Table of Contents:

Chapter 1

Abstract

1.0 Abstract:

The main objective of this project is to model a small enterprise network using different topologies and characteristics (such as delay, bandwidth and the amount of traffic) and study its performance, in order to fully understand the effects of changing each of these characteristics on the Quality of Service (QoS). The second objective is to understand and implement the concept of priority within the modeled network, and also study its effect on the Quality of Service. The OPNET modeler will be used to model the different networks, run different simulations and output the results.

Chapter 2

Introduction

2.0 Introduction:

Due to the rapid increase in all types of communications and information transmission in the recent years there was a need to invent some kind of solution that would conserve time and money before making sure of the feasibility of building such networked systems. Thus, simulation programs appeared in order to help the network manager to build the most efficient network for the type of job that is intended for it to be used in. That way, by testing the model that the network modeler designed and by subjecting it to different attributes and configurations such as changing the traffic and performance parameters, would show the network manager the pros and cons of this specific network that has been designed and how to improve it. The OPNET simulation program will be used in this project to model the different networks, run simulations and output the results. The main issue that will be discussed is the effect of using different parameters when modeling the network and the implementation of priority on the Quality of Service.

2.1 Main Challenges:

- Modeling two different networks based on different topologies.
- Understanding the effects of changing the different parameters (such as traffic type and links bandwidth) on the two networks.
- Finding the similarities and difference between the different topologies.
- Understanding and implementing the concept of priority.
- Integrating the priority concept with the modeled networks.

2.2 Chapters Overview:

Chapter 3 – Background and Theory: This chapter contains all the information that the reader has to know in order to be able to understand all the technical terms and theories mentioned inside the report.

Chapter 4 – System Overview: Deals with explaining the requirements and specifications of the OPNET program and the modeled networks.

Chapter 5 – Theoretical Background: Explains the theory behind priority in details. It also shows how the basic OPNET editors work.

Chapter 6 – Design Issues: It shows examples of network components and what the networks that have been designed in the project.

Chapter 7 – Implementation and Testing: This chapter explains the details of all the network scenarios were implemented in this project as well as the implementation priority concept that was introduced to the new network.
This chapter explains also two types of testing that were applied to the system.

Chapter 8 – Discussion: Deals with investigation of the results obtained from running different simulations on the both of the networks that were implemented (star and cascade topologies)

Chapter 3

Background and Theory

This chapter contains introductory and basic information that will help the reader to understand the more complicated topics (different network components and explanation of many other theories used in details) that will be pointed out throughout the report.

Networks:

Number of workstations and/or other end-systems connected to each other to facilitate communications between each other. [8]

Open System Interconnect (OSI) Model Layers:

It is divided into seven layers in order to make networking easy to conceptualize and understand. It is called a model because usually vendors combine certain layers with each other in order to reduce the redundancy and to neglect unnecessary layers. The layers are:

1. Physical Layer:

   This is the actual physical connection between network devices. It deals with pulses and the volts as 0s and 1s. Ethernet cables and wires (10BaseT and 100BaseT) are examples of physical layer.

2. Data-Link Layer:

   This responsibility of this layer is to transmit packets from the network device to physical connection. In this level, the information is divided into frames. These frames go through error checking and flow control to make sure that the packets are in the correct format once they are transmitted. Switch is an example of this layer.

3. Network Layer:

   Routers and IPs are examples of network layer. Here, the packets are directed and route is assigned to each one of them. It also makes sure that the packets travel from node to another in the correct form (without damage or loss).

4. Transport Layer:

   This layer ensures that the packets reach the destination with no errors. Lost or damaged packets are retransmitted, as this layer requires from the source. Transport Control Protocol (TCP) is an example of transport layer.

5. Session Layer:

   This layer makes sure that the connection is still alive, even though there is no transmission on that connection. Usually, this layer is combined with the transport layer because there is no need to have it in a separate layer.

6. Presentation Layer:

   Establishing common data format is done here. It works as a translator between, for example, EBCDIC format and ASCII format. Encryption is an example of this layer.

7. Application Layer:

   It is the layer where normal users of network features are familiar with. For example, transferring files and emails are network applications. [2]

Traffic:

Is the load on a communications device or system. One of the principal jobs of a system administrator is to monitor traffic levels and take appropriate actions when traffic becomes heavy. [8]

Load:

In networking, load refers to the amount of data (traffic) being carried by the network.[8]

Throughput:

The amount of data transferred from one place to another or processed in a specified amount of time. Data transfer rates for disk drives and networks are measured in terms of throughput. Typically, throughputs are measured in kbps, Mbps and Gbps.[8]

Utilization:

Percentage of the total resources of the device (such as bandwidth, storage or memory) that has been used.[8]

Routers:

They are devices, which forward packets. They are used to connect LANs to each other, WANs to each other or a LAN to the Internet provider. This means that routers have the ability to deal with traffic coming from the Internet. Also, routers receive packets from the source and retransmit them to their exact destination according to certain routing tables stored inside the routers. Therefore, packets are not distributed to all peers. To connected end-systems to routers, a unique IP must be assigned to each of them. Routers are layer-3 equipment. [2] [3] [8]

BRouters:

They are the same as routers, except that they are able to deal with MAC addresses along with IPs. [2]

Switches:

They are LAN dedicated equipment. They are used to connect end-systems via star topology. The received packet is retransmitted to exact destination. This is done by using MAC addresses. Switches are not able to be connected with the Internet provider because that requires an equipment having the capability to identify IP addresses, while switches are not able to do so. Switches are layer-2 equipment. [2] [8]

Local Area Networks (LANs):

They reside in a specific limited geographical area, such as a building and small enterprise. The purpose of them is to connect end-systems within the limited area to each other. [2] [8] [9]

Wide Area Networks (WANs):

WANs consist of multiple of LANs. They are not limited by the geographical constraints. [2] [8]

Servers:

They are the network equipment, which provide certain services to other network equipment and workstations within the network. [2] [8]

Packets:

When data are to be transmitted through a network (e.g. the Internet), it is divided into pieces. It is not necessary for these pieces to use the same path in order to reach the destination. These pieces are called packets. The advantage of having packets is to avoid congestion and crowded paths. Also (in certain networks), an error control is assigned to handle the lost and corrupted packets. Generally, packets are divided into two major parts, headers payloads. The payload contains the data to be transmitted, while the header usually consists of error and flow control bits. [2] [8]

Workstations:

This is another name for computers. It is used to differentiate computers connected to a network from personal computers, which are not connected to a network. [8]

X.25:

It is a standard in which divides data into variable-size packets. It supports error correction and flow control. The flow control is implemented by using a sliding window protocol. On the other hand, errors are detected depending on not receiving the acknowledgement of a specific packet or receiving its negative acknowledgement. X.25 supports two error correction facilities, reset and restart facilities. The reset is used to correct minor errors, while the restart is used when a major error occurs. [2] [8] [9]

Asynchronous Transfer Mode (ATM):

It is a standard, which divides data into fixed-size cells (packets). Each cell is divided into two parts, 5-byte header and 48-byte payload. The fixed size simplifies the requirements of processing the cells. The error control of the ATM is minimal and also it has flow control capabilities. The physical transmission path is divided into multiple virtual paths. Each virtual path is a bundle of virtual channels. Each of the virtual path and virtual channel has an identifier so that the cells do not get lost. ATM is used for all types information, all the way from data to video conferencing. It benefits from the periods when the link is not fully used. For example, when people talk on phone, there are certain periods that none of the both sides talk. ATM makes use of these periods and employs them in other services. [2] [8] [9]

Sliding Window Protocol:

Here, both sender and receiver have a window, which is a buffer. For example, if the buffer size for both sender and receiver is 3, three packets will be sent from source to destination. If the destination receives the correct packet, it will send an acknowledgement to the source. If this packet is at the beginning of the window, then the window will move one step. If the packet is lost or damaged, it is retransmitted and the window does not move until the packet is correctly received. [9]

Internet Protocol (IP):

IP is the most widely used protocol. It is usually employed in networks, which are to be connected to the Internet. The IPv4 consists of four bytes. It uniquely identifies each host. [2] [8]

Media Access Control (MAC) Address:

This address is unique for each network interface card (NIC). It is assigned to the NIC when it is manufactured. It is represented in six bytes. The first three bytes identify the manufacturer, while the last three bytes identify the product of that specific manufacturer. [2] [8]

Priority:

Generally it means treating certain packets that belong to specific services more importantly than any packets that belong to other services. In this project the network implements two types of time priority, which deals with the order in which cells leave the waiting area and enter the server for onward transmission. Thus the main focus for time priority is on the delay performance that is how long does a packet wait inside the queue.

Two types of time priority were introduced to the network in order to provide several class of services are Head of line which indicate that there is high priority packet at the head of the queue that server must serve, the other type of time priority is threshold base which was designed to solve the problem with the first type where the low priority packets will have a poor service since the sever most of the time is serving the high priority packets, this problem can be solved by assigning a threshold value which tell the server when to sever only the high priority packet (number of high priority packets greater than threshold value) while keep serving both type of packet equally in the normal case.

Chapter 4

System Overview

4.1 Requirements:

A UNIX / Windows system is needed for the OPNET simulation program to run (depends on the version).

C++ language and some built in functions (system calls) in the OPNET simulation program are used to program the different processes that are going to be used.

Function: The simulation program is used to model a network in order to anticipate errors before they happen and increase the maximum efficiency of the network built by analyzing the output of the different testing scenarios.

Performance and Behavior: The performance depends on the size of the network, since the time taken to simulate the network is directly proportional to its size.

Constraints: The OPNET simulation program runs only on one workstation due to license constraints, which forces sharing the workstation one at a time.

Interfaces: The OPNET simulation program uses a graphical user interface, except for the process programming part, which uses C++.

4.2 Specification:

4.2.1 Functionality of the system:

The OPNET program allows the user to design (model) a network that is going to be built in real life. The program then paves the way for the user to change and configure many of the network attributes before starting the simulation. When the user reaches the end of the simulation phase graphs will be shown depending on the aforementioned configurations. Finally the user is allowed to change the configuration and statistics until the most efficient network design is reached.

One of the main objectives to be achieved is to introduce the concept of priority to the network model where the important user such as VIP users will have more priority than normal user in term of bandwidth, delay and such.

4.2.2 System Interfaces:-

Input Data:

The inputs were broken up into stages. The network components used were workstations, a server, 3com switches with 102 interfaces and Cisco routers with two interfaces.

a. The first stage represents the first floor, which consists of star topology as a connection of workstations, a switch and a server. The links used are 10Baset links.
b. The second stage is the first stage with the addition of two more floors. Each of these extra floors has a number of offices, which are represented by ready-made

LANs. The offices are connected to each other in star topology with a centralized switch.

c. In the third stage, the offices in the second stage are connected to each other in cascade topology instead of star.
d. The offices in the third stage are substituted with custom-made LANs, resulting in the fourth stage.
e. The fifth and sixth stages are the same stages two and three, but the links are changed from 10BaseT to 100BaseT.
f. In the seventh and eighth stages, the number of floors in the stages five and six is increased to 6 floors.

Output Data: - The program will show different graphs depending on the parameters that were chosen to be displayed.

Chapter 5

Theoretical Foundations

5.1 Theoretical Model:

In order to model and analysis a network using OPNET program, four basic steps must be considered:

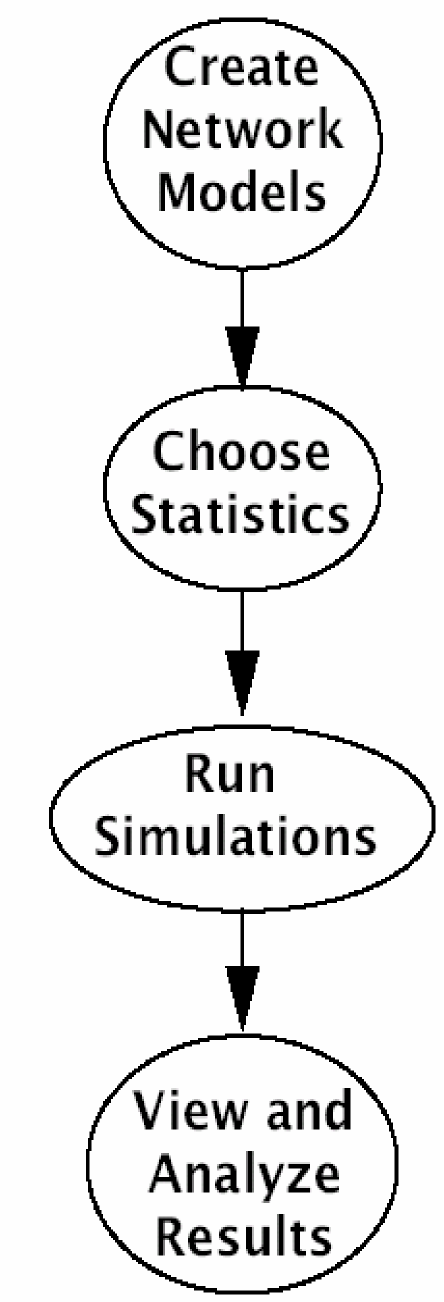


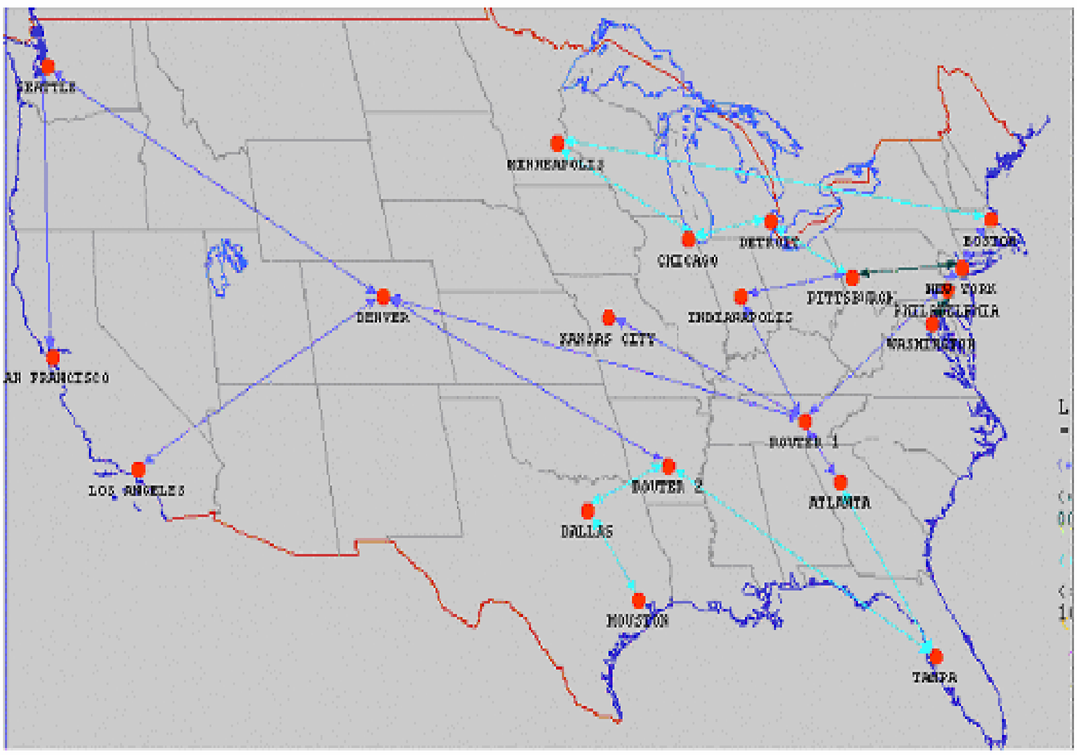
Fig 5.1 basic steps to create any network model
[1]

5.1.1 Create Network Model:

OPNET modeler has several editors, which allow building network model in a proper way such as project, node, process, packet and link editors.

Project Editor:

It is the main staging area for creating a network simulation. From this editor, the user can create a network model using models from the standard library, collect statistics about the network, run the simulation, and view the results. The user can also create node and process models, build packet formats, and create filters and parameters, using specialized editors accessible from the project editor.



A network model built in the Project Editor

Fig 5.2 sample of network model

[1]

Node Editor:

The Node Editor lets you define the behavior of each network object. Behavior is defined using different modules, each of which models some internal aspect of node behavior such as data creation, data storage, etc. Modules are connected via packet streams or statistic wires. A single network object is typically made up of multiple modules defining its behavior.

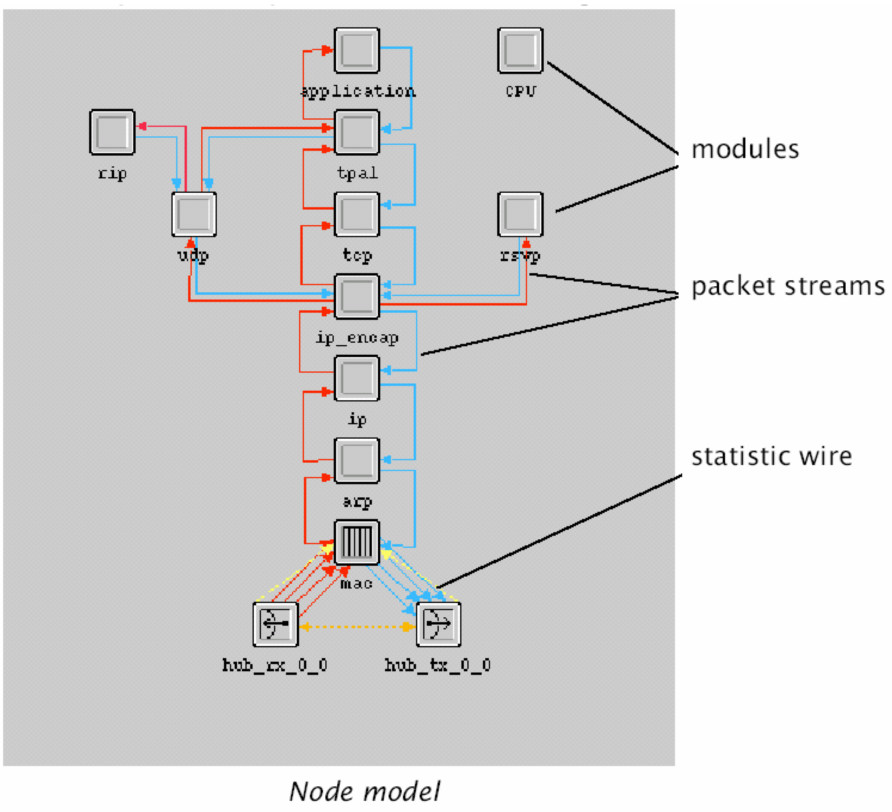
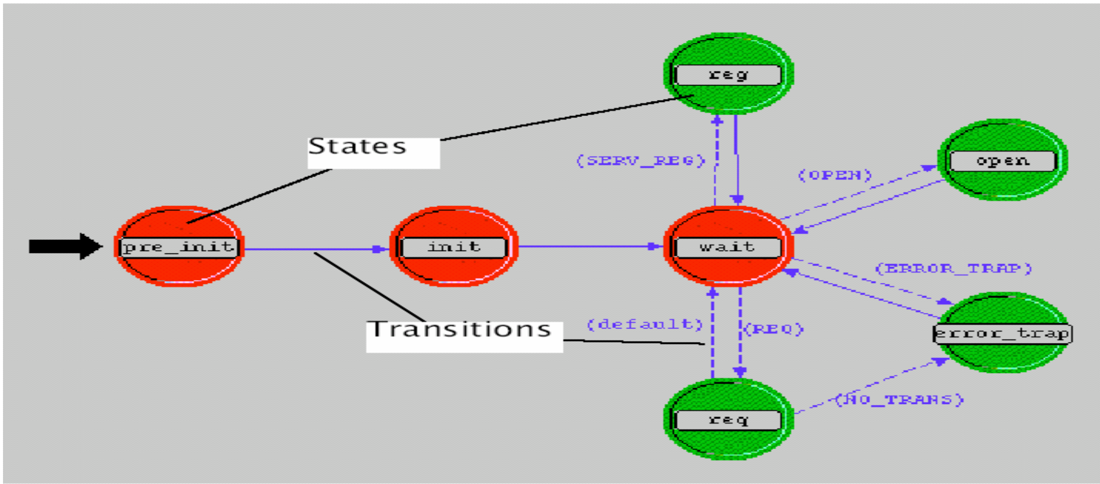


Fig 5.3 sample of node model

[1]

Process Editor:

The Process Editor lets you create process models, which control the underlying functionality of the node models created in the Node Editor. Process models are represented by finite state machines (FSMs), and are created with icons that represent states and lines that represent transitions between states. Operations performed in each state or for a transition are described in embedded C or C++ code blocks.



Process model

Fig 5.4 sample of process model

How does process model work?

OPNET simulations are made up of events. Process models respond to events and may schedule new ones.

When an event occurs that affects a module, the Simulation Kernel passes control to the module's process model via an interrupt. The process model responds to the event, changing state and executing related code, and then returns control to the Simulation Kernel (invocation 1).

The next time OPNET invokes the process model, it determines the state in which it was left (invocation 2), responds to the new event, and passes control back to the Simulation Kernel.

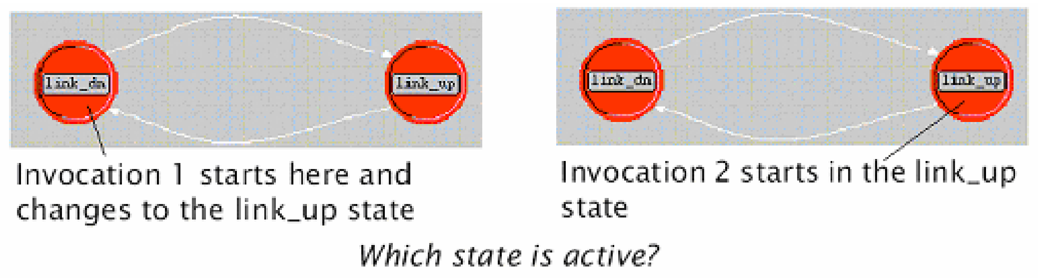


Fig 5.5 invocation process

When a process model enters a state, it executes the enter executives and then, if the state is unforced (as above), the process model stops execution and returns control to the simulation.

* The previous operation applied to the unforced state (Red state) , in case of the forced (Green state) the same operation take place but rather than stopping the execution and returns control to the simulation it execute the rest of the code (exit executive code) and move to the next state. [1]

State Coding (C/C++):

Three places where the code can be attached in OPNET(enter, exit and header block)

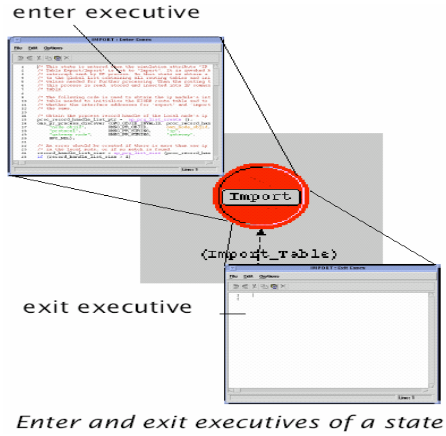


Fig 5.6 Enter & Exit executive

[1]

Other editors:

They are used for other purposes like choosing special statistics or editing the wireless devices (antenna) or viewing the results in different way …etc.

5.1.1.1 Priority Model:

One of the main objectives of this project is to introduce the priority concept to the network model using the priority model which is created using both node and process editors. This model deals with normal packets and has two types of time priority that the user can choose among them during the simulation; these are head of line and threshold base. The idea behind using this model is to give the system the appropriate flexibility to provide different classes of service:

1- Provide more bandwidth for the costumers at certain time (Football matches on line)
2- Support traffic management**.**
3- Quality of service to support delay sensitive applications such as voice, video and multimedia application. Support delay insensitive data application
4- Support delay insensitive data applications.

<u>How does it work?</u>

Simply the thirty workstations will generate number of packets and each packet will be sent with random destination address to the switch, the switch then will check the packet priority value which is assigned manually in the switch program and insert it in the first subqueue if the priority value is high otherwise it will insert it in the second subqueue, the switch server will start serving the two subqueues based on the priority type that is set by the user.

When the service is finished the switch will extract the destination address from the packet field and forward the packet to its destination, and the process goes on.

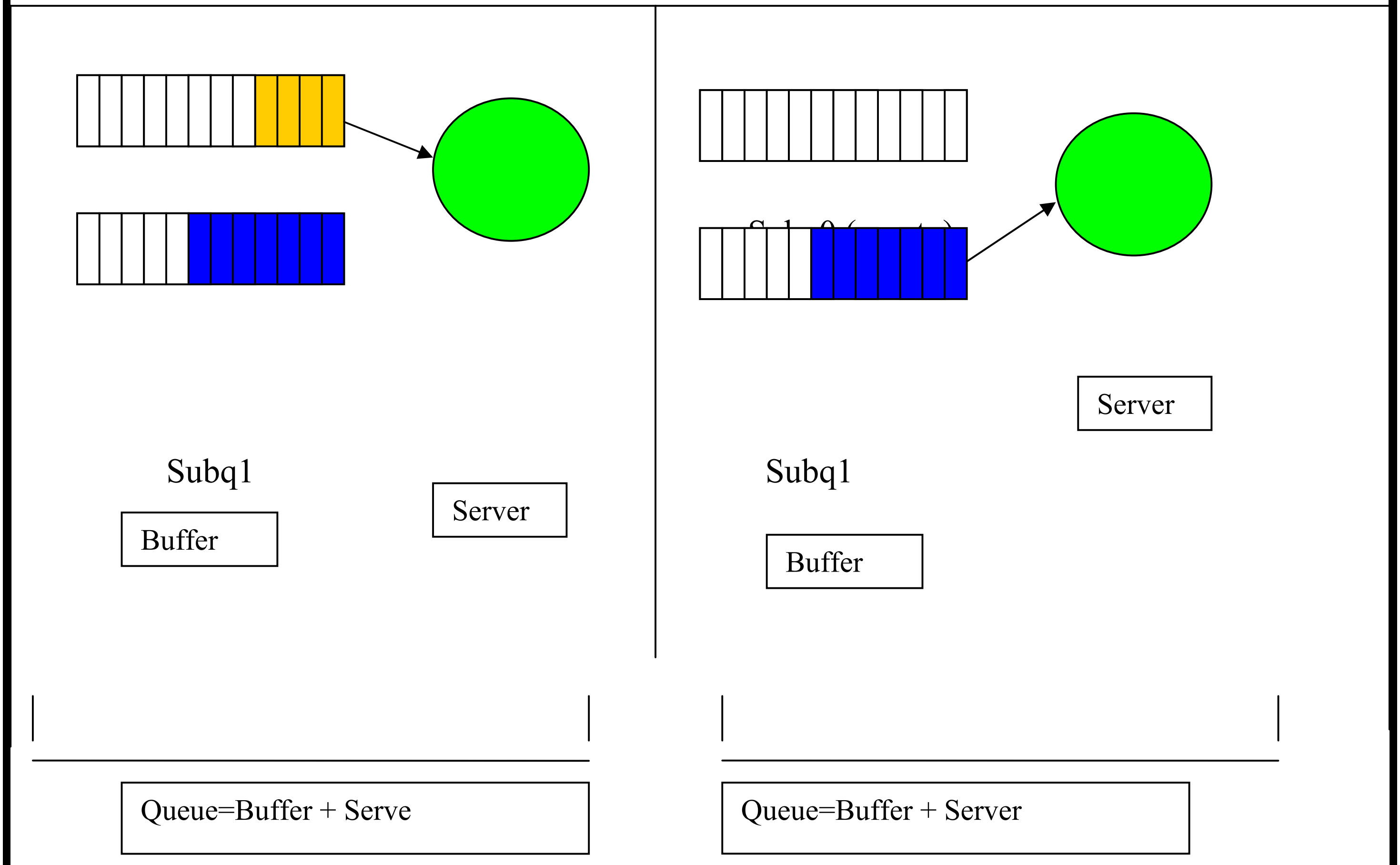
<u>Priority Type:</u>

<u>1-</u> Time priority (Head of line):

<u>Idea:</u> The queue buffer of the switch is divided into two subqueues:

- *Subqueue 0* is dedicated for High priority packet
- *Subqueue 1* is dedicated for Low priority packet

The concept states that queue server will serve high priority packets as long as subqueue 0 is not empty, once it gets empty the server will serve the low priority packets until a high priority packet is inserted in subqueue 0, then the server will go back and serve subqueue 0 and the same operation will be repeated till both subqueues are empty or the simulation time is finished.



Key: High priority packets   Low priority packet

Fig 5.7 Time Priority Head Of line

Priority Type:

2- Time priority (Threshold base):

The previous design does not give the low priority packets a chance to be served very regularly therefore its delay will increase.

Threshold concept was introduced to solve this problem where in this case the queue server will serve both of subqueues equally until the subqueue0 occupancy go beyond the threshold value (limit reached) in this case the server will only serve the subqueue 0 till the its occupancy drops below the threshold then the server will go back and serve them equally.

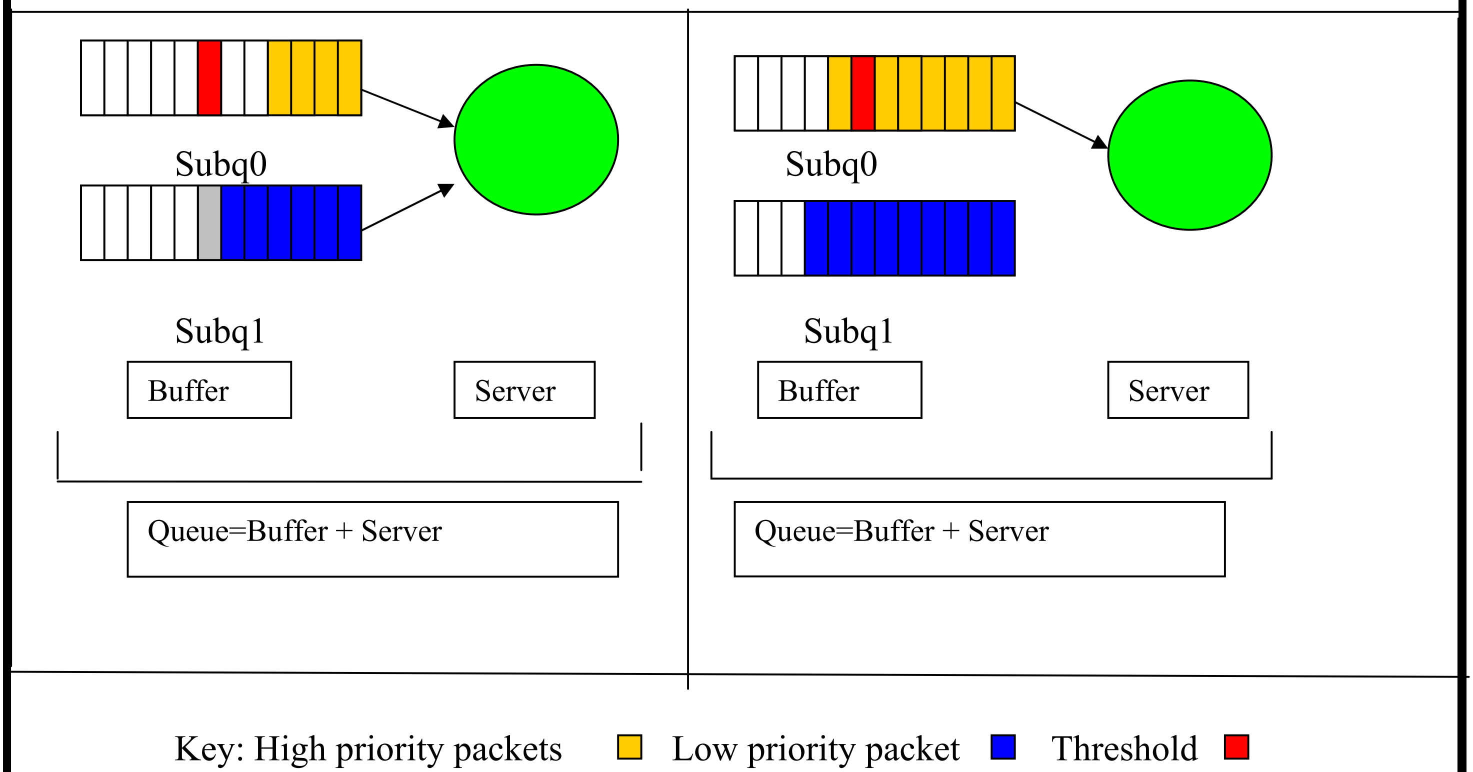


Fig 5.8 Time Priority Threshold base

5.1.2 Choosing statistics:

Statistics can be collected in two ways either from the individual nodes using object statistics (Local statistics) or from the entire network as a whole (global statistics).

5.1.3 Run Simulation:

Statistics concerning the simulation phase can be set by choosing the ‘ Configure Simulation ‘ option from the ‘Simulation’ menu. The user then could change different parameters such as ‘Simulation Duration’ and such before choosing ‘run’ to start the simulation. The user then waits until the simulation completes where then the results will be displayed.

5.1.4 View Results:

The collected results could be viewed at any time by choosing ‘View Results’ option after ‘right clicking’ on the workspace. The user could view the global and local statistics by ticking on the shown parameters where then the program will display each parameter on its own.

The results that were obtained by us will be explained in detail in chapter eight (Discussion).

5.2 Limitation and Assumptions:

Network model designed using OPNET modeler has the following limitations:

1. There is only once license for OPNET and that delays the working schedule. This means, when the network is being modeled, it is not possible to model in the process level and vise versa. Also, the current license does not include certain features, such as debugging the code in the process level.
2. It is very difficult to integrate the process level (priority model) with the network modeled due to complexity in addition to the limited resources to refer to. Therefore, the priority model is separated from the modeled network.
3. The networks modeled are not capable of having safe Internet access because security measures were not applied

Chapter 6

Design Issues

This chapter shows certain components and network equipment, which can be used in different network designs. It also represents the block diagrams of the networks that were modeled throughout this project. In addition to this, it gives an idea about the processing and simulation part of the program. Moreover, it illustrates what kind of output is expected from the simulation.

Modeling a network is separated into three segments:

1- **Input:**

All networks have the same structure. Each of them contains a specific number of end-systems. End-systems are sub-classed into workstations, printers and devices (repeaters, hubs, switches, bridges and routers and servers). The way the end-systems and network equipment are connected to each other is called network topology. Connecting these devices is done using links (wired or wireless). The usage of each link type differs from one network to another according to the specification of each link. It also depends on which connections they are to be used for. In addition to this, the environment of the network is able to limit the choice of the link types. Information transmitted and received among networks is divided into packets. The size of the packets and how fast they get transmitted affect the performance of the network, either positively or negatively. The user is able to specify the period of time that the simulation represents in real time. Going deep inside the network equipment, it is permitted to design user defined process models. In the process level, the programming language, C++, along with certain built-in functions (system calls) are involved to give the user the luxury of dealing with the packets. For example, it is possible to write a code that gives the packets of a specific end-user more priority than other users, so that once they get to the network equipment, they are processed and then retransmitted before anyone else. For this project, the topology types used are star and cascade topologies, while the links are 10BaseT and 100BaseT.

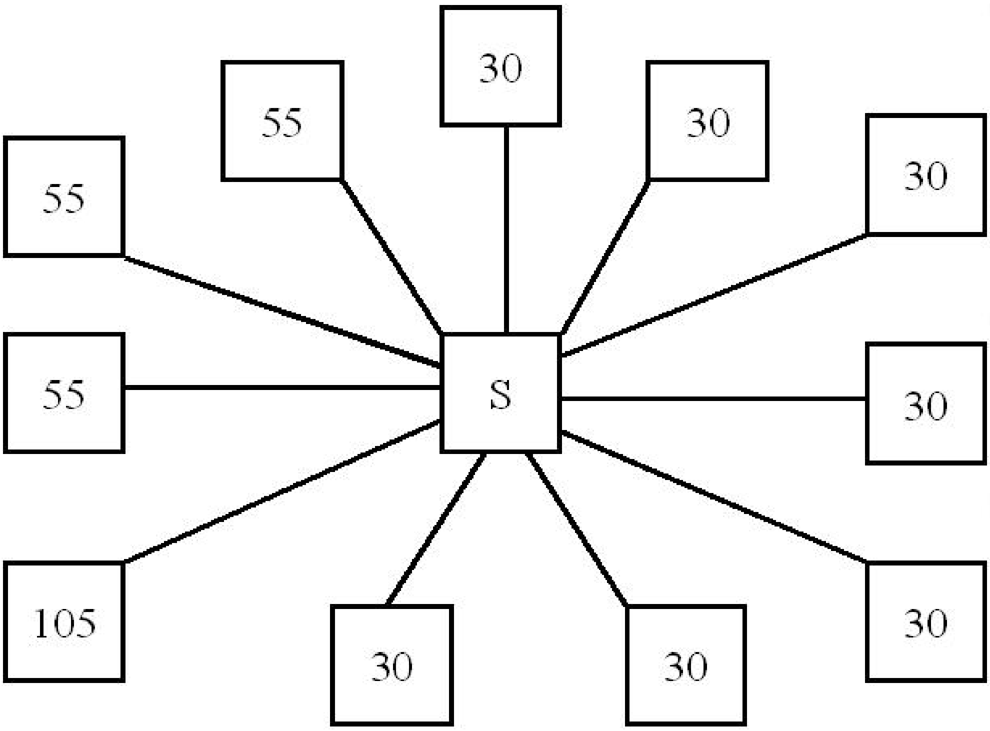


Figure 6.1 Block diagram of star topology.

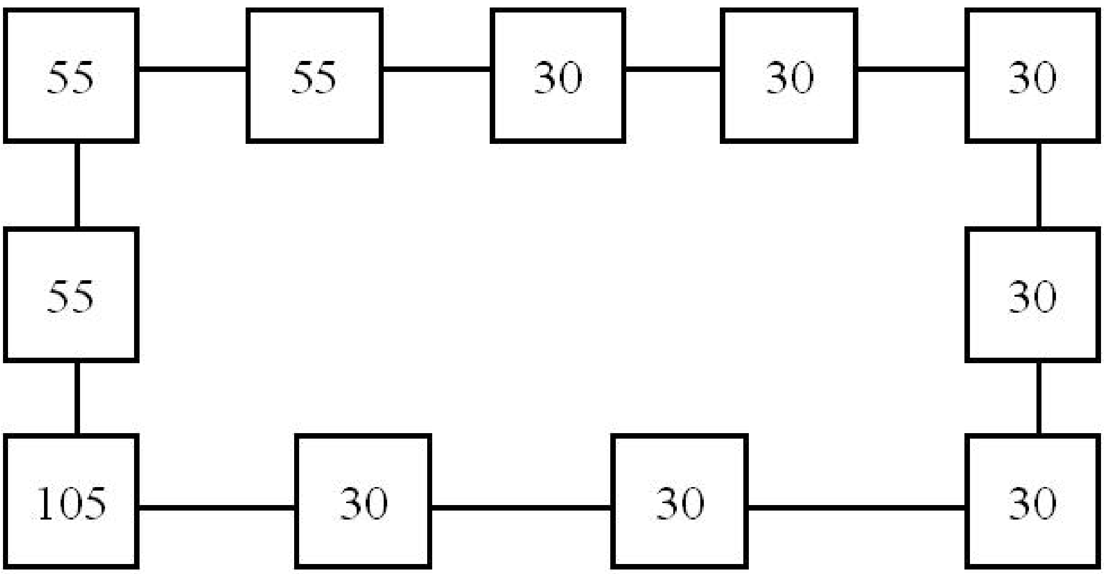


Figure 6.2 Block diagram of cascade topology.

2- Processing:

After the program receives the required input, it runs the simulation. The simulation takes care of checking whether the input parts are suitable for real-life networks or not. If the input parts fall under the correct criteria of networking, the program starts calculating and producing the output results. Otherwise, a message is sent on the screen indicating that an error has occurred. The program is capable of specifying the exact error to the user, like link errors.

3-* Output*:

After the simulation is completed with no errors, the results are produced, which are graphs. These graphs represent detailed information about performance of the network. Modeling several networks allows investigating and studying the performance differences between each of these networks.

Chapter 7

Implementation and Testing

This chapter includes four parts where the first two parts explain in details of the implementation of both the network model with the scenarios and the priority model where a new network, switch, workstation, packet format and link model were created from scratch using the OPNET editors.
The last two parts explain the validation and verification, which belong to the testing section.

7.1 Implementation of the Network Model:

UNIX platform is used to implement all the parts of the project. Although another version of OPNET has been developed, the UNIX version has been used because it is the one available in the college.

For all designs of the network, there are several floors. The first floor consists of 30 workstations. Each floor of the other floors consists of 480 workstations, where each floor has also number of floors. The number of workstations varies from an office to another, while the topology applied to the workstations within an office is star topology all the way. The first floor workstations are always connected in star topology. On the other hand, different topologies are applied to the connection between the offices. These offices are connected to each other via a switch. The first floor contains a server and a router. The server is connected with the first floor star topology. The switch and the first floor star are connected through the router in order to facilitate routing the traffic. After making sure that both of the different topologies were working correctly, the effects of changing the link types and adding extra floors to the networks were obtained after running different simulations.

There are common characteristics between all the designs, which are the following:

The first floor consists of 30 workstations connected in star topology with a centralized switch. From this switch, there are other two connections, one to the server and another one to the main router of the enterprise, as in figure 7.1.

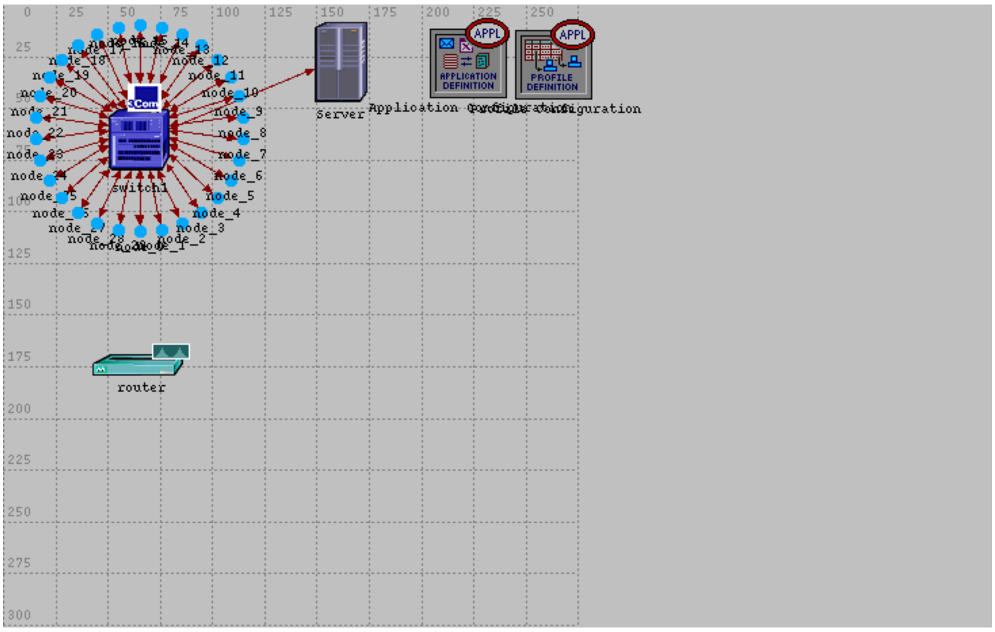


Figure7.1

The first floor consists of a router, a server and 30 workstations connected in star topology with a switch

For all of the other floors, there is a fixed number of offices in each floor (11 offices). Out of these eleven offices, there are 7 offices whose number of workstations is 30, 3 offices with 55 workstations and one office consisting of 105 workstations. The topology used in each office is start topology with centralized switch. Initially, all the links are 10BaseT (10 Mbps-Twisted Pair links), however, the link type is changeable according to the utilization (as shown in the outputs section). Moreover, most of the different topologies and models are applied on how the offices and connected to each other. Usually (exceptions are mentioned separately in each design as they occur) all the connections are from the switch of each office to another equipment outside that office, not from a workstation to another. The subsequent describes the models that have been accomplished.

a) This structure has number of floors of three. In each floor (except the first floor), the eleven offices, which are represented by the ready-made star LANs, are connected to each other on star topology with centralized switch, as in figure 7.2 below. The router used here has only two ports (interfaces). One port is already connected with the switch of the first floor. Therefore, the other floors are connected to a switch, and this switch is connected to the router. The results are in the outputs section.

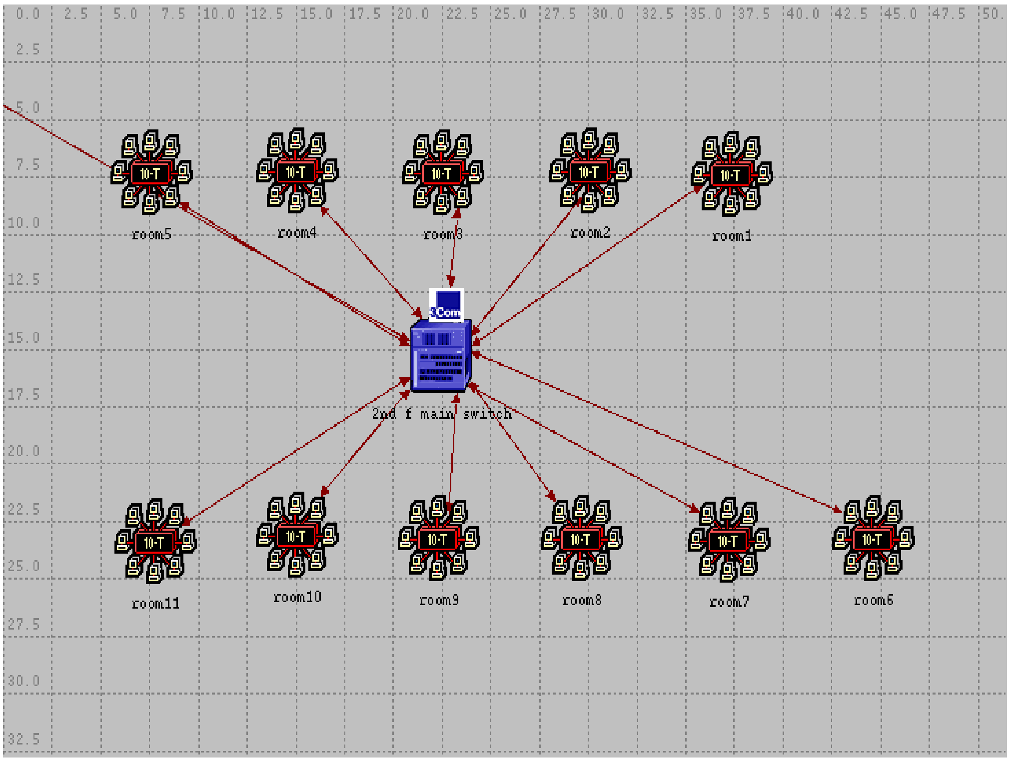


Figure 7.2

The eleven offices of each floor are connected in star topology.

b) Here, the topology of the floors in (a) is reconfigured from star topology to another topology. This topology looks like a circle or cascaded topology (Figure 7.3). This is first done by connecting each office directly to its neighboring office. It has been discovered that this is an inefficient way in applying the configuration. The solution is to place a router between each two offices because of its ability to route the traffic. The routers used here are the same as the one described previously. To connect this floor to the whole network, a switch is placed in this floor. It is connected to one of the workstations in the floor and also connected to the switch in the first floor. This allows future additions in the floor.

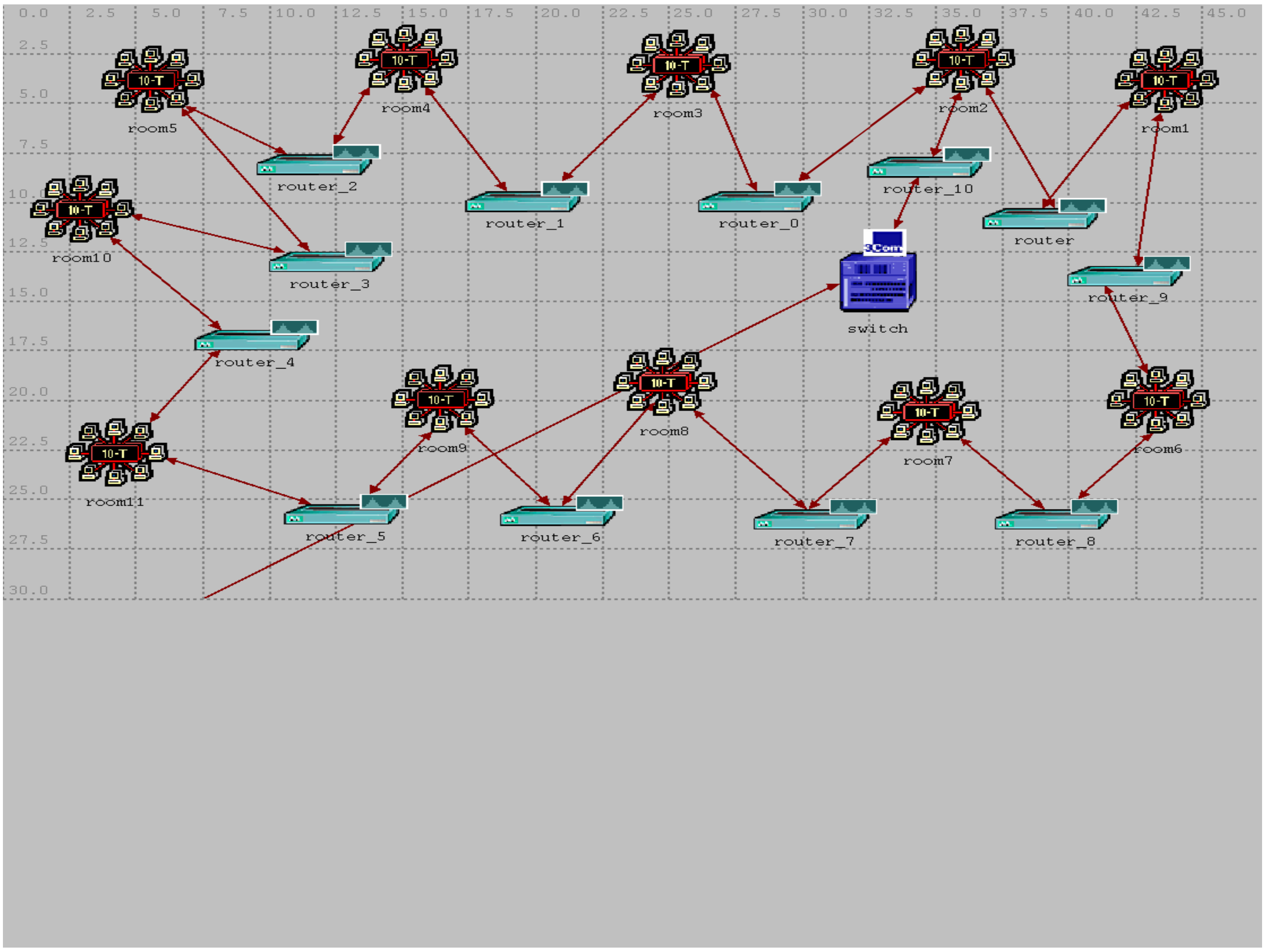


Figure 7.3

The eleven offices of each floor are connected in cascade topology.

c) For this design, the 10BaseT links of structure (a) are substituted with 100BaseT links. The reason is that only with three floors, the utilization of the former link exceeded %80 (as explain in the outputs section). A dramatic decrease in the link utilization has occurred because of increasing the bandwidth of the link. Therefore, it is possible to increase the number of floors of the enterprise.

d) In this structure the links of design (b) are changed from 10BaseT to 100BaseT for the same reasons explained in (c). This results in having a higher threshold of bandwidth in order to increase the number of floors.

e) In design (c), the number of floors has been increased from three to 7. New results have been gathered (output section).

f) In design (d), the number of floors has been increased from three to 7. New results have been gathered (output section).

g) In design c, the ready-made star LANs were substituted by custom made LANs using one of the features supported by OPNET (Rapid configuration). It allows choosing the topology of the custom-made network. The difference between ready-made LANs and custom-made LANs is that the ready-made LANs has a switch with 105 interfaces, while in the custom made LANs, 3com switches with 102 interfaces were used. Only one problem occurred. One of the offices should have 105 workstations, while the switch used here had only 102 interfaces. In order to solve this problem, the network of the office was split into two sub-star networks, one with 50 workstations and another with 55 workstations. Each of the sub-networks had the same switch mentioned earlier. The two switches were then connected to each other via a router, so that the two networks became connected to each other.

7.2 Implementation of priority model:

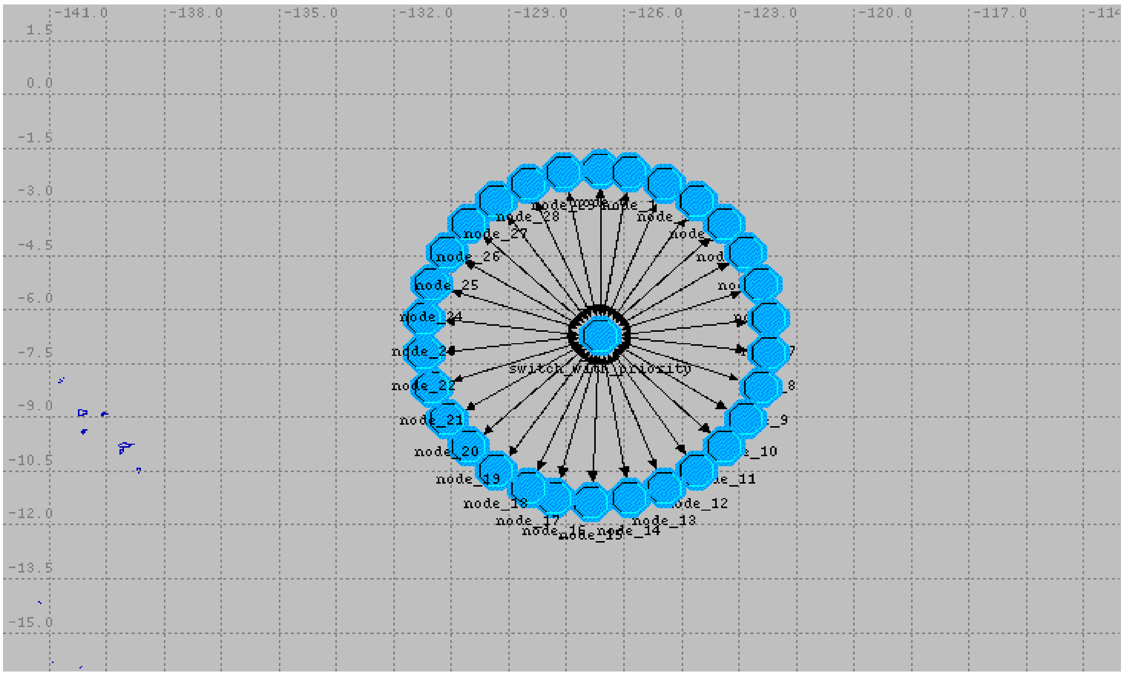


Fig 7.4 Network model with priority

Procedures:

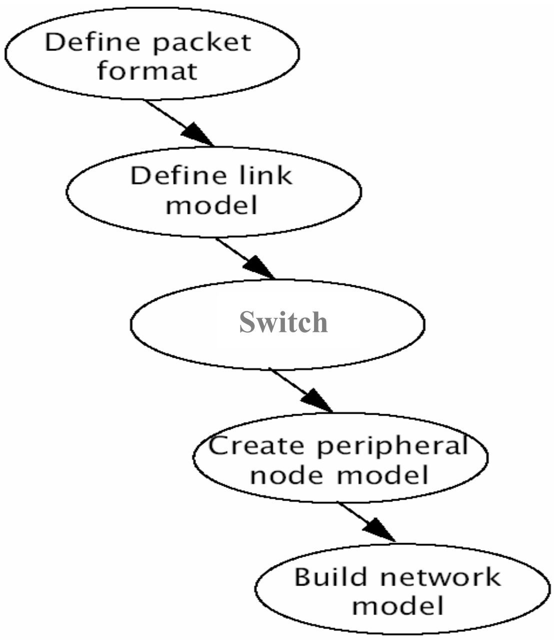


Fig 7.5 Procedures to create Network model with priority

1.Packet format:

By using packet format editor, a field of 32 bit is assigned for the destination address, changing the set creation attribute to unset (This ensures that the field will not be assigned a default value when the packet is created), the packet format will look like the following:

Fig 7.6 Packet field

2.Link Model:

By using Link model editor, it is possible to create a link that support the above packet format, In addition to setting the supported packet format type, it is desirable to set the following attributes in the attribute table:

1) Setting the data rate attribute to 9600.

2) Setting the ecc model to ecc_zero_err. (error correction model).

3) Setting the error model to error_zero_err.

4) Setting the propdel model to dpt_propdel (point-to-point propagation delay model).

5) Change txdel model attribute to dpt_txdel (point-to-point transmission delay).

3. Switch node model:

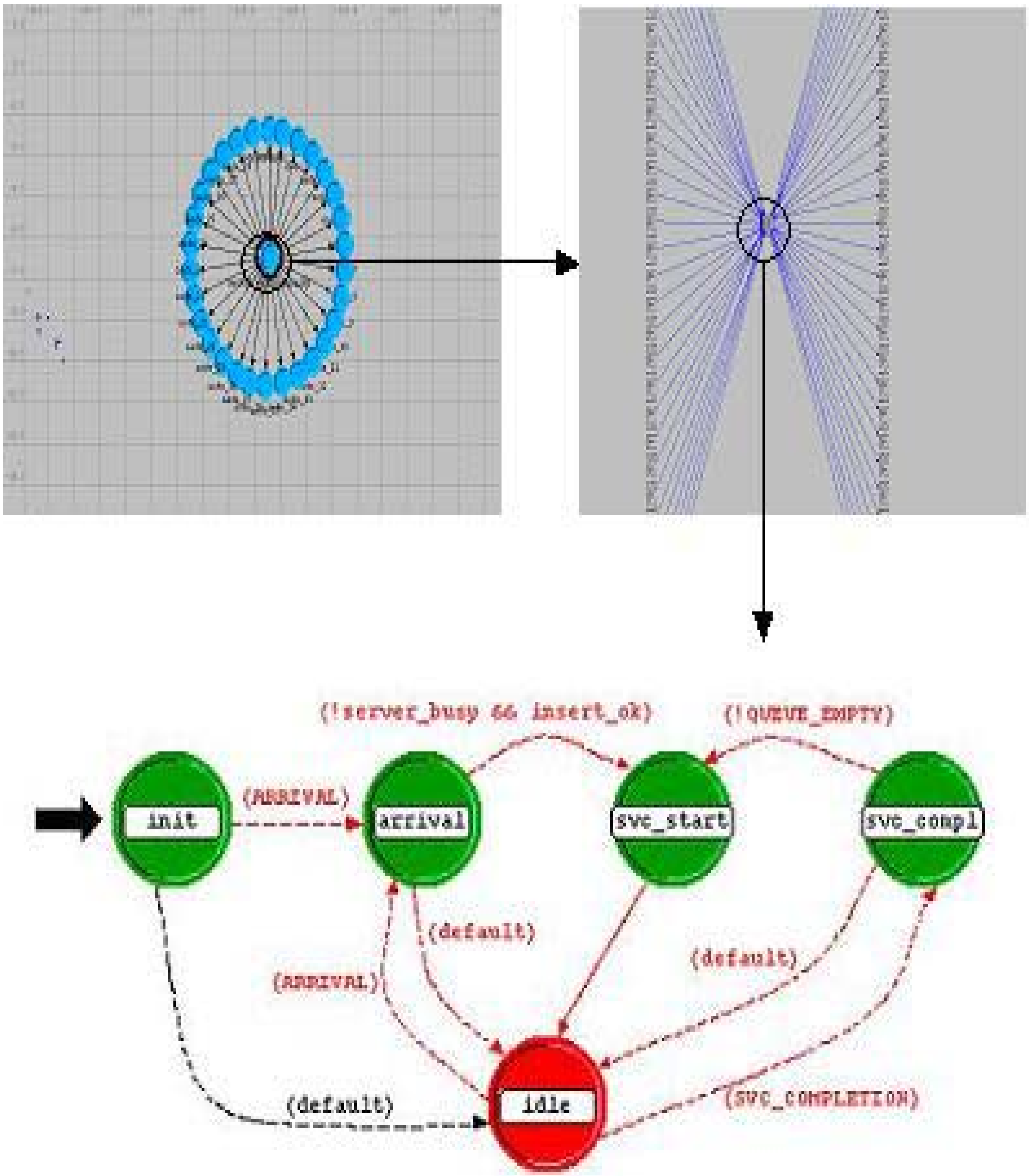


Fig 7.7 Switch model

A- Node Model

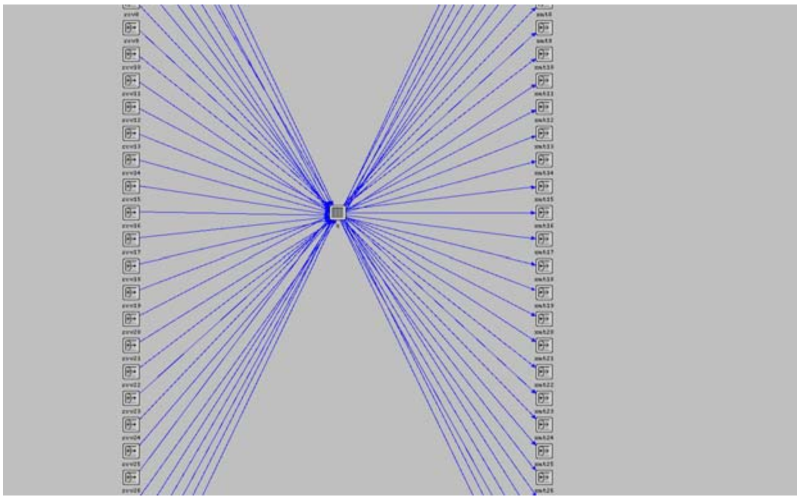


Fig 7.8 Switch node model

-As seen from the picture above the switch consists of one queue and 32 point to point receiver and transmitter (32 ports) each rcv and xmt module is dedicated for one workstation (i.e. rcv1 and xmt1 is for receiving and transmitting packets from/to workstation 1), the link between rcv -> Q -> xmt is called packet stream which is used to transfer the packet from one module to another.

Attributes of the queue such as service rate, buffer size and process model can be set manually by clicking the right button of the mouse on the queue module -> edit attribute.

The default settings of the queue are:

- Service rate = 9600
- Number of Rows = 1
- Buffer Size (in terms of packets) = infinity
- Buffer Size (in terms of bits) = infinity

* Buffer size means that how many packets or bits it can receive.

B- Process Model

By clicking twice on the queue module the following process model will appear:

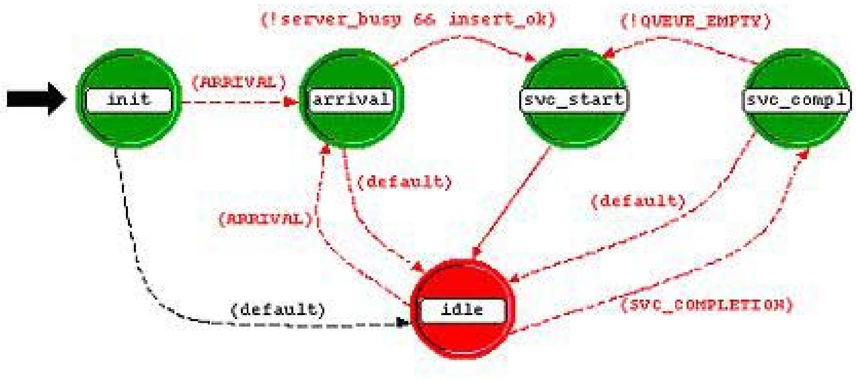


Fig 7.9 Switch process model

The process model of the queue module is made up of five states (four unforced + one forced), once the queue process model is invoked by the kernel in response to an event it the process will start moving from one state to another, executing the code that describe the job of each state and finally going to the forced state where it waits for another invocation.

i- Initial state:

The initial state is the starting point of the queue process and it is used to initialize the queue variables which are:

- **Server_busy:** ***it is used to indicate whether the server is busy or not***
- **Packet counters**: ***count the number of packets received and transmitted.***
- **Packet handlers*: they are used to read and write the results of the packet counter.***
- **Limit:** ***indicates subqueue 0 occupancy in case of (Head of line priority).***
- **Threshold:** ***indicates the threshold value entered by the user (used in threshold base).***

After executing the initial state code the process will move to the next state which is arrival in response to receiving packet or move to the idle state in response to other type of interrupts or events.

ii-Arrival state:

It is the state where the queue deals with inserting the packets into the buffer, first of all the queue will create a pointer for the packet, assign priority value to the packet based on the user choice (manually in the program code) for example the packets received from workstation1 will be assigned a high priority value while the rest will be assigned low priority value, at the same time the queue process will count number packets received and record it.

The next step is checking the packet priority value so high priority packets will be inserted into subqueue 0 or buffer 1 and low priority packets will be inserted into subqueue 1.

If the insertion was successful and the server is not busy the then queue process will move from the arrival state to the svc_start state, but if the insertion failed due to the full occupancy of the buffer then the packet is destroyed and the queue process will move back to the idle state.

iii-Svc_start state:

This corresponds to the start of serving the packets by the queue server, the behavior of the server is based on the priority type therefore the type of priority is checked first:

-Threshold Base:

The server will check whether the subqueue 0 (contains high priority packets) occupancy has reached the threshold value or not, if the answer is yes then the server will access this subqueue and start serving the packet at its head by doing the following:

1- Determining the packets length
2- Determining the time required to complete the service of the packets
3- Schedule an interrupt for this process t the time where service ends.

Then the queue process will move to the idle state waiting for another interrupts Once the process model enters this state again it will repeat the same procedure as long as the answer is yes.

If the answer is no then the server will repeat the same operation but with different subqueues each time for example (first time with subqueue 0, the next time will be with subqueue 1, then subqueu0 and so on) until the answer becomes yes.
In Both cases the server will be busy (Server_busy=1);

-Head of Line:
Same operation applied here the only difference is that the server will always serve the subqueue 0 till it gets empty then it start serving the subqueue 1 until subqueue 0 receives high priority packets.

iv-Svc_compl state:
This corresponds to the completion of service, the queue process will move from the idle state to the svc_compl state in response to the service completion events which was scheduled in the svc_start state, now the queue will firstly check the priority type and perform different action based on that:

-Threshold Base:
Same operation as in the case of svc_start but rather than accessing the subqueue for serving it will remove the packets from the head of the subqueue, extract the destination address, forward the packet to its destination and count the number of forwarded packets.
Then the queue process will move to the idle state waiting for another interrupts

In case the queue is empty (both subqueu 0 and 1 are empty) or go back to the svc_start state in case the queue is not empty for serving the rest of the packets.

- Head of Line:

Same operation applied here the only difference is that the queue process will always start removing packets from subqueue 0 till it gets empty then it start removing the low priority packets within the subqueue 1.

In both cases the server will be idle (Server_busy=0);

v- Idle state:

It is the state where the queue process go to in response to unwanted events or interrupt, the then process will wait there until wanted events occur like receiving a packet.

2.Workstation (peripheral).

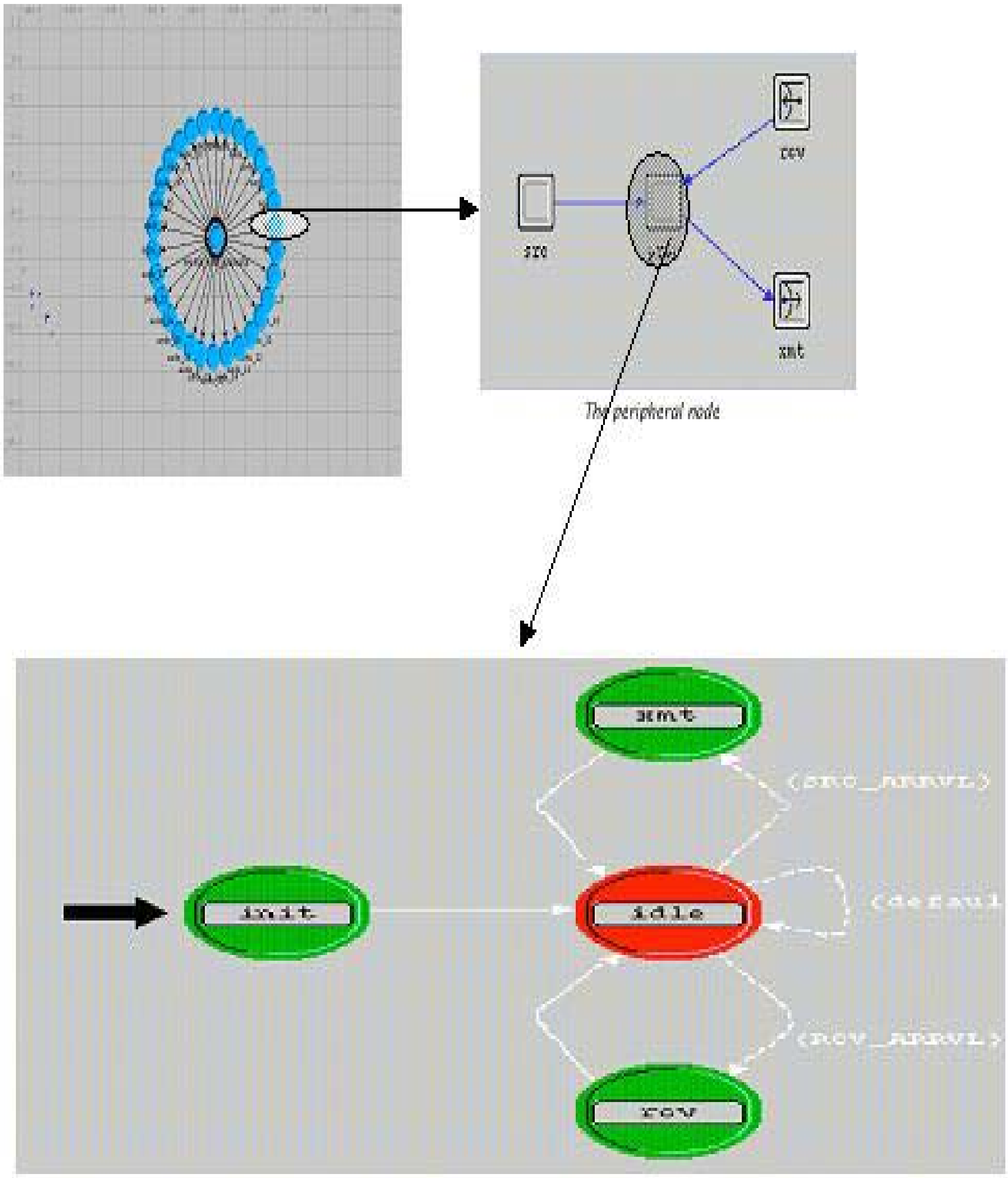


Fig 7.10 Workstation model

A- Node model:

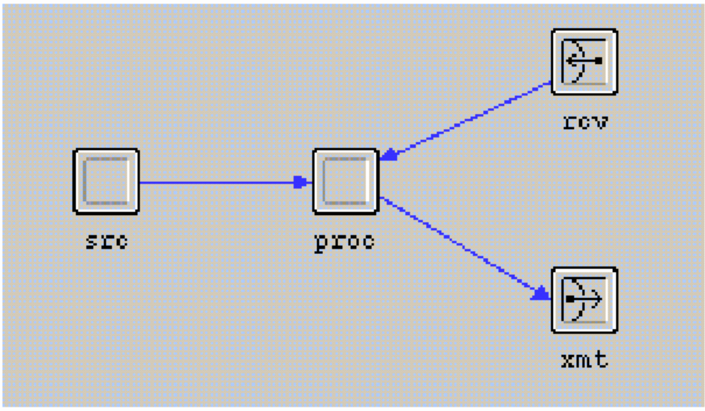


Fig 7.11 Workstation node model

The workstation or peripheral model must generate packets, assign destination addresses, and process received packets. The workstation node model will employ an OPNET generator module to create packets. It will employ a user-defined process model to assign destination addresses to packets and send them to the node's point-to-point transmitter. This process model will retrieve packets arriving from the point-to-point receiver. Upon receiving a packet, the same process model will calculate the packet's end-to end delay and write the value to a global statistic (a global statistic is accessible to multiple processes throughout the system).

B- Process model:

By clicking twice on the proc module the following process model will appear:

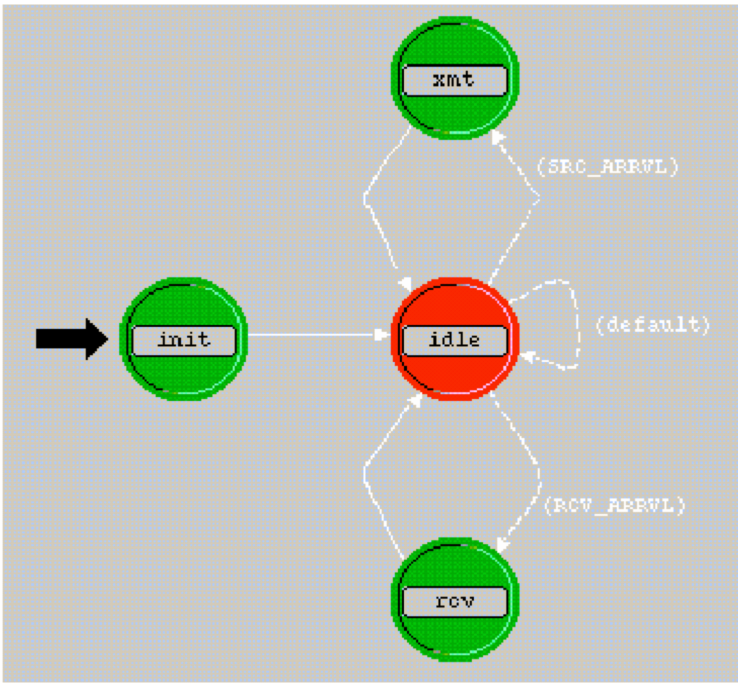


Fig 7.12 Workstation process model

The process model of the peripheral or workstation consists of four states (3 unforced + one forced), Once the peripheral process model is invoked by the kernel in response to an event, the process will start moving from one state to another, executing the code that describe the job of each state and finally going to the forced state where it waits for another invocation.

i- Initial state:

The initial state is the starting point of the queue process and it is used to initialize the peripheral variables which are:

- **Address_dist:** ***generate a random destination address from 0 up to 30.***
- **Ete_gsh**: ***it is a handler used to record the end-to-end delay.***

After executing the initial state code the process will move to the next state, which is idle.

ii-Idle state:

It is the state where the queue process go to in response to unwanted events or interrupt, the then process will wait there until wanted events occur like receiving a packet.

iii-Xmt state:

It is the state that handles transmitting the packet from the peripheral node to the switch, the peripheral process will move from the idle state to the this state in response to receiving a packet to forward it so at this state the process obtains the packet pointer, assign random destination address and finally send the packet to the switch using send function. Then it goes back to the idle state waiting for an event.

iv-Rcv state:

It is the state that handles receiving the packet from the switch, the peripheral process will move from the idle state to the this state in response to receiving a packet to process it (calculate end to end delay) so at this state the process obtains the packet pointer, determine end to end delay by subtracting packet creation time from the simulation time , record the result and finally destroy the packet. Then it goes back to the idle state waiting for an event.

**** For the codes, please see (appendix B).***

7.3 Verification:

All of the six scenarios were tested thoroughly by checking all of the different links again and going through all of the components used by checking their parameters again. Finally, all of the scenarios were simulated one more time in order to double-check their consistency. The new outputted results were identical to the ones that were obtained before.

7.4 Validation:

Concerning validation, the different results were anticipated before obtaining them; this enabled us to form a rough idea about how the system will change, lets say, if extra floors were added. After the simulations were run and after comparing the different graphs together, the final solution that was obtained was expected. Thus, this shows that the right system was built.

Chapter 8

Discussion

This chapter will put the spotlight on the different results that were obtained after running the simulation on the six different scenarios and on the priority part. These results show the relations between the different parameters such as the load and server, and the inputs used such as the number of floors or the type of links implemented. It also explains the results obtained from the priority part.

8.1 Work Done:

8.1.1 Output Analysis and Research:

The following parameters were tested and compared in order to know what affects them, and how to reduce unwanted components such as delay. Also these tests were conducted in order to notice where the two topologies differ and where do they appear the same:

1- Server Load
2- Overall Network Delay
3- Links Utilization
4- Links Throughput
5- Traffic Received
6- Sever Delay

The first parameter that will be discussed is the ‘ load on the server’:

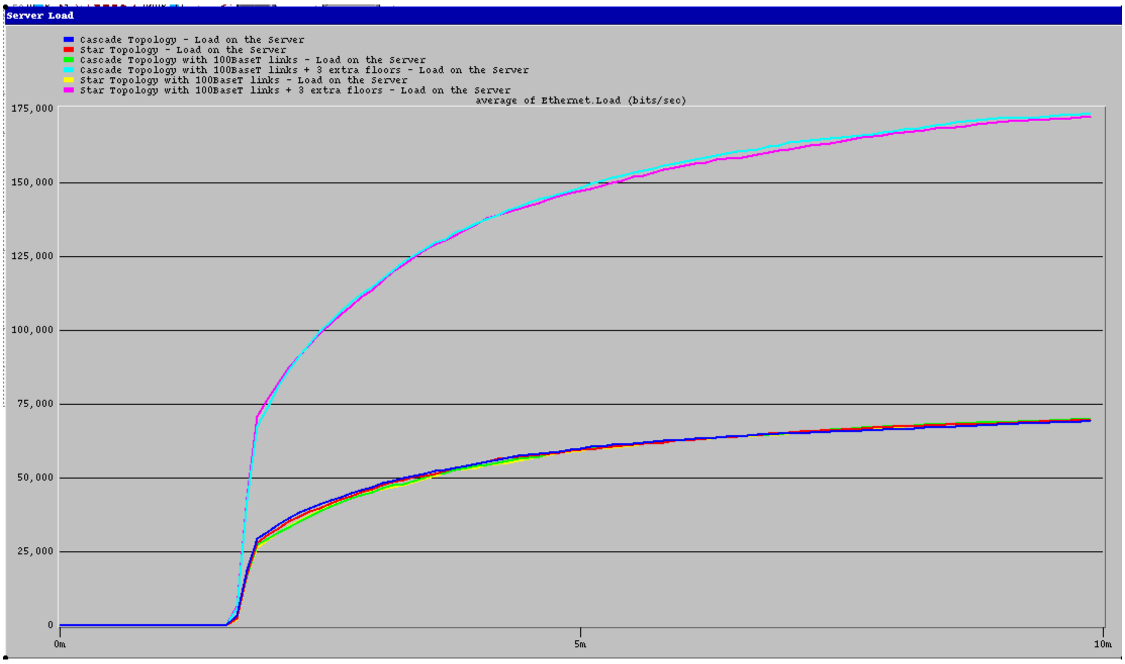


Fig 8.1 shows the load on the server in the six different scenarios

The previous figure shows that when changing the links from 10BaseT to 100BaseT in both the star and the cascaded topologies, no increase in the load on the server happens. However, when three more floors were added, this resulted in an increase in the load on the server by nearly 2.5 times the original maximum load. This shows that the load is directly proportional to the amount of traffic (ie, number of workstations) going through the network and has no relation to the capacity of the links used. However, the capacity of the links would affect the load on the server if already the 10BaseT links have reached maximum utilization of 100%

The second parameter that will be discussed is the End-to-End Delay:

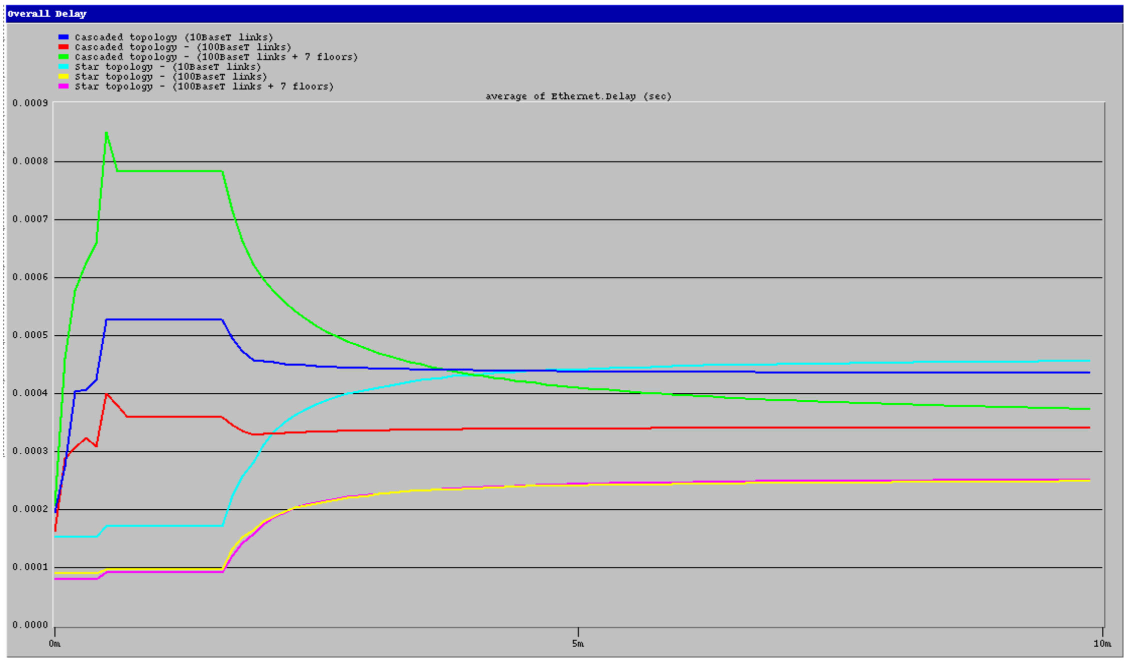


Fig 8.2 shows the overall end to end delay in the six different scenarios

This figure shows that the delay is directly proportional to the amount of traffic that passes through the network and the types of the links used. Also, both of the different topologies gave nearly the same maximum delay values when different variables were kept the same, such as the number of workstations, link types, etc. The blue curves represents the end to end delay when using 10BaseT links were used in the cascade topology, while the red curve shows the end to end delay when 100BaseT links were used in the cascade topology. The green curve shows the increase in the delay when extra floors were introduced to the network (cascade topology with 100BaseT links). It can be noticed that the light blue curve (which represents the end to end delay in the star topology with 10BaseT links) is slightly higher than the cascade topology when 10BaseT links were used too. But when using 100BaseT links, the delay in the star topology was considerably lower than the delay in the cascaded topology (using 100BaseT links). This shows that there could be an error in the network, since the output of the delay in the star topology should be higher than the delay in the cascaded topology (since all of the variables such as, the number of floors and links bandwidth) were the same in both of the topologies. A good explanation would be that when simulating the cascaded topology with 100BaseT links, collusion occurred, which increased the delay because the packets were retransferred again. Also the graph shows that when increasing the traffic in the star topology (by increasing the number of floors), the delay was exactly the same as when there were only three floors. This also shows a slight error, because there must be an increase in delay, however in this case, it is not shown clearly. This problem could be solved by running the simulation for a longer time in order to notice if there will be an increase in the end-to-end delay.

The third parameter that will be discussed is the 'Links Throughput':

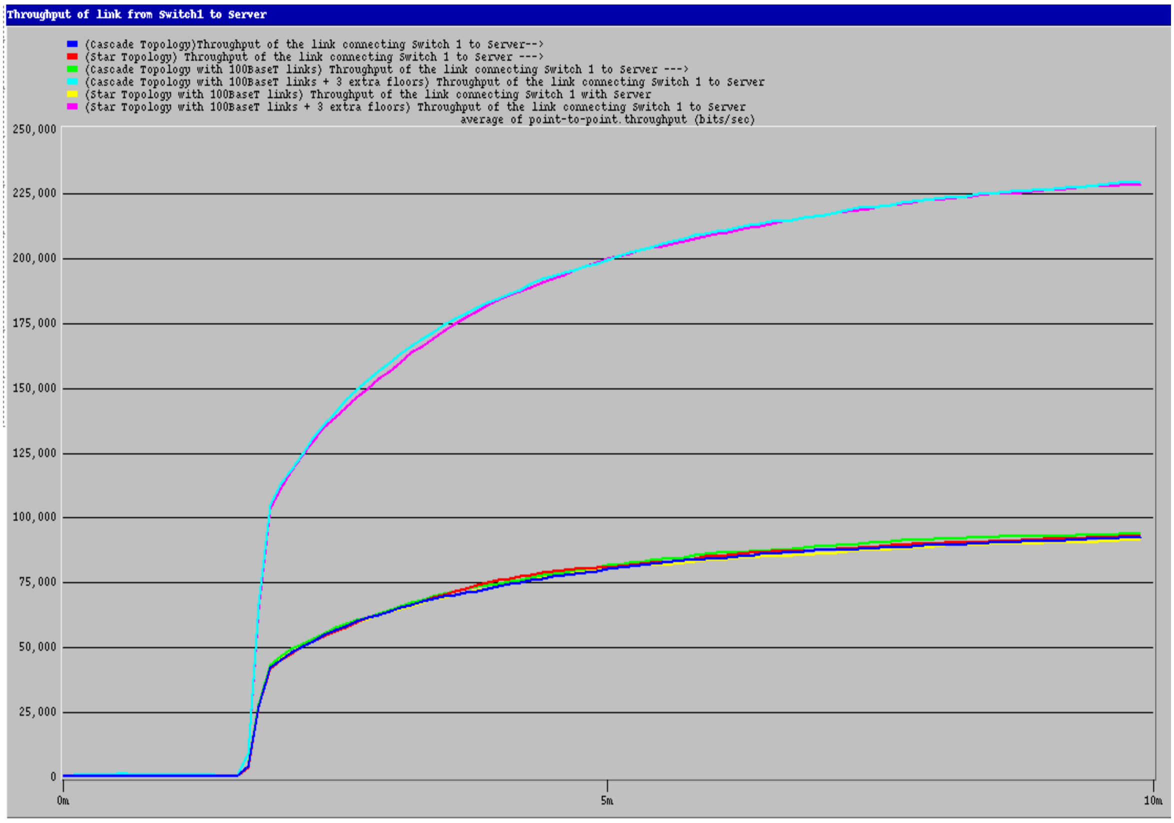


Fig 8.3 shows the throughputs of the links used in the different scenarios

The previous figure shows that the throughput only increased when more floors were added, it also could be noted that when three extra floors were added, the maximum throughput increased by 2.5 the original maximum throughput value. The light blue and purple colors represent the two topologies after changing the links from 10BaseT to 100BaseT and adding three extra floors, while the other colored graphs represents the two original topologies and the topologies when we only changed the links from 10BaseT to 100BaseT. This shows that the throughputs of the links are directly proportional to the amount of traffic that is passing through the network.

The fourth parameter that will be discussed is 'Links Utilization':

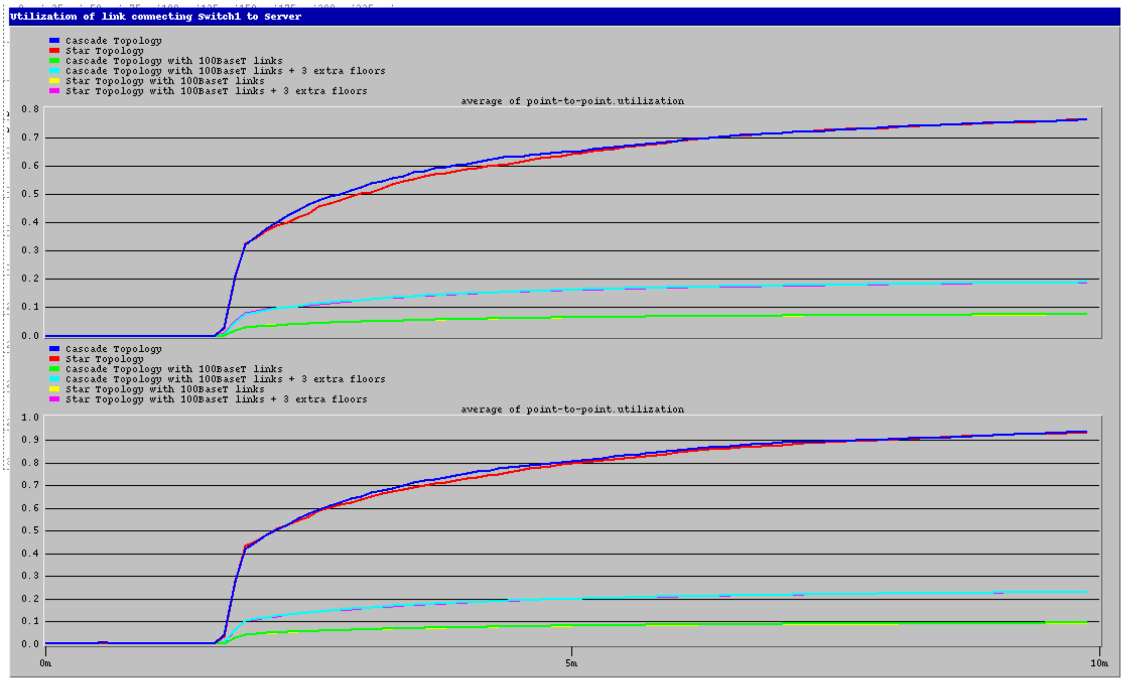


Fig 8.4 shows the utilization of the links used in the six different scenarios

This graph shows that utilization of the links depends on the capacity of the links used. This means that if the links capacity were increased, the links utilization would decrease due to the extra bandwidth available. This is clearly shown in the graph, where the dark blue and red curves show the utilization of the link that connects Switch1 to the Server when 10BaseT links were used. The other curves represent the utilization of the links when 100BaseT links were used. What also can be deduced from the graph is that the link utilization is nearly the same when 100BaseT links were used even when three extra floors were added. That is because the 100BaseT links could accommodate even much more traffic that what was added, so the increase was barely visible. The increase in utilization would be visible if, lets say, thirty or forty extra floors were added.

The fifth parameter that will be discussed is the 'Traffic Received':

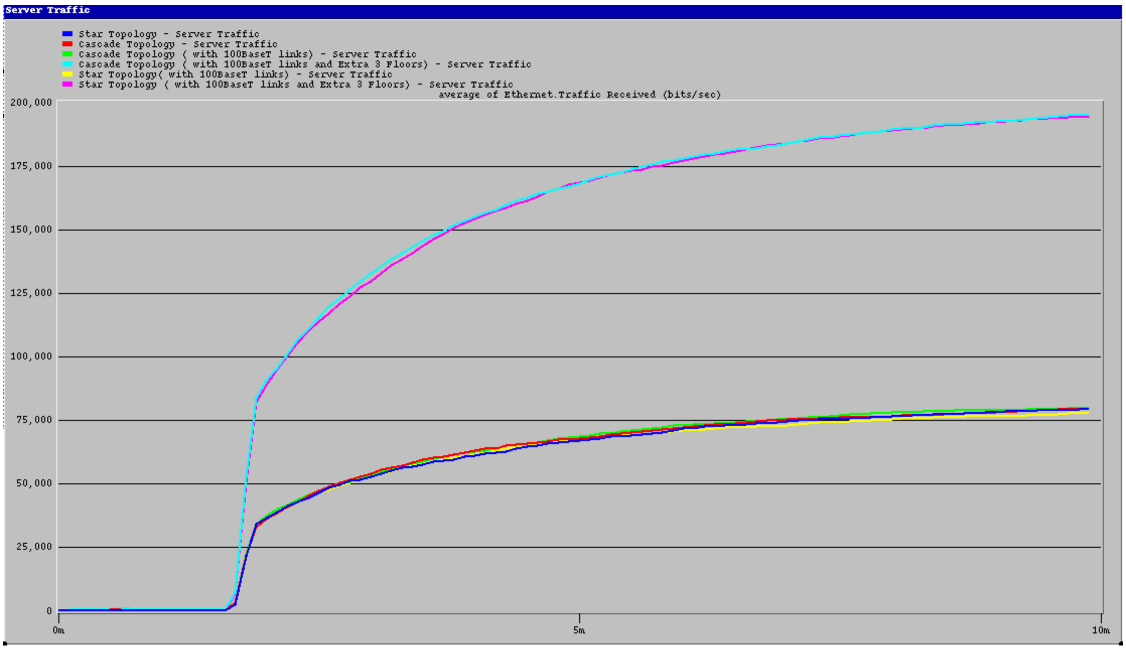


Fig 8.5 shows the traffic generated in the six different scenarios

This graph shows that the amount of traffic depends on the number of workstations used, because Internet traffic was not introduced to the networks that were built. Thus, it can be noticed that when three extra floors were built, the maximum traffic value that was generated, was nearly equal to 2.5 of the original maximum traffic value. Since the maximum utilization of the 10BaseT links was not reached, using100BaseT links did not change the amount of traffic that reached the Server.

The sixth parameter that will be discussed is the 'Server Delay':

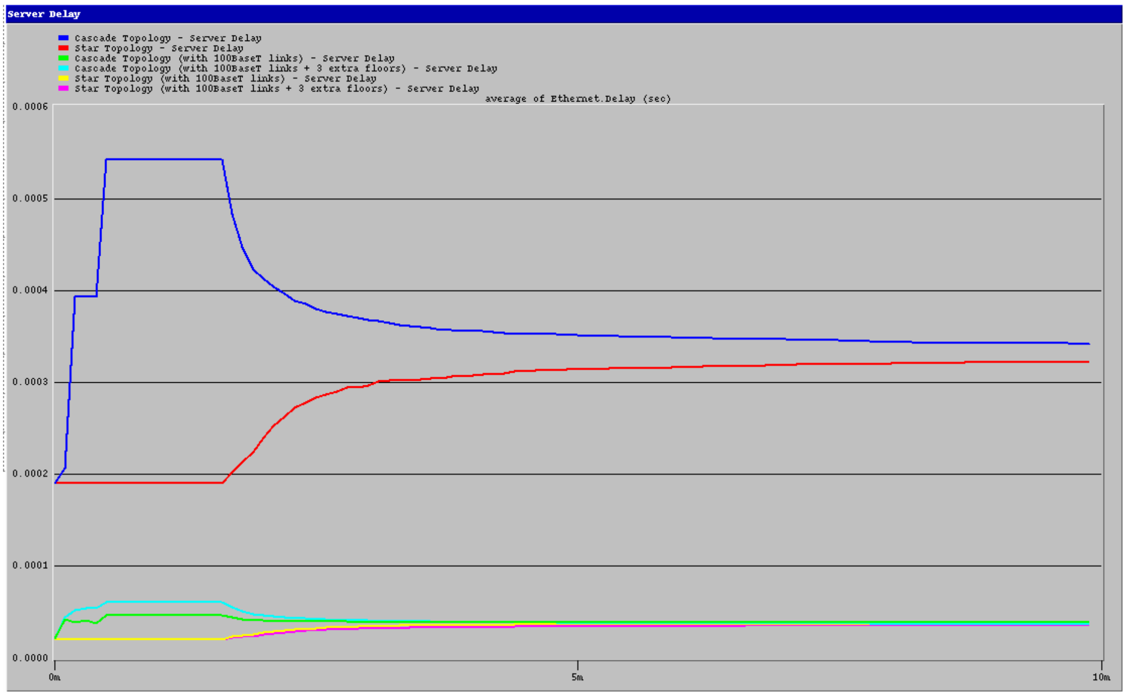


Fig 8.6 shows the server delay in the six different scenarios

This graph shows that the delay that happens in the server is directly proportional to the capacity of links used. After changing the links from 10BaseT to 100BaseT, the delay decreased dramatically. The 100BaseT links were able to accommodate the extra traffic after adding three extra floors, that's why there was no increase in delay. An increase in the server delay would appear, if many floors were added such as thirty or forty floors.

Another scenario that was tested is where both of the two different topologies could differ. So, in order to achieve this goal, room four in both of the topologies was tested thoroughly and here is what was found:

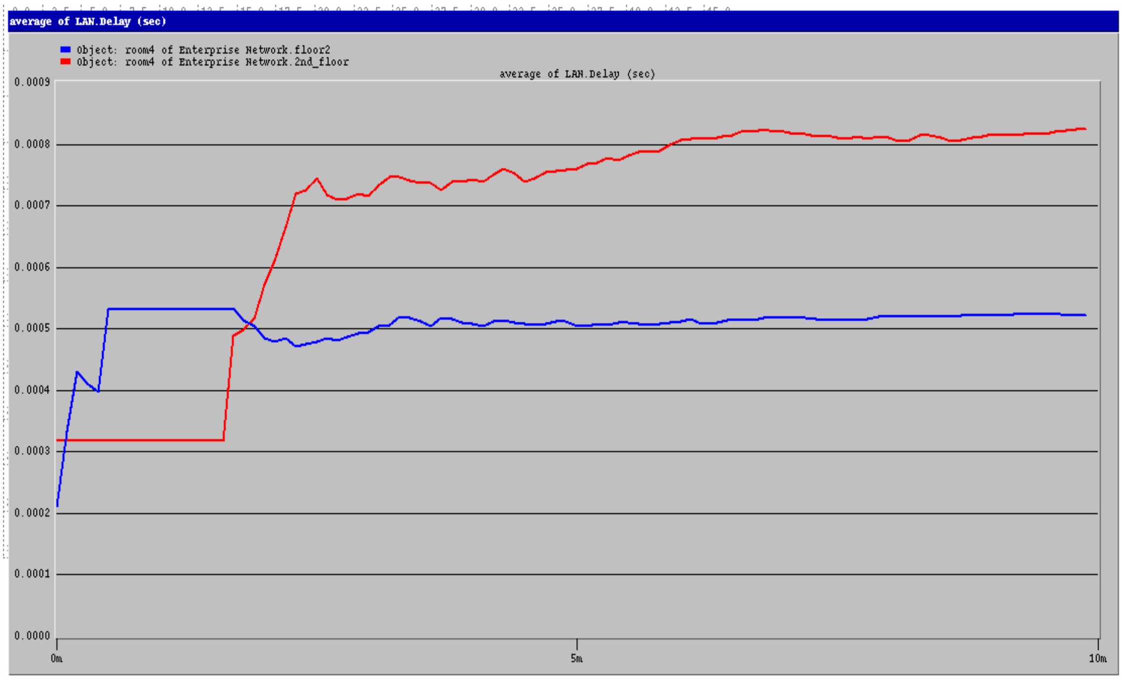


Fig 8.7 shows the difference between star and cascade topology when comparing the delay in room 4

The above graph shows that the delay, which appeared in room four in the Star topology is higher than the delay in room four in the cascaded topology. This increase in delay was caused because of the devices that are connected to room four which are used to route the traffic out or into room four. In the Star topology, room four is connected only to a switch; on the other hand, room four in the cascaded topology is connected to two routers. This allows the traffic (packets) not to 'stack' up and wait (queuing delay) for their turn to exit or enter to room four, so that's why in the cascaded topology there is lower queuing and propagation delay! Also, the cascaded topology links suffer from a higher utilization percentage, since the LANs are connected in series (traffic passing through the links increases). On the other hand, the link utilization is lower in the Star topology.

8.1.3 Performance under priority model:

The priority introduced to the network model made its performance much better in terms of quality of service for certain applications such as video and audio packets while keeping poor the performance of the other application which based on the low priority packet, for this reasons many type of priorities were introduced to solve this problems for example time priority head of line is a good choice as far as high priority is concerned since it give a good chance for the high priority packets to be served well, but this is not the case with low priority packets since very poor services are provided for these packets. The other type of time priority, which is the threshold base gives a balance solution for this issue since it has special value that makes the service switch to the priority mode when it is exceeded while keeping a fair service in case this value is not reached.

Parameters and Results:

Throughout experiencing the parameter of the switch model, it is found that there are certain parameters that have a great effect on the priority and therefore causing the performance of the system to reach its optimum state if the values of these parameters were chosen appropriately or the other way around where the performance goes to its minimum level and therefore the priority effects are not noticed when the value of the parameters are not properly chosen.

1- **Service rate:** Queue service rate that serves the packets stored in the buffer with a constant rate, the effect of this parameter is that the higher the service rate the better service for the application, however the priority effect will not be noticed if this rate is very high since both high and low priority will be served approximately equally due to high speed of the queue server unless there are large number of the workstations that generate low priority packets, now in order to observe or make the priority effect takes place in the network the value of this parameter should be reasonable (neither very high nor low).

Results 1and 2 shows the effect of changing the service rate on the following aspects respectively:

- High priority service
- Low priority service
- Buffering delay (packet waiting time inside the buffer)
- End to end delay

Default service rate (9600 bit/sec):

<u>Units</u>

packets

packets

sec

sec

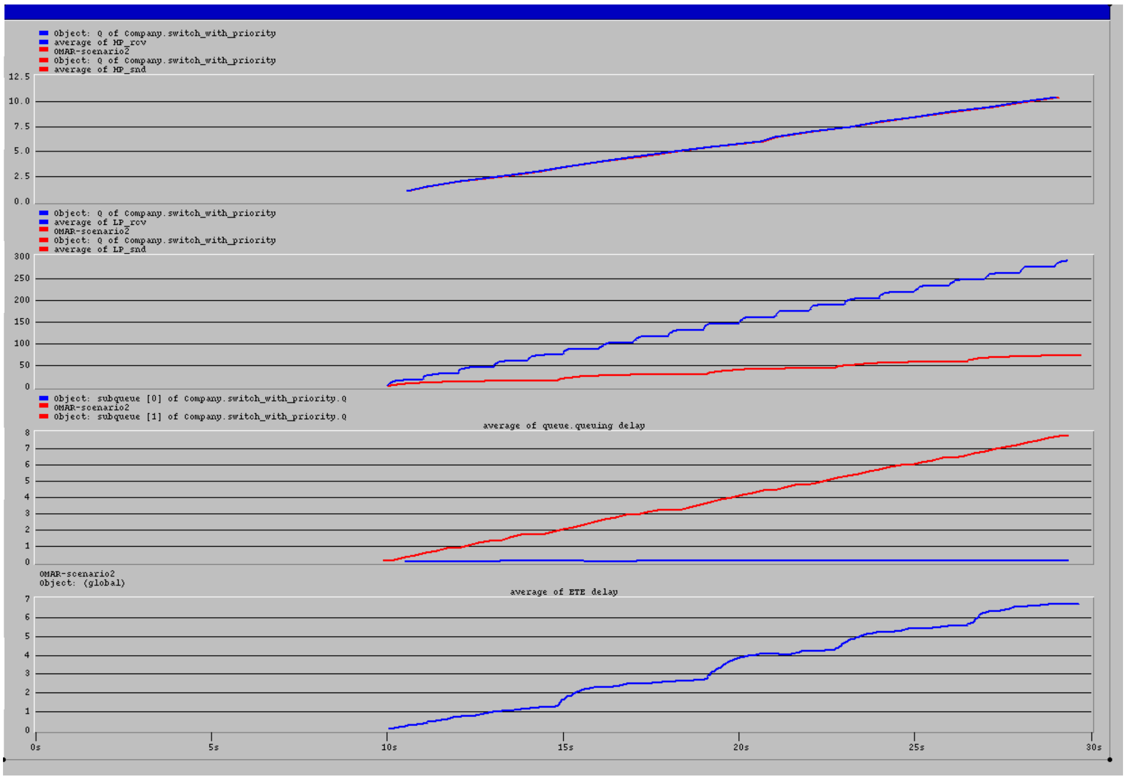


Fig 8.?? **(Result 1)**

<u>Graph 1</u>: (High priority service)

The two curves (blue one indicates the load received by the switch, red one indicates the amount of packets transmitted by the switch) are exactly over each other which indicates that the amount received equals the amount sent therefore giving the high priority packets good service due to the time priority head of line that serve the high priority packet most of the time.

Graph 2: (Low priority service)

The two curves (blue one indicates the load received by the switch, red one indicates the amount of packets transmitted by the switch) are not matched which indicates poor service for the low priority packets due to the time priority head of line that often serve the low priority packet (time when there are no high priority packets at buffer 0 of the queue).

Graph 3: (Subqueue Delay)

This corresponds to the time that a packet waits inside the buffers before it gets processed or served by the queue server, the blue curve indicates delay of the high priority packets inside the subqueue 0 and from the graph it looks constant which is expected since the server most of the time is serving the high priority packets, while the red curve indicates delay of the low priority packets inside the subqueue 1 and from the graph it looks linear which is true since the low priority packets will take more time waiting for the service as long as the server is serving the high priority.

Graph 4: (end-to-end delay)

It is desirable that the end to end delay to be as small as possible to ensure the quality of service, in this case the end to end delay is reasonable since the service rate is high enough to adapt the low priority packet delay.

Result 2 (service rate 2500 bit/sec):

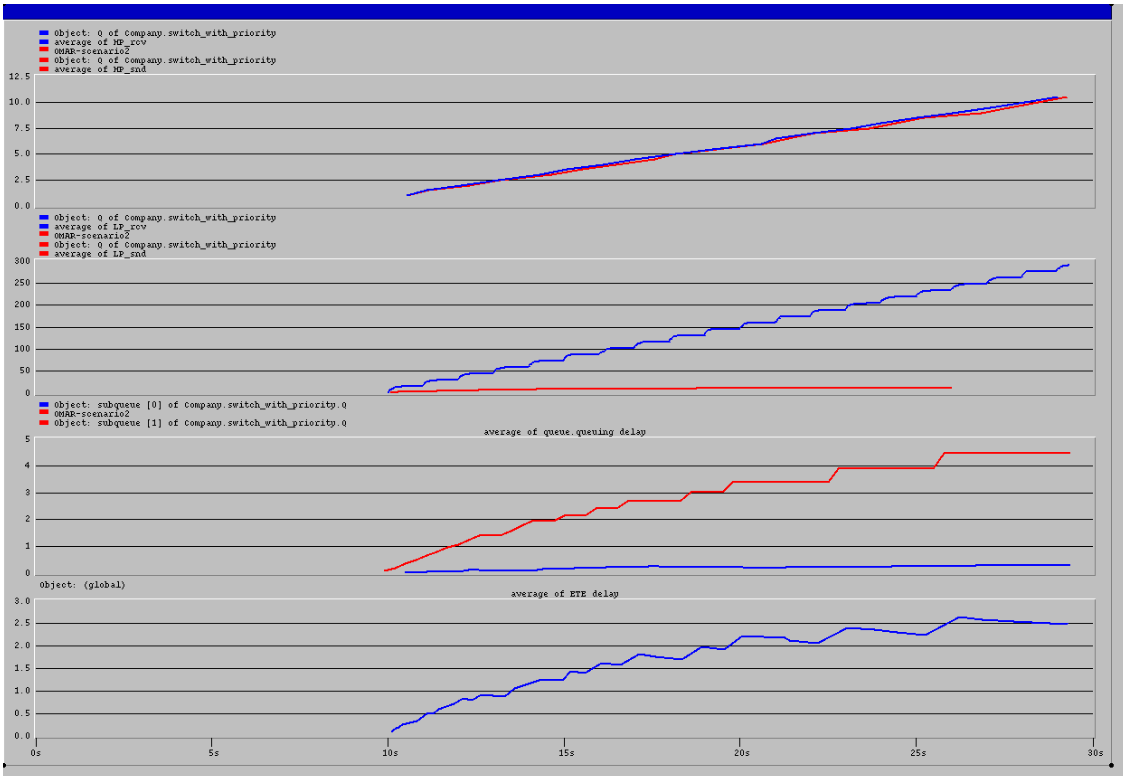


Fig 8.?? **(Result 2)**

<u>Graph 1:</u> (High priority service)

The two curves are nearly over each since the service rate in this case is slower than the one in the previous case but he results shows that the service for high priority packets is good enough since the priority type is head of line.

<u>Graph 2:</u> (Low priority service)

The two curves are further away from each other than the case in 9600, since the service rate in this case is slower than the one in the previous case in addition to that the server dose not frequently serve the low priority packet therefore the results will be obvious

Graph 3: **(Subqueue Delay)**

Same as the previous time but he the maximum delay value for the high priority packets will be a little bigger, while the maximum delay for the low priority packets will much higher than the previous case.

Graph 4: **(end-to-end delay)**

In this case the end-to-end delay is a little bit large which considered a quite normal.

* It is found that the higher the services rate the better performance but this will be expensive in terms of the switch price.

2- **Buffer size:** the size of the buffer has a quite good effect on the space priority where the space is the main issue, but it also affects the priority of the system, for example if the size of the buffer is small and the service rate is not high enough to handle all the packets at certain time, this will lead to filling the buffer till its limit and after that there will be large number of loss in the high priority packets which will cause the performance of the system to goes down.

3- **Packet Arrival time:** is the time that a packet is generated which could be constant or exponential, this effect will be noticed when the service rate and the buffer size of the queue are very low to handle the fast packet arrival rate.

4- **Others:**

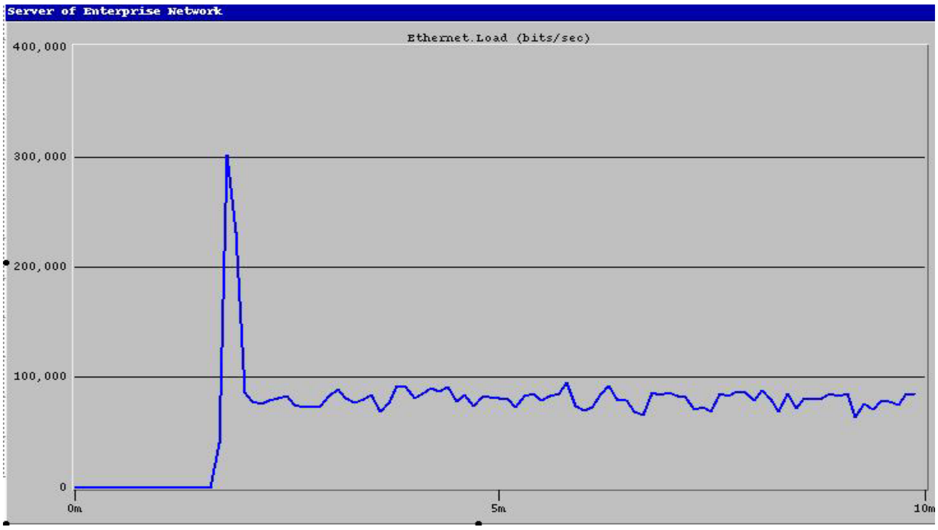
There are other parameters that affect the priority concept like type of the links packet format, etc.

** In general the above parameters are very related to each other so if the user change in one of them it will affect the rest and therefore affect priority concept which leads to poor performance as far as the high priority service is concern, therefore keeping all the parameters at a balance state when changing any other* *parameter will ensure the priority effect on the system which will lead to better performance*

- In case of threshold base the result will be similar but the effect of the low priority packets will be less since the threshold base was designed in order to balance the effect of the high priority over low priority, but this is not always the case because this depends on the threshold value which if it is high then the above explanation will be applied while if it is low then the high priority packet will take over the low priority packet completely in terms of service.

8.4 Problems Encountered:

The first problem that was encountered was the strange looking graph that showed a sudden increase of the load on the Server.



But then it was concluded that this phenomena occurred because all of the workstations starting up at the same time and requesting from the server to be served. After the spike, randomization happens to the connections going from the workstations to the server and then the load settles at about 80,000 (bits/s).

Another problem that was noticed was the “decaying” graphs of the throughput and traffic that was obtained after simulating the newly built cascaded network. Building on the facts that the graphs obtained should look and measure exactly like the Star network, since all of the variables such as the number of offices, number of workstations, number of floors, etc, were kept the same, the cascaded network was thoroughly checked, and each link was tested to see if the 10BaseT connections were enough to carry the traffic. The problem appeared that one of the links that connected Room 3 with the router, was down, which affected the whole cascaded network connections (which is theoretically correct, since we know that if any link goes down, the whole cascade topology including all its components will be greatly affected). Thus after replacing the 10BaseT link, the cascaded network was up and running again. Enabling us to run all of the simulations again and obtaining the results, which were inspected alongside the Star network simulation results.

When the problem of using a switch with 102 interfaces has occurred in the 105-workstation office, splitting the network into two sub-networks was the action taken to solve this problem. Another solution is to use a switch with more number of interfaces than the one used. This is not possible because in Rapid Configuration, only the switch of 102 interfaces is supported. Therefore, the first solution is chosen for this problem

The concept of priority wasn't introduced to the enterprise network due to the complexity of modifying the priority model (Q module in the switch) to be compatible with the original network.
The reason is that the Q module in the switch (see figure 7.8) was designed to deal with plain packets (mainly has one field) that are generated with a simple source (see src in fig 7.11) while the packets generated in the original network by the workstation (fig 5.3) are not plain packets and passed through several stages before they gets transmitted, at each stage the packets will be processed and given an extra information like TCP header, MAC and IP addresses. Modifying the q model to deal with these types of the packet requires more time than the available for this project since dealing with each fields of the packet requires large amount of coding and understanding of the where each extra information will go and how is going be processed.

The best solution was to create a new network model (fig 7.4)from scratch by creating the switch, workstation, packet format and link model using the node and process editor. This network is suppose to resemble the original one but it deals with simple type of packets at the beginning and at a later stages extra information (TCP, MAC and IP) will be added to it in stages until the packets used in the original network is reached then study the performance of this network after applying the concept of priority to it. (Due to time limitation and the amount of effort in designing, coding, testing and managing both node and process models for each network object only first floor was finished).
This is the best way for the students to learn more about the infrastructure of the network components and how they work.

8.3 Proposal for enhancement or redesign:

- The project can be enhanced by integrating the priority concept with the two different networks, where the effects of different priority could be intensively analyzed and researched.

- Creating new topologies but with different packet formats and protocols, in order to analyze the effect of changing the packet format or protocols on the whole network.

- Introducing internet traffic to the network to test its effect.

- Adding security measures to the network in order to minimize attacks from inside and outside the company.

- Testing the effect of connecting the different networks together (LANs) in order to form a WAN.

References

[1] OPNET official Manual Tutorial..

[2] Computer Networking: A Top-Down Approach Featuring the Internet, Second Edition, by James F. Kurose and Keith W. Ross.

[3] www.howstuffworks.com/router.html

[4] Disaster recovery for LANS: A Planning and Action Guide, by Regis J. Bates

[5] Netware LANs Performance and Troubleshooting, by Ian Theakston.

[6] Introduction to IP and ATM Design and Performance, by J.M Pitts and J.A Schormans.

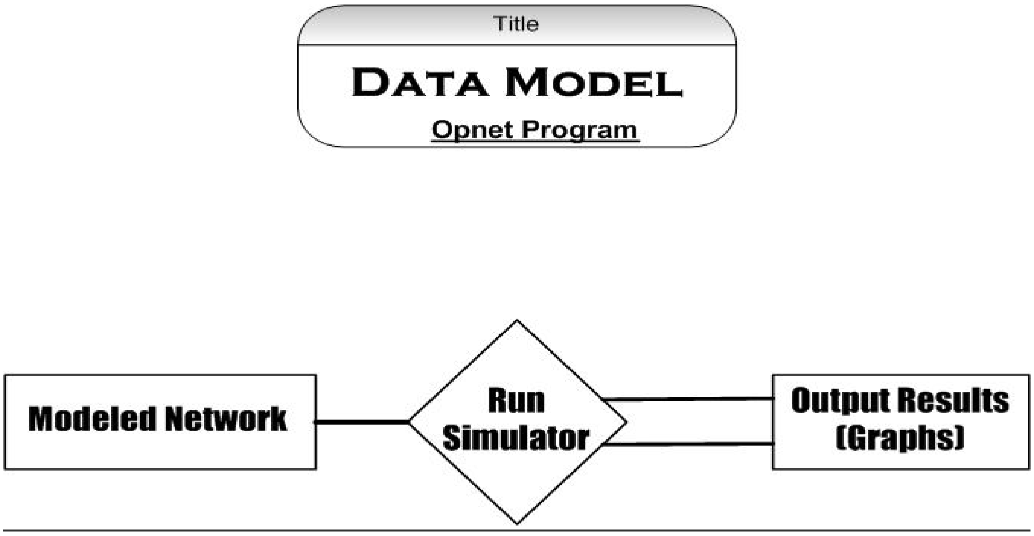
[7] Computer Networks: A Systems Approach, by Larry L.Peterson and Bruce S. Davie

[8] www.webopedia.com

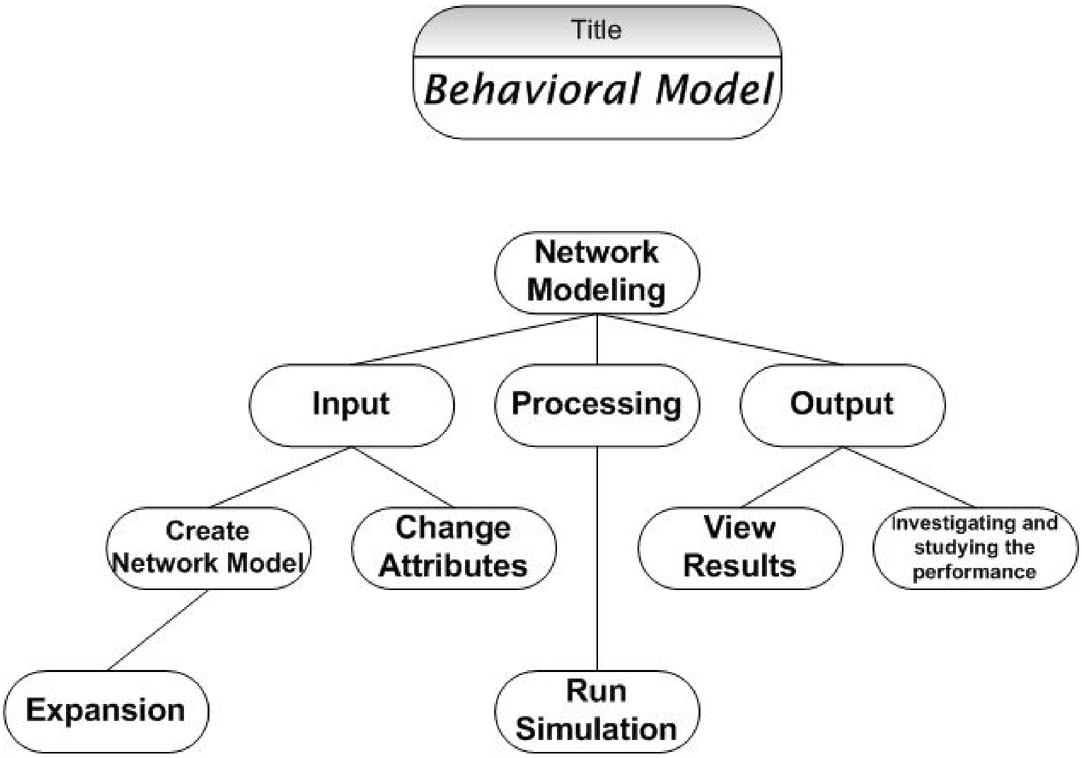
[9] Data and Computer Communications, Sixth edition, by William stallings

Appendix A

Data Model:



10.2 Behavioral Model:



.

Appendix B

Appendix C