Evaluation of the Performance of the Simulation of a Campus Network in OPNET

by

Aziz Alshehri

Bachelor of Computer Science, King Abdulaziz University, 2006

A REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Computer science

In the Graduate Academic Unit of Computer Science

Supervisor(s):	Przemyslaw Pochec, PhD, Faculty of Computer Science John DeDourek, PhD, Faculty of Computer Science
Examining Board:	Wei Song, PhD, Faculty of Computer Science, Rod Cooper, PhD, Faculty of Computer Science

This report is accepted by the

Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

January, 2015

©Aziz Alshehri, 2015

Abstract

Simulation is widely used to test the performance of computer networks under different operating conditions. The aim of this report is to evaluate the accuracy of the network simulation using OPNET. Measuring the data transfer rate of an FTP session on the FCS campus network was used as the case study. Two approaches were used to validate the results of simulation. The first approach compared the simulation results with the calculations based on an analytical model. The second approach was based on the comparison of the results obtained from the simulator with the network measurements. The results show that in the case of a simple scenario (three clients connecting to a server) the simulator's network model was close to the actual network performance (the average transfer rate difference between individual flow data rates of 24.7 Mbps) and the analytical results were almost the same as those of the simulation (difference of 0.1 Mbps). In a more complex scenario (up to thirteen clients), the results from the simulator's network model differed from the measurement of the actual network (average transfer rate difference of 404.7 Mbps) and from the analytical calculations (average transfer

rate difference of 326.7 Mbps). Furthermore, this difference was large when measured between the data rates of individual (pairs of) flows, while the difference between the averages of all flows in one time slot between OPNET and Iperf was small.

Dedication

There are a number of people without whom this thesis might not have been written, and to whom I am greatly indebted.

To my mother(Asheeh), who has been a source of encouragement and inspiration to me throughout my life.

To my father (Abdallah), who has supported me all the way since the beginning of my studies.

My wife (Nadiah) I can not thank her enough for her patience, and support. I would not have been able to accomplish this without her in my education.

My children (Raghad, Ahmad and Rima) they are the joy of my life, they are the ones who kept me going through hard times.

To all my brothers and sisters, thank you for your encouragement.

Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisors Professor John DeDourek and Professor Przemyslaw Pochec for their limitless guidance, patience, motivation, enthusiasm, and immense knowledge. My sincere thanks also go to the FCS IT department for their help, especially to Sean Seeley and John Peterson. I would like to thank OPNET for providing me with the license for OPNET Modeler. Finally, I would like to thank the Ministry of Higher Education of Saudi Arabia through the Saudi Arabian Cultural Bureau in Canada for their sponsorship.

Table of Contents

Abstract ii

Dedication iv

Acknowledgments v

Table of Contents ix

List of Tables x

List of Figures xii

Abbreviations xiii

1 Introduction 1

1.1 Introduction 1

1.2 Motivation 2

1.3 Project Description and Objectives 2

1.4 Layout of The Report 3

2 Network Evaluation Tools 5

2.1 OPNET Simulator 5
2.2 Network Topology in OPNET 6
2.3 Gathering statistics for an OPNET Simulation 7
2.4 Support for Modelling in OPNET 7
2.4.1 Electronic mail (Email) 8
2.4.2 Voice 9
2.4.3 Video Conferencing 10
2.4.4 File Transfer Protocol (FTP) 11
2.4.5 Hypertext Transfer Protocol (HTTP) 12
2.5 Iperf 13
2.6 Summary 15

3 Modeling a Campus Network **16**
3.1 The Model 16
3.2 The Links 19

4 Simple Validation Experiment **21**
4.1 Introduction 21
4.2 Analytical Model 22
4.3 Simulator Model in OPNET 23
4.4 Experiment with Iperf 26
4.5 Result and Analysis 26

5 Time of Transfer Experiment **31**

5.1 Introduction 31
5.2 Experiment with Iperf 32
5.3 Analytical Model 32
5.4 Simulator Model in OPNET 33
5.5 Results 33

6 Investigation of Throughput on the Complex Campus Network 36

6.1 Introduction 36
6.2 Analytical Approach 38
6.3 Simulation Model in OPNET 39
6.4 Experiment with Iperf 40
6.5 Results 40
6.6 Background Traffic 48
6.6.1 Generating Background Traffic in OPNET 52
6.6.2 Results and Discussion 53

7 Evaluation of a Video Session on a Campus Network Under Heavy Load 57

7.1 Introduction 57
7.2 Video Application on the Campus Network 58
7.2.1 Implementation 58
7.2.2 Results and Analysis 59

8 Conclusion and Future Work 61

8.1 Future Work 62

Bibliography 64

Vita

List of Tables

4.1 Table for profile parameters 25

4.2 Table for analytical validation 27

4.3 Table for OPNET result 27

4.4 Table for Iperf result 28

5.1 Table for OPNET transfer one file 35

6.1 Transfer rate for 13 flows based on analytical calculation . . . 42

6.2 Transfer rate for 13 flows in Iperf 43

6.3 Transfer rate for 13 flows in OPNET 44

6.4 The lowest throughput in Iperf 50

6.5 Calculation of the background traffic based on the difference between highest total data rate recorded in Iperf and OPNET 51

6.6 Calculation of the background traffic based on the difference between lowest total data rate recorded in Iperf and OPNET . 51

6.7 OPNET with background 55

6.8 Average flows of OPNET with and without background traffic 56

List of Figures

3.1 FCS network topology 17

3.2 Gillin Hall sub-network 18

3.3 ITC sub-network 20

3.4 Servers room 20

4.1 Three computers send file to one server 22

4.2 Three computers send file to one server in OPNET 24

4.3 FTP configuration table 25

4.4 Iperf throughput chart for three flows 28

4.5 OPNET throughput chart for three flows 29

5.1 Time to transfer one File in Iperf 34

5.2 Comparison of throughput between Iperf and OPNET 35

6.1 13 Clients in GH 37

6.2 13 Servers in ITC 38

6.3 FTP pre-configured application 39

6.4 Transfer rate at time interval T5 with 8 flows 45

6.5 Transfer rate at time interval T9 with 13 flows 46

6.6 Comparison of average data rate 48
6.7 Attributes of traffic load 53
6.8 Generated the traffic load 53
6.9 Comparison 13 flows in OPNET at T9 56

7.1 Video configuration table of high resolution 59

List of Symbols, Nomenclature or Abbreviations

FCS	Faculty of Computer Science
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
IPERF	Internet Performance Working Group
OPNET	Optimized Network Engineering Tool
QoS	Quality of Service
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VoIP	Voice over IP

Chapter 1

Introduction

1.1 Introduction

Network simulation is one of the evaluation methodologies for testing the performance of computer networking. Complex networks are modeled by specifying the topology and the behavior of the network nodes. The Faculty of Computer Science (FCS) network on the UNBF campus is an example of a complex network. The Faculty of Computer Science network has a large number of computers and links connecting multiple floors and buildings. Students and staff use computers on the network to run many applications such as Email, file sharing, printer sharing, video conference, and Voice over Internet Protocol (VoIP). Some data intensive applications such as VoIP and video-conferencing, may affect the performance of the network [1].

In this research, the UNBF FCS network is modeled using simulation to determine if the network has the capability to handle the extra traffic caused by video-conferencing. To increase confidence in the results of simulations, validation of the simulation process is performed.

1.2 Motivation

The motivation of this report is to model a complex network and validate these results with the performance of the actual network. There are several approaches to validation. One approach to evaluating simulation accuracy is to attempt to compare simulator-observed performance results with a real machine. Another approach to validate the result of simulation is to model and analyze simple situations where the results are known (or can be calculated) before the simulation is conducted. We began by investigating the performance of a simple scenario with three users on a portion of the FCS campus network followed by the investigation of a more complex scenario with many users.

1.3 Project Description and Objectives

This Master's report demonstrates the simulation of a university campus network using OPNET. To validate these results two approaches were used. The first approach was to run simulations for which the results were known

before the simulation was modeled. The second approach was to attempt to reproduce observed performance results from a real network. We created a scenario for a simple network and validated the results with an actual network; then the same was performed on a more complex network. An OPNET simulator was used to model the campus network and the results were compared with Iperf measurements and with analytical calculations. In the experiments the Iperf tool was used to measure the throughput of the actual network. In the experiments, we generated a number of competing flows that shared the bottleneck between ITC and Gillin Hall. Then we applied the same scenario in OPNET and compared the results. Finally, we used OPNET to investigate quality of service (QoS) for individual video streaming users on a campus network under congested traffic conditions, to test the capability of the network. This Master's report is a follow-up to the Alsaiari's report [2] that investigated the average network traffic load level when deploying new services such as Video Streaming, for the FCS UNB campus. Alsaiari's report tested the network with only the low resolution video, while we tested the network with high resolution video under congested traffic conditions.

1.4 Layout of The Report

Chapter 2 introduces an overview of the network evaluation tools including OPNET and Iperf software. Chapter 3 is about a simple validation

experiment using FTP. Chapter 4 describes the model of the FCS campus network, with descriptions for the Faculty of Computer Science, including the ITC and Gillin Hall subnetworks. Chapter 5 shows the analysis of the time of transfer experiment using an analytical approach, which was then observed and measured using Iperf, and finally simulated in OPNET. Chapter 6 presents investigation of the QoS on the campus network to validate the results between OPNET, Iperf and the analytical model. Chapter 7 includes the evaluation of a video session under a heavy load. Conclusions are presented in Chapter 8.

Chapter 2

Network Evaluation Tools

This chapter describes the tools that were used to evaluate the performance of the campus network at FCS. Iperf and OPNET software are the main tools that were used in this research. Section 2.1 explains in detail OPNET and how it was used to evaluate the performance of a network. Section 2.3 provides an overview of Iperf used to measure the throughput of the network.

2.1 OPNET Simulator

OPNET software is a professional tool used to simulate and analyze various computer networks, from simple networks to complex networks such as a campus networks. It includes a graphical user interface. OPNET can be used for evaluating the performance of newly developed products. OPNET contains a massive library of models that simulate and test network compo-

nents and services. For non-commercial use, the OPNET University Program offers both the IT Guru Academic Edition and OPNET Modeler. Both these products allow researchers to study complex computer networks. IT Guru can be downloaded free of charge, but it has a limitation on the number of events it can simulate (50 million). In contrast, OPNET Modeler has an unlimited number of simulation events [3], and is available for academic use through an extendable six months licence. In this research, we applied for a license and used the full version of the OPNET modeler, OPNET Modeler 17.5.

2.2 Network Topology in OPNET

OPNET has the ability to create the physical/logical topology of a simulated network. OPNET supports seven topologies: star, tree, ring, bus, hybrid, fully-connected mesh and not full mesh. End nodes, such as workstations, may be connected to more than one network for example, when workstations may have multiple links to improve reliability. OPNET has a large library of network models and components; for instance, devices, links, LAN and network cloud. These models are differentiated based on their type; for example, switch, link, router and server. OPNET supports components with various interfaces including Ethernet server, point-to-point protocol (PPP) workstation, IP router with Ethernet, 100BASE-T LAN, Fiber Distributed Data Interface 100BASE-T Ethernet link and many others. Moreover, OP-

NET allows users to create custom models. One of the features for creating topology in OPNET is the Rapid Configuration Tool. This tool creates networking topology with a large number of nodes.

2.3 Gathering statistics for an OPNET Simulation

After creating the topology, the first step before running the simulation is selecting the statistics that we want to observe during the simulation. There are two types of the statistics available in OPNET, global statistics and object statistics. Global statistics describe the behavior of the particular protocol in the whole simulation, while object statistics describe the behavior within a specific object. The simulation is controlled by specifying the amount of time to run the simulation, and the initial seed used by the simulator when generating the network traffic during the simulation.

2.4 Support for Modelling in OPNET

OPNET has various standard applications such as FTP, Email, HTTP, VoIP and Video Conferencing. This section describes the specifications and default values of these applications.

2.4.1 Electronic mail (Email)

OPNET provides configurations for the Email application to exchange messages between an Email server and an Email client. The Email configurations include Low Load, Medium Load, and High Load. Email uses TCP transport protocol as a default setting in OPNET. Client-to-Client Email message exchange is not modeled by OPNET [4]. The Email configuration table has many attributes. such as: send interarrival time, send group size, receive interarrival time, receive group size, Email size, type of service (TOS), etc. The five main attributes are:

1. Send Interarrival Time (seconds), which specifies the time to upload the next Email to the Email server.

2. Send Group Size, which specifies the number of Email messages grouped before transmission.

3. Receive Interarrival Time (seconds), which specifies the time to download the next Email from the Email server.

4. Receive Group Size, which specifies the number of e-mail messages grouped before reception.

5. Email Size (bytes), which specifies the size of a single Email in bytes.

2.4.2 Voice

Some applications, such as FTP, HTP, and Email are not affected by data delay of transfer. On the other hand, other applications, like voice, may be affected by an increase in delay. Voice over IP (VoIP) uses the Internet Protocol (IP)for the delivery of voice communication. VoIP has the potential to replace a regular phone system. When a network provides voice over IP, the network administrator must maintain the quality of service balance. It is necessary to assess the network before adding VoIP and to ensure the network is ready for voice communications. Voice QoS depends on a number of factors in a given network, including delay, jitter, and packet loss [5]. By default, voice applications run over user datagram protocol (UDP) in OPNET. However internally, voice application can run between two clients and there is no need for a server to establish the call. The voice Configuration Table has many attributes and we describe nine main attributes:

1. Silence Length (second) specifies the length of the silence period during voice conversation. There are two sub-attributes incoming silence length and outgoing silence length. These sub-attributes follow the exponential distribution by default.

2. Talk Spurt Length (second) specifies the length of a talk spurt for the incoming and outgoing calls along with the associated distributions.

3. Symbolic Destination Name defines the symbolic destination name of the client.

4. Encoder Scheme specifies the algorithm used for encoding voice. OPNET supports 30 different encoding schemes.

5. Voice Frame per Packet specifies the number of voice frames that can be sent in a single packet into an application layer before the voice data is sent to the lower layers.

6. Signaling specifies the method used for establishing and tearing down a voice call, which can be H.323 or session initiation protocol (SIP).

7. Compression Delay (seconds) specifies the delay to compress a voice packet.

8. Decompression Delay (seconds) specifies the delay to decompress a voice packet.

9. Conversation Environment Specifies the quality of the environment in both clients [4].

2.4.3 Video Conferencing

The video conferencing application transmits data between two nodes in the network. This application runs over UDP transport protocol by default. The user can modify the video conferencing application to run over transmission control protocol (TCP) but it causes more delay and requires more connection management.

The video conferencing application has several attributes. The two primary

attributes are frame interarrival time information and frame size information [4]:

1. Frame interarrival time information characterizes two traffic flows: incoming and outgoing traffic. Both incoming and outgoing traffic specify the frequency of video frame transmissions. Incoming stream interarrival time (seconds) sets the time between two consecutive frames in the incoming stream. In contrast, the outgoing stream defines the time between two consecutive video frames in the outgoing stream.

2. Frame size information (byte) consists of two sub-attributes for incoming and outgoing stream frames in units of bytes. The value of this attribute is defined as a pixel. For instance, when the frame size is 128* 120 pixels and each pixel requires 9 bits, then 128*120*9 bits =138,240 bits which is equal to 17,280 bytes.

2.4.4 File Transfer Protocol (FTP)

OPNET simulates only the two most important FTP operations for data transfer: put and get. The task that is associated with a "put" operation is up-loading a FTP file into the server that supports the FTP application and the "get" operation is the opposite of a "put" operation, which occurs when the client downloads an FTP file from the FTP server. An FTP application in OPNET only transfers one file at a time, and by default it runs over TCP [6]. The FTP put operation and FTP get operation have two messages that

are called control and data. When the operation is in FTP get mode (which means download of a file, in this case) the control message is a request for a file. In contrast, when the operation is in FTP put mode, the control message requests acknowledgments of file transfer. Data messages carry a file that is being transferred between the client and server, and the size of the data messages is specified by the user, while the control message size is always 512 bytes [4]. The FTP application model has several attributes. Here, we only explain three primary attributes. These attributes include command mix (get/total), inter-request time and file size.

1. Command Mix (Get/Total) defines the percentage of get operation and the total number of FTP operations. For instance, when the value of this attribute is set to 60%, then 60% of all FTP operations will be get operations, while ratio of the put operations will be 40%.

2. Inter-Request Time (Second) defines the amount of time between sequence FTP operations in seconds. When the previous FTP operation starts the value of this attribute will start to count the time for next file to transfer.

3. File Size (byte) determines the size of the file in bytes.

2.4.5 Hypertext Transfer Protocol (HTTP)

OPNETs HTTP application simulates a web browser that enables a client to contact the web server to view web pages. Each web page contains text, im-

ages, data files and multiple inline objects. HTTP uses TCP as its transport protocol. The client sends an HTTP request for a web page and the server responds to the request by sending the web page to the client [4]. There are two versions of HTTP that preconfigured: HTTP 1.0, and HTTP 1.1. There are four main attributes that describe the properties of the HTTP application:

1. HTTP Specification defines the HTTP protocol used by the web browser. This attribute has five sub-attributes including HTTP Version , Max Connections, Max Idle Period Number of Pipelined Request and Request Size (byte).

2. Page Interarrival Time (seconds) specifies the time between any HTTP operations that are in consecutive sequence.

3. Page Properties describes the attributes of the web page. This attribute has information that specifies the size of the base HTML and the size of inline objects.

4. Server Selection specifies whether the web pages with embedded web page links will contact the same server or another one [4].

2.5 Iperf

Iperf is an open source tool (i.e, Iperf is not a part of the OPNET simulator) used for the evaluation of both TCP and UDP traffic performance on an

actual network. It is written in C and it was developed by the Distributed Applications Support Team (DAST) at the National Laboratory for Applied Network Research (NLANR). Iperf works based on a client and server model. By default, the Iperf client connects to the Iperf server on the TCP port 5001 and measures the bandwidth from the client to the server. The user can run Iperf over UDP tests by using the -u argument butand it can also measure bandwidth being sent from the server back to the client in a dual test. Iperf can run multiple threads to simultaneously send and receive data. Iperf reports various statistical measurements including throughput, delay, jitter and datagram loss. Typical Iperf output contains a timestamped report of the amount of data transferred [6]. Iperf is a useful tool that can be used to troubleshoot congestion and bottlenecks between various computers on the network and it is also a valuable tool for network administrators. Iperf can be installed very easily on any operating system such as UNIX, Linux or Microsoft Windows. One host must be set as client, and the other one as server [7].

2.6 Summary

Some advantages of two network evaluation tools that discussed in this chapter: OPNET and Iperf. OPNET is the simulation tool that can model many different networks on a computer. Iperf is used to test the actual network. Using both tools will allow comparison of the results from the actual network with the OPNET simulator.

Chapter 3

Modeling a Campus Network

This chapter presents modeling of the topology a campus network in OPNET. This model is based on the FCS local area network.

3.1 The Model

In this report, a campus network of the University of New Brunswick is modeled. It is based on the network used by FCS. The actual campus network is implemented using different networking technologies, like switched Ethernet and Wireless LAN. The campus network model has 512 users: 412 are students, and 100 are faculty members. Faculty members are using various applications such as HTTP, database, Email and FTP, while students are using the same applications except database and FTP. The network is divided into three sub-networks and a gateway: Gillin Hall, the ITC, and the server

room as shown in Figure 3.1. There are eleven switches in the Gillin Hall

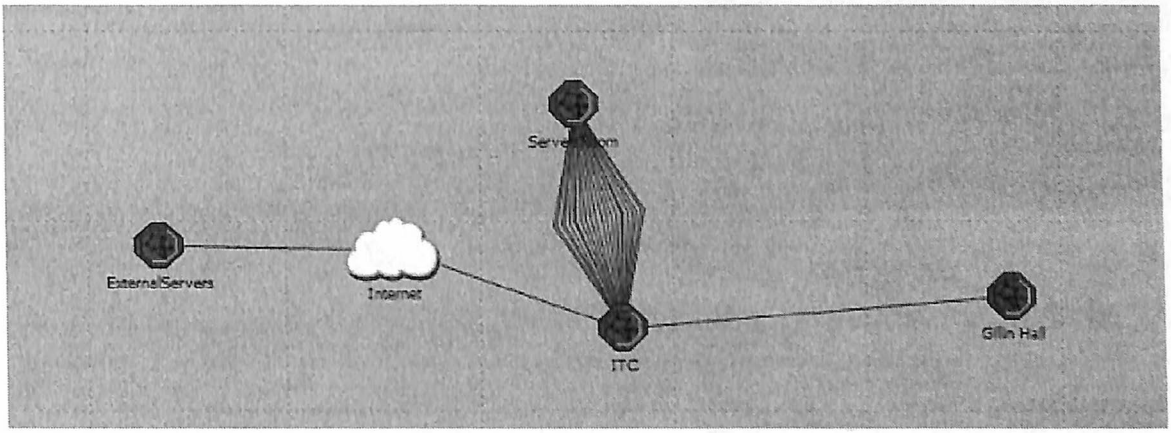

Figure 3.1: FCS network topology

sub-network. It consists of 225 student workstations and 58 faculty workstations. While there are a number of ways to connect various network switches in a LAN, stacking a group of network switches is used in FCS. Stacking switches allows for efficiency and ease of management. On the Gillin Hall sub-network five of the switches are stacked, so logically they work as one switch; therefore, in the simulation model they are configured as one switch connecting 125 workstations. Among the other six switches, there are four switches where each switch connects 21 workstations. Each of the remaining two switches connects 37 workstations. The Gillin Hall subnetwork has only one connection to the outside sub-network, which is the ITC subnetwork.

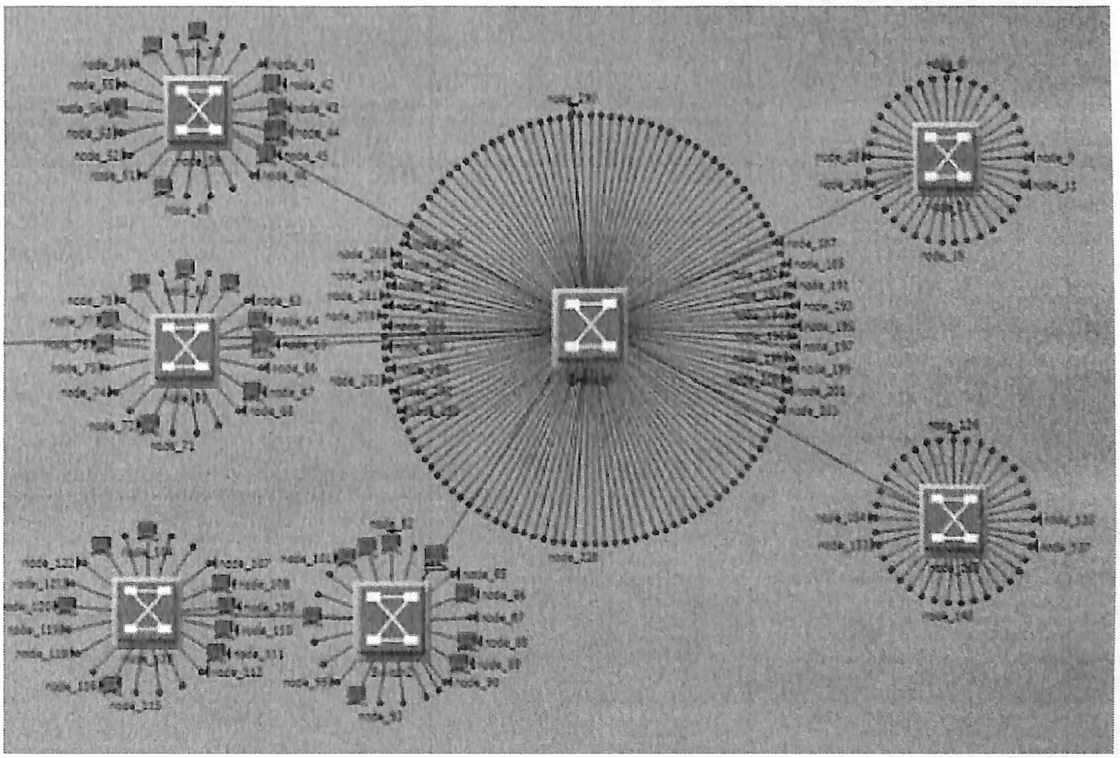

Figure 3.2: Gillin Hall sub-network

The second sub-network in FCS is the ITC sub-network that consists of nine switches. The switches in this sub -network are less numerous than in the Gillin Hall sub-network since the users are less numerous than in the Gillin Hall sub-network. The total number of users in the ITC sub-network is 229, including 187 workstations for students, and 42 workstations faculty users.

Five of the switches are stacked, therefore they work as a single switch that connects 125 workstations. The total numbers of user for the four remaining switches is 104 workstations. One of these four switches connects 41 workstations, while the three remaining switches connect 21 workstations. The third sub-network covers the server room. The total number of servers that are in the server room is 20 servers. These servers are split into four groups of five, each group supporting a different application. The server room sub-network consists of five servers for Email, five servers for Database, five servers for FTP, and five servers for HTTP. Four of the servers run the UNIX operating system while 16 servers run the Windows operating system. The switch that connects all servers is located in the ITC sub-network. This switch is the same one that connects the ITC sub-network to the Gillin Hall sub-network.

3.2 The Links

The link speed between each computer and a switch is 1 Gbps, while the link speed between the Gillin Hall sub-network and the ITC sub-network is 10 Gbps. We applied the same link speeds in OPNET Modeler.

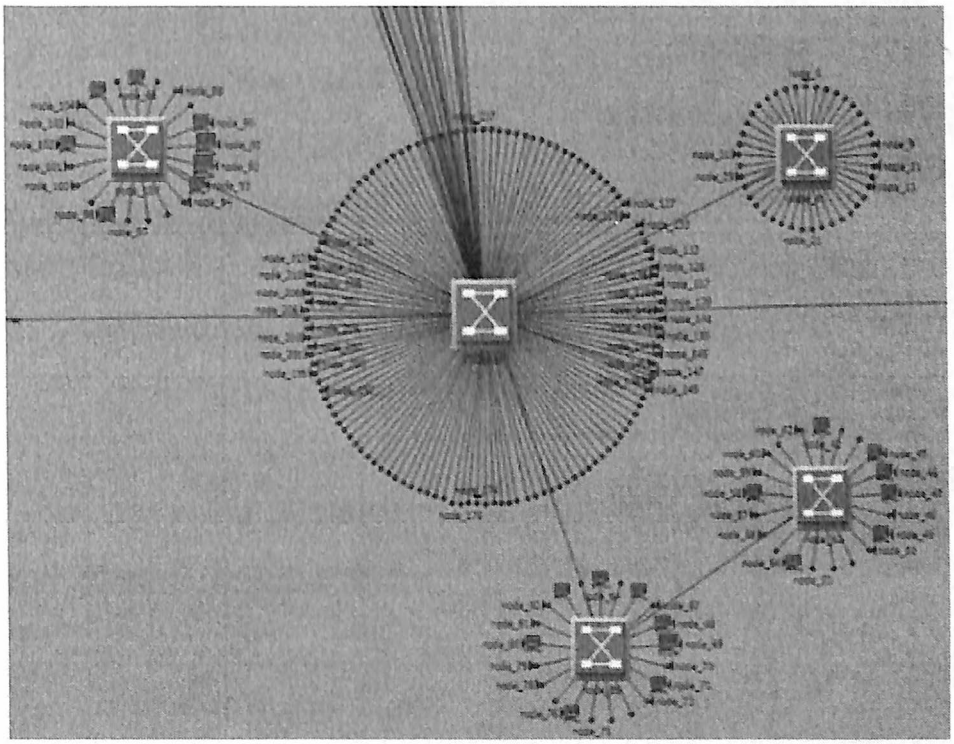

Figure 3.3: ITC sub-network

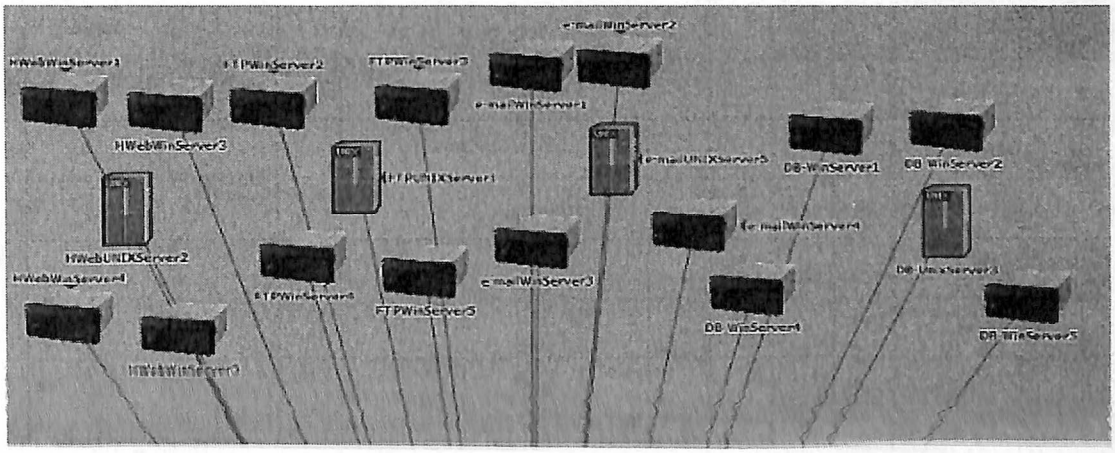

Figure 3.4: Servers room

Chapter 4

Simple Validation Experiment

This chapter describes a simple validation experiment using the FTP application. An OPNET simulator was used to model an FTP transfer on the campus network and Iperf was used to measure the FTP performance on the actual network.

4.1 Introduction

In order to increase confidence in the results of simulations, a validation of the simulation process should be performed. There are several approaches to validation [8]. Two approaches to validate the result of simulation were used in this report. The first approach was to compare the results from the OPNET simulator with an analytical model. The second approach was to attempt to compare the results observed from a real network with the simu-

lator.

In this simple validation scenario there are three workstations operating as sources and one as a server. The three workstations send data to the server. The three workstations have different start times for commencing the transfers: 1s, 5s and 10s respectively. They run the same application which is FTP. In this experiment, the link speed between each workstation and the switch is 1 Gbps and this reflects the actual network setup on the FCS network in the Gillin Hall building.

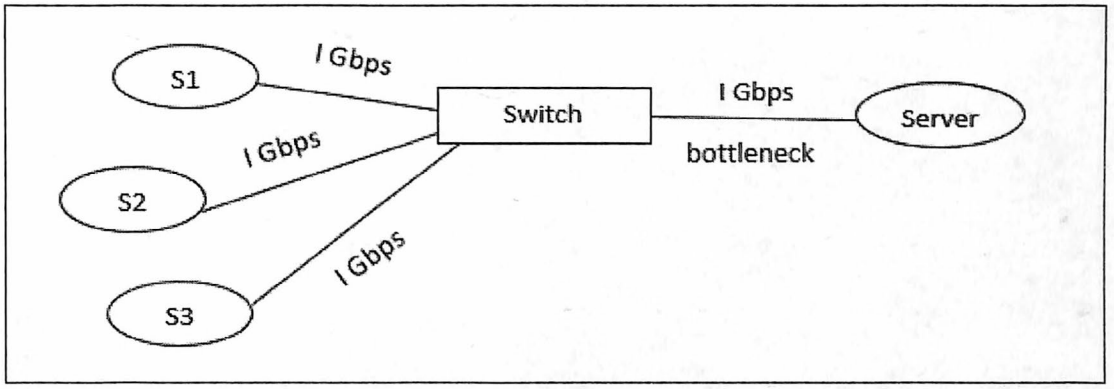

Figure 4.1: Three computers send file to one server

4.2 Analytical Model

Consider the scenario shown in Figure 4.1. First, the packet header overhead must be considered when calculating the maximum FTP throughput. To calculate the FTP throughput, the number of data bits carried in a frame (represented by Ldata) is divided by the total number of bits, including

the overhead (represented by Lmax), then multiplied by the link speed to determine the highest practical amount of FTP data throughput that could possibly be achieved on the link [9]. For 1518 byte packets with 1460 byte payload sent over 1 Gbps link we have:

$$FTP\ Throughput = \frac{Ldata}{Lmax} \times Linkspeed \tag{4.1}$$

$$FTP\ Throughput = \frac{1460}{1518} \times 1Gbps$$

$$FTP\ Throughput = 962Mbps$$

which gives 962 Mbps total available data throughput on the bottleneck link Figure 4.1. When the first flow starts, the FTP throughput should theoretically be the entire 962 Mbps, but when the second flow starts, the throughput for each FTP flow decreases to 50 percent of the full capacity of the bottleneck link as the two TCP flows share the bottleneck fairly (Table 4.2). In addition, when the third client starts, the throughput for each flow decreases to 33.3 percent of available data bandwidth to reflect sharing of the bottleneck equally between three flows.

4.3 Simulator Model in OPNET

In this simple simulation scenario, there were three workstations operating as sources and one as a server, as shown in Figures 4.1 and 4.2. The workstations and the destinations use 1000BaseX links modeled in OPNET. A Profile

Object and an Application Object were created to define traffic for each workstation. These objects, the Application Object and the Profile Object, must be included in the OPNET model to generate the traffic in the network models to be tested and evaluated. An FTP application was set in the Application Object. Figure 4.3 shows attributes of the FTP application. The file size is 4112431185.92 bytes as shown in Figure 4.3 in attribute file size. The file size was 4112431185.92 bytes to match the file size observed in the next experiment using Iperf (Section 4.4) which was 3.83 Gbps. The value of the attribute command mix (Get/Total) is set to 0, so that the simulated workstation only uploads a file to the FTP server, and there is no download operation. Three profiles were defined in the Profile Object: FTP5, FTP10, and FTP15 respectively. Table 4.1 shows the parameters used in the Profile Object for each profile. We set FTP5, FTP10, and FTP15 profiles to client1, client2, and client3 respectively. Before running the simulation, the link statistics were chosen and then the throughput of the link was selected to be monitored by the simulator.

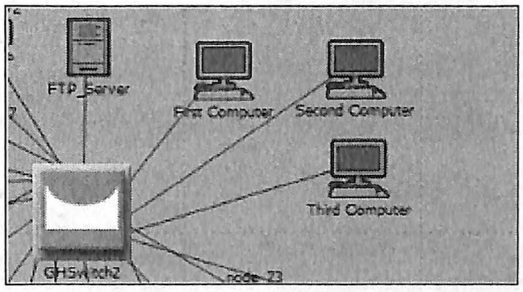

Figure 4.2: Three computers send file to one server in OPNET

Table 4.1: Table for profile parameters

FTP5	
Start Time	1 Second
Application	FTP
Duration	End of profile
FTP10	
Start Time	5 Second
Application	FTP
Duration	End of profile
FTP15	
Start Time	10 Second
Application	FTP
Duration	End of profile

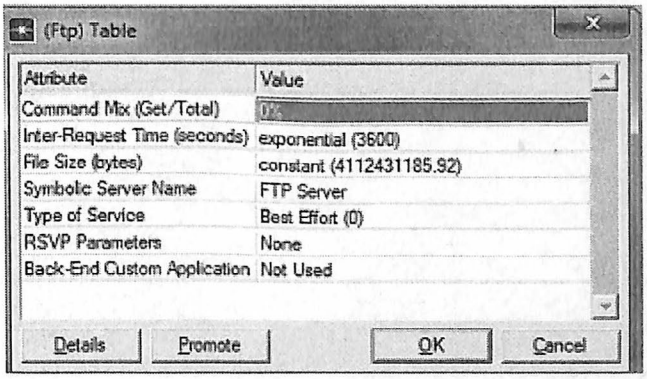

Figure 4.3: FTP configuration table

4.4 Experiment with Iperf

On the actual network Iperf was used to monitor the FTP transfer and to validate the results from the OPNET simulation. In this simple scenario, three workstations ran over TCP and sent data to one server. We used four workstations in the same ITC lab: three as clients and one as the server. The three client workstations had different start times: 1s, 5s, and 10s, respectively. By default, each Iperf client connects to the Iperf server on TCP port 5001. The first step is to run Iperf on the server side using the following command:

```
Iperf -s
```

The -s argument is used for specifying server. The above command starts an Iperf server on the machine, and the server runs on the port 5001 by default.

```
Iperf -c Server IP -t 30 -i1
```

This command runs the clients. The -t argument is used to initiate the data transfer for 30 seconds and the -i arguments tells the Iperf client to monitor the transfer rate over each interval of 1 second, of the entire 30 second transfer.

4.5 Result and Analysis

The three results are displayed in Tables 4.2, 4.3, and 4.4. The average transfer rate over each three second period is recorded, i.e., each time slot Ti represents the average of three samples (either from OPNET or Iperf) taken

Table 4.2: Table for analytical validation

Time	Flow1	Flow2	Flow3	Total
T1	962	0	0	962
T2	481	481	0	962
T3	320.7	320.7	320.7	962
T4	320.7	320.7	320.7	962
T5	320.7	320.7	320.7	962
T6	320.7	320.7	320.7	962
T7	320.7	320.7	320.7	962
T8	320.7	320.7	320.7	962
T9	320.7	320.7	320.7	962

Table 4.3: Table for OPNET result

Time	Flow1	Flow2	Flow3	Total
T1	962.4	0	0	962.4
T2	481.3	481.2	0	962.5
T3	481.2	481.2	0	962.4
T4	320.9	320.9	320.7	962.4
T5	320.8	320.8	320.8	962.4
T6	320.8	320.8	320.8	962.4
T7	320.8	320.8	321	962.4
T8	320.8	320.8	320.8	962.4
T9	320.8	320.8	320.8	962.4

Table 4.4: Table for Iperf result

Time	Flow1	Flow2	Flow3	Total
T1	941.7	0	0	941.7
T2	827.5	378.5	0	1206
T3	466.7	476.3	0	943
T4	371.7	285	335.3	992
T5	317	303.3	318	938.3
T6	271.3	318.7	351.7	941.7
T7	306.7	363.7	223.3	893.7
T8	358.7	303	333.3	995
T9	284	318.7	285.7	888.4

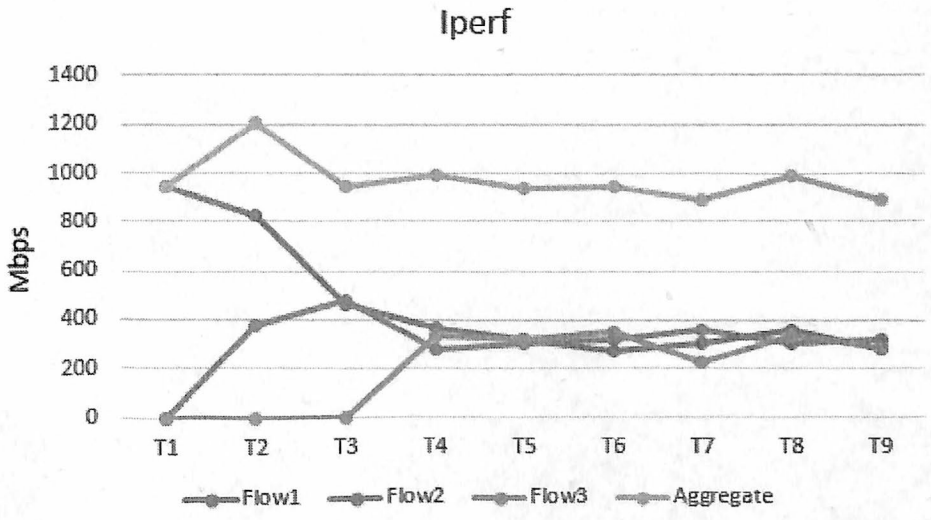

Figure 4.4: Iperf throughput chart for three flows

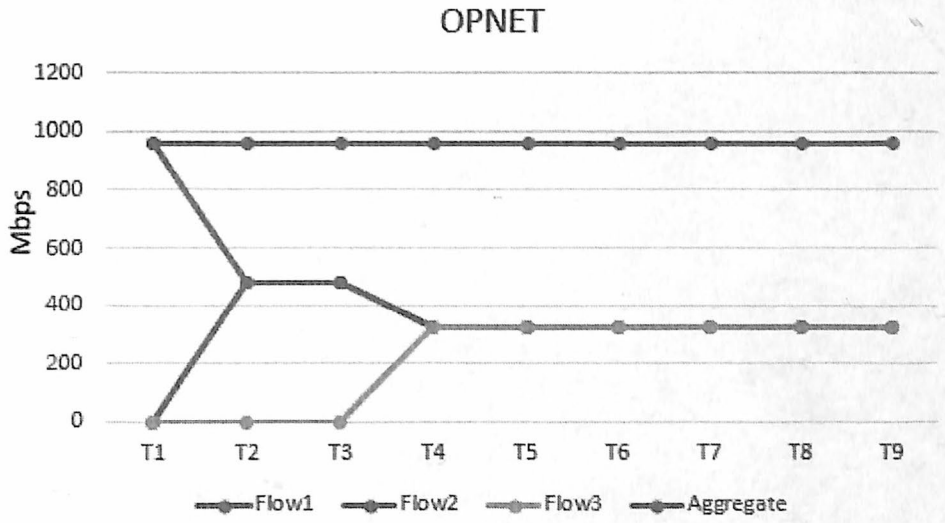

Figure 4.5: OPNET throughput chart for three flows

at a one second intervals. Recall that the main point of this experiment is the comparison between the actual network and the simulation in which three active flows compete for bandwidth over the same link. The bottleneck represents the link between the server and switch in this simple scenario. In Table 4.3 OPNET shows an almost fair sharing of the bottleneck link capacity, as shown in Figure 4.5 with the first flow streaming initially (i.e. time slot T1) at 962 Mbps. In Figure 4.4, the throughput of the first flow measured by Iperf at time T1 is 941 Mbps. This indicates that on the actual network the FTP transfer does not use the full physical capacity of the link because of the overhead and the background traffic present. The three flows in the Time slot T9 in OPNET are almost the same as the flows in the analytical model (difference of 0.1 Mbps), but the three flows in OPNET

are not same as the flows in Iperf. In general, the throughput in the actual network is lower than in the analytical model and in the simulation. The average difference between the three flow data rates (n=3, i=1,2,3) in the time slot T9 measured in OPNET and in Iperf and calculated using the data rate values from Tables 4.3 and 4.4 is

$$\text{diff}_{OPNET-\text{Iperf}} = \frac{\sum |OPNET\ rate(i) - \text{Iperf}\ rate(i)|}{n} \tag{4.2}$$

$$= \frac{74}{3} = 24.7Mbps$$

The reasons for these throughput differences include overheads, the space between packets, errors from collisions or electronic interference, variations in TCP receiver window size, and operating system limitations. We were unable to measure all of these individually.

Chapter 5

Time of Transfer Experiment

This chapter describes another validation test called the time of transfer experiment and includes the comparison of results from the analytical approach, from Iperf measurements, and finally, from the simulation in OPNET.

5.1 Introduction

In this experiment, an FTP operation was examined by transferring a file (using FTP) from one location to another, and then measuring the time taken to complete the transfer. The main purpose of this experiment is to determine the time of transfer for one flow using different methods. This measured (or estimated) time taken to transfer a file was compared between the actual network, the simulation, and the analytical model. First we ran the FTP transfer using Iperf on the actual campus network for 30 seconds

and we measured the transfer size. Then we modeled transferring a file of the same size in OPNET and using the analytical approach.

5.2 Experiment with Iperf

In this experiment, only one FTP client runs over TCP and sends data to the server. We run the client for 30 seconds and record the number of bytes transferred. By default, the Iperf client connects to the Iperf server on the TCP port 5001. The first step to is to run Iperf on the server side using the following command (same as Section 4.4):

```
Iperf -s
```

The above command starts the Iperf server on the machine, and it runs on the port 5001 by default.

```
Iperf -c Server IP -t 30 -i1
```

This command runs the client. The -t argument is used to transfer data for 30 seconds and the -i argument tells the Iperf client to show the transfer rate at an interval of 1 second, for a 30 second transfer. The size of the file transfer in this experiment reported by Iperf was 3.27 GB.

5.3 Analytical Model

In this analytical model we calculate the time required to transfer the 3.27 Gbytes file over a 1 Gbps link. The overhead was calculated for the 1 Gbps

link in Chapter 3 Section 3.2 giving the effective FTP transfer data rate of 962 Mbps. The transfer time of a 3.27 GB file can be calculated as:

$$The\ total\ time\ to\ \text{transfer}\ file\ = \frac{File\ size}{Link\ speed} = 29.2\ second \qquad (5.1)$$

5.4 Simulator Model in OPNET

Next, OPNET was used to observe how long it took in the simulation to transfer 3.27 Gbytes. The source workstation and the destination server were connected by a 1 Gbps 1000BaseX link. The Application Object and the Profile Object were created in models to generate the traffic. The file size was set to 3511135764.48 bytes, therefore, the FTP application that was defined will transfer one 3.27 Gbyte file at a time.

5.5 Results

In the Iperf experiment, it took 30 seconds to transfer 3.27 Gbytes. The total time in OPNET was also 30 seconds as shown in Table 5.1. The first recorded link throughput in OPNET was 935.8 Mbps and the last throughput was 885.9 Mbps. OPNET started and finished the flow in slow (startup) mode as shown in Figure 5.2. Both times recorded in Iperf and OPNET were slightly higher than the calculated theoretical minimum of 29.2 seconds.

```
"Client connecting to gc127n04, TCP"    port    5001
TCP window size: 23.2 KByte (defaul     t)
-----------------------------------     -----
-------------------
[  3] local 131.202.242.46 port 43377 connected with
131.202.242.49 port 5001
[  3]  0.0- 1.0 sec    114 MBytes       955     Mbits/sec
[  3]  1.0- 2.0 sec    112 MBytes       936     Mbits/sec
[  3]  2.0- 3.0 sec    112 MBytes       939     Mbits/sec
[  3]  3.0- 4.0 sec    111 MBytes       933     Mbits/sec
[  3]  4.0- 5.0 sec    110 MBytes       926     Mbits/sec
[  3]  5.0- 6.0 sec    112 MBytes       940     Mbits/sec
[  3]  6.0- 7.0 sec    112 MBytes       937     Mbits/sec
[  3]  7.0- 8.0 sec    111 MBytes       930     Mbits/sec
[  3]  8.0- 9.0 sec    112 MBytes       938     Mbits/sec
[  3]  9.0-10.0 sec    111 MBytes       927     Mbits/sec
[  3] 10.0-11.0 sec    112 MBytes       942     Mbits/sec
[  3] 11.0-12.0 sec    111 MBytes       931     Mbits/sec
[  3] 12.0-13.0 sec    112 MBytes       938     Mbits/sec
[  3] 13.0-14.0 sec    111 MBytes       929     Mbits/sec
[  3] 14.0-15.0 sec    111 MBytes       932     Mbits/sec
[  3] 15.0-16.0 sec    112 MBytes       938     Mbits/sec
[  3] 16.0-17.0 sec    111 MBytes       935     Mbits/sec
[  3] 17.0-18.0 sec    112 MBytes       939     Mbits/sec
[  3] 18.0-19.0 sec    110 MBytes       926     Mbits/sec
[  3] 19.0-20.0 sec    112 MBytes       940     Mbits/sec
[  3] 20.0-21.0 sec    112 MBytes       937     Mbits/sec
[  3] 21.0-22.0 sec    111 MBytes       932     Mbits/sec
[  3] 22.0-23.0 sec    112 MBytes       939     Mbits/sec
[  3] 23.0-24.0 sec    111 MBytes       928     Mbits/sec
[  3] 24.0-25.0 sec    112 MBytes       941     Mbits/sec
[  3] 25.0-26.0 sec    111 MBytes       932     Mbits/sec
[  3] 26.0-27.0 sec    111 MBytes       934     Mbits/sec
[  3] 27.0-28.0 sec    111 MBytes       929     Mbits/sec
[  3] 28.0-29.0 sec    112 MBytes       941     Mbits/sec
[  3] 29.0-30.0 sec    111 MBytes       933     Mbits/sec
[  3]  0.0-30.0 sec  3.27 GBytes        935     Mbits/sec
```

Figure 5.1: Time to transfer one File in Iperf

Table 5.1: Table for OPNET transfer one file

Time	Throughput	Time	Throughput
1	935.8	16	962.4
2	962.4	17	962.4
3	962.4	18	962.4
4	962.4	19	962.4
5	962.4	20	962.4
6	962.4	21	962.4
7	962.4	22	962.4
8	962.4	23	962.4
9	962.4	24	962.4
10	962.4	25	962.4
11	962.4	26	962.4
12	962.4	27	962.4
13	962.4	28	885.9
14	962.4	29	885.9
15	962.4	30	885.9

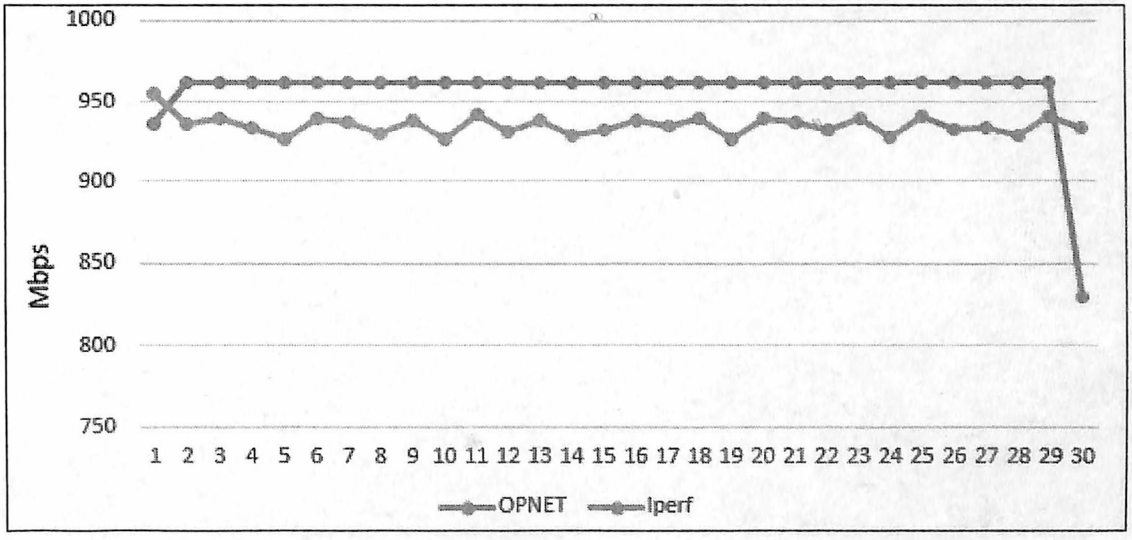

Figure 5.2: Comparison of throughput between Iperf and OPNET

Chapter 6

Investigation of Throughput on the Complex Campus Network

In this chapter, Section 6.1 provides an overview of the complex network under investigation. Section 6.2 shows analytical calculations. Section 6.3 explains the parameters that were used to run the OPNET Modeler. Section 6.4 explains the implementation in Iperf. Section 6.5 shows the results of the experiment. Section 6.5 describes the analysis and the modeling of the background traffic.

6.1 Introduction

The objective of this investigation is to assess what would happen on a network with complex topology (FCS campus network) when multiple FTP

clients saturate a bottleneck link. In the investigated scenario there are 13 clients placed in GH (Gillin Hall), as shown in Figure 6.1, and 13 servers placed in ITC, as shown in Figure 6.2, sharing a 10 Gbps bottleneck link between GH and ITC. The experiment uses as many as 13 clients since in order to saturate a 10 Gbps link at least 10 sources sending at the 1 Gbps rate are required. Each node on this network is connected to the switch with 1 Gbps. Each node sends data to a separate server. This scenario was tested on the actual FCS campus network and modeled in OPNET and the results between the actual network and the simulation were compared. The clients start transmitting in two second intervals (except the 2nd one): 2s, 2s, 4s, 6s, 8s, 10s, 12s, 14s, 16s, 18s, 20s, 22s, and 24s respectively.

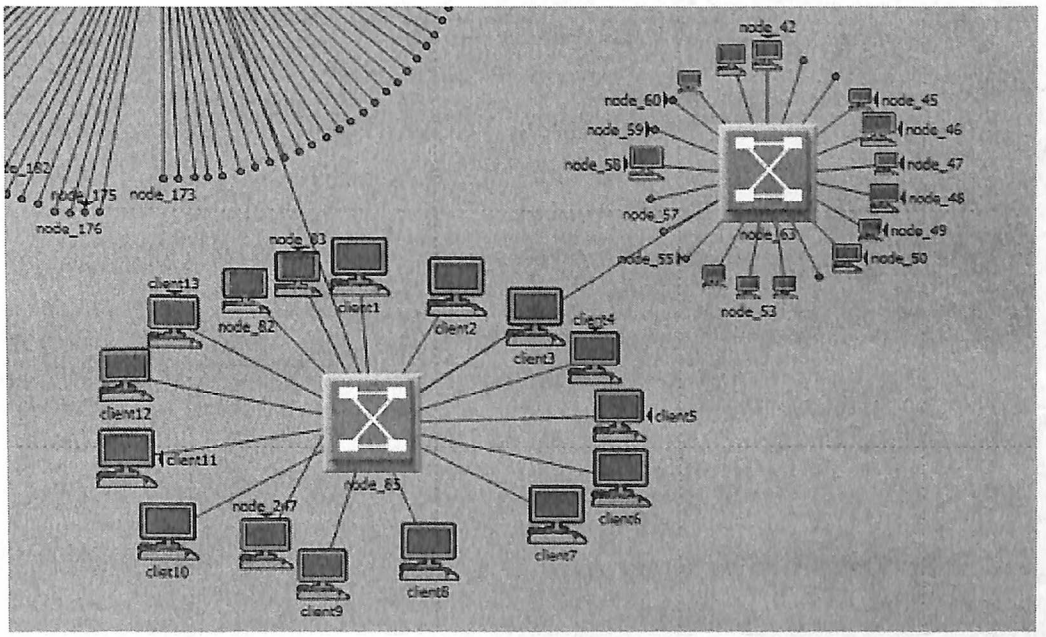

Figure 6.1: 13 Clients in GH

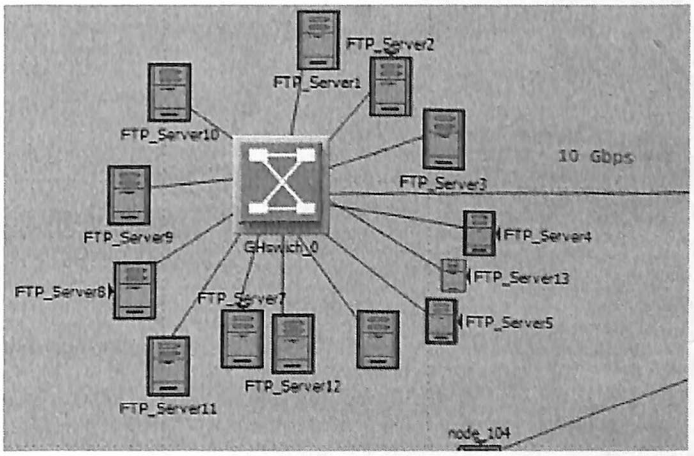

Figure 6.2: 13 Servers in ITC

6.2 Analytical Approach

The analytical maximum data throughput is 962 Mbps for the link speed of 1 Gbps, based on the calculation from Section 4.2. To saturate the link between GH and ITC that has 9620 Mbps available to send data, the number of flows will need to be at least 10 (10 * 962=9620Mbps). Because TCP has the ability to share the bottleneck [10], the analytical calculations were performed to tabulate the data rate for each flow in this experiment. For example, when there are 13 flows, each flow should be 9620/13= 769.2 Mbps, as shown in Table 6.4. The 13 flows have different start times: 2s, 2s, 4s, 6s, 8s, 10s, 12s, 14s, 16s, 18s, 20s, 22s, and 24s respectively.

6.3 Simulation Model in OPNET

The simulation ran for 30 seconds. All 13 clients ran FTP pre-configured applications provided by OPNET. The FTP applications ran with unlimited repeatability. The Command Mix parameter was set to 0% to upload a file from a client onto the FTP server as shown in Figure 6.3. The file size was 3511135764.48 Bytes, which is equal to 3.27 Gbytes. The file size in OPNET was set to 3.27 Gbytes to allow for sending the file over a period of time of at least 30 seconds required for this experiment.

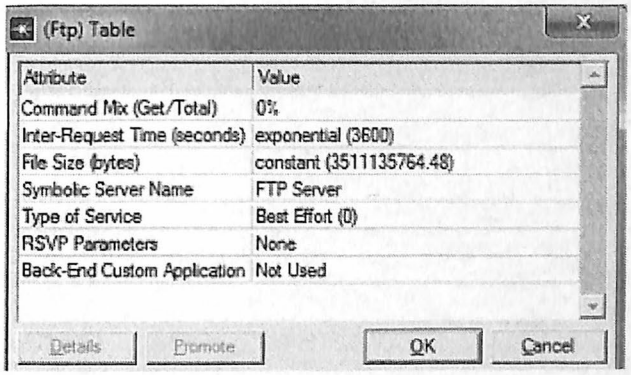

Figure 6.3: FTP pre-configured application

Each workstation sent a file to a distinct server. Application destination preference attributes were used to provide mapping between the workstation and the server; for instance, the first workstation sent a file to the first server. Multiple simulations were run with different seeds that were automatically generated by OPNET. The 13 workstations had different start times: 2s, 2s, 4s, 6s, 8s, 10s, 12s, 14s, 16s, 18s, 20s, 22s, and 24s respectively.

6.4 Experiment with Iperf

In order to validate the results from the OPNET simulator the same scenario was implemented on the actual campus network using Iperf. As before, many active flows competed for bandwidth over the same link. Using Iperf terminology, the workstation was the transmitter and the server was the receiver. In this experiment, the data was sent from the workstation to the server. The client connected to the server on the TCP port 5001. In the first step 13 servers were started in the Gillin Hall building. The following command was used to start each server (same as Section 4.4):

```
Iperf -s
```

The -s argument was used for designating a server. Then thirteen clients were run on separate machines in the ITC building by using the following command:

```
Iperf -c server IP -t 30 -i1
```

The -t argument was used to initiate the transfer of data for 30 seconds, and the -i argument set the Iperf client to show the transfer rate at an interval of 1 second for the entire 30 second transfer.

6.5 Results

The results for the analytical, actual and simulated networks are shown in Tables 6.1, 6.2, and 6.3 respectively. The average transfer rate over each three second period is recorded, i.e. each time slot Ti represents the average of

three samples (either from OPNET or Iperf) taken at a one second interval. Both OPNET and Iperf experiments lasted 30 seconds. OPNET was run four times with four different seeds giving almost identical result in each run. Iperf was run for four different times and the average data rate result was recorded.

Table 6.1: Transfer rate for 13 flows based on analytical calculation

Time	Flow1	Flow2	Flow3	Flow4	Flow5	Flow6	Flow7	Flow8	Flow9	Flow10	Flow11	Flow12	Flow13	Total
T1	962	962	0	0	0	0	0	0	0	0	0	0	0	1924
T2	962	962	962	962	0	0	0	0	0	0	0	0	0	3848
T3	962	962	962	962	962	0	0	0	0	0	0	0	0	4810
T4	962	962	962	962	962	962	962	0	0	0	0	0	0	6734
T5	962	962	962	962	962	962	962	962	0	0	0	0	0	7696
T6	962	962	962	962	962	962	962	962	962	962	0	0	0	9620
T7	874.6	874.6	874.6	874.6	874.6	874.6	874.6	874.6	874.6	874.6	874	0	0	9620
T8	740	740	740	740	740	740	740	740	740	740	740	740	740	9620
T9	740	740	740	740	740	740	740	740	740	740	740	740	740	9620

Table 6.2: Transfer rate for 13 flows in Iperf

	Flow1	Flow2	Flow3	Flow4	Flow5	Flow6	Flow7	Flow8	Flow9	Flow10	Flow11	Flow12	Flow13	Total
T1	874.79	947.5	0	0	0	0	0	0	0	0	0	0	0	1574
T2	874.08	932.58	943.2	797.0833	0	0	0	0	0	0	0	0	0	3126.7
T3	872.92	933.83	933.3	934	947.38	0	0	0	0	0	0	0	0	4364.7
T4	872.5	935.83	934.8	934.8333	935.33	924.92	960.25	0	0	0	0	0	0	5937
T5	866.25	933.17	934.5	934.5833	932.83	933.67	934.33	946.13	0	0	0	0	0	7167
T6	871.83	931.92	932.9	924	934.42	932.08	932.33	934.67	942.42	874.5	0	0	0	8739.3
T7	760	826.75	831.3	835.1667	842.67	832.25	826.58	862.17	841.17	797.42	911.38	0	0	8829
T8	622.5	720.67	722.6	711.25	715.33	712.92	623.83	722	707.17	726.83	917.67	915.5	275.23	8741
T9	531.83	706.42	710.3	694	704.42	700.17	535.58	713	715.67	773.17	919.08	919.92	314.17	8809

Table 6.3: Transfer rate for 13 flows in OPNET

Time	Flow1	Flow2	Flow3	Flow4	Flow5	Flow6	Flow7	Flow8	Flow9	Flow10	Flow11	Flow12	Flow13	Total
T1	0	959.4	0	0	0	0	0	0	0	0	0	0	0	959.399
T2	896.4	962.43	960.41	956.36	0	0	0	0	0	0	0	0	0	3775.57
T3	962.4	962.43	962.43	962.43	959.39	0	0	0	0	0	0	0	0	4809.12
T4	962.4	962.43	962.43	962.43	962.43	960.41	956.36	0	0	0	0	0	0	6728.92
T5	962.4	962.43	962.43	962.43	962.43	962.43	962.43	959.39	0	0	0	0	0	7696.42
T6	962.4	962.43	962.43	962.43	962.43	962.43	656.22	962.43	960.4	956.34	0	0	0	9309.99
T7	960.1	960.12	962.43	962.43	530.94	962.43	268.98	657.63	961.76	962.43	959.34	0	0	9148.61
T8	962.4	962.43	960.12	962.43	595.16	962.43	623.19	0.3596	962.43	962.43	14.711	643.83	950.1	9562.05
T9	962.4	962.37	962.43	961.41	593.37	960.95	609.57	25.994	962.43	962.43	0.0041	0.0041	961.8	8925.19

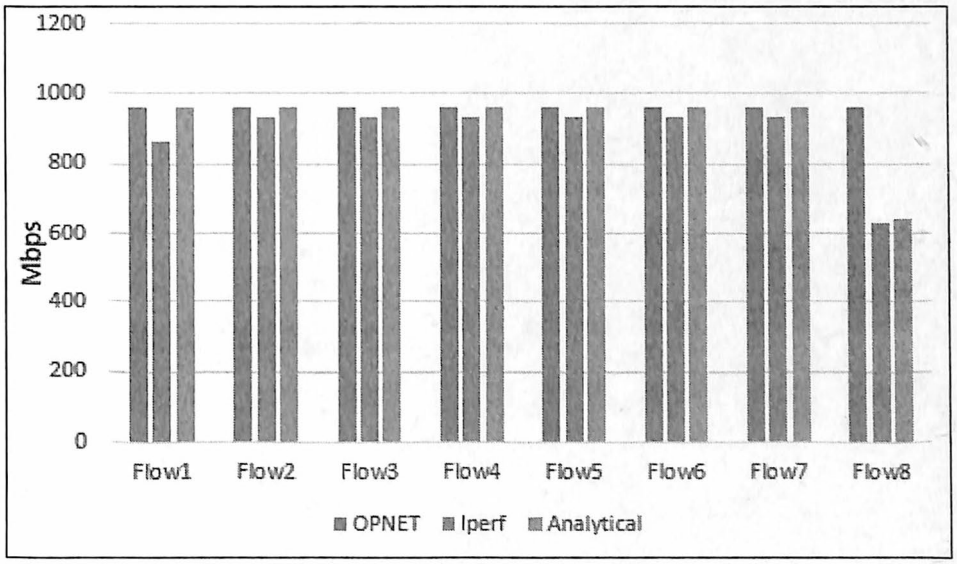

Figure 6.4: Transfer rate at time interval T5 with 8 flows

The maximum throughput for a flow recorded in Iperf was 947 Mbps as shown in Table 6.2, while 962 Mbps was the maximum throughput recorded in OPNET giving in Table 6.3. Figure 6.4 shows the comparison between eight flows at time T5 and Figure 6.5 shows the comparison between 13 flows at time T9. For the comparison between the results from the OPNET simulator and from other measurements (Iperf, analytical) we used the average of flow data rate difference calculated as follow:

$$\text{diff}_{OPNET-\text{Iperf}} = \frac{\sum |OPNET\ rate(i) - \text{Iperf}\ rate(i)|}{n} \tag{6.1}$$

which was

$$\text{diff}_{OPNET-\text{Iperf}} = \frac{596.8}{8} = 74.5 Mbps \tag{6.2}$$

at T5 for 8 flows, (n=8, i=1..8) and

$$\text{diff}_{OPNET-\text{Iperf}} = \frac{5261}{13} = 404.7Mbps \tag{6.3}$$

at T9 for 13 flows (n=13, i=1..13). For comparison with the analytical model we used:

$$\text{diff}_{OPNET-\text{analytical}} = \frac{\sum |OPNET\ rate(i) - \text{analytical}\ rate(i)|}{n} \tag{6.4}$$

which was

$$\text{diff}_{OPNET-\text{Iperf}} = \frac{321.1}{8} = 40.1Mbps \tag{6.5}$$

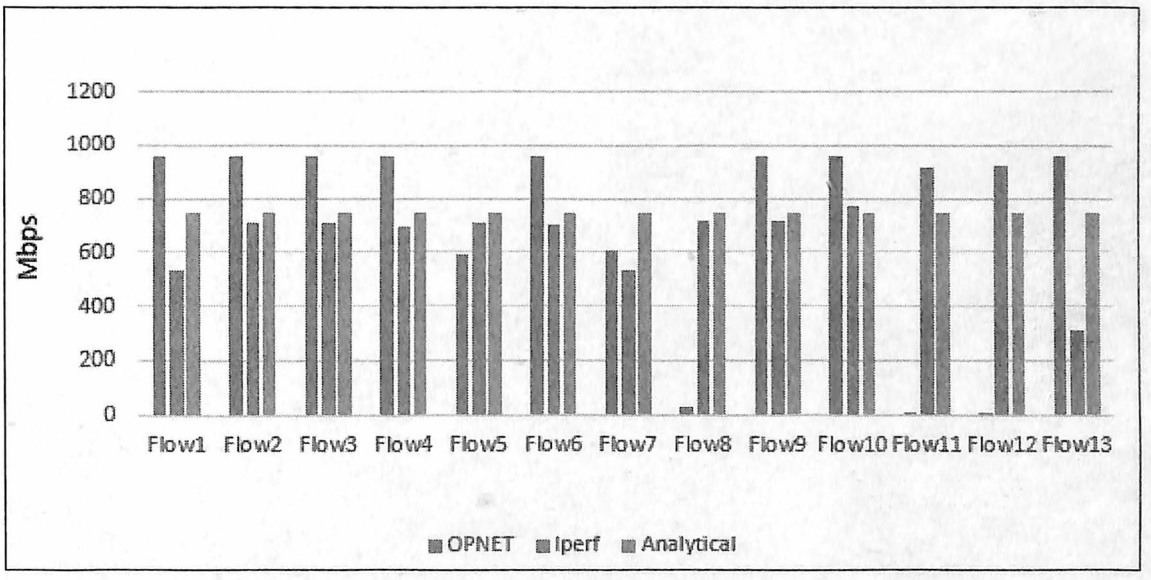

Figure 6.5: Transfer rate at time interval T9 with 13 flows

at T5 for 8 flows, and

$$\text{diff}_{OPNET-\text{Iperf}} = \frac{4247.3}{13} = 326.7Mbps \tag{6.6}$$

at T9 for 13 flows.

When there were 8 flows at T5, the results for the analytical calculations in Iperf and OPNET were almost the same (average transfer rate difference of 74.5 Mbps or $\frac{74.5}{962} = 8\%$ and 40.1 Mbps or $\frac{40.1}{962} = 5\%$ respectively). This is because the network was not saturated and ran at its maximum speed. On the other hand, when there were 13 flows at T9, the results from OPNET and Iperf are quite different with the average difference of 404.7 Mbps or $\frac{404.7}{962} = 42\%$. This is because with 13 active flows the network was saturated and the flows were subject to throttling due to the congestion control mechanism in TCP. The congestion control mechanism operates independently for each flow causing the large observed deference in the results from the models and the actual network.

In most cases the throughput of the actual network was lower, and different, from the simulation. The reasons include overheads, the space between packets, errors from collisions or electronic interference, variations in TCP receiver window size, and operating system limitations. We were unable to measure all of these individually. Figure 6.6 shows the average data rate for an active flow, for each time slot in Iperf, in the analytical model, and

in OPNET calculated based on Tables 6.1, 6.2, and 6.3. The diagram in Figure 6.6 shows that the results for OPNET are close to those for Iperf and the analytical model. It should be noted on Figure 6.5 that the difference between the data rates of individual (pairs of) flows in one time slot (T9) is big (42%), while the difference between the averages of all flows in one time slot between OPNET and Iperf is small as can be seen in Figure 6.6 (for T9 they are the same).

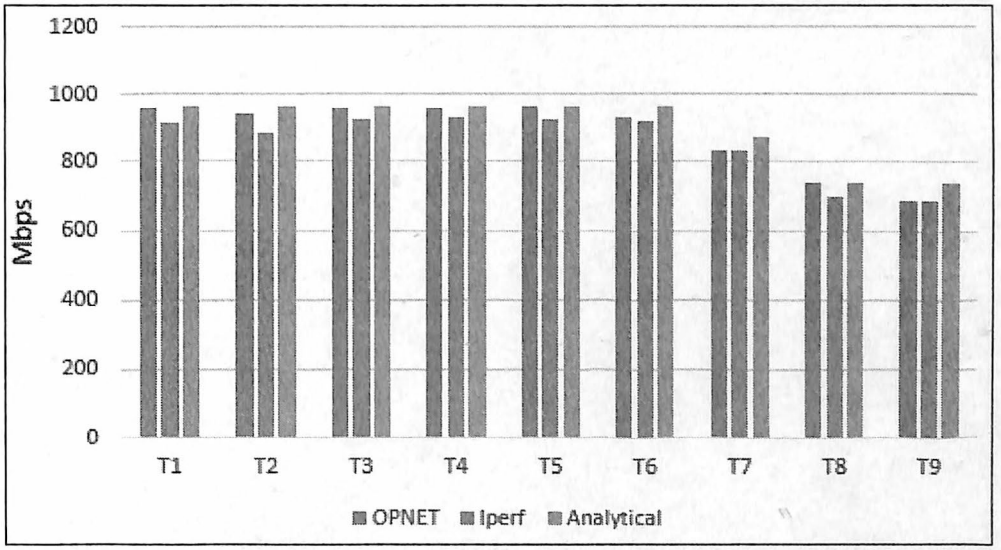

Figure 6.6: Comparison of average data rate

6.6 Background Traffic

On the actual network the bandwidth available to a user is typically less than the link capacity because a large number of other users present on the

network can produce important background traffic by using a variety of data intensive applications on the network, such as Email and web browser applications. This background traffic can have an impact on the actual network performance. It is difficult to estimate the background traffic because the behavior of real users is difficult to model. Another challenge in modeling is that the OPNET simulator has no mechanism for generating the background traffic with the data rate defined with standard statistical distributions [11]. The background traffic can be estimated by comparing the achievable data rate in a simulator (scenario without any background traffic) with the actual data rate recorded on the actual network with Iperf. We set up an experiment to observe the background traffic in the actual network with Iperf and ran it at eight different times (morning, afternoon, etc.). This experiment was the same as the previous experiment that had 13 workstations and 13 servers, but it was run multiple times to observe the background traffic at different times of day. The results for the lowest throughput in Iperf are shown in Table 6.4.

Table 6.4: The lowest throughput in Iperf

Time	Flow1	Flow2	Flow3	Flow4	Flow5	Flow6	Flow7	Flow8	Flow9	Flow10	Flow11	Flow12	Flow13	Total
T1	941.33	629.67	0	0	0	0	0	0	0	0	0	0	0	1571
T2	937.33	935.33	943.67	321.33	0	0	0	0	0	0	0	0	0	3137.7
T3	932.67	935	933.33	933.33	630	0	0	0	0	0	0	0	0	4364.3
T4	936	934.67	935.33	935.33	936.3333	941.67	320.3	0	0	0	0	0	0	5939.7
T5	933	931	932.33	931.33	931.6667	934	931.3	631.67	0	0	0	0	0	7156.3
T6	933	929	935.33	935.67	934	935	935.3	934.67	941.3	321	0	0	0	8734.3
T7	883.67	750.33	769.67	751.67	760.3333	758.33	754.7	810.33	792.3	824	583.67	0	0	8439
T8	685.67	670.33	693.67	634	673.6667	663.67	544.7	694.67	685.7	689.33	913.67	917.667	133.33	8600
T9	689.67	685.33	689	676	672.6667	658.33	534.7	705.33	694	697	913.67	914	383	8912.7

Table 6.5: Calculation of the background traffic based on the difference between highest total data rate recorded in Iperf and OPNET

Time	OPNET	Iperf
T6	9310	9366
T7	9148.6	9341.5
T8	9562	8913.7
T9	8925.2	8897
Average	9236.5	9129.5
Standard Deviation	268.3	259.2
The difference between averages		107 Mbps

Table 6.6: Calculation of the background traffic based on the difference between lowest total data rate recorded in Iperf and OPNET

Time	OPNET	Iperf
T6	9310	8734.3
T7	9148.6	8439
T8	9562	8600
T9	8925.2	8912.7
Average	9236.5	8671.5
Standard Deviation	268.3	201.1
The difference between averages		565 Mbps

The background traffic was calculated for two scenarios:

- Low background traffic (e.g. at night).
- High background traffic (e.g. day time).

The high background traffic estimate was calculated by subtracting from the OPNET simulated data rate the lowest recorded data rate in Iperf during

the period T6 to T9 when the network (the bottleneck) is used at full capacity. The low background traffic estimate was calculated by subtracting from OPNET simulated data rate the highest data rate in Iperf.

Tables 6.5 and 6.6 show the calculation based on the data rate average over 12 seconds covering the intervals T6-T9 (i.e when the bottleneck link is saturated). The low estimate for the background traffic was 107 Mbps and the estimate was 565 Mbps.

6.6.1 Generating Background Traffic in OPNET

When generating the background traffic in the OPNET simulator we used the high estimate of 565 Mbps based on Table 6.6. The background traffic was generated in one direction corresponding to a bottleneck from the ITC sub-network to the Gillin Hall sub-network. Figure 6.7 shows the attributes used to define the background traffic in OPNET. Two main attributes are explained:

- Average Packet Size (byte): By default, the average packet size is 567 bytes.
- Traffic Load (bps): This specifies the load of the background traffic that changes over the course of time in bits per second.

Figure 6.8 shows the background traffic load set in OPNET. Start time for the background traffic was at 2 seconds, the end at 30 seconds and the intensity

(rate) was 565 Mbps. The traffic load did not change with time, which means the background traffic was constant during the running of the simulation.

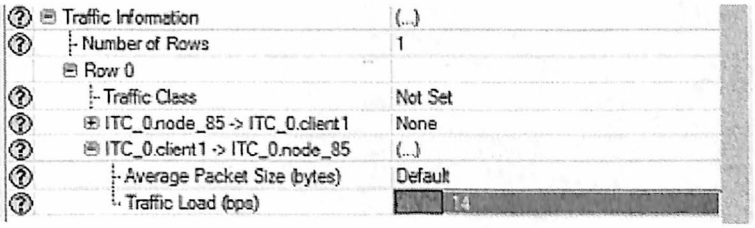

Figure 6.7: Attributes of traffic load

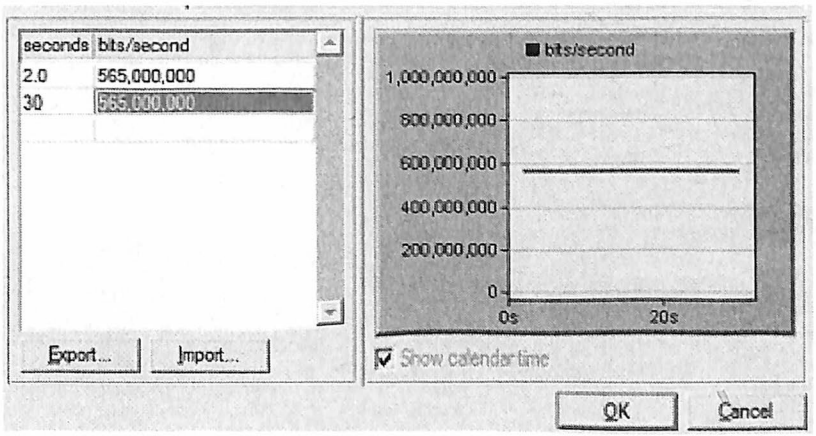

Figure 6.8: Generated the traffic load

6.6.2 Results and Discussion

The results for the simulated networks with background traffic are shown in Tables 6.7. Table 6.8 shows the average of active flows in each time interval for OPNET both with and without background traffic. Table 6.9 shows that

the average of flows, both with and without background traffic, were the same except for in time slots T6, T7, T8 and T9. This is expected because the impact of the background traffic is visible only when the network is saturated (T6-T9). Figure 6.9 shows the detailed comparison of the data rate, with and without background traffic, of each of the 13 flows during the last time interval T9 when the network is saturated. In this case, the throughput for OPNET with background and without background are not similar. Not only the simulation without background traffic resulted in a higher data rate value for each flow, but the individual flow intensities differ significantly. Again, this is expected as the flows are independently responding to competition congestion.

Table 6.7: OPNET with background

Time	Flow1	Flow2	Flow3	Flow4	Flow5	Flow6	Flow7	Flow8	Flow9	Flow10	Flow11	Flow12	Flow13	Total
T1	0	959.4	0	0	0	0	0	0	0	0	0	0	0	959.399
T2	896.37	962.43	960.41	956.36	0	0	0	0	0	0	0	0	0	3775.57
T3	962.43	962.43	962.43	962.43	959.39	0	0	0	0	0	0	0	0	4809.12
T4	962.43	962.43	962.43	962.43	962.43	960.41	956.36	0	0	0	0	0	0	6728.92
T5	962.43	962.43	962.43	962.43	962.43	962.43	962.43	959.39	0	0	0	0	0	7696.41
T6	664.21	962.19	659.84	664.28	664.16	656.49	960.79	962.02	655.2	45.54	0	0	0	6894.7
T7	5.5042	962.43	0.1181	5.6986	0.1181	18.569	962.43	962.43	800.09	799.9	959.34	0	0	5476.62
T8	960.41	962.2	896.21	925.9	6.5574	66.162	962.16	962.43	962.01	958.1	14.892	295.95	350.19	8323.19
T9	876.95	890.36	834.49	920.78	489.45	377.95	818.63	827.88	811.07	787.2	0.1508	517.41	442.22	8594.49

Table 6.8: Average flows of OPNET with and without background traffic

Time	OPNET without background	OPNET with background	The difference
T1	959.4	959.4	0
T2	943.9	943.9	0
T3	962.4	962.4	0
T4	962.4	962.4	0
T5	962.4	962.4	0
T6	931	689	242
T7	832	498	343
T8	736	640	96
T9	687	661	26

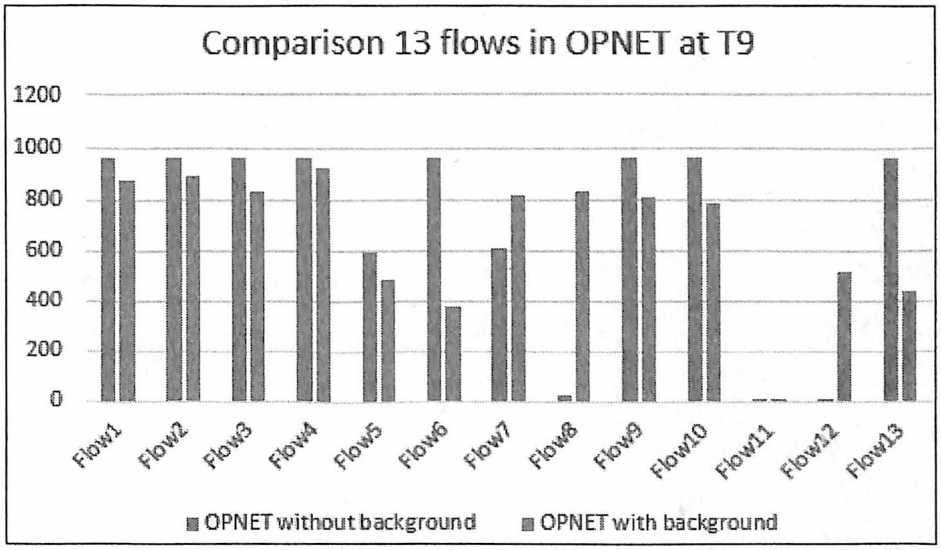

Figure 6.9: Comparison 13 flows in OPNET at T9

Chapter 7

Evaluation of a Video Session on a Campus Network Under Heavy Load

In this Chapter, we apply the network model developed in this report to the FCS campus network and evaluate the performance of a video session.

7.1 Introduction

In this experiment, the quality of service for 13 video conferencing clients on the ITC sub-network connecting to 13 servers in the Gillin Hall sub-network was tested using OPNET. OPNET supports three pre-configured video conferencing types: low-resolution video, high-resolution video, and VCR quality

video. The high-resolution quality was used in this simulation. Two scenarios were considered to ensure that new applications could be added to the network without affecting other applications. The first scenario included only video applications deployed on the campus Network. In the second scenario, in addition to the video application, we had the FTP application, and the background traffic present on the FCS network at the same time.

7.2 Video Application on the Campus Network

In this scenario, there were 13 workstations operating as sources and 13 as servers. The topology in this chapter was the same topology as in Chapter 6. We deployed video in high resolution and monitored the QoS.

7.2.1 Implementation

In the first experiment all workstations ran the same application: video conferencing with high resolution. Each source had different start time : 2s, 2s, 4s, 6s, 8s, 10s, 12s, 14s, 16s, 18s, 20s, 22s, and 24s respectively (same in previous FTP experiment). The frame size was 128 * 240 pixels as shown in Figure 7.1, the pixel depth was 9 bits therefore each video frame consisted of 128 * 240 * 9 = 276,480 bits, which corresponds to 34,560 bytes and considering the frame rate of 15 frames/sec, the data rate for this video session

was 518,400 bytes/sec.

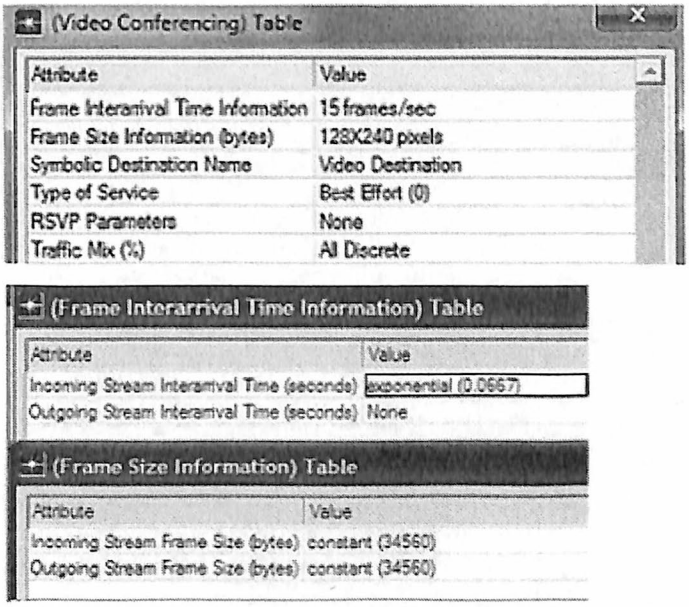

Figure 7.1: Video configuration table of high resolution

In the second experiment each workstation ran the video session and the FTP transfers at the same time. Each source had different start time (as in the pervious experiment). Additionally the background traffic, as determined in Chapter 6, equal to 565 Mbps was placed on the bottleneck link. Three metrics were used to determine the quality of video service. The metrics used were end-to-end delay, packet delay variation (jitter), and packet loss.

7.2.2 Results and Analysis

The two experiments were run for 30 seconds each. In the first experiment the received data rate recorded for the video application was 518,400

bytes/sec exactly the same as analytically calculated. The end-to-end delay was recorded as 0.679 ms and the delay jitter was zero. After adding the FTP sessions and the background traffic OPNET reported exactly the same results. The QoS recommended for the video conferencing [12] requires less than 1% of packet loss, end-to-end delay no more than 150 ms and packet delay variation no more than 30 ms. The results from OPNET satisfy these criteria.

Chapter 8

Conclusion and Future Work

This research investigated the performance of the OPNET simulator for modeling the FCS campus network. The OPNET simulator was used to estimate the data transfer rate of an FTP session. Two methods were used to validate the results of simulation: analytical calculation and the results from the measurements on the actual network. The results from the validation tests run on the lab networks in a simple scenario with only three FTP clients and a simple network topology indicated that the simulator's network model was close to actual network (average difference 24.7 Mbps or 5%), and that the analytical results were almost the same to those from the simulation (average difference 0.1 Mbps). In a more complex scenario, with 13 FTP clients and a complex networks topology closely resembling the actual FCS network, the results from the simulator's network model, when comparing individual flows, differed from the results of the actual network (average difference 404.7 Mbps

or 42%) and from analytical calculations (average difference 326.7 Mpbs or 34%). Thedifference between OPNETand Iperfforthe average data transfer rate is very small. The throughput of the actual network was lower than the values attained from the simulation because of the additional overhead present in the actual network and not modeled in the simulator. A method was developed for estimating (the overhead due to) the background traffic on the bottleneck link connecting the Gillin Hall sub-network to the ITC Hall sub-network. The background traffic was estimated to be 107 Mbps in one experiment and 565 Mbps in another. The modeled network was successfully able to handle video streaming, even with the congested traffic and the background traffic. The video delay measured in the OPNET simulator was 0.7 ms, which normally allows for a very good video conferencing quality because, in order to maintain acceptable service, the delay should be equal to or less than 100 ms.

8.1 Future Work

This report provides results that give some insight into the future work on the simulator and the actual network. Further investigation will be on how to change some parameters to make the results of the simulator closer to the results of an actual network under complex network conditions. We tested the network only with default parameters.

Bibliography

[1] Mohd Raziff Abd Razak and Abdul Halim Ali. The impact of voice traffic on unikl-bmi campus network. In *Electrotechnical Conference, 2006. MELECON 2006. IEEE Mediterranean*, pages 688–691. IEEE, 2006.

[2] Omar Alsaiari. *Modeling a Campus Network and Provisioning Video and VoIP Services.* University of New Brunswick, 2012.

[3] Aleksandar Mišković and Srđan Atanasijević. Analysis of the pull method for crl download by the pki simulation model.

[4] Adarshpal S Sethi and Vasil Y Hnatyshin. *The Practical OPNET User Guide for Computer Network Simulation*, pages 127–136. CRC Press, 2012.

[5] Andre L Alexander, Alexander L Wijesinha, and Ramesh Karne. An evaluation of secure real-time transport protocol (srtp) performance for voip. In *Network and System Security, 2009. NSS'09. Third International Conference on*, pages 95–101. IEEE, 2009.

[6] Andrew Tyler. Using iperf to troubleshoot speed/throughput issues, 2011.

[7] Richard Blum. *Network Performance Open Source Toolkit: Using Netperf, tcptrace, NISTnet, and SSFNet.* John Wiley & Sons, 2003.

[8] M. W. Pearson A. J. McGregor and D. H. T. R. Lawson. Validation of the wand simulator through comparison with laboratory tests. *PAM2002 Workshop on Passive and Active measurements, (Ft Collins, Colorado, March 2002)*, 2002.

[9] Rachel Weiss. File transfer protocol (ftp) throughput testing. *JDSU*, page 3, 2008.

[10] Robert Morris. Tcp behavior with many flows. In *Network Protocols, 1997. Proceedings., 1997 International Conference on*, pages 205–211. IEEE, 1997.

[11] Shabana Razak, Mian Zhou, and Sheau-Dong Lang. Network intrusion simulation using opnet. In *OPNETWORK2002 conference*, 2002.

[12] Christina Hattingh and Tim Szigeti. End to end qos network design quality of service in lans, wans, and vpns. *Cisco Press*, 2004.

Vita

Candidate's full name: Aziz Abdullah Alshehri.
University attended: King Abdulaziz University, Jeddah City, Kingdom of Saudi Arabia, 2002-2006, Bachelor of Computer Science.

University of New Brunswick, Fredericton NB Canada, 2013 - 2014 Masters in Computer Science.