A Lab report
On

SMALL INTER NETWORKS

By

NAME**:** **TOBENNA UZOMA, IHEGBOROW**
STUDENT ID: **A00267346**
INSTRUCTOR: **JACKIE STEWART**

Aim: A company has a star topology network on the first floor of its office building and plans to add an additional star topology network on another floor. The aim of this report is to see how OPNET Modeler can be used for organizational scaling to build and test a “what-if” scenario for the company ensuring that the load added by the second network will not cause the network to fail and also confirming that the server will be able to handle the additional load from the second network.

Objectives:

The simulation of a real world scenario to the test the following;

- To test the load caused by the second network on the server.
- To test the network delay caused by the network from the second floor.

Introduction/Backgrounds:

The scope of this practical was carried out with OPNET MODELER SUITE, now known as RIVERBED MODELER. OPNET is now part of Riverbed since October 2012 for about $1 billion US dollars. OPNET Technologies, including network simulators, build upon Riverbed's strong heritage of delivering industry-leading solutions to drive application performance. "*OPNET*" was Alain Cohen's (co-founder, CTO & President) graduate project for a networking course while he was at MIT. OPNET stands for “Optimized Network Engineering Tools”. Alain, along with brother Marc (co-founder, CEO & Chairman) and classmate Steven Baraniuk, decided to commercialize the software. The company's first product was OPNET Modeler, a software tool for computer network modelling and simulation. Since then, it has been diversified to provide a range of solutions for:

- **Application performance management** which include ACE Analyst Standard, ACE Analyst Plus, ACE Enterprise Management Server, OPNET Panorama, ACE Live Appliance, ACE Live Rover, ACE Live on RSP, ACE Live VMon, IT Guru Systems Planner
- **Network planning and engineering** which include IT Guru, SP Guru Network Planner, SP Guru Transport Planner, IT Guru Network Planner, NetOne, VNE Server, Report Server, IT NetMapper, IT Sentinel, SP Sentinel, OPNET nCompass for Enterprises, OPNET nCompass for Service Providers.
- **Network Research & Development** which include OPNET Modeler, OPNET Modeler Wireless Suite, OPNET Modeler Wireless Suite for Defence.

OPNET works on Microsoft (Windows 2000, Server 2003, XP, Vista, 7 and 10) and LINUX (Red Hat Enterprise and Fedora) platforms. The supporting software for OPNET Modeler are Microsoft (Visual C/C++ 6.x, Visual Studio .NET 2002, Visual Studio .NET 2003, Visual Studio 2005, Visual Studio 2008, or Visual C++ 2008 Express Edition) and Linux (gcc 3.4 or higher).

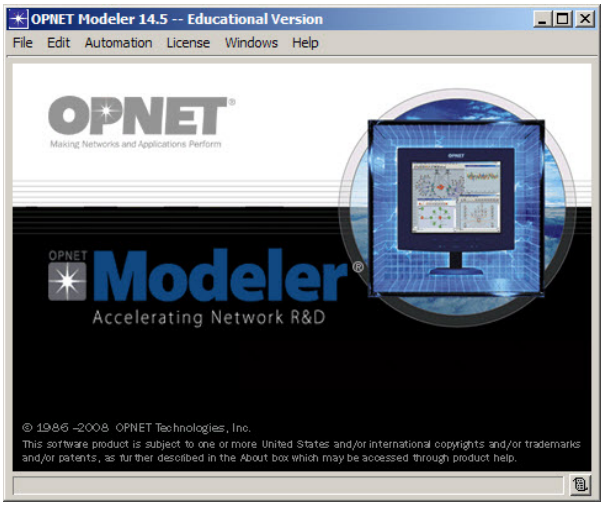


illustration1: OPNET Modeler Interface

OPNET is one of the wide variety of tools available for network planning and design. Network planning and design is an iterative process, encompassing topological design, network-synthesis, and network-realization, and is aimed at ensuring that a new telecommunications network or service meets the needs of the subscriber and operator. The process can be tailored according to each new network or service. Not all networks are the same. As data is broken into component parts (often known frames, packets, or segments) for transmission, several factors can affect their delivery;

- **Delay:** It can take a long time for a packet to be delivered across intervening networks. In reliable protocols where a receiver acknowledges delivery of each chunk of data, it is possible to measure this as round-trip time in seconds.
- **Packet loss:** In some cases, intermediate devices in a network will lose packets. This may be due to errors, to overloading of the intermediate network, or to the intentional discarding of traffic in order to enforce a particular service level.
- **Retransmission:** When packets are lost in a reliable network, they are retransmitted. This incurs two delays: First, the delay from re-sending the data; and second, the delay resulting from waiting until the data is received in the correct order before forwarding it up the protocol stack.
- **Throughput:** The amount of traffic a network can carry is measured as throughput, usually in terms such as kilobits per second. Throughput is analogous to the number of lanes on a highway, whereas latency is analogous to its speed limit. In other words, Throughput is the rate at which data is traversing a link. For example, take a look at the virtual router output below. The throughput rate is 643Kbps in each direction. Goodput is the rate at which useful data traverses a link. Assuming an uncongested path between endpoints, goodput and throughput will be as close as they are theoretically able to be.

  The difference between goodput and throughput is that throughput is the measurement of *all* data flowing through a link whether it is *useful* data or not, while goodput is focused on useful data only. Throughput measurements, such as those reported by router interface statistics, cannot distinguish the nature of the data flowing through the interface — merely that bits have gone past. Throughput is not the same as goodput because throughput can include undesirable data such as data retransmissions, or overhead data such as protocol wrappers.

The practical was written based on the OPNET Modeler 17.1.A. The workflow for OPNET Modeler (that is, the steps used to build a network model and run simulations) centers around the Project Editor. Using the Project Editor, I created a network model, chose statistics to collect from each network object or from the whole network, executed the simulation, and viewed and compared the results.

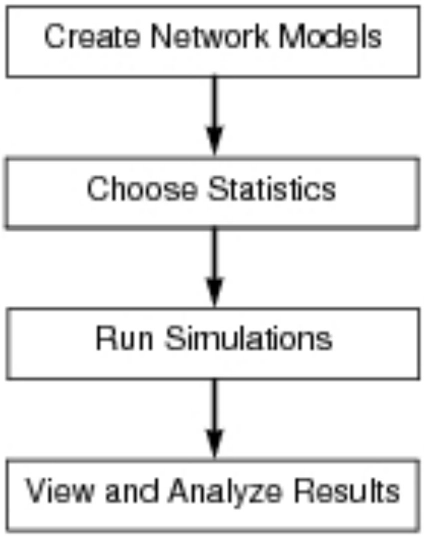


illustration 2:Process diagram for Profile Editor.

I went through the OPNET Modeler in Lesson 1tutorial to show me how to use the Project Editor to build a small internetwork. There I also found out I can also use advanced editors for specialized needs. For the nuts-and-bolts aspects of modeling, such as programming the underlying process of a particular network object or defining a new packet format, one would have to use additional editors.

Method:

- Created the scenario for the first floor. Here, I defined the size, topology, link, nodes and scale of this floor. The first floor is made up of a single server, switch and 30 nodes (hosts/end users) sending and receiving files into/from the server.

  Specify Size: *100 m x 100 m*
  Select Technologies: *Sm_Int_Model_List*
  Rapid Configuration: *Star*
  Centre Node Model: *3C_SSII_1100_3300_4s_ae52_e48_ge3*
  Periphery Node Model: *Sm_Int_wkstn*
  Number (of periphery nodes): *30*
  Link Model: *10BaseT*
  Centre X x Y: *25 x 25*
  Radius: *20*

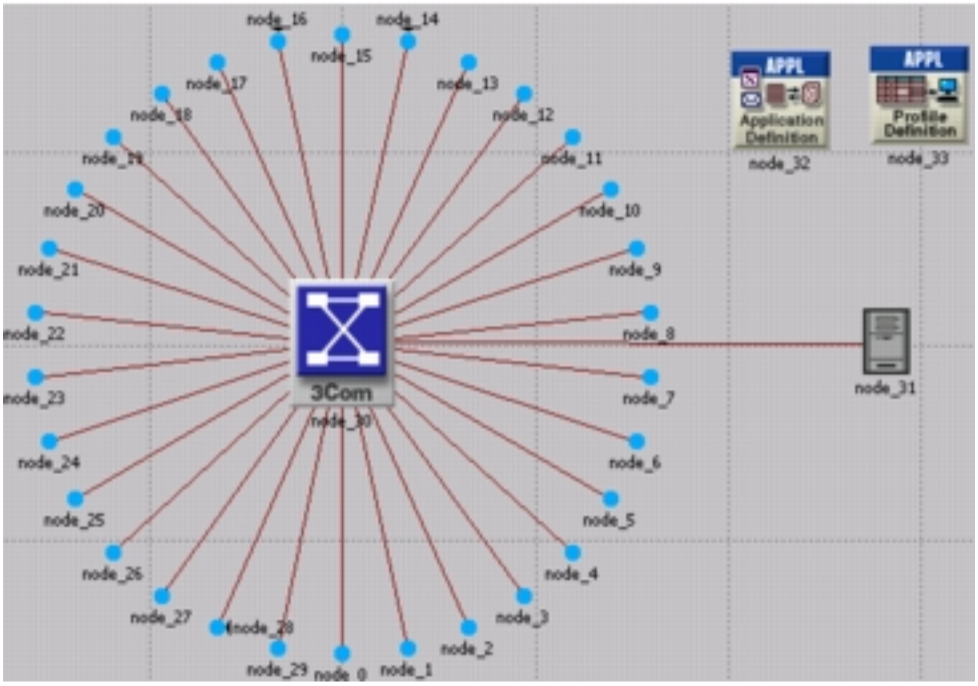


Illustration 3: the first floor.

- Selected the Object and Global stats which were Server load and network delay.
- Ran the simulations and captured the results.
- Duplicated the first floor, called it Expansion. Expanded the network by adding 15 more nodes, a router and connected both networks using a switch. Saved the Expansion floor, ran and compared the results of the server load and the network delay (Object and Global statistics) to answer the “what-if” question.

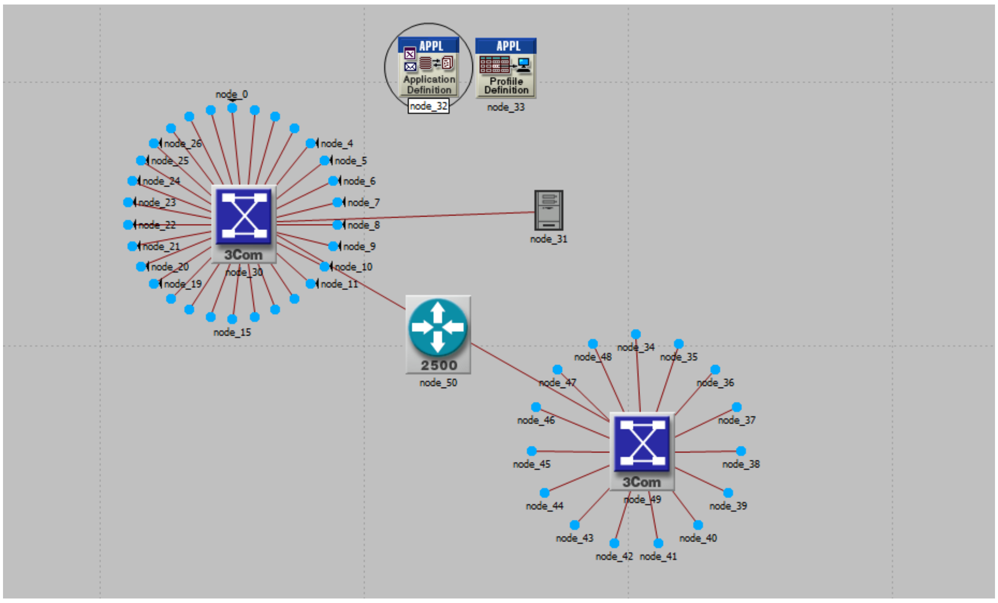


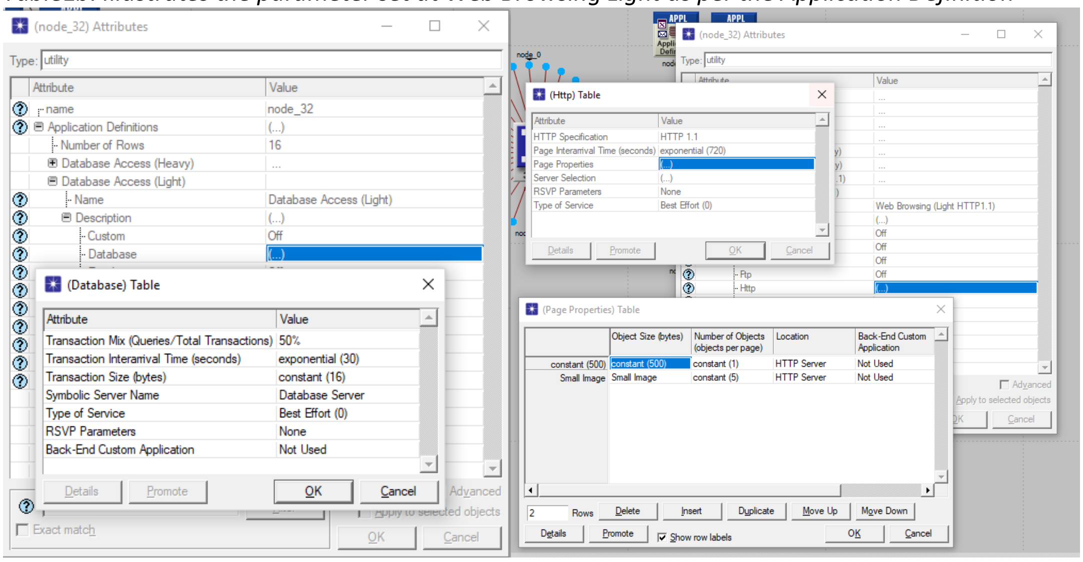
Illustration 4: First & Expansion floor

- Duplicated the Expansion network making this the third floor. Changed the load on the network (Web Browsing Heavy / Database Heavy)). Ran the whole network again and compared the results of the Objects and Global statistics across all three floors.

Results:

- **Theoretical Result**:

Table1a: Illustrates the parameter set at Database Light as per the Application Definition
Table1b: Illustrates the parameter set at Web Browsing Light as per the Application Definition



The above *tables1(a & b)* show the **network load at Database Light and Web Browsing Light** results for the **First & Second (Expansion) floors** in the “*What-If*” scenario for the company.

Inter-arrival time is set at – “Exponential” 30 seconds.

Transaction size (file or packet size) is 16 bytes constant.

Calculations of Load on network based on the above traffic type:

$$\frac{\textbf{File or packet size}}{\textbf{Inter-arrival time}}$$

That is;

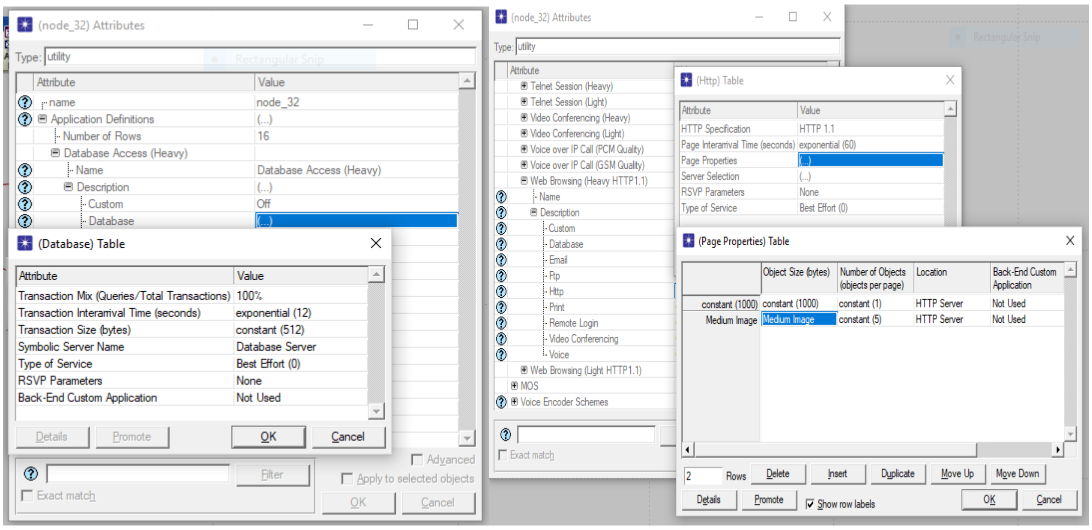
$$\frac{16\text{Bytes} \times 8 \text{ bits per byte}}{30 \text{ seconds}} = \textbf{4.267 bits/s per workstation}$$

Hence:

Utilising 45 workstations (nodes) = 192 bits/sec - onto the network

Utilising 30 workstations (nodes) = 128 bits/sec - onto the network

Table2a: Illustrates the parameter at Database Heavy as per the Application Definition
Table2b: Illustrates the parameter set at Web Browsing Heavy as per the Application Definition



The above *Table2(a & b)* shows the **network load at Database Heavy and Web Browsing Heavy** results for the **First, Second (Expansion) and Third (Database Heavy) floors** in the "*What-If*" scenario for the company.

Inter-arrival time is set at – "Exponential" 12 seconds.

Transaction size (file or packet size) is 512 bytes constant.

Calculations of Load on network based on the above traffic type:

$$\frac{\textbf{File or packet size}}{\textbf{Inter-arrival time}}$$

That is;

$$\frac{512 \text{ bytes x 8 bits per byte}}{12 \text{ seconds}} = \textbf{341.33 bits/s per workstation}$$

Hence:

Utilising 45 workstations (nodes) = 15,360 bits/sec - onto the network

Utilising 30 workstations (nodes) = 10,240 bits/sec - onto the network

- **Simulation Result**

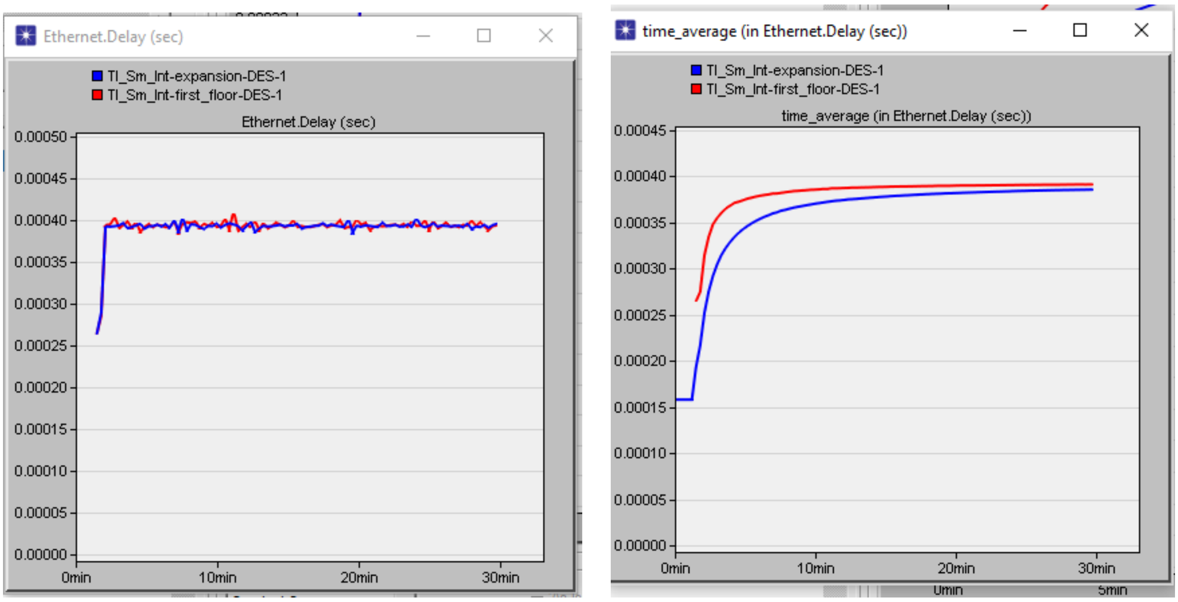


Fig1a: Illustration of the Network delay for first & second floor (As-Is)
Fig1b: Illustration of the Network delay for first & second floor (Time Avg)

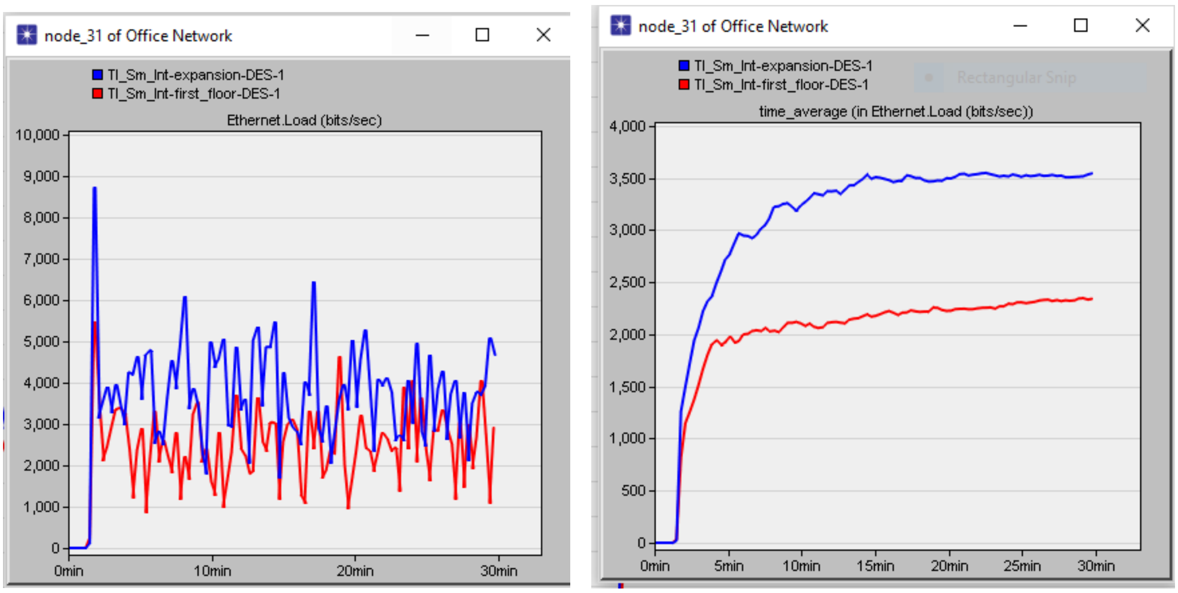
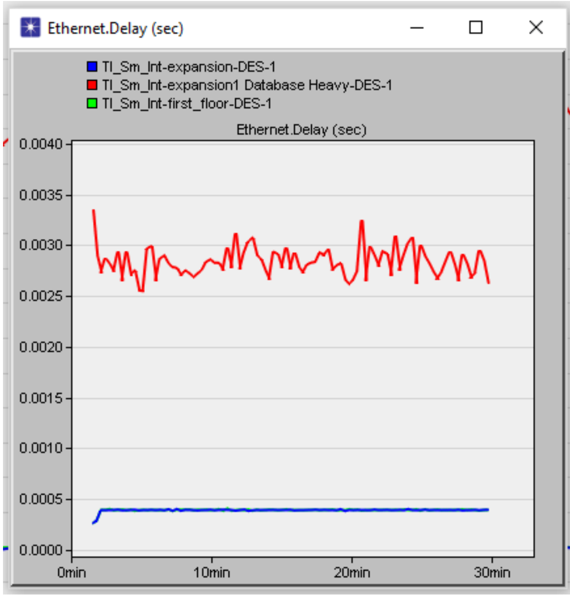


Fig2a: Illustration of the Server Load for first & second floor (As-Is)
Fig2b: Illustration of the Server Load for first & second floor (Time Avg)

The above illustrations *Fig1(a, b) and Fig2(a & b)* compares the **network delay** and **server load** results for the **First & Second (Expansion) floors** in the "*What-If*" scenario for the company which was set at Default Database Light.



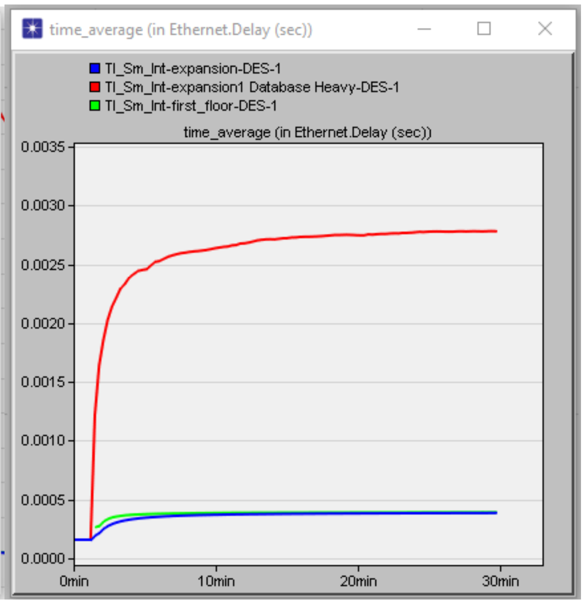
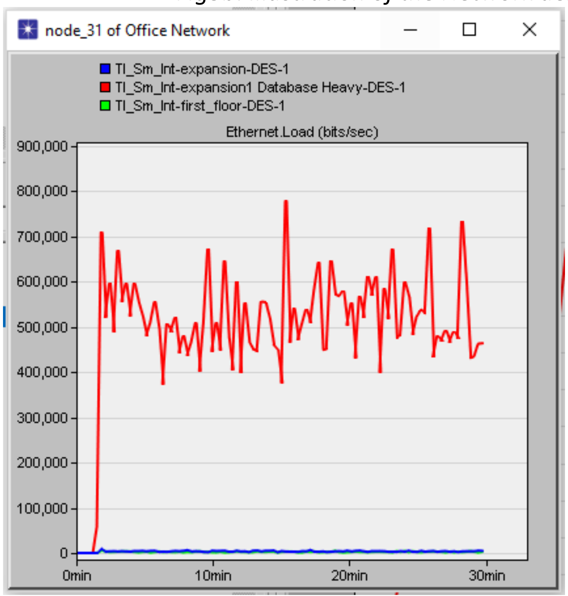


Fig3a: Illustration of the Network delay for first, second & third floor (As-Is)
Fig3b: Illustration of the Network delay for first, second & third floor (Time Avg)



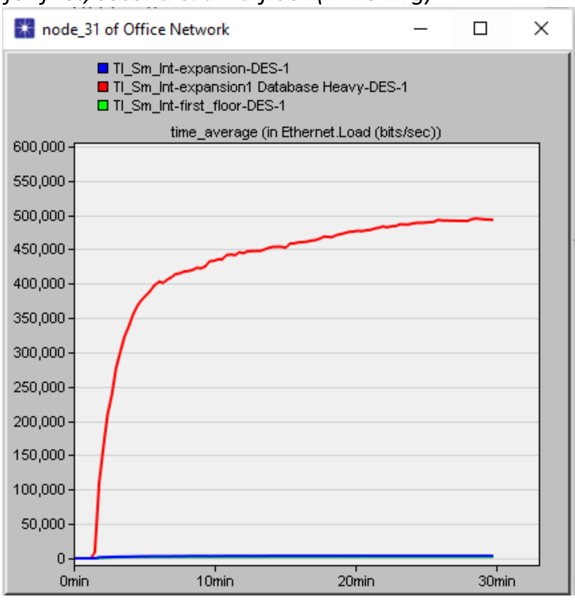


Fig4a: Illustration of the Server Load for first, second & third floor (As-Is)
Fig4b: Illustration of the Server Load for first, second & third floor (Time Avg)

The above illustrations Fig*3(a, b,) and Fig4(a,b)* compares the **network delay** and **server load** results for the **First, Second (Expansion) and Third (Database Heavy) floors** in the "*What-If*" scenario for the company which was set at Default Database Heavy.

Discussion:

This report is to test a company's star topology network on the first floor of its office building having an already existing 30 nodes (end users). The company planned on adding an additional star topology network on another floor with an addition of 15 extra nodes. Using OPNET Modeler, I built and tested a "what-if" scenario for the company to ensure that the load added by the second network will not cause the network to fail and also confirming that the server would be able to handle the additional load from the second network. In addition, I also tested another duplicate second floor on the network to see if the server could handle all three floors with a heavier application traffic.

Looking at the graphs in the graphs in *Fig1 & Fig2*, it compares the network delay and server load for the first and second floor. It reveals there is a drop in the network delay, which was expected, from about 0.00040secs before levelling out at about 0.00037secs. This delay is still well within the threshold for latency. It also shows and increase in the server load, also expected, from about 2,400bits/sec before levelling out at 3,500bits/sec. Also, looking at the application traffic in *Table1*, where the network was set at database light, The inter-arrival time for 16bytes packets is at 30secs exponentially where service is running at best effort. Also, the Web browsing on each node, the minimum objects per page is set at 1 where the object size of at least 500bytes having at least 5 small images. A simpler way of saying all of this is, the server is performing optimally with the inclusion of the additional nodes from the second floor, hence, there is no reason to purchase a new server at this level.

To test the additional (duplicate) third floor, I duplicated the second floor again and also changed the application type to database heavy. That is, increasing the network traffic on this floor. Comparing the graphs in *Fig3 & Fig4*, we can observe that the network traffic was increased. The delay changed from 0.0004secs to 0.00027secs before eventually levelling out remaining stable. Again, this delay is still well within the threshold for latency. There was also a change in the server load, showing a significant increase in value when the traffic is increased to database heavy rising to about 500,000bits/sec and eventually remaining stable on this level. With the network traffic now set at database heavy, the inter-arrival time increases to 512bytes packets at 12secs exponentially where the service is once again running at best effort. Looking at the Web browsing heavy, each node has a minimum objects per page set at 1, an increased size of at least 1000bytes and 5 medium sized images. With all of this, we can see even with the heavier workload on the server it still functions perfectly so getting a new server is not required.

Also note, another difference in both scenarios is as a result of the headers. The first theory result doesn't contain headers. The headers were included as a result of the increase in the traffic types set to Database Heavy.

Conclusion:

I built the scenario in the first floor, built the expansion network, duplicated the expansion network and analysed them on a different traffic type, all aimed to find out if the business man would require a new server if he was to expand his company by another floor. From this I can say that the business man does not require a new server as his already existing server can carry enough load even if he expands his network by 15 end users and changes the network to different traffic types.