Journal of Kerbala University , Vol. 7 No.1 Scientific. 2009

Modeling Computer Laboratories in University that Contains Eight Colleges by Using OPNET Software

Assistant Lecturer / Ahmed Abdul Hadi Ahmed
Kerbala University / College of Engineering / Electrical and Electronic Department
ahmedh1333@yahoo.com

Abstract :

This paper presents the University Network Model with eight colleges, each college has a computer laboratory connected with each other, with university servers and also connected to the internet. The main applications that used in this model are web browsing by using Hypertext Transfer Protocol (HTTP), file transfer (FTP), electronic mail (E-mail), Remote Login and finally Messenger. According to the time of scholastic year in the university, three types of user profiles are defined. They are Exam Time, Weekdays and Weekends. By applying these profiles to the university model and using some main statistical to collect results, the best use of the network will get. This means that maximum throughputs with minimum time response and delay. The statistical that used in this model are delay time (response time) and throughputs.The university network model is modeled and simulated by using OPNET MODELER software package and all the results are collected from this simulator.**Keywords:** computer laboratory, HTTP, FTP, E-mail, Remote Login, Messenger, OPNET Modeler.

المستخلص :

هذا المشروع يبين نموذج لشبكة داخل جامعة تحوي ثمانية كليات, كل كلية تحوي مختبر للحاسبات. هذه المختبرات مرتبطة مع بعضها البعض ومع مجهز خدمة خاص بالجامعة وأيضا مع شبكة المعلومات الدولية. التطبيقات الرئيسية المستخدمة في هذه الشبكة هي خدمة تصفح الانترنيت, خدمة نقل الملفات, البريد الالكتروني, الدخول و التواصل عن بعد و كذلك خدمة التحدث. اعتمادا على السنة الدراسية في الجامعة, هنالك ثلاثة ملفات للمستخدمين سيتم تعريفها. هذه الملفات هي: وقت الامتحان و أيام الأسبوع و أيام نهاية الأسبوع. بتطبيق هذه الملفات و اعتمادا على بعض الإحصائيات (المتغيرات) لجمع النتائج, نستطيع ملاحظة الاستخدام الأفضل الشبكة, هذا يعني أعلى نتائج بأقل وقت تأخير و زمن الاستجابة. الإحصائيات المستخدمة في المشروع تشمل مقدار النواتج و زمن التأخير (زمن الاستجابة). نموذج الشبكة قد تم صياغته و إجراء المحاكاة عليه باستخدام برنامج الاوبنت حيث كل النتائج أخذت عن طريق هذا البرنامج.

1. Introduction

Simulation Modeling is becoming an increasingly popular method for network performance analysis. Generally, there are two forms of network simulation: analytical modeling and computer simulation. The first is by mathematical analysis that characterizes a network as a set of equations. The main disadvantage is its over simplistic view of the network and inability to simulate the dynamic nature of a network. Thus, the study of a complex system always requires a discrete event simulation package, which can compute the time that would be associated with real events in a real-life situation. Software simulator is a valuable tool especially for today's network with complex architectures and topologies. Designers can test their new ideas and carry out performance related studies, therefore freed from the burden of the "trial and error" hardware implementations [1]. Many companies have a substantial number of computers. For example, a company may have separate computers to monitor production, keep track of inventories, and do the payroll. Initially, each of these computers may have worked in isolation from the others, but at some point, management may have decided to connect them to be able to extract and correlate information about the entire company [2].The computer-communications revolution has produced several remarkable facts [3]:

- There is no fundamental difference between data processing (computers) and data communications (transmission and switching equipment).

- There are no fundamental differences among data, voice, and video communications.
- The lines between single-processor computer, multi-processor computer, local network, metropolitan network, and long-haul network have blurred.

This paper will describe the data communication networking and explains the coverage area of each type of the networks in section 2. Section 3 describes OPNET simulator and explains the important tools that used in this project. Section 4 describes the methodology of the project and how it configured and modeled. Section 5 explains the result obtained after running the simulation. Section 6 describes the conclusions that obtained from the results and sections 7 and 8 describe future work and the references that used in this project.

2. OPNET Simulator

OPNET (Optimized Network Engineering Tool) provides a comprehensive development environment for the specification, simulation and performance analysis of communication networks. A large range of communication systems from a single LAN to global satellite networks can be supported. Discrete event simulations are used as the means of analyzing system performance and their behavior. The key features of OPNET are summarized here as [1]:

- **Modeling and Simulation Cycle** OPNET provides powerful tools to assist user to go through three out of the five phases in a design circle (i.e. the building of models, the execution of a simulation and the analysis of the output data), as shown in figure 1.
- **Hierarchical Modeling** OPNET employs a hierarchical structure to modeling. Each level of the hierarchy describes different aspects of the complete model being simulated.
- **Specialized in communication networks** Detailed library models provide support for existing protocols and allow researchers and developers to either modify these existing models or develop new models of their own.
- **Automatic simulation generation** OPNET models can be compiled into executable code. An executable discrete-event simulation can be debugged or simply executed, resulting in output data.

OPNET provides four tools called editors to develop a representation of a system being modeled. These editors, the Network, Node, Process and Parameter Editors, are organized in a hierarchical fashion, which supports the concept of model level reuse.

2.1Network Model

Network Editor is used to specify the physical topology of a communications network, which define the position and interconnection of communicating entities, i.e., *node* and *link*. The specific capabilities of each node are realized in the underlying *model*. A set of parameters or characteristics is attached with each model that can be set to customize the node's behavior. A node can either be fixed, mobile or satellite. Simplex (unidirectional) or duplex (bi-directional) point-to-point links connects pairs of nodes. A *bus link* provides a broadcast medium for an arbitrary number of attached devices. Mobile communication is supported by *radio links*. Links can also be customized to simulate the actual communication channels [7].

2.2 Node Model

Communication devices created and interconnected at the network level need to be specified in the node domain using the Node Editor. Node models are expressed as interconnected *modules*. These modules can be grouped into two distinct categories. The first set is modules that have predefined characteristics and a set of built-in parameters. Examples are packet generators, point-to-point transmitters and radio receivers. The second group contains highly programmable modules. These modules referred to as *processors* and *queues*, rely on process model specifications [7].

2.3 Process Model

Process models, created using the process editor, are used to describe the logic flow and behavior of processor and queue modules. Communication between processes is supported by *interrupts*. Process models are expressed in a language called Proto-C, which consists of state transition diagrams (STDs), a library of kernel procedures, and the standard C programming language. The OPNET Process Editor uses a powerful state-transition diagram approach to support specification of any type of protocol, resource, application, algorithm, or queuing policy. States and transitions graphically define the progression of a process in response to events. Within each state, general logic can be specified using a library of predefined functions and even the full flexibility of the C language. Process may create new processes (*child process*) to perform sub-tasks and thus is called the *parent process* [8].

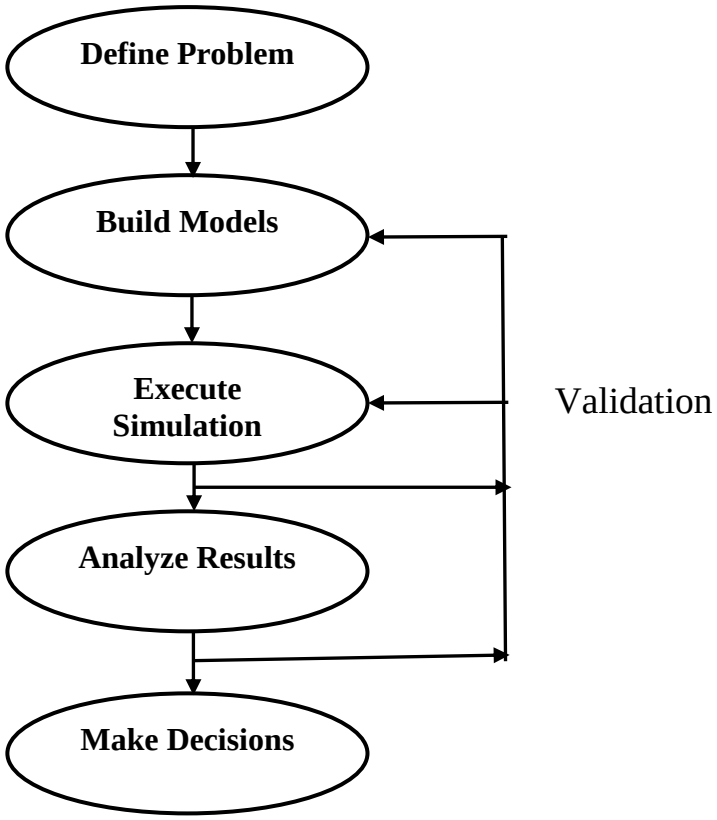


Figure 1: Modeling and Simulation Cycle

3. Methodology

The university model is developed using OPNET simulator, then this model is studied and analyzed to get the better performance and high quality of the services that provided by the this network.

3.1 Network Topology

The physical topology of the university model is shown in figure 2. It consists of a CISCO 2948G switch that used to connect all the colleges in the university and also this switch connects the colleges to the internet gateway that has two branches, one to connect the colleges to the internet and the other branch to the university's servers. The university's servers provide the students by the necessary services, these services includes browsing the university web page by HTTP protocol, file transfer by FTP protocol, e-mail and finally the remote login applications. The internet is simulated using IP32 cloud, a built-in OPNET model. To access application services over the internet, the traffic that originates from the colleges must travel via university internet-gateway router, the IP32 cloud, and the internet servers' gateway. Figure 2 illustrates the topology of the university network and its internet connections.

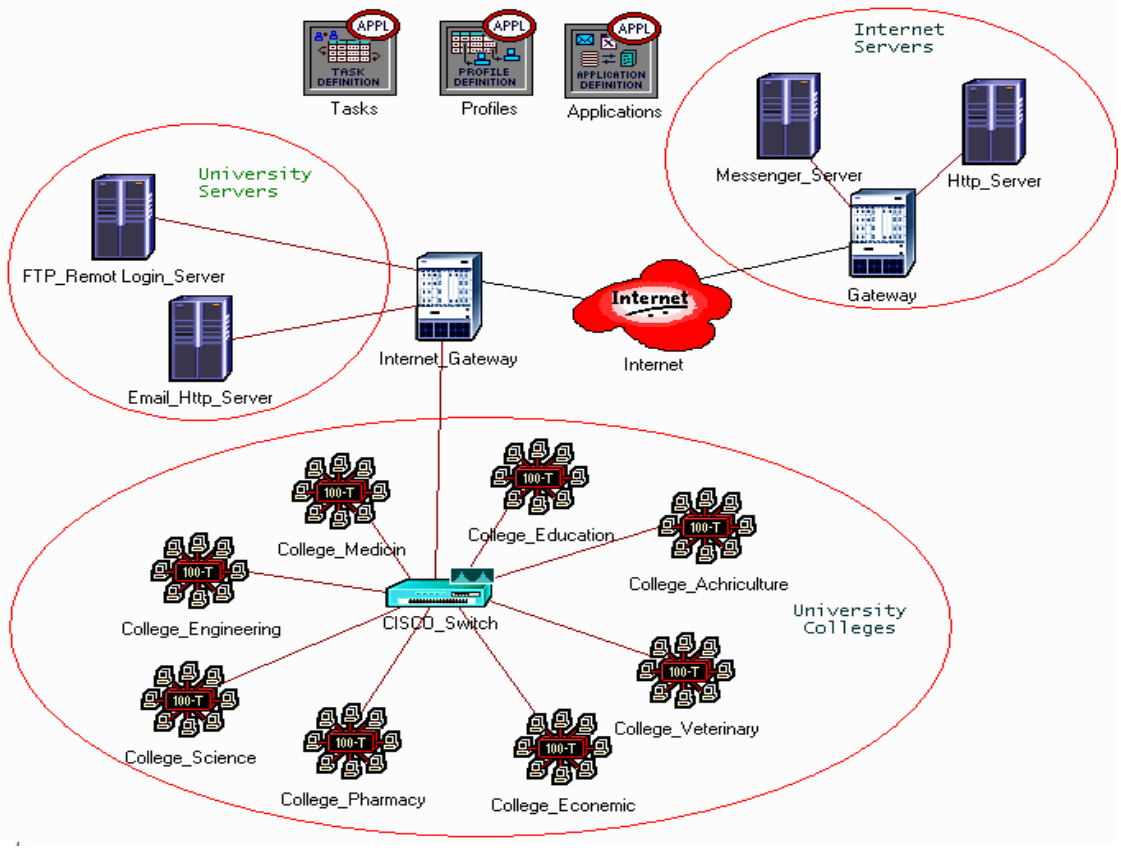


Figure 2: Physical Topology of the University Model

3.2 Applications

There are two categories of applications that require network access: (1) those that require internet access, which is called the internet applications and (2) those that obtain the necessary services on the local university network, which is called the local applications.

Internet application traffic must travel across the university internet gateway, the IP32 cloud, and finally via the internet servers gateway to reach the internet servers. The internet applications are web browsing and instant messaging. The local applications access the services they need across the CISCO 2948G switch, internet gateway and finally the university servers. The local applications are remote login, e-mail, browsing the university web pages and FTP. Students use remote login and FTP applications to access local university servers to complete various programming assignments or save data on the network drives. Students access local e-mail servers to read their daily e-mail. Finally, students access the university web pages for various information about their college.

The internet and local applications are classified further into two broad categories: Leisure and Homework to clarify and simplify the basic operation of each class. Each category divided into additional sub-class as follows:

- **Leisure:** *chatting* – which is 50% of students accessing e-mail and 50% using the Instant Messenger,
- **Leisure:** *web-browsing* – which is 60% of students browsing the Internet, 20% accessing e-mail, and 20% using Instant Messenger,
- **Homework:** *writing a term paper* – which is 60% of students using FTP program to save data on the local network drive, 10% reading e-mail, 25% browsing the Internet to find information on the subject, and 5% browsing University web pages to retrieve course-related information.

- **Homework:** *programming* – which is 60% of students using remote login to complete programming assignments, 10% reading e-mail, 25% browsing the Internet to find information on the subject, and 5% browsing University web pages to retrieve course-related information.

Table 1 summarizes the application classifications. These applications are the most frequently used by the university students.

Table 1 Application Classification

Application Category	Sub-category	Local Application	Internet Applications
Leisure	Web-browsing	E-mail (20%)	Web (60 %). Internet Messenger (20%).
	Online Chatting	E-mail (50 %)	Internet Messenger (50 %)
Homework	Writing a paper	FTP (60 %) E-mail (10 %) Web (5 %)	Web (25 %)
	Programming	Remote login (60 %) E-mail (10 %) Web (5 %)	Web (25 %)

3.3 User Profiles

There are a set of user profiles which specify how the network applications are being used in the computer laboratory for each college in the university. The user profiles are divided into: exam time, weekdays and weekends. These categories determine lab occupancy (e.g. the number of simultaneously active users in the laboratory) and the application selection (e.g. which applications are being used). The exam time happens twice during the scholastic year, at midterm and at end of scholastic year, and, during these periods, the lab is completely occupied with students working on their homework and programming assignments only. The weekdays time period models student activity on Sunday through Thursday. During this time period, the laboratory is not completely occupied, and students mostly work on their homework assignments and occasionally run leisure applications. Finally, the weekends time period includes Fridays and Saturday, when the laboratory is primarily empty. On weekends students mostly run leisure applications and occasionally work on homework assignments. User profiles are further divided based on the time of the day into three sub-categories: morning, day, or evening. Table 2 summarizes the user profiles and lists the application distribution and laboratory occupancy. For each application category, students execute the sub-category applications with equal frequency. For example, at evening during the weekdays period the laboratory is only 80% occupied and out of those users, 80% of users are doing homework assignments; 45% are writing papers and another 45% are working on programming assignments. The other 20% of the users run leisure applications; 10% are chatting online and another 10% are doing leisure web browsing. The simulation model can be easily modified to have sub-category applications used with different frequencies.

Table 2 User Profiles

User Profile	Time of the Day	Lab Occupancy	Application Categories
Exam Time	Morning	0 %	N/A
	Day	100 %	100 % Homework
	Evening	100 %	100 % Homework
Weekdays	Morning	20 %	100 % Homework
	Day	60 %	100 % Homework
	Evening	80 %	80 % Homework 20 % Leisure
Weekends	Morning	0 %	N/A
	Day	20 %	30 % Homework 70 % Leisure
	Evening	10 %	10 % Homework 90 % Leisure

3.4 Modeling Applications in OPNET

User applications are modeled via OPNET Modeler simulation software. OPNET Modeler provides standard built-in models for software applications such as *web (HTTP)*, *e-mail*, *FTP*, and *remote login*, which can be easily configured to simulate applications used in the computer laboratory However, there are no built-in models for applications such as the Instant Messenger. Thus, this application has to be implemented and configured via OPNET's Custom Application feature. To set-up and configure any application in OPNET, the user must add the Application Configuration and Profile Configuration modules. The Application Configuration module contains the application definitions, while the Profile Configuration module contains the profiles of user behavior, e.g. describes how the users employ the applications defined in the Application Configuration module.

3.4.1 Configuring Web (HTTP) Applications

To configure a web browsing application, the application named HTTP should be selected from the list of built-in models. OPNET provides pre-set configurations such as: Light Browsing, Heavy Browsing, Searching, or Image Browsing. In addition, OPNET allows configuring web applications via parameters such as: HTTP Specification which defines the version of HTTP protocol, Page Interarrival Time which specifies time in seconds between consecutive webpage downloads, Page Properties which models properties of the webpage by specifying the number and the type of objects. Figure 3 illustrates steps for configuring a web application in OPNET.

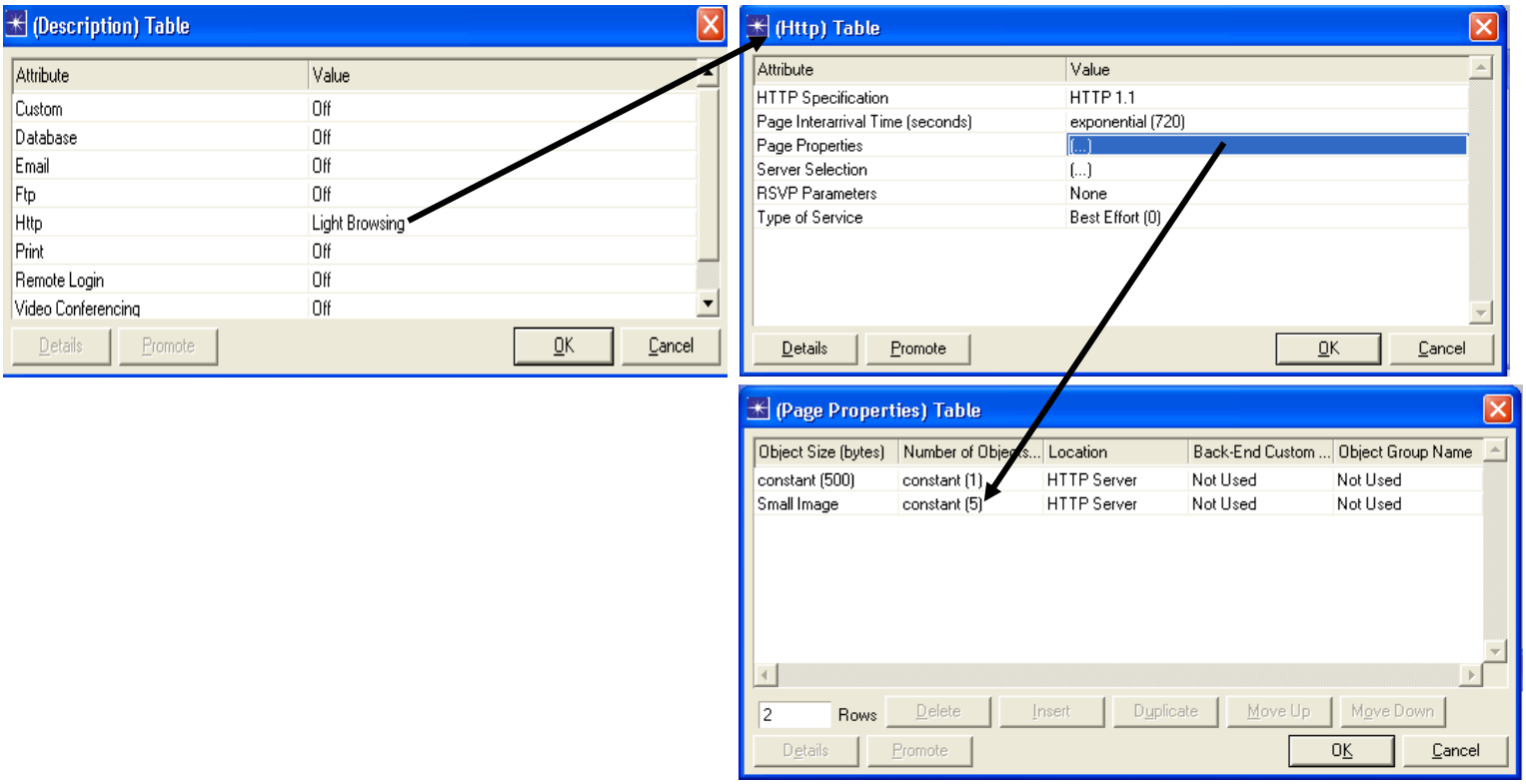


Figure 3: Configuring a Web Application in OPNET

3.4.2 Configuring E-mail Applications

OPNET provides preset configurations for e-mail application as well. The preset E-mail configurations include Low Load, Medium Load, and High Load as shown in figure 4. The model also allows custom configuration via parameters such as Send(Receive) Interarrival Time which specify the amount of time in seconds between consecutive sent (receive) operations, Send(Receive) Group Size which determine the number of e-mails messages per single sent(receive) operation, and E-Mail Size which defines the size of e-mail message in bytes.

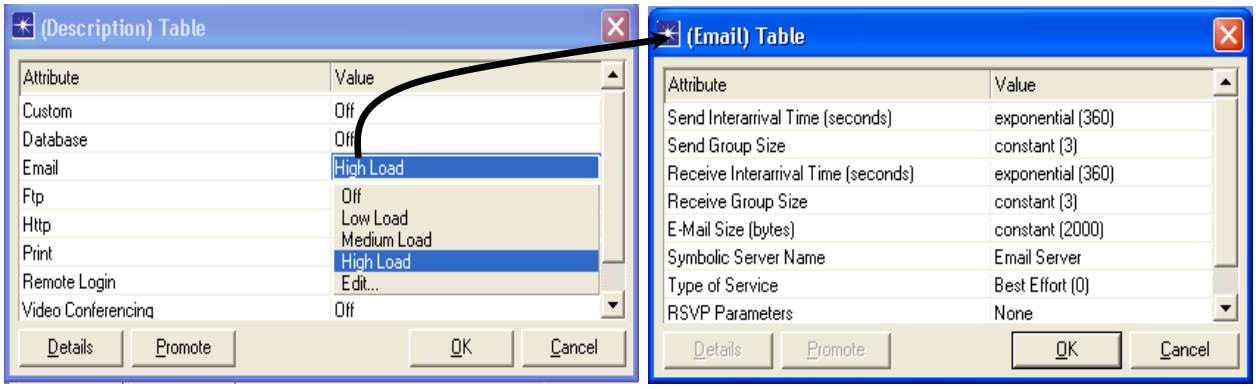


Figure 4: Configuring an E-mail Application in OPNET

3.4.3 Configuring FTP Applications

OPNET also provides preset configurations for FTP application. The preset FTP configurations include Low Load, Medium Load, and High Load as shown in figure 5. The model also allows for a custom configuration of FTP applications. The Command Mix parameter specifies the ratio

between the number of get commands and the total number of FTP requests. For example, when the Command Mix parameter is set to its default value of 50% then half of FTP requests will be to get (download) data and the other half to put (upload) data. The Inter-Request Time parameter is the time in seconds between consecutive FTP requests. The File Size defines the size in bytes of the FTP file to be transferred.

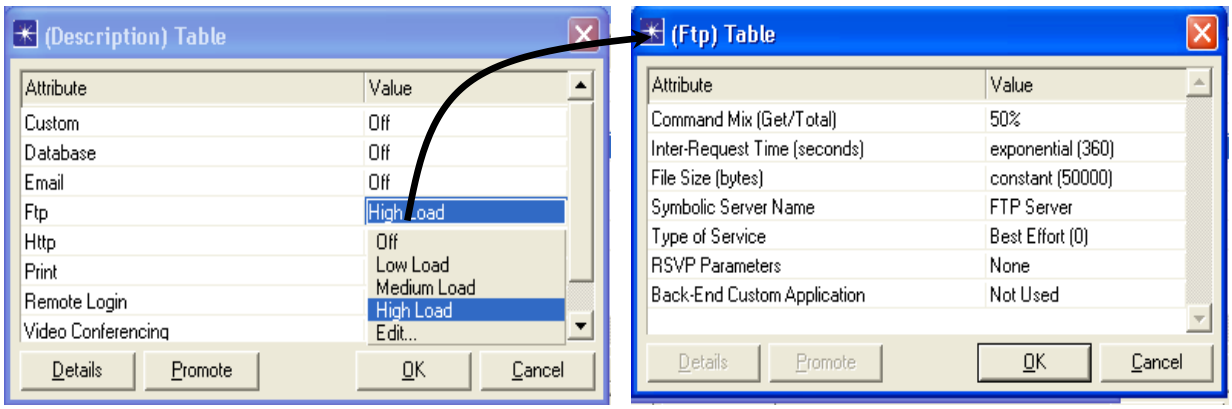


Figure 5: Configuring an FTP Application in OPNET

3.4.4 Configuring Remote Login Applications

Similar to E-mail and FTP applications, OPNET provides preset Low Load, Medium Load, and High Load configurations as well as the ability to specify user-defined settings for Remote Login application as shown in figure 6. The Inter-Command Time parameter specifies the time in seconds between consecutive commands during the remote login session. The Terminal Host) Traffic parameter defines the size in bytes of each command generated at the terminal (host).

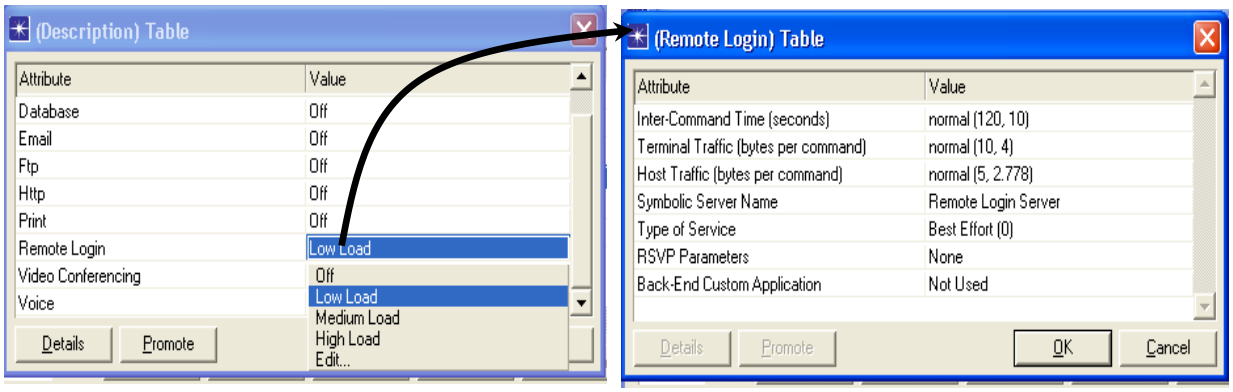


Figure 6: Configuring a Remote Login Application in OPNET

3.4.5 Configuring Instant Messenger Applications

Instant messenger is not a part of the standard OPNET applications models and it is more complex to simulate. The instant messenger software is modeled using OPNET's Task Configuration module which specifies the packet exchange between the application nodes.

The Task Configuration module specifies a basic unit of user activity wuhtin the custom application such as a server login or reading an e-mail message. The custom Application module specifies applications that use the tasks defined in Task Configuration module. The instant messenger behavior is modeled via three basic tasks:

i. Authentication: login in to authorization (Messenger) server.

ii. Server Login: after successful authentication the messenger user login to the messenger server.

iii. Server Messaging: message exchange among messenger users.

These tasks used to implement Messenger application that used to exchange messages among users. The Authorization and Server Login tasks are configured as a request-response message exchanges

between a single client and the corresponding server. Figure 7 shown Application and Task Configuration.

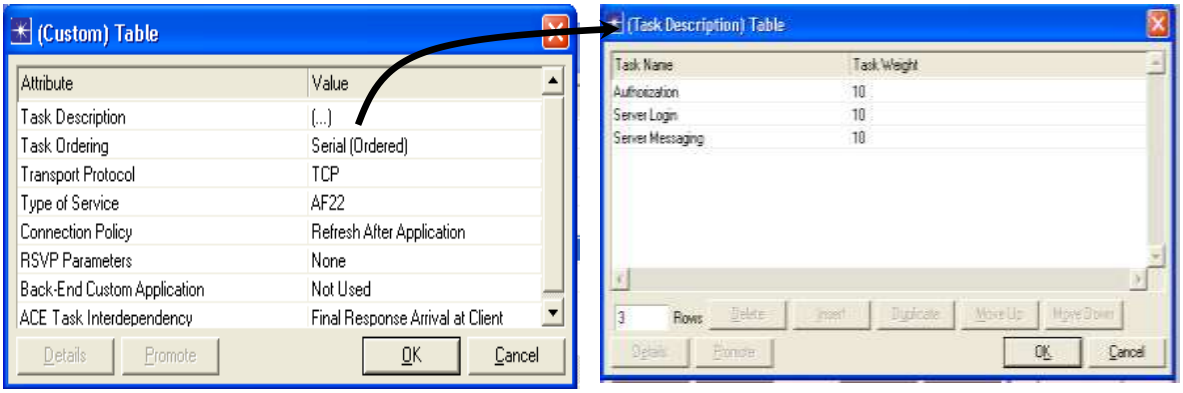


Figure 7: Configuring a Messenger Application in OPNET

3.5 University Network Node Model

The node model specifies the internal structure of a network node. Typical nodes include workstations, packet switches, satellite terminals, remote sensors. Nodes can be fixed, mobile or satellite type. The node model of the college of engineering is shown in figure 8. The node model is organized according to OSI model (Open System Interconnection) and is divided into layers beginning from application layer and ending with physical layer.

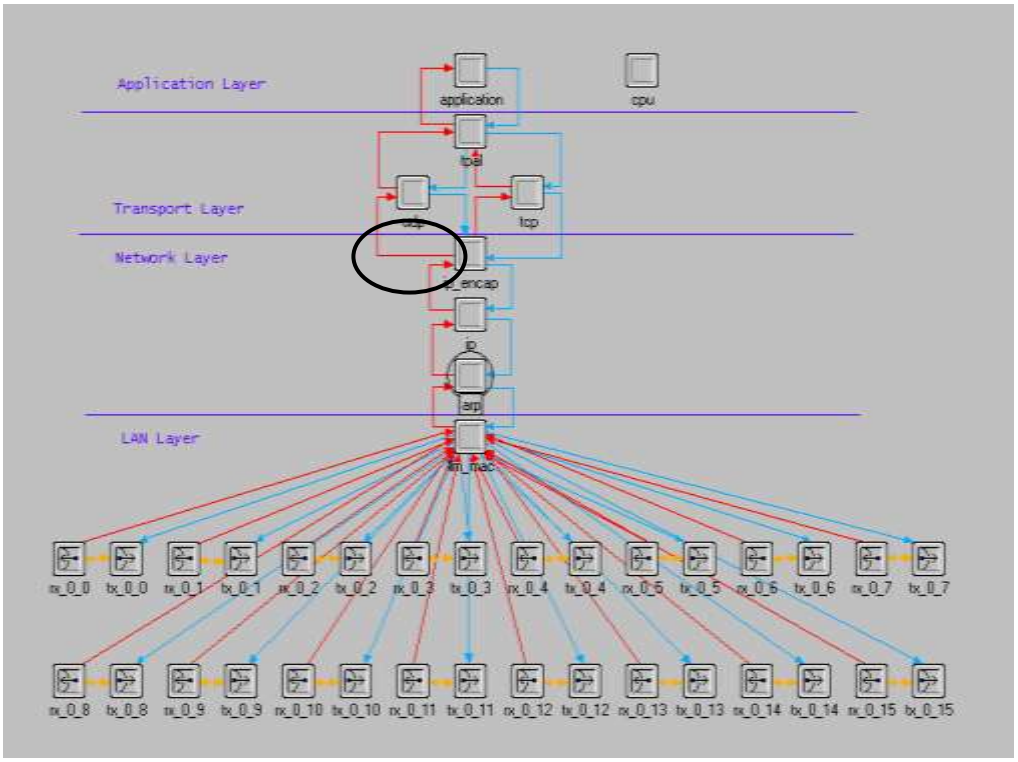


Figure 8: The Node Model of the College of Engineering

3.6 University Network Process Model

Process models are used to specify the behavior of a processor and queue modules, which exists in the Node Domain. A module is modeled as a finite state machine (FSM). FSM consists of states with transitions and conditions between them. From figure 8, the process model of the network layer (represents by ip_encap node) is shown figure 9. This figure represents the FSM of of the

network layer. The source code for the initial state of it is programmed by using C++ languages , as shown in figure 10.

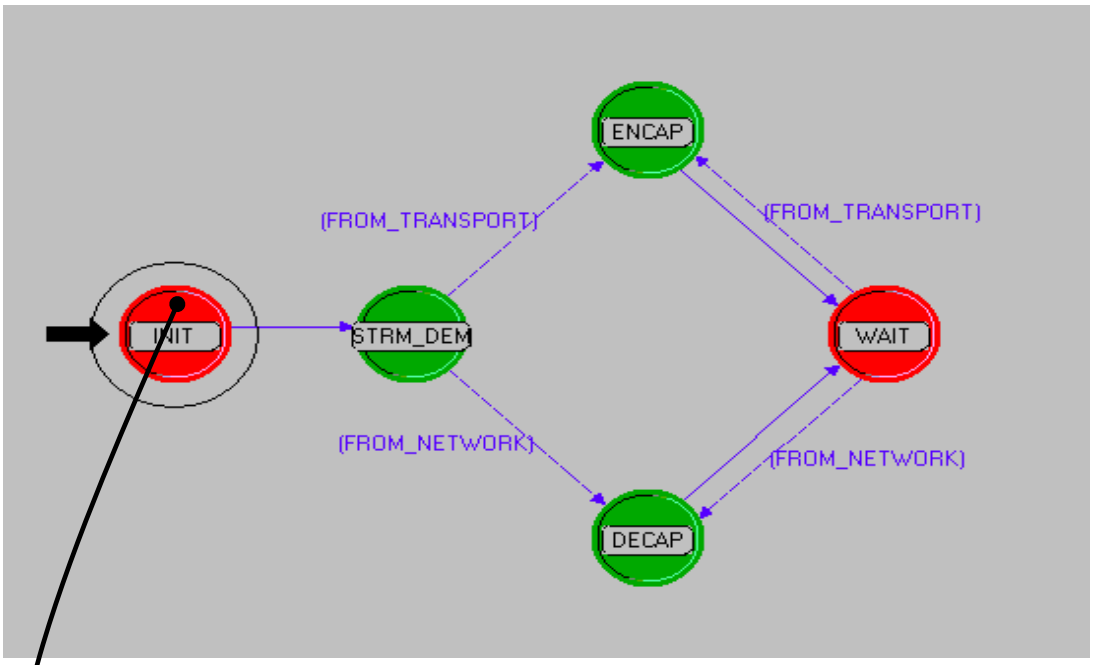
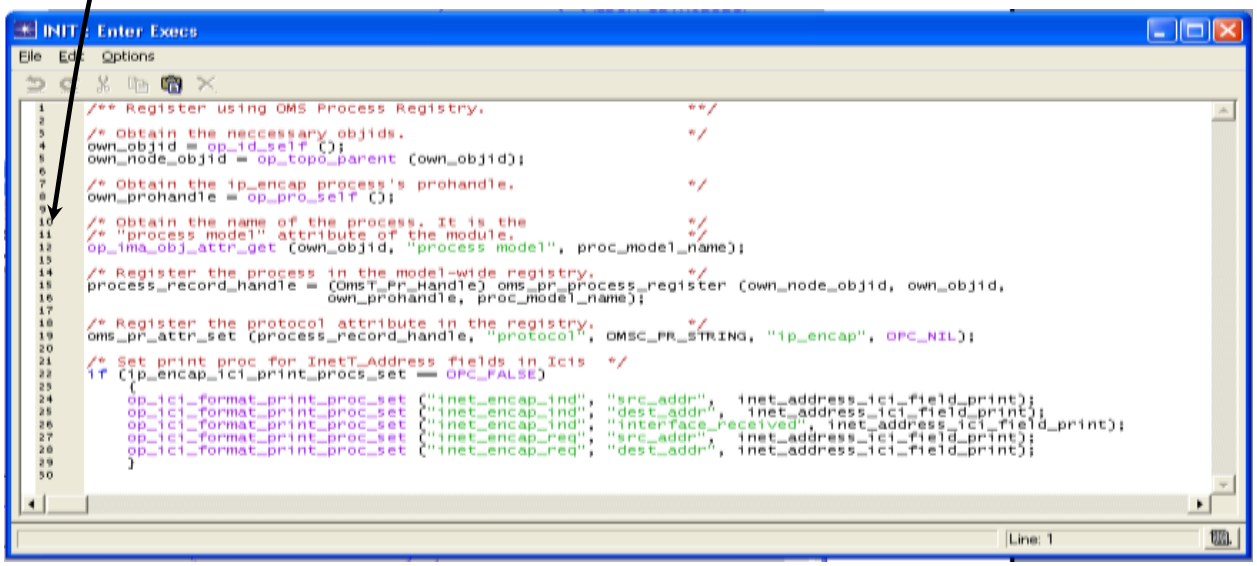


Figure 9: The Process Model of the ip_encap Node



4. Results

After running the simulation model for 1800 second and according to the user profiles, the results are collected and analyzed. The parameters that used to make a comparison among the results is delay time and throughput. So, following is the result of each user profiles:

1.**Exam Time**: this profile has no user in the morning (0 % occupancy) because of the examination, while in the day and evening the lab is full and all the clients are used homework applications. The result of this profile is shown in figures 11, 12, 13, 14.

2.**Weekdays**: in this profile the lab has different occupancy according to the time of day. In the morning the lab occupancy is 20 %, while in the day 60 % and in the evening 80 %. The type of application in morning and day is 100 % homework where in the evening 80 % homework and 20% leisure. The result of this profile is shown in figure 15.

3.**Weekends**: in this profile the lab has also different occupancy according to the time of day. In the morning the lab occupancy is 0 %, while in the day 20 % and in the evening 10 %. The type of

application in day is 30 % homework and 70 % leisure where in the evening 10 % homework and 90% leisure. The result of this profile is shown in figures 16, 17 and 18.
The comparison among the three user profiles is shown in figures 19, 20, 21 and 22.

5. Conclusion

This project discusses the steps in modeling the University network and describes the methodology developed for modeling applications and user profiles in a computer laboratory, and explains the steps for modeling a non-standard application such as Instant Messenger using OPNET Modeler. This project has three scenarios and each scenario has sub-scenarios depends on the time of the day and the time of the scholastic year of the University.
By making a comparison between the user profiles that used in this project; Exam Time, Weekdays and Weekends; the results obtained is as follows: the throughput is maximum when the lab is fully occupancy and the users obtained their services from University servers and Internet severs, this can be shown in figure 16, where the maximum throughput is about 160 (Packet/second) in the weekday and evening user profile. This result leads to that the traffic sent and received between any college and the university servers and internet severs throughout the CISCO switch and the gateways in this profile is maximum. The next important statistical is the delay time, and from the comparison of the three user profiles, the conclusion is that the delay has minimum difference between the three scenarios and the average of them is that 0.0003 second. This leads to the benefit that says, the download and upload response time for email application have minimum difference between the user profiles, and the page response time for http application also has minimum difference between the user profiles.

6. Future Works

This paper has described the first step for modeling a university network by using OPNET Modeler Simulator that have been developed for a Computer Laboratory for each college in the university.
For the future the university network can be enhanced by adding wireless and mobile techniques to connect the colleges with other parts of the university and to connect the university with other governmental companies in the town. The network can be also enhanced by using fast links and more than one switch to connect the university parties to overcome the bottleneck that might happen.

7. References

1. OPNET Technologies, Inc., "*OPNET Modeler Product Documentation"*, release 10.5.
2. Andrew S. Tanenbaum," *Computer Networks*", Fourth Edition, 2003.
3. William Stallings, *"Data and Computer Communications"*, Fifth Edition, 2003.
4. Halsall, F. *"Data Communications, Computer Networks", and Open Systems.* 1996.
5. Stallings, W. "*Local and Metropolitan Area Networks"*, Fifth Edition. 1997.
6. Larry L. Peterson and Bruce S. Davie. *"Computer Networks: A System Approach"*, 3rd Edition. Morgan Kaufmann, 2005.
7. The World's *"Leading Network Modeling and Simulation Environment"*, OPNET Technologies, 2004, http://www.opnet.com/products/modeler/home.html.
8. J. Potemans, J. Theunis, B. Rodiers, B. Van den Broeck, P. Leys, E. Van Lil and A. Van de Capelle, "*Simulation of a Campus Backbone Network, a case-study*", *OPNETWORK 2002*, Washington D.C., USA, 2002.

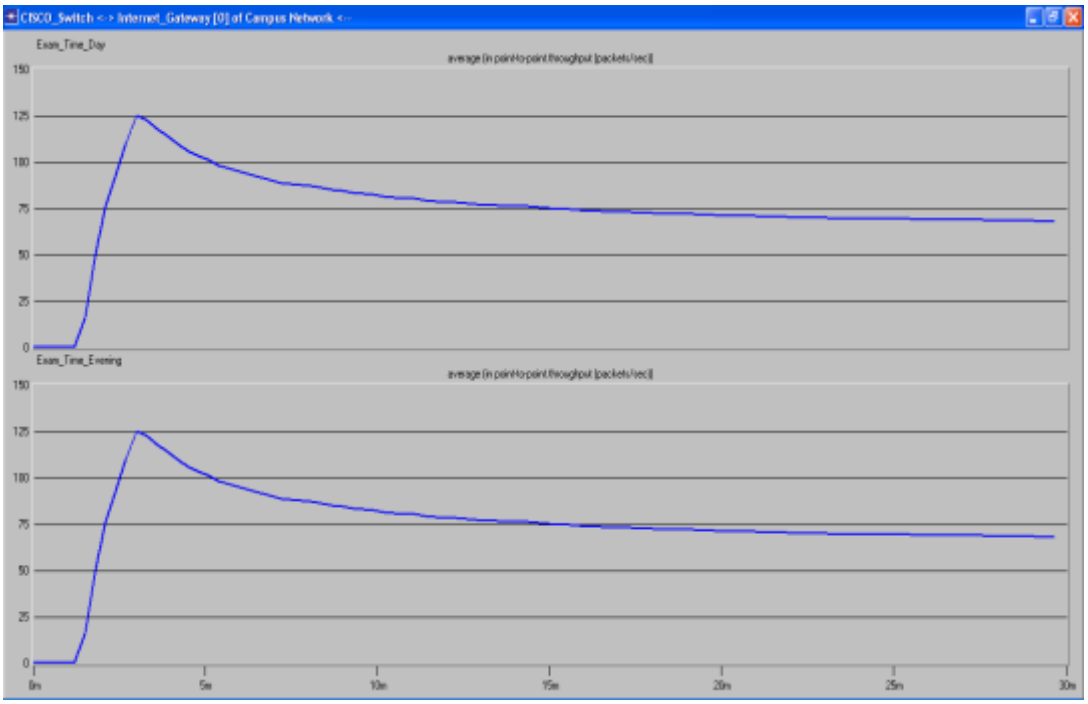


Figure 11: The Average Throughput (Packet/Sec) between Internet_Gateway → CISCO Router

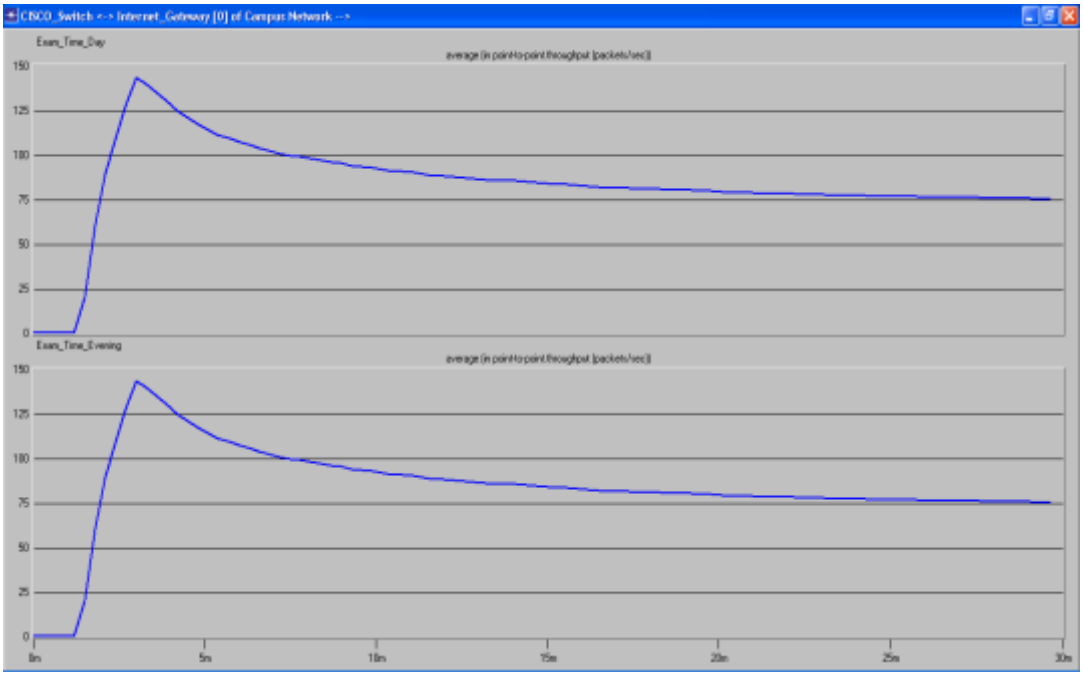


Figure 12: The Average Throughput (Packet/Sec) between CISCO Router → Internet_Gateway

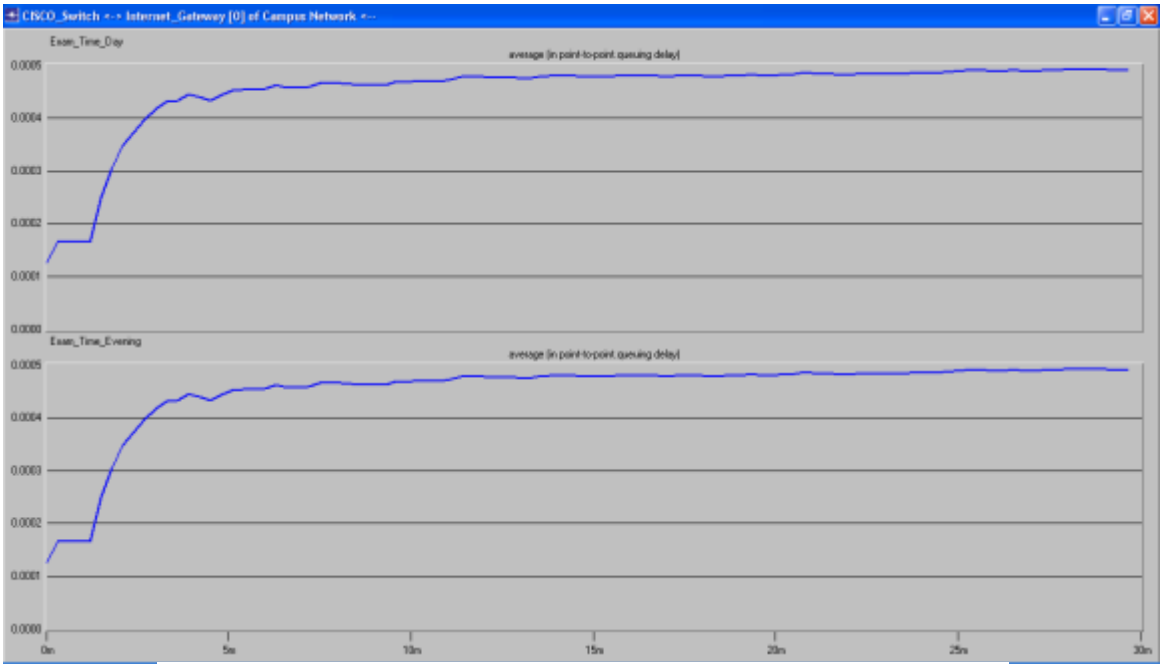


Figure 13: The Average Queuing Delay (Sec) between Internet_Gateway ⟶ CISCO Router

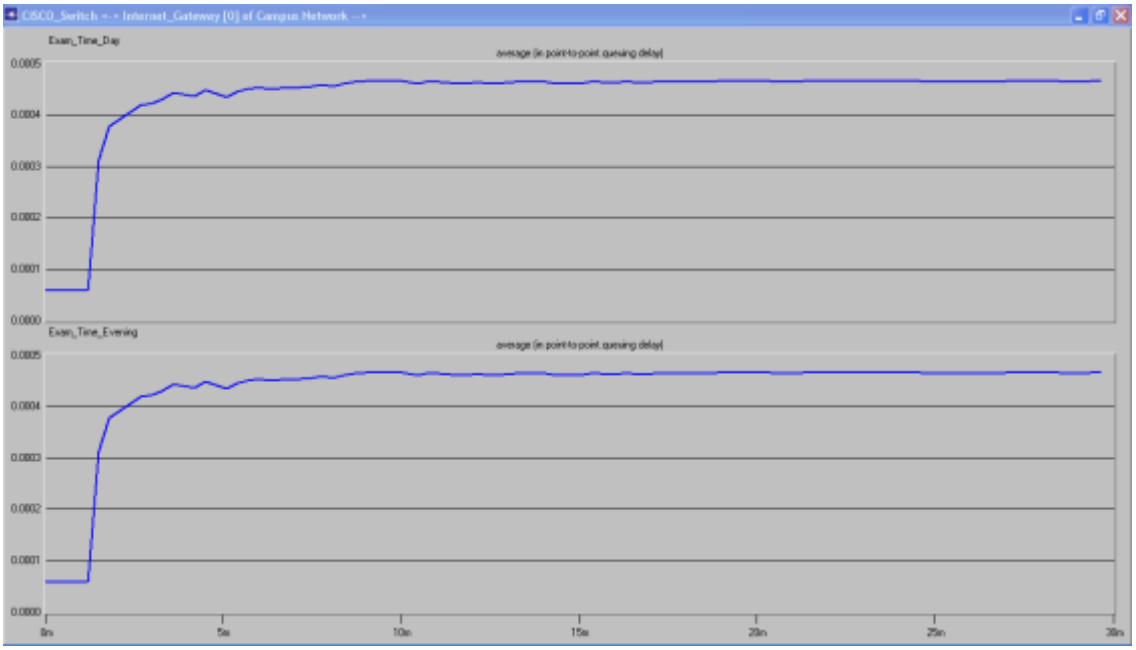


Figure 14: The Average Queuing Delay (Sec) between CISCO Router ⟶ Internet_Gateway

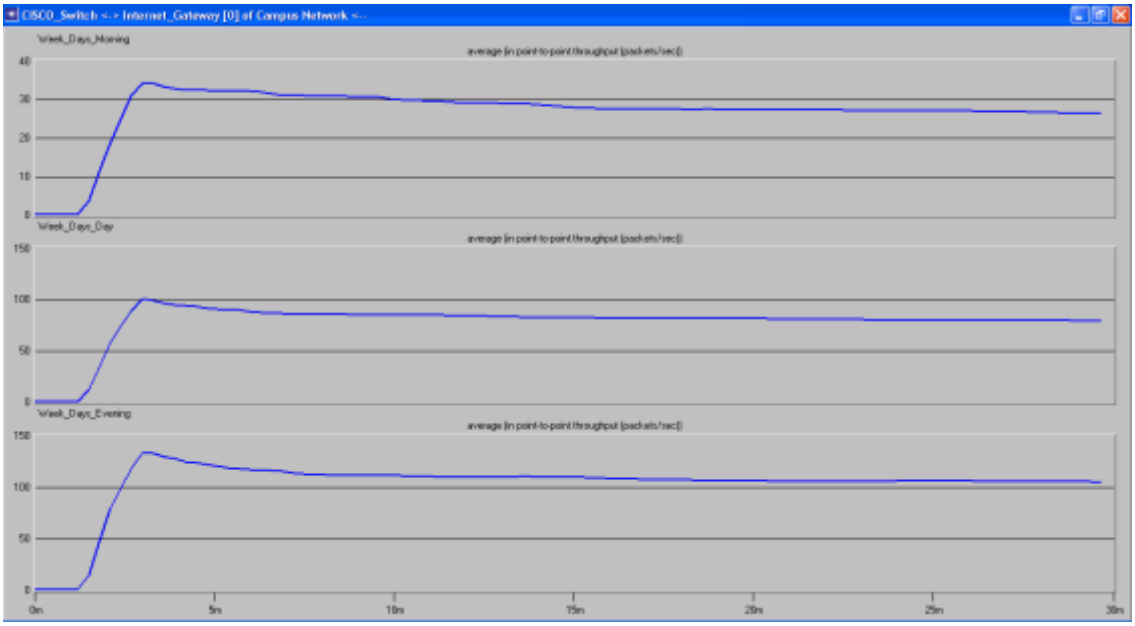


Figure 15: The Average Throughput (Packet/Sec) between Internet_Gateway ⟶ CISCO Router

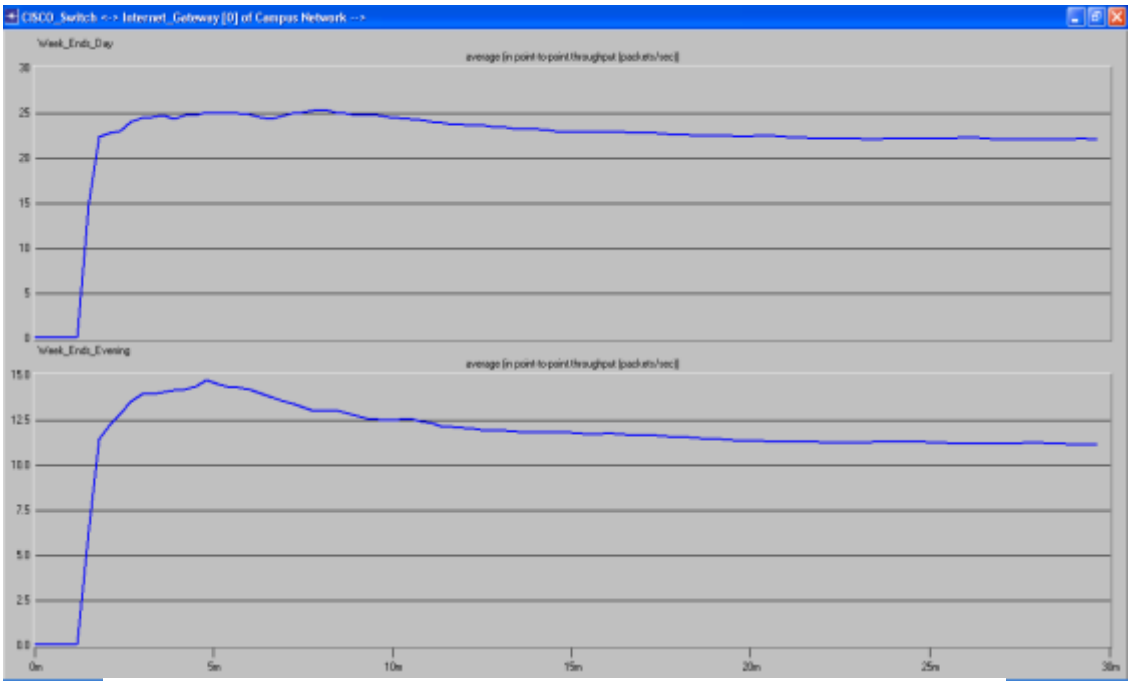


Figure 16: The Average Throughput (Packet / Sec) between CISCO Router ⟶ Internet_Gateway

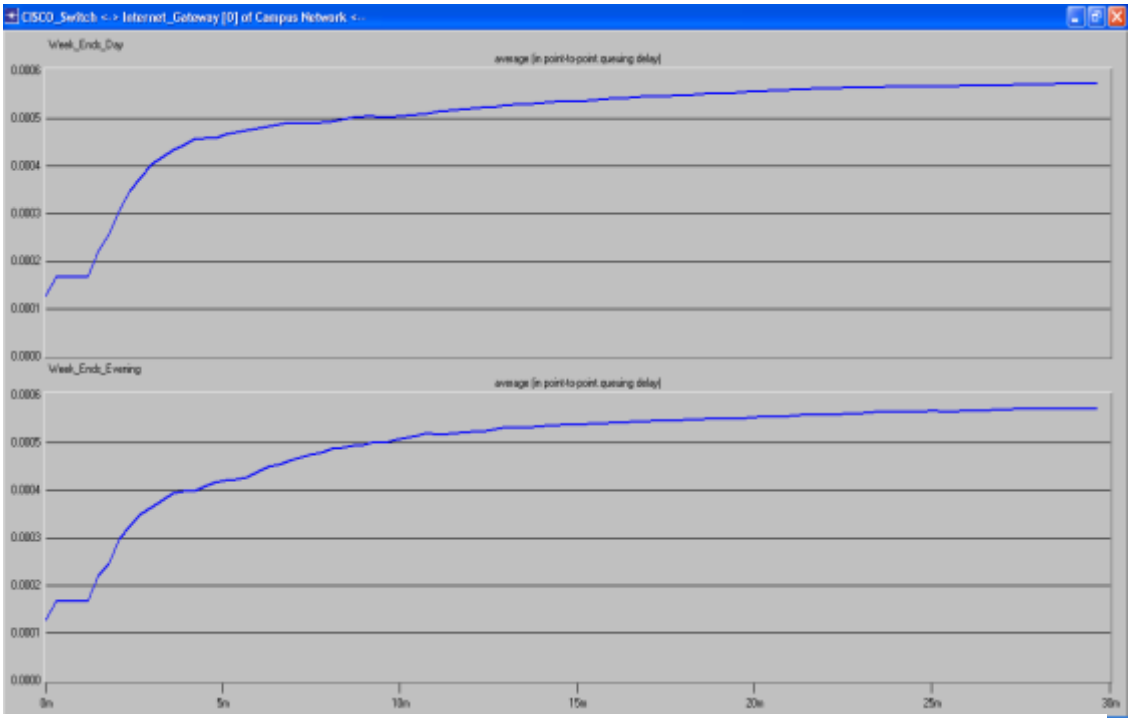


Figure 17: The Average Queuing Delay (Sec) between Internet_Gateway ⟶ CISCO Router

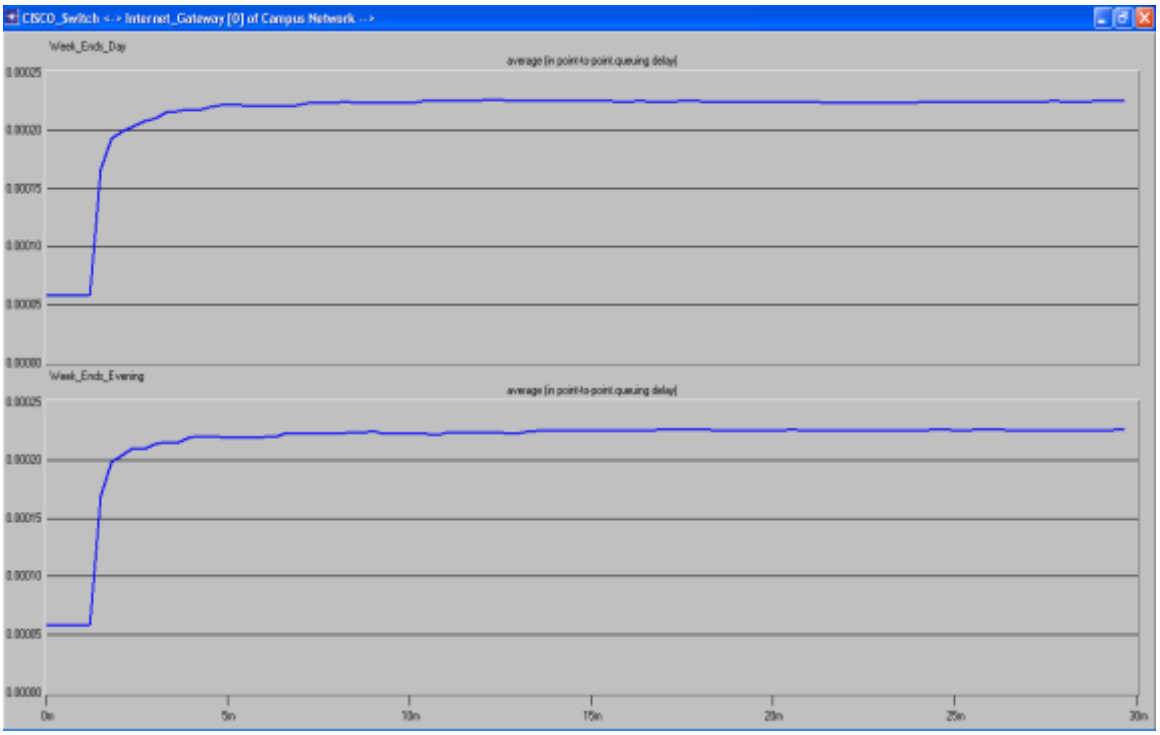


Figure 18: The Average Queuing Delay (Sec) between CISCO Router ⟶ Internet_Gateway

Figure 3: Configuring a Web Application in OPNET

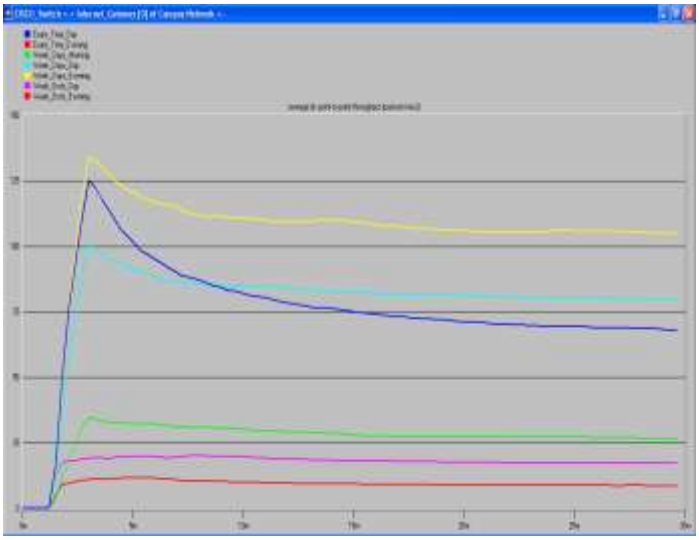


Figure 19: The Average Throughput (Packet / Sec) between Internet_Gateway ⟶ CISCO Router

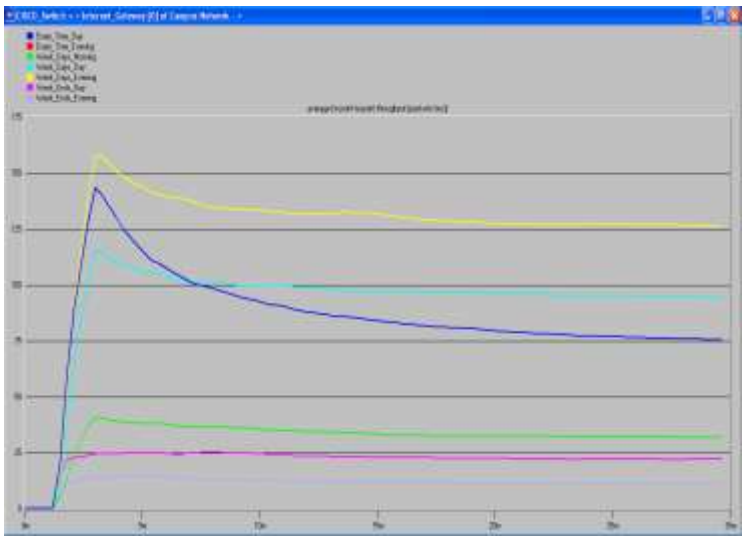


Figure 20: The Average Throughput (Packet / Sec) between CISCO Router ⟶ Internet_Gateway

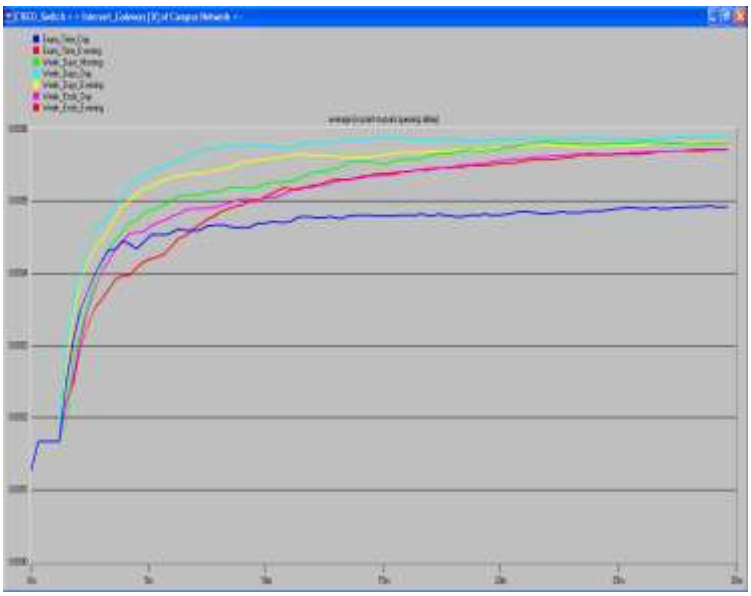


Figure 21: The Average Queuing Delay (Sec) between Internet_Gateway ⟶ CISCO Router

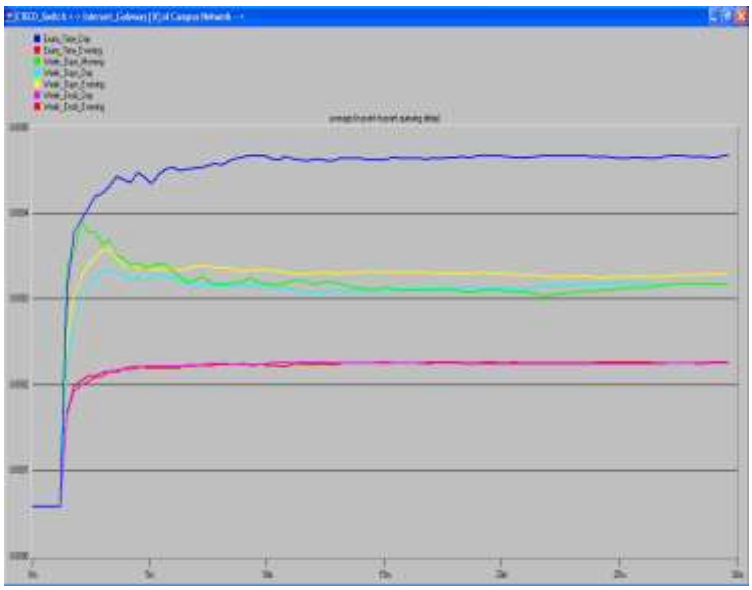


Figure 22: The Average Queuing Delay (Sec) between CISCO Router ⟶ Internet_Gateway