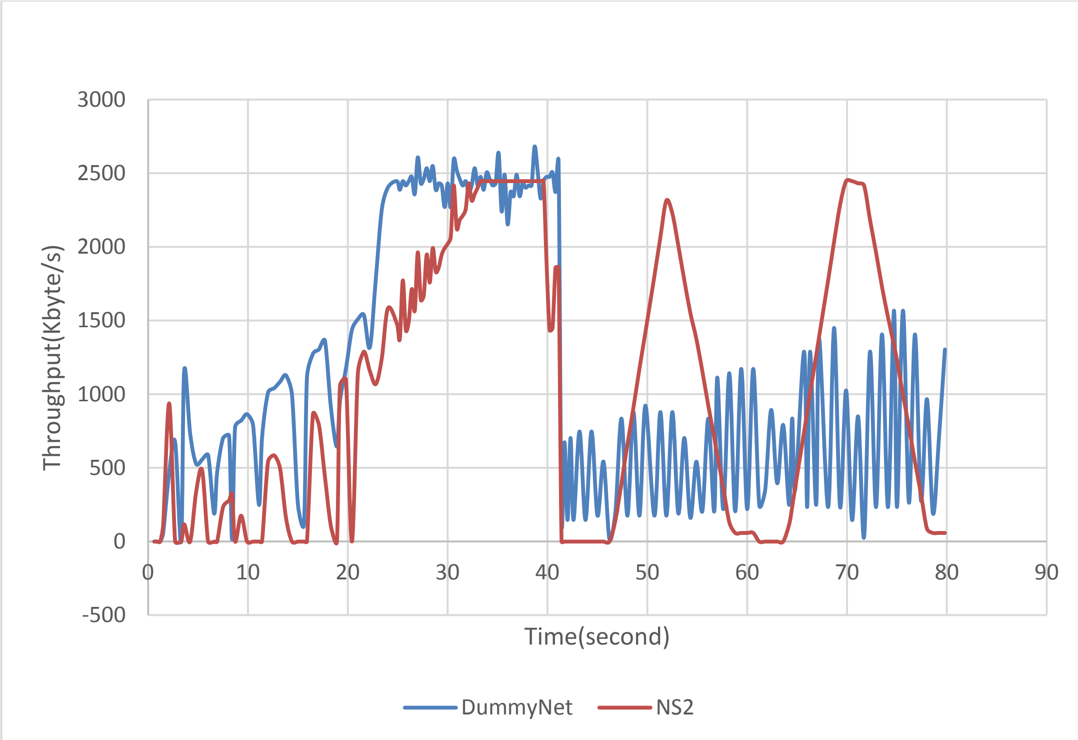


Figure 32: DummyNet Vs NS2 of changing delay (100-10-200ms with 20Mb/s).

CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1 Conclusion

It is reported that Internet paths are significantly less stable than previously reported. Therefore, this thesis evaluates the impact of changing routing path (stability of Internet path) for the TCP protocol. The study includes the change of bandwidth and delay during TCP session. The change includes a prompt change which may be a result of changing the path routes which is transparent to TCP protocol. To analyze the effect of these changes on TCP performance, we utilized the Network simulator 2 (NS2) and DummyNet emulator to apply real scenarios for changing pathways. The experimental results showed that the changes in an Internet path affect TCP performance. This effect leads to a reduction to TCP throughput depending with respect to the expected one considering the path parameters including delay or bandwidth. As a conclusion, our experimental results indicate TCP protocol will not adapt fast to utilize fully the available resources. Thus, we recommend an update in TCP protocol to sense the change and to adapt its parameters more effectively.

5.2 Future work

We intend to focus on TCP performance when the path is changed, as there are many complexities that require further analysis. For example the variability in the packet drop ratio can also be estimated when a path between a source-destination pair changes. Therefore, we suggest the following as future work:

1- To study the total time needed to send a large data from source to destination when path parameters are changed concerning whether the path changes to higher or lower bandwidth.

2- We propose a change in TCP and feedback mechanisms in order to notify TCP source of a path changing. For TCP protocol to react to this change, we suggest to consider modifying policy of congestion.

3- We propose to study the effect of path change in a real system with real session. Through collecting a data logs from the Internet and analyzing it.

4- Finally, to study the effect of complete cutoff of the path between source and destination for a short period of time.

REFERENCE

[1] . Jain, Manish, and Mangesh M. Kasbekar."Preventing TCP from becoming too conservative too quickly." U.S. Patent Application No. 14/011,021, (2013).

[2] . Gettys, Jim, and Kathleen Nichols. "Bufferbloat: Dark buffers in the internet."Queue 9.11 (2011): 40.

[3] . Ling, Xiang, et al. "Global dynamic routing for scale-free networks." Physical Review E 81.1 (2010): 016113.

[4] . Dukkipati, Nandita, et al. "An argument for increasing TCP's initial congestion window." Computer Communication Review 40.3 (2010): 26-33.

[5] . Shah, Abhi U., et al. "Effect of Packet-Size over Network Performance."International Journal of Electronics and Computer Science Engineering, vol 1 (2012): 762-766.

[6] . Ibrahim, Tharwat, GamalAttiya, and Ahmed Hamad. "Fuzzy based Tuning Congestion Window for Improving End-to-End Congestion Control Protocols."International Journal of Computer Applications 87.1 (2014): 1-8.

[7] . Shaikh, Junaid, Markus Fiedler, and Denis Collange. "Quality of Experience from user and network perspectives." annals of telecommunications-annales des télécommunications 65.1-2 (2010): 47-57.

[8] . Roberts, Doug, et al. "Method and apparatus to determine the amount of delay in the transfer of data associated with a TCP zero window event or set of TCP zero window events." U.S. Patent No. 8,849,994. 30 Sep. 2014.

[9] . John, Samuel Ndueso, R. E. Okonigene, and AdebiyiAdelakun. "Impacts of latency on throughput of a corporate computer network." Unpublished (2010).

[10] . Kretz, Tobias. "The dynamic distance potential field in a situation with asymmetric bottleneck capacities." Cellular Automata. Springer Berlin Heidelberg, 2010. 480-488.

[11] . Islam, MdShariful, et al. "EFT: a high throughput routing metric for IEEE 802.11 s wireless mesh networks." annals of telecommunications-annales des télécommunications 65.5-6 (2010): 247-262.

[12] . Canonico, Roberto, Donato Emma, and Giorgio Ventre. "Extended NAM: an NS2 compatible network topology editor for simulation of web caching systems on large network topologies." European Simulation and Modelling Conference (ESMc 2003)-October. 2003.

[13] . Mehta, Vikram, and DrNeena Gupta. "Performance analysis of QoS parameters for WiMAX networks." International Journal of Engineering and Innovative Technology (IJEIT) Volume 1 (2012): 105-110.

[14] . Jurgelionis, Audrius, et al. "An empirical study of netem network emulation functionalities." Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on. IEEE, 2011.

[15] . Afanasyev, Alexander, et al. "Host-to-host congestion control for TCP."Communications Surveys & Tutorials, IEEE 12.3 (2010): 304-342.

[16] . Parvez, Nadim, AnirbanMahanti, and Carey Williamson. "An analytic throughput model for TCP New Reno." IEEE/ACM Transactions on Networking (TON) 18.2 (2010): 448-461.

[17] . Davidson, John. An introduction to TCP/IP. Springer Science & Business Media, 2012.

[18] . Pathak,Abhinav,etal."Latency inflation with MPLS-based traffic

engineering." *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. ACM, 2011.

[19] . Mohammed AL-Sarayreh, “Internet Path Stability: Exploring the Impact of MPLS Deployment,” M.S. thesis, Dept. Computer. Eng., Yarmouk Univ., Irbid, Jordan, 2014.

[20] . Zhou, Dizhi, Wei Song, and Minghui Shi. "Goodput improvement for multipath TCP by congestion window adaptation in multi-radio devices."*Consumer Communications and Networking Conference (CCNC), 2013 IEEE*. IEEE, 2013.

[21] . Chen, Jianyong, Cunying Hu, and Zhen Ji. "Self-tuning random early detection algorithm to improve performance of network transmission."*Mathematical Problems in Engineering* 2011 (2010).

[22] . "Configuring Windows XP As A Network Bridge". WindowsNetworking.com. N.p., 2002.

[23] . "TCL/OTCL". *Mathcs.emory.edu*. N.p., 2016.

[24] . Benton, Mark C., et al. "Gigabit wet mate active cable." U.S. Patent No. 8,792,759. 29 Jul. 2014.

[25] . Carbone, Marta, and Luigi Rizzo. "Dummynet revisited." *ACM SIGCOMM Computer Communication Review* 40.2 (2010): 12-20.

[26] . Marsic, Ivan." *Computer Networks Performance And Quality of Service"*. New Jersey, 2013.

[27] . Henderson, Tom, et al. *The NewReno modification to TCP's fast recovery algorithm*. No. RFC 6582. 2012.

تقييم أثر تغير وسط المسار على الأداء بروتوكول ال TCP

الملخص

في هذه الاطروحه، قمنا بدراسة اثر تغيير المسار في أثناء اتصال بروتوكول الTCP، وتغيير المسار قد تحدث لعدة أسباب منها الفشل في الإرسال أو توزيع الأحمال وغيرها،كما أن برتوكول الTCP يقوم بالتركيز على والتحكم بالازدحام وكيفية توزيع الأحمال ليقوم باستغلال المسار وخصائصه بأفضل طريقه ممكنه، وسنركز في هذه الاطروحه على أثر تغيير خصائص المسار ومنها الوقت المستغرق لنقل حزمه من البيانات والنطاق الترددي بين المرسل والمستقبل وغيرها على أداء البرتوكول.

عند اتصال المرسل والمستقبل والاستعداد لنقل البيانات باستخدام بروتوكول ال TCP، يعمل البرتوكول على نقل البيانات وفقاّ لاستنباط خصائص المسار المتوفر بين المرسل والمستقبل، فمن الممكن تغيير المسار المستخدم أثناء إرسال المعلومات لسبب من الأسباب وبالتالي سوف يستخدم مسار جديد لإرسال المعلومات، والمسار الجديد يكون لديه خصائص مختلفة عن المسار القديم.

قمنا بدراسة اثر تغيير المسار أثناء إرسال المعلومات على بروتوكل الTCP وكيف سيتعامل مع تغيير المسار إلى مسار جديدة ذو خصائص مختلفة عن السابق،وهل سيكون هناك تأثير على الفعالية للإرسال أو جودة الإرسال أم لن يكون هناك تأثير يذكر ،صممنا تجربه تقوم على تغيير خصائص المسار عدة مرات أثناء الإرسال لنقوم بدراسة التأثيرات وسلوكيات البرتوكول وكيفية التعامل معها .

قمنا باستخدام Network Simulator version 2 و DummyNet Emulator لمحاكاة إرسال البيانات باستخدام بروتوكول الTCP وإرسال المعلومات، وأثناء الإرسال قمنا بتغيير خصائص المسار لتتحول إلى مسار جديد ومراقبة النتائج وتحليلها.

وجدنا أن هناك تأثير على جودة وأداء البرتوكول إذا تم تغيير المسار وخصائصه ، كما ان الTCP قد يستغرق وقت طويل حتى يتمكن من استغلال كامل المصادر المتوفرة لديه والعمل على استخدامها في إرسال البيانات للمستقبل .