UNIVERSITY OF HERTFORDSHIRE

Faculty of Science, Technology and the Creative Arts

Modular BSc Honours in Computer Science

6COM0287 – Networks Project

Final Report

April 2012

Learning about computer networks using network simulation software

S. Collings

Supervised by: J. Malcolm

Abstract

This report compares four different network simulators. It does so by implementing four networks in each simulator. It lists additional features of the simulators that may be of use to a user and describes any problems encountered during the implementation phase.

The simulators are then compared against a set of criteria to determine what simulator is the best for implementing the networks mentioned in the specification. The winner was Packet Tracer as it offered the best combination of attributes when the networks were implemented in the four simulators.

Each of the four simulators is built for different purposes and requires different skill levels to successfully implement and run a simulation of a network.

Contents

1.0 Introduction

Learning how to implement a network can be a complex task, especially if a person is more specialised in other areas of Information Technology (IT) such as programming or databases. For people who are interested in computer networking (such as network administrators and network technicians) network implementation may be an easy task (dependent on their level of skills and experience), but before a new network goes live, the network architect/administrator may wish to try out the planned configuration of the network within a simulator to see if any unexpected issues arise.

This project will look at a variety of different network simulation software that can assist with learning about the implementation and configuration of computer networking hardware and software. A search for suitable software will be carried out, and then several networks will be created in each of the chosen network simulation programs. The steps taken to implement the sample networks will be documented for each piece of network simulation software that has been chosen. Finally, the best simulator for the task will be identified.

The scenario for this project is to implement four different networks into network simulators. The networks will use a variety of protocols (such as DNS and DHCP) and media (such as Ethernet and Wireless Networks) to provide network services and connectivity to the network.

1.1 Aims

The aims of this project are to:

- Identify suitable network simulation software
- Define the network that will be implemented in each of the simulators
- Implement a small computer network in each of the network simulation programs
- Detail the steps taken to implement the network in each simulator
- Compare the network simulators
- See if the simulators can help with troubleshooting networks

1.2 Structure of this Report

Chapter 1 introduces the report to the reader, along with setting out the aims of the project.

Chapter 2 introduces the network simulation software, along with the research methodologies and choosing the simulators to be used in the project.

Chapter 3 discusses some of the best practices for implementing a computer network.

Chapter 4 sets out the specification for each of the networks to be implemented in the simulators, as well as covering a little network security and the types of attacks a network could face.

Chapters 5, 6, 7 and 8 detail the process of implementing the networks within each of the different simulation software packages.

Chapter 9 compares the statistics from each simulator against each other to see how much of a difference there is between the results.

Chapter 10 compares each of the simulators against a list of criteria.

Chapter 11 includes discussion about the project and the conclusion.

2.0 A Brief Introduction to Each Network Simulator

2.1 Research Methodology

To find the network simulation software used in this project, search engines were used along with the search term ‘Network Simulator’. This search produced results for GNS3, OPNET and NS. Hyper-V, ESXi and Cisco Packet Tracer were all found through other sources such as friends (ESXi/Hyper-V) and training courses (Packet Tracer).

2.2 Virtualisation software that could be useful for network implementation

The software mentioned below was also found during the research into network simulators but from the initial research, it was found that these products would be more suited to learning about networking settings and configuration of client and server operating systems (such as Linux, Microsoft Windows 7 and Microsoft Windows Server 2008 R2)

VMware ESXi

VMware ESXi is a bare metal hypervisor which allows for virtualization of operating systems such as Windows and Linux. ESXi supports multiple vSwitches allowing for an isolated test network to be set up to learn about how different operating systems can be configured for use on a network. ESXi supports VLANS and traffic shaping (VMware, 2011).

Microsoft Hyper-V

Hyper-V is a similar product to VMware ESXi, but only runs on Windows Server operating systems (unlike ESXi which is based on Linux). It has fewer networking features than ESXi, but still allows for users to create an isolated network for testing (Microsoft, 2009).

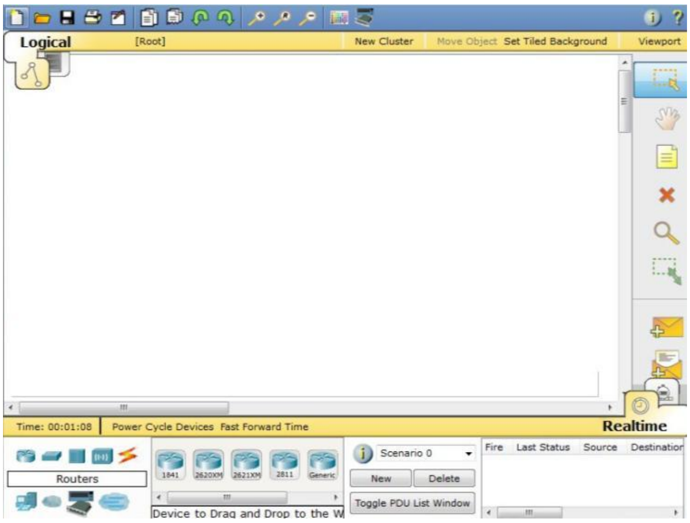
Oracle VirtualBox

Oracle Virtualbox originally started out as a program by Innotech GmbH (which then got taken over by Sun Microsystems, then Sun Microsystems was bought by Oracle) to allow for virtualisation of 32 and 64 bit operating systems. Virtualbox currently runs on a variety of host operating systems such as Windows, Mac, Linux and Solaris (these examples can all currently be run as a guest within a Virtualbox virtual machine) (Oracle, 2012).

The virtualisation software mentioned above will be looked at briefly to establish the usefulness of these programs for simulating the networking configuration options of operating systems, and to a smaller degree, using the programs to connect hosts to an isolated network.

2.3 Cisco Packet Tracer

Packet Tracer is produced by Cisco Systems and is offered for several different operating systems such as Windows and the Ubuntu/Fedora Linux Distributions (Cisco Systems, 2010). It also features a logical and physical workspace with the logical workspace allowing the user to see the topology of the network and the physical workspace allowing the user to see what the network devices would look like in real life (Cisco Systems, 2010). Packet tracer has 2 different modes of which one provides a real time mode (to simulate real equipment) and the other mode allows for users to see how data is sent across a network and how it is affected by things such as propagation delay and switching times introduced by the network (Cisco Systems, 2010). Packet tracer also supports a variety of application layer, transport layer and network layer protocols (Cisco Systems, 2010).



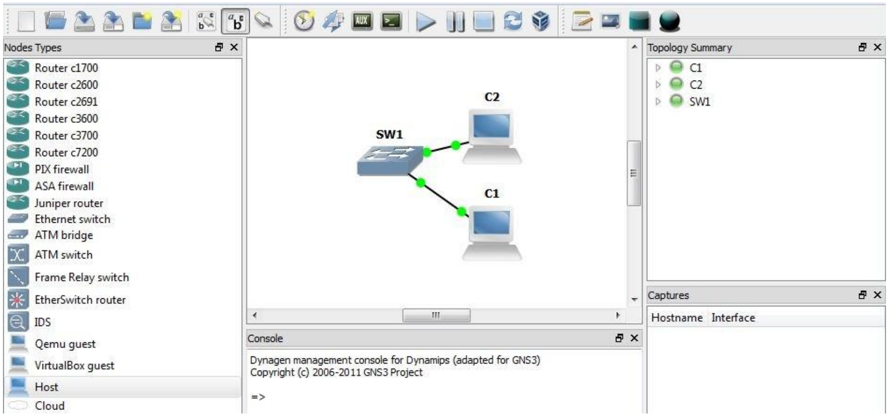
The main interface for Packet Tracer

2.4 GNS3

Graphical Network Simulator 3 (GNS3) is another network simulator that is similar to Packet Tracer. GNS3 supports both Cisco and Juniper equipment, however the user has to provide their own Cisco IOS or Juniper JunOS images before routers can be used within the GNS environment (GNS Project, 2007).

GNS3 also supports the Virtualbox and Qemu virtualization products allowing for users to build networks with virtual computers that can run operating systems such as Microsoft Windows and Linux Distributions such as Debian and Centos (GNS Project, 2007).

Graphical Network Simulator is a cross platform program (there are Windows, Mac and Linux Versions available). Due to the program being open source, and licence agreements of the proprietary Cisco IOS and Juniper JunOS, users have to provide their own IOS or JunOS images to use the router functionality within GNS3.



```
Dynagen management console for Dynamips (adapted for GNS3)
Copyright (c) 2006-2011 GNS3 Project

=>
```

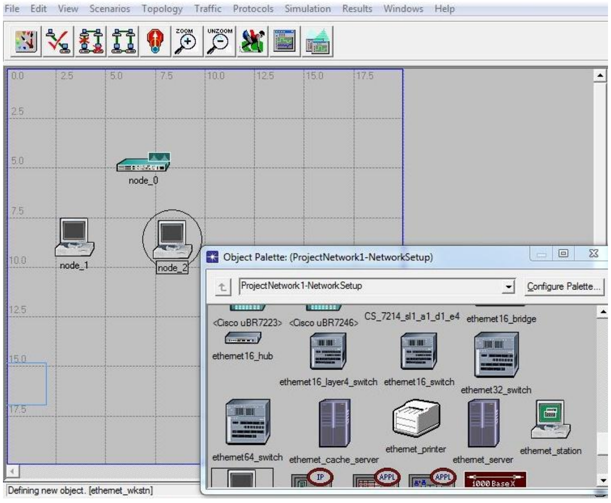
The main user interface of GNS3

2.5 OPNET IT Guru Academic Edition

IT Guru is a commercial simulator produced by OPNET. OPNET also offers a university program to allow universities to teach about networks using the OPNET range of programs.

The academic version of OPNET IT Guru has limited functionality such as limited wireless functions and limited import/export capabilities (Dunaytsev, 2010).

It allows for simulation of computer networks and various factors that may affect computer networks such as node failures or link failures. Some of the features of IT Guru include generating a network topology (e.g. Bus, Star, Mesh, and Ring) quickly as well as creating network traffic to test situations and how well the network copes with the simulated changes (Dunaytsev, 2010).



Defining new object. [ethernet_wkstn]

The main interface of IT Guru showing the main workspace and object palette

2.6 NS2

NS (Network Simulator) (University of Southern California, 2011) is probably one of the more well-known network simulators within the networking community. Development was originally started in 1989. NS2 is coded in the C++ programming language.

Network simulator has builds available that run on Linux, BSD and Macs. NS will run within a Windows environment, but Windows users will have to install Cygwin before Network Simulator will run on Windows. Network Simulator lacks a graphical user interface, so users will have to have knowledge of how to use the command prompt before using this software.

2.7 Selecting Network Simulators for Use within the project

The main pieces of network simulation software that will be compared are:

- Cisco Packet Tracer
- Graphical Network Simulator 3
- OPNET IT Guru
- NS2 (Network Simulator 2)

The above programs have been chosen as they all offer the ability to set up and simulate a complete network complete with switches, routers and computers that will typically be found on a network.

3.0 Best practices for network implementation and maintenance

There are several different guides and papers available that list a set of best practices for network upkeep/maintenance. Although the first document (National Security Agency, 2011) is mainly based around network maintenance, some of the facts presented are still relevant during the network implementation phase (such as security methods and security issues).

The first document was produced by the National Security Agency (NSA) in the United States, and lists a variety of methods as to how a home network can be secured.

For the network security section of the guide, the NSA recommends that:

- Network administration should only be carried out within the internal network (not through the internet using remote access functionality)
- Users should buy their own modem/router rather than use ISP supplied modems/routers to have maximum control over their network (as ISP supplied equipment may have restricted settings that users cannot change)
- WPA2 encryption should be used for wireless networks
- Networked devices should have strong passwords to prevent unauthorised access

The second guide (University of Tennessee, 2008) relates more to network infrastructure security and provides the following information and tips:

- Logs should be reviewed frequently and transferred to a remote server over a secure connection
- Network devices should have redundancy and any default passwords should be changed
- Unneeded network services should be disabled to enhance security
- Externally accessible services (such as SSH or RDP) should be restricted to known IP addresses/ranges
- An IDS (Intrusion Detection System) and/or an IDPS (Intrusion Detection and Prevention system) should be installed and updated regularly on networks that hold sensitive data.

The document was produced by the University of Tennessee and applied to their network and users of the network to ensure that the network remains secure whilst university data is transferred using the network.

4.0 Network Design

According to Chapter 1 of (Peterson & Davie, 2012), a computer network should be implemented with the following principles in mind:

- To be Reliable
    - By continuing to operate even in the case of node failures or other types of failures
- Manageable
    - By allowing changes as the network expands
    - By allowing troubleshooting to take place
- Cost effective
    - By not breaking down frequently
- Able to perform well
    - By having a high bandwidth throughput and low latency
    - By sending data efficiently
- Scalable
    - By allowing network expansion for new hosts and networking equipment
    - By allowing extra bandwidth to be added to accommodate new equipment being added to the network

The CCNA Discovery Learning Guide (Stewart, et al., 2008) also lists network requirements and principles. In addition to some of the points made in the previous set of requirements/principles, the Cisco guide states some additional requirements relating to network security:

- That the network should be able to protect against security incidents that are not expected
- That the network should be secure and security devices such as firewalls and security devices should be placed appropriately within the network

The CCNA guide lists the benefits of having a hierarchal network layout rather than a flat network layout. The benefits that the guide lists are:

- Good performance will be maintained, as,
- Hierarchal networks are broken down into sections (reducing the amount of broadcasts on the network), making the network easier to manage compared to flat networks

In addition to the more general performance requirements of networks listed above, talking to the client will give a good idea of what the specific needs from the network are and how these needs can be implemented in the network. The CCNA guide suggests splitting the needs and goals into two different categories, technical requirements (e.g. supporting video streaming) and business goals (e.g. allowing the company to generate a profit from internet based transactions).

Although these principles and requirements are more suited to physical networks rather than virtual networks built in network simulators, they still apply to virtual networks, especially if a network built in a simulator is then carried over and implemented in the real world.

There are two main types of computer networks:

- Peer to peer networks
- Client/server networks

Peer to peer (P2P) networks are more suited to smaller networks, as each client can act as both a client and a server. Managing a peer to peer network could become incredibly hard as more hosts get added to the network due to the fact that it is not a centrally managed network and each host has its own set of users and security policies, for example.

According to (Zacker, 2006, pp. 340-341), the number of PCs connected to a peer to peer network should be limited to around 10-15 nodes or hosts. Others recommend that peer to peer networks are used for connecting just two hosts together. Compared to a centralised client/server network, a P2P network is decentralised as the “individual nodes organise themselves into a network without any centralised coordination” (Peterson & Davie, 2012, p. 770).

An example of a peer to peer network could be a network with 2 computers connected to each other via an Ethernet link. Another example of a peer to peer network could be the typical home network with 3 or 4 PCs and a router.

Client/server based networks are more suited to larger networks with hundreds or thousands of computers. Managing a large amount (20-25+) of computers within a P2P network would take a long time. With a Client/server network, the client computers can be centrally managed meaning that user and software management could be carried out from the server to allow software updates/patches to be distributed to client computers. One of the major disadvantages with client/server networks is that the entire network could go down due to a server fault (if replication/load balancing has not been configured on two or more servers) compared with P2P networks which can generally stay operational if one node or host goes down (Zacker, 2006, pp. 340-341).

An example of a client/server network could be a company with 1 server and 100 PCs, with the server being the central point for things like DHCP/DNS, user management, software management and internet access via a content filter/proxy server.

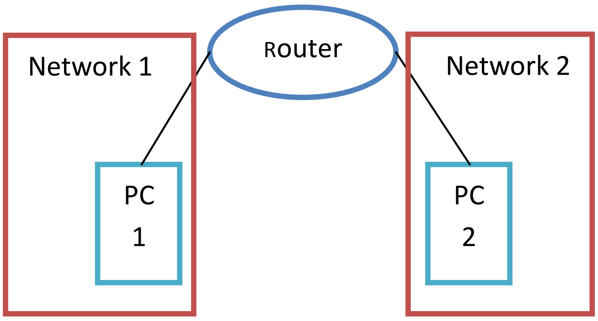
4.1 The Network Specifications

Based on the usage for the networks (implementing a small network in several network simulators), the networks will be of a small size.

There will be 4 different networks that will be created in each program. The first will consist of 2 computers and a router linked via an Ethernet cable. The second will consist of 4 computers, a switch and a router, again linked via Ethernet cable. The third network will consist of 4 computers, 1 server and 1 switch to link all the devices together (with the server providing important services such as DHCP to clients). The final network will be similar to the third network, but with an added wireless access point for connecting wireless devices to the network.

4.1.1 Network 1

The first network will look similar to the following:



The above network will have the following IP addresses and subnet masks set manually as follows:

Network 1:

- PC 1 Will have the IP address of 192.168.100.101 and gateway of 192.168.100.1
- The router will have the IP address of 192.168.100.1
- The subnet mask will be 255.255.255.0 for all hosts on network 1

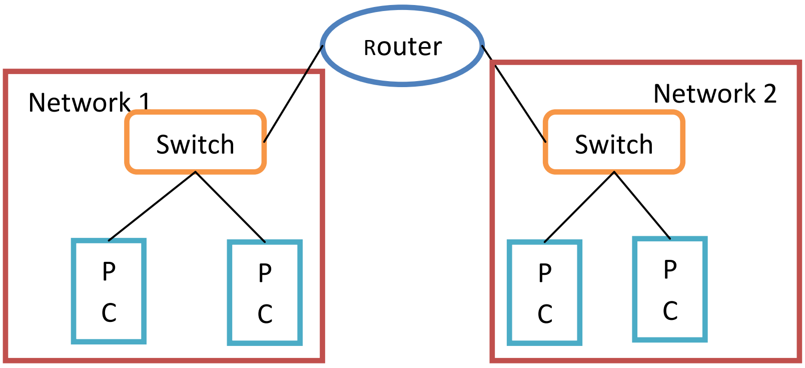
Network 2:

- PC 2 Will have the IP address of 192.168.200.200 and gateway of 192.168.200.254
- The router will have the IP address of 192.168.200.254
- The subnet mask will be 255.255.255.0 for all hosts on network 2

The router within this network should forward packets between each local area network and allow each host to communicate with hosts on the other network.

4.1.2 Network 2

The second network will be similar to the first, but will introduce two more hosts and two switches to the network. The second network will look like the following diagram:



The second network will be configured as follows:

Network 1:

- PC 1 Will have the IP address of 192.168.1.101
- PC 2 Will have the IP address of 192.168.1.102
- The router will have the IP address of 192.168.1.1
- The switch will have the IP address of 192.168.1.2
- The subnet mask will be 255.255.255.0 for all hosts on network 1

Network 2:

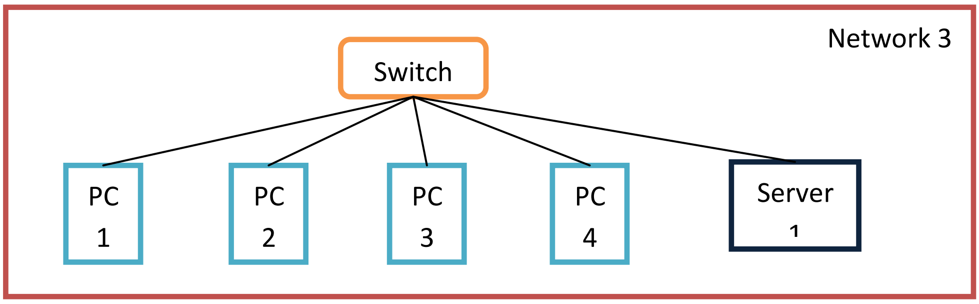
- PC 3 Will have the IP address of 192.168.100.10
- PC 4 Will have the IP address of 192.168.100.11
- The router will have the IP address of 192.168.100.1
- The switch will have the IP address of 192.168.100.2
- The subnet mask will be 255.255.255.0 for all hosts on network 2

Again all addresses will be set manually due to the small size of the network. All devices on the two networks should be able to communicate with each other. The network will be a peer to peer network rather than a client/server network.

The third network will be a simple switched network with 5 hosts and a network switch. There will be 4 clients and one server, which will handle services such as HTTP (Hypertext Transfer Protocol), DHCP (Dynamic Host Configuration Protocol) and DNS (Domain Name System).

4.1.3 Network 3

The third network will look similar to the following:



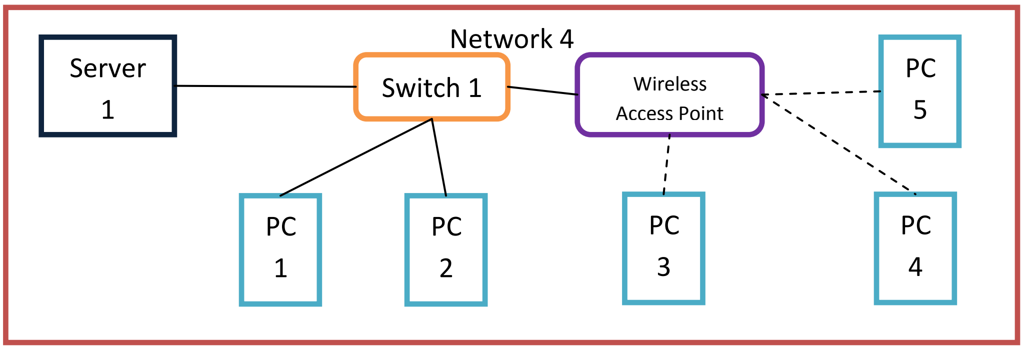
The above network will be configured as follows:

- Server 1 has a static IP address of 192.168.100.1 and a subnet mask of 255.255.255.0
- Switch 1 has a static IP address of 192.168.100.254 and a subnet mask of 255.255.255.0
- All four clients obtain their IP address/subnet mask via DHCP (provided by the server)

All hosts should be able to communicate with each other (as DHCP will be in use on this network) and the network will be based on the client/server model rather than being a peer to peer network. As well as hosting DHCP, the server will also provide DNS and HTTP services to clients.

4.1.4 Network 4

Where possible, there will be a fourth network which will introduce wireless networking. The network will be similar to the following:



The fourth network will be configured as follows:

- All PCs will have a dynamically assigned IP/Subnet mask
- The server will have a static IP address of 192.168.0.1
- The subnet mask will be 255.255.255.0 for all hosts on network 2
- The wireless network will be secured with WPA2-PSK encryption and a password

This network will also be a client/server network and should allow all computers to communicate with each other over the different types of links in this network.

4.1.5 Network Security

Security is a big issue within networks (as well as most, if not all, other areas of computing). If a network is not properly secured, then data and users could be put at risk of attack from viruses, malware and hacking attempts to gain access to networks/computers. The UK has the Computer Misuse Act (CMA) (Legislation.gov.uk, 2012) which allows for people to be prosecuted if they are caught hacking into computers of UK companies/agencies/citizens. Other countries, such as the USA, have different laws (with the US law being called the 'Computer Fraud and Abuse Act' (Cornell University Law School, 2012)).

Some of the methods to secure the networks mentioned in the previous section will be to:

- Use authentication methods, such as strong passwords for user authentication
- Shut down unused ports on routers and switches
- Disable unused services on routers and switches
- Using access control lists (ACLs)

In the guide about hardening the security of Cisco routers (Cisco Systems, 2011), various security methods are discussed (such as physical access to equipment) that aren't applicable to networks built in network simulators, however, are very important to address in real world scenarios to prevent breaches of network security (which could have further problems for home users and businesses).

4.1.6 Types of threats to networks

There are many ways that performance and security of a network could be affected and/or compromised by an attacker. Here are just a few of the many ways an attacker could degrade the performance or compromise the security features of the network:

- DoS (Denial of Service) and DDoS (Distributed Denial of Service) attacks:

  These are one of the most common types of attacks against computers and networks, which are designed to overload a specific computer or network. A DDoS attack works by getting a group of computers together (or a bot net of hijacked computers) to constantly ping a website until the server's performance is degraded to the point where it cannot handle the amount of requests it receives per second and effectively shuts the website down (McDowell, 2004).

  A recent case of DDoS attacks on websites (at the time of writing) was the retaliation of the 'Anonymous' hacking group against the US Motion Picture Association of America, FBI and Department of Justice websites for taking down the megaupload.com website (Kelion, 2012).

- DNS (Domain Name System) spoofing

  DNS is a system that translates a domain name (e.g. www.herts.ac.uk) into an IP address (e.g. 147.197.200.94 is the IP address for www.herts.ac.uk) that computers use to connect to a website. DNS spoofing is where a rogue entry has been added to a DNS server to point to another IP address hosting a fake website. A rogue DNS server could also be set up by a malicious user with rogue entries to popular websites such as Google and Facebook to try and harvest user credentials to use for other purposes (Stanford University, 2001) (Sanders, 2010).

- ARP (Address Resolution Protocol) Spoofing/Poisoning

  ARP spoofing can be used to perform MITM (Man in the Middle) attacks, which allow for an attacker to intercept and monitor communications between hosts. This type of attack is bad for networks, as all kinds of data could be gathered by the attacker such as passwords and configuration information.

  ARP spoofing can be used in conjunction with DNS spoofing to perform MITM attacks on users (that think they are using a genuine site such as Twitter) to redirect them to a fake web page (to harvest users credentials for example) (Sanders, 2010) (King & Lauerman, 2010).

  - Physical attacks

  Gaining unauthorised access to physical networking could also put the security of a network at risk in a big way. When an attacker gains physical access to a computer or network device, security features can be bypassed by booting a computer from removable media such as a CD or USB flash drive for example. If an attacker gained unauthorised access to equipment, they could steal the equipment or vandalise it if the attacker's aim was to do some serious damage to a network (Convery, 2004).

  An example of a case where physical security was breached is the case where the business arm of Verizon had equipment stolen from a datacentre in London in 2007. Thieves, dressed up as policemen, claimed that there were reports of people on the roof and managed to gain access to the data centre before stealing computing equipment (Espiner, 2007).

4.1.7 Protecting against network threats

To protect against the threats listed in the previous section (Chapter 4.1.6), this section outlines some of the ways the attacks listed above can be stopped or reduced to limit the impact on the network.

According to (IWS, 2000), to reduce the likelihood of a DOS/DDOS attack from taking out a network, a network administrator should:

- Block packets with spoofed IP addresses
- Only allow incoming traffic on certain ports (e.g. port 80 for HTTP) and use the firewall/packet filtering to block any other incoming traffic

To reduce the risk of DNS record spoofing, a network administrator could:

- Restrict zone transfers to prevent leakage of DNS records
- Turn on any DNS cache pollution protection systems on the DNS server (such as the one in Microsoft DNS servers (Microsoft Help & Support, 2007))
- Limit the amount of sources that can do dynamic updates to the DNS server (NIST, 1998)

To reduce the risk of ARP Spoofing the following items can be implemented:

- Use an intrusion detection system (IDS) to protect against these attacks
- Use a static ARP cache (if the network is small) (Fewer, 2007)

To reduce the likelihood of an attack on the physical hardware:

- Access control to rooms containing the equipment could be implemented
- Bolting the equipment to a server cabinet, and keeping the cabinet locked would make it harder for an unauthorised user to access the equipment

5.0 Implementing the networks in Cisco Packet Tracer

This section will look at the steps to implement each of the networks detailed in the specifications (Chapter 4.1) in the simulation software listed in Chapter 2.

5.1 Cisco Packet Tracer

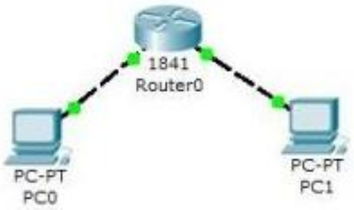
5.1.1 Network 1

Using the logical view in Cisco Packet Tracer allows for network devices to be placed into a new network.

Two computers and a router were placed onto the logical network view in packet tracer. The two computers are standard computers as emulated by packet tracer. The router used was a Cisco 1841 router.

The router and the two computers were then linked to each other using a standard crossover cable (‘Copper crossover’ as per the packet tracer definition). PC0 was connected to port 0 of the router (the ‘FastEthernet0/0’ interface) and PC1 was connected to the routers FastEthernet0/1 interface.

The finished network can be found in the screenshot below:

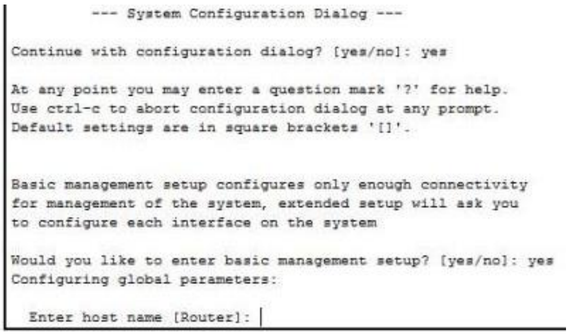


The topology for network 1

Device Configuration

Router

The next step was to configure the router using the command line interface. As this was the first time the router had been switched on, it showed the system configuration dialogue as shown in the screenshot below:



```
          --- System Configuration Dialog ---

Continue with configuration dialog? [yes/no]: yes

At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.


Basic management setup configures only enough connectivity
for management of the system, extended setup will ask you
to configure each interface on the system

Would you like to enter basic management setup? [yes/no]: yes
Configuring global parameters:

  Enter host name [Router]:
```

The System Configuration Dialogue from the router

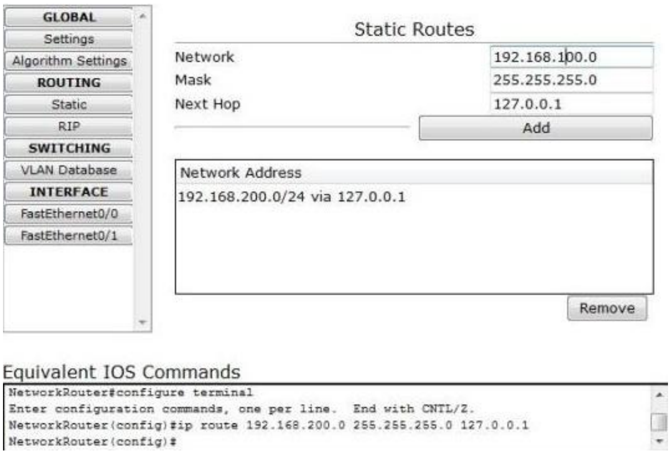
The system configuration dialogue asked if the basic management should be run to set up the router. This process asked for the hostname, passwords for the enable secret and enable password, a password for the virtual terminal mode, whether SMNP should be set up and which interface should be configured.

Once the system configuration dialogue had been completed, the settings were saved on the device and the second interface configured manually.

A full log of the router setup process (including passwords and configuration information) can be found in Appendix 1.1.

Configuration using the graphical user interface

In addition to the command line interface (CLI), Packet Tracer also features a graphical user interface (GUI) to help set up a router (or any other supported equipment in packet tracer). Although the GUI is limited when compared with the CLI, it is useful as for every configuration change a user makes in the GUI section it shows the command that would be used on real Cisco routers under ‘Equivalent IOS commands’.



Network 192.168.100.0
Mask 255.255.255.0
Next Hop 127.0.0.1

Network Address
192.168.200.0/24 via 127.0.0.1

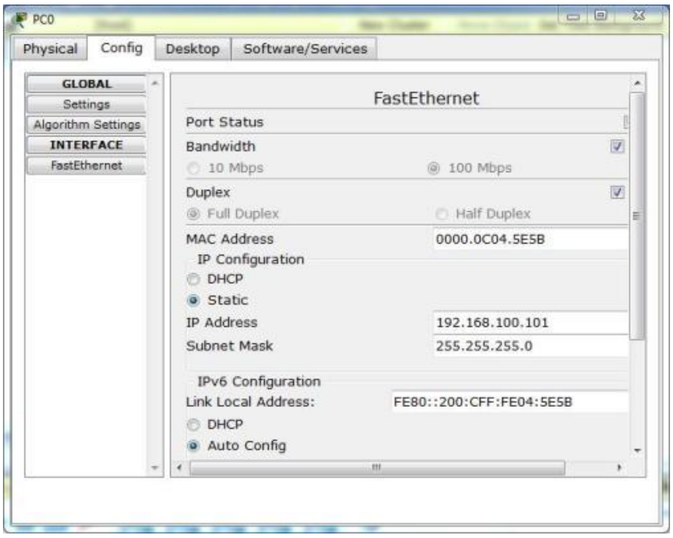
Equivalent IOS Commands

```
NetworkRouter#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
NetworkRouter(config)#ip route 192.168.200.0 255.255.255.0 127.0.0.1
NetworkRouter(config)#
```

The static routes GUI configuration screen

Configuring hosts, servers and other networking devices can only be done via a graphical user interface as packet tracer does not simulate a full PC running Windows or Linux (using a virtualization product such as VMware Player or Oracle Virtualbox).

The next two screenshots show the different ways of configuring IP addresses on hosts in packet tracer:



PC0

Physical | Config | Desktop | Software/Services

GLOBAL
Settings
Algorithm Settings
INTERFACE
FastEthernet

FastEthernet

Port Status

Bandwidth ☑

○ 10 Mbps ● 100 Mbps

Duplex ☑

● Full Duplex ○ Half Duplex

MAC Address 0000.0C04.5E5B

IP Configuration

○ DHCP

● Static

IP Address 192.168.100.101

Subnet Mask 255.255.255.0

IPv6 Configuration

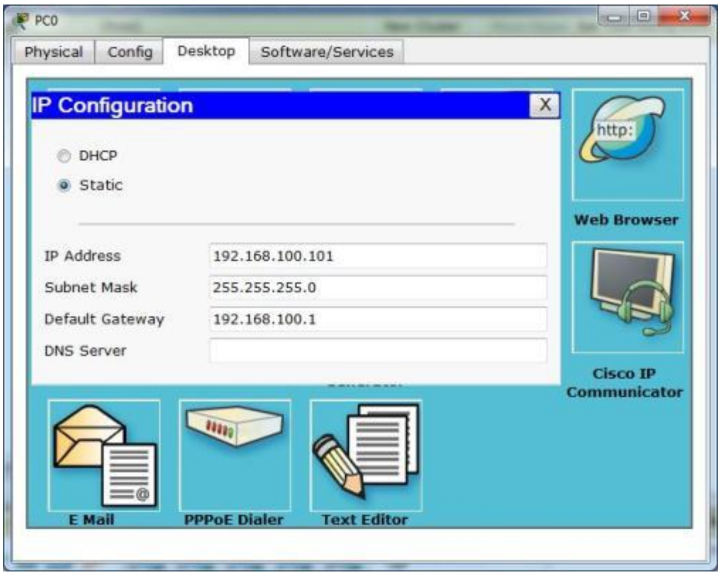
Link Local Address: FE80::200:CFF:FE04:5E5B

○ DHCP

● Auto Config

Using the GUI to configure settings

To configure the networking settings of a host (in this case, the IP address, Default Gateway and Subnet Mask) the user could use the ‘Config’ tab which allows for more detailed settings compared to the IP configuration window under the ‘Desktop’ tab (Screenshot below).



PC0

Physical | Config | Desktop | Software/Services

IP Configuration

○ DHCP

● Static

IP Address 192.168.100.101

Subnet Mask 255.255.255.0

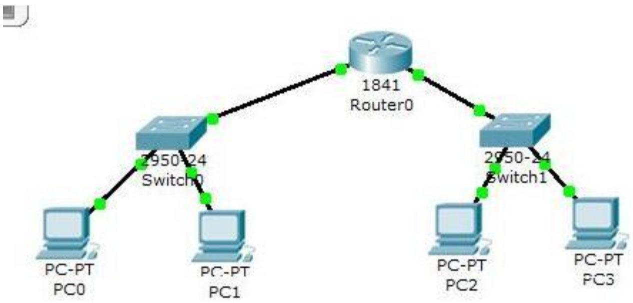
Default Gateway 192.168.100.1

DNS Server

Using the ‘Desktop’ tab to configure IP addresses

5.1.2 Network 2

This network consists of 1 Cisco 1841 router, 2 Cisco 2950 switches and 4 desktop PCs, all linked together by patch cables (‘copper straight through’ in packet tracer).



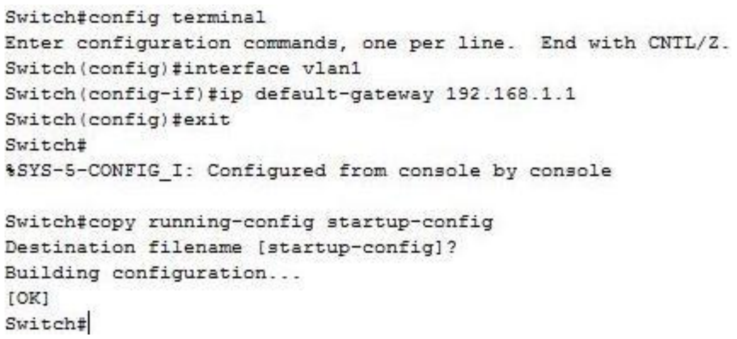
The layout of network 2

Device Configuration

Router and Switches

The router was the first network device to be configured for this network. The command line interface (CLI) was used to set up the router by using the system configuration dialogue (SCD) to set up general settings and one interface. Once the SCD had been completed, the configuration was saved, and then the second interface was configured manually with an IP address/subnet mask.

The two switches were then configured, again using the CLI, to set the IP address and change default passwords to increase security. The switches also had the default gateway set (something which isn’t available in the limited GUI interface) for VLAN1 to allow for remote management of the switches (Davis, 2008).

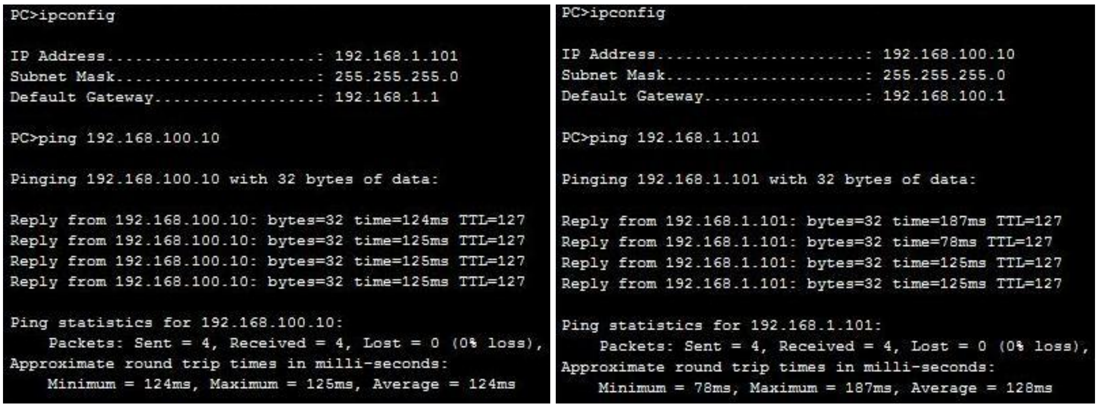


```
Switch#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#interface vlan1
Switch(config-if)#ip default-gateway 192.168.1.1
Switch(config)#exit
Switch#
%SYS-5-CONFIG_I: Configured from console by console

Switch#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Switch#
```

Setting up the default gateway for switch0 using the CLI.

Nodes and computers

Nodes that connected to the network were then configured with the settings as per the specification for network 2. Each computer received a static IP address within the range for the network they are connected to (192.168.1.x for network 1 and 192.168.100.x for network 2) to allow communication with other devices and networks.



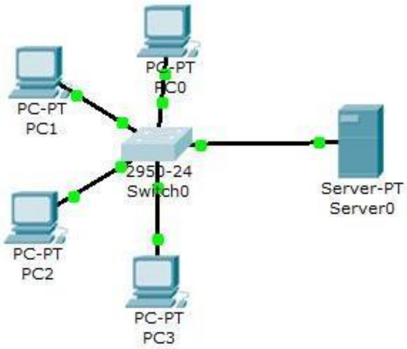
```
PC>ipconfig

IP Address......................: 192.168.1.101
Subnet Mask.....................: 255.255.255.0
Default Gateway.................: 192.168.1.1

PC>ping 192.168.100.10

Pinging 192.168.100.10 with 32 bytes of data:

Reply from 192.168.100.10: bytes=32 time=124ms TTL=127
Reply from 192.168.100.10: bytes=32 time=125ms TTL=127
Reply from 192.168.100.10: bytes=32 time=125ms TTL=127
Reply from 192.168.100.10: bytes=32 time=125ms TTL=127

Ping statistics for 192.168.100.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 124ms, Maximum = 125ms, Average = 124ms
```

```
PC>ipconfig

IP Address......................: 192.168.100.10
Subnet Mask.....................: 255.255.255.0
Default Gateway.................: 192.168.100.1

PC>ping 192.168.1.101

Pinging 192.168.1.101 with 32 bytes of data:

Reply from 192.168.1.101: bytes=32 time=187ms TTL=127
Reply from 192.168.1.101: bytes=32 time=78ms TTL=127
Reply from 192.168.1.101: bytes=32 time=125ms TTL=127
Reply from 192.168.1.101: bytes=32 time=125ms TTL=127

Ping statistics for 192.168.1.101:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 78ms, Maximum = 187ms, Average = 128ms
```

Testing connectivity between LAN 1 and LAN 2 from host 1 (Left screenshot) and testing connectivity between LAN2 and LAN1 from Host 4 (Right Screenshot)

5.1.3 Network 3

This network consists of 1 server, a four hosts and a switch. As there is a server connected to this network the server was configured to host DHCP (Dynamic Host Configuration Protocol), DNS (Domain name system) and a HTTP (Hypertext Transfer Protocol) web server for clients connected to the network to use. As the server provides automatic IP address assignment via DHCP, the hosts all had their networking settings set to dynamically obtain addresses.



The topology for network 3

Device Configuration

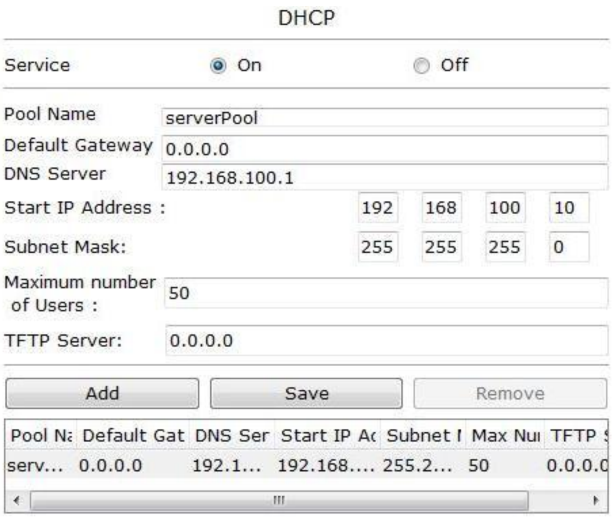
Network Switch

Again, the switch was configured using the more powerful CLI rather than the GUI. The IP address of the switch was configured by entering the '`ip addr 192.168.100.254 255.255.255.0`' command, giving VLAN1 (Virtual Local Area Network) an IP address of 192.168.100.254 and subnet mask of 255.255.255.0. The switch was also configured with a hostname of 'Network3Switch' using the GUI. After the switch had been configured, the settings were saved using the '`copy running-config startup-config`' command so that settings remained after reboots. The steps taken to configure the switches can be found at Appendix 1.3: Network 3 & Network 4 Switch configuration.

Server

The server was the next device to be configured, starting with the IP address. As there is a very limited command line interface, the GUI was used to configure the server. Configuring the DHCP server was fairly simple as it requires:

- An IP address to start leases from (e.g. start the IP address pool from 192.168.100.50)
- The amount of leases available (e.g. only let up to 20 devices to have a dynamic IP at a time)
- The subnet mask (255.255.255.0 for this network)
- A DNS server address (to let clients resolve hostnames)
- A Default gateway (0.0.0.0 in this case as there are no routers)
- A TFTP server for PXE booting (Also 0.0.0.0 as there is no need for TFTP in this LAN)
- A Pool name (e.g. DHCP Addresses for LAN 4 or serverPool)


DHCP

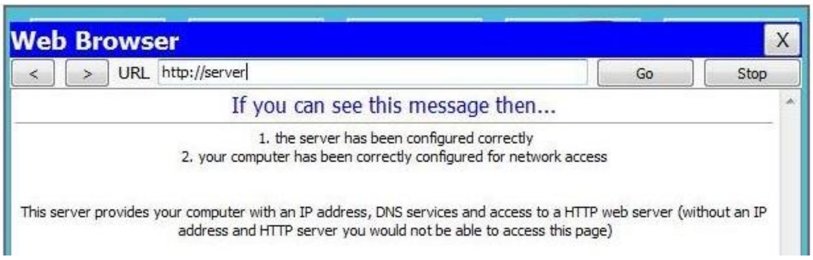
Service ● On ○ Off

Pool Name: serverPool
Default Gateway: 0.0.0.0
DNS Server: 192.168.100.1
Start IP Address : 192 168 100 10
Subnet Mask: 255 255 255 0
Maximum number of Users : 50
TFTP Server: 0.0.0.0

Pool Na	Default Gat	DNS Ser	Start IP Ac	Subnet I	Max Nu	TFTP S
serv...	0.0.0.0	192.1...	192.168....	255.2...	50	0.0.0.0

The servers DHCP configuration as shown in the GUI

The HTTP server was the next item to be configured. A simple HTML page was created to allow clients to see that the HTTP server was working and that the networking components were working fine. The example page can be found in a screenshot below as it is seen by a client.


Web Browser

URL http://server

If you can see this message then...

1. the server has been configured correctly
2. your computer has been correctly configured for network access

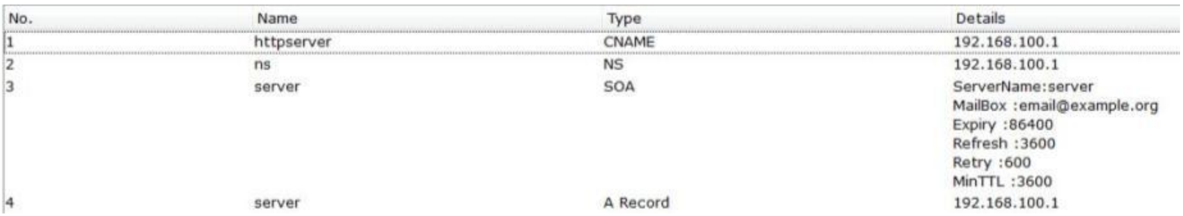
This server provides your computer with an IP address, DNS services and access to a HTTP web server (without an IP address and HTTP server you would not be able to access this page)

Testing to ensure that the HTTP server (and DNS server – note the URL) work

Finally, the DNS server was configured with:

- A Host (A) record for the server
    - To allow clients to access the server via the hostname rather than the servers IP address
- A CNAME alias for the server
    - To provide an additional name/subdomain to access the server by (in this case, it is httpserver)
- A Start of Authority record
    - To provide details about the DNS zone (email address to contact domain administrator, what the primary nameserver for the zone is, how secondary nameservers get updated and a time to live (TTL) value). For more detailed information on SOA records, see (Perry, 2011).

For more information on all DNS record types (including MX mail server records), see (Google, 2012) and (Microsoft Help and Support, 2007).



No.	Name	Type	Details
1	httpserver	CNAME	192.168.100.1
2	ns	NS	192.168.100.1
3	server	SOA	ServerName:server MailBox :email@example.org Expiry :86400 Refresh :3600 Retry :600 MinTTL :3600
4	server	A Record	192.168.100.1

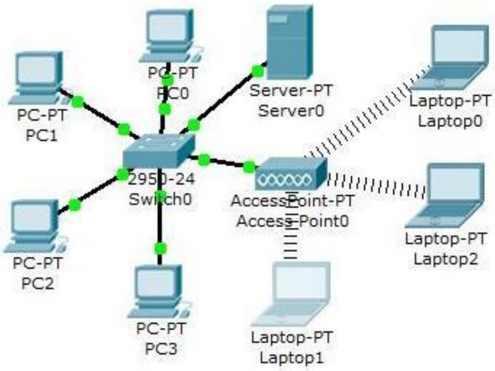
The DNS server configuration as shown in packet tracer

Client PCs

Once the server had been configured, the clients had all been configured to obtain addresses via DHCP. Once the IP address had been allocated, then the clients were used to test out the HTTP and DNS servers to ensure they were reachable from a remote host.

5.1.4 Network 4

The final network is essentially the same as network 3, but with an added wireless link for clients.



The topology for network 4

Device Configuration

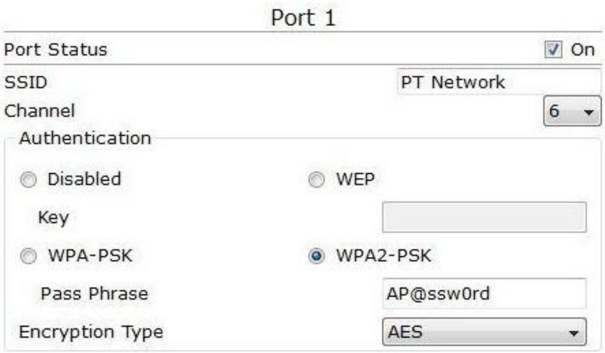
Servers, Clients and Switches

The servers, switches and wired clients were configured in the same way as network 3. All clients on network 4 were allocated a dynamic IP address by the server to reduce the likelihood of an IP address conflict between permanent devices (wired clients) and portable devices (wireless clients) that may be configured for other wireless networks.

A wireless access point was added to the network to allow authorised wireless devices to gain access.

Wireless Network

The wireless access point was configured so that it would broadcast on wireless channel 6, with the SSID (Service Set Identifier) of ‘PT Network’. The wireless network was secured using WPA2-PSK (Wireless Protected Access 2 Pre shared Key) and a passcode of ‘AP@ssw0rd’ so that unauthorised clients would not be able to access the network.



Port 1

Port Status	☑ On
SSID	PT Network
Channel	6

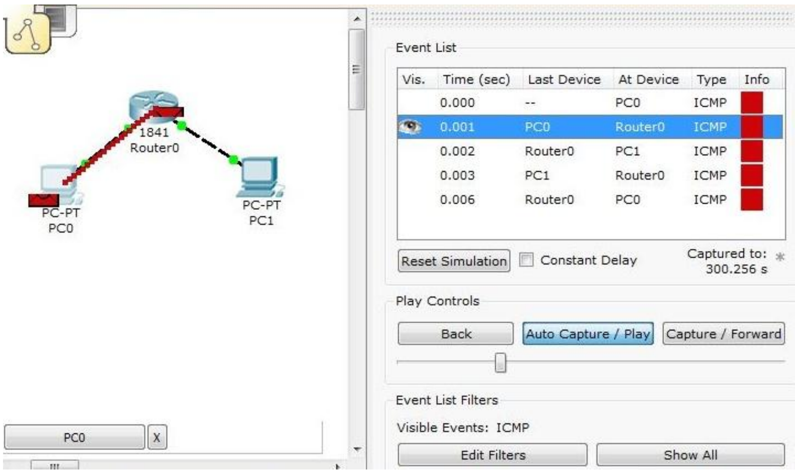
Authentication

○ Disabled	○ WEP
Key	
○ WPA-PSK	● WPA2-PSK
Pass Phrase	AP@ssw0rd
Encryption Type	AES

The wireless access point settings

5.1.5 Other functionalities of Packet Tracer

Packet tracer has a simulation mode that allows a user to see how a packet gets transmitted across a network along with seeing a diagram of the frame that was sent onto the network. In the screenshots below, a ping command was run to generate traffic between the two hosts PC0 and PC1.



Event List

Vis.	Time (sec)	Last Device	At Device	Type	Info
	0.000	--	PC0	ICMP	🟥
👁	0.001	PC0	Router0	ICMP	🟥
	0.002	Router0	PC1	ICMP	🟥
	0.003	PC1	Router0	ICMP	🟥
	0.006	Router0	PC0	ICMP	🟥

□ Constant Delay

Captured to: 300.256 s

Play Controls

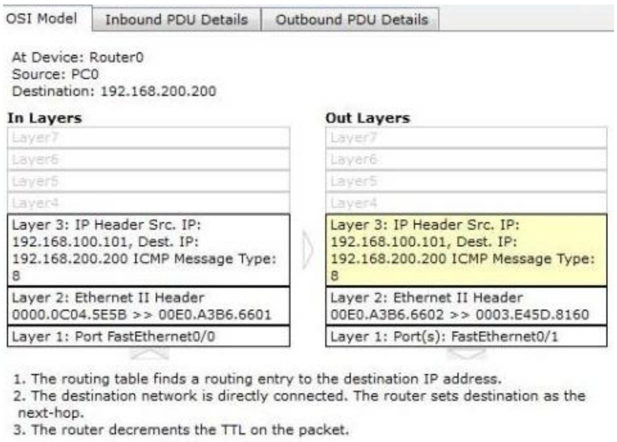
Event List Filters

Visible Events: ICMP

The simulation mode showing the process of sending an ICMP packet over the network

The red envelope (packet) is shown in the screenshot whilst being transferred from PC0 to Router0. Once Router 0 had processed the packet, the packet would then get sent onto PC1 (who would then reply and send a message (an ACK packet) back to PC0).

The test above had filters that were configured to only show ICMP Ping requests between the two hosts. By clicking on one of the events in the event list; the user would then see a screen similar to the screenshot below:

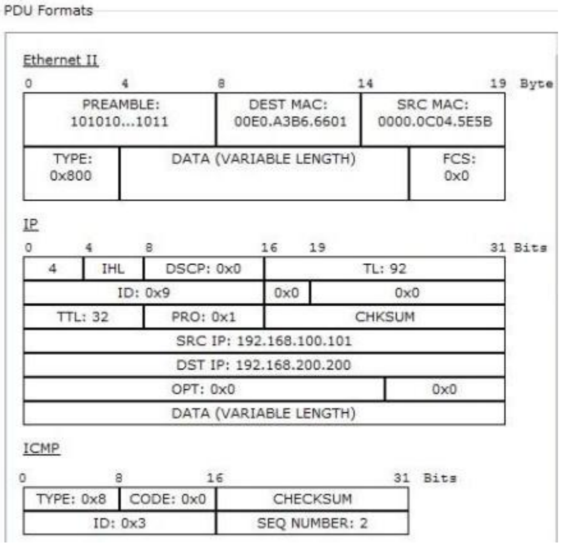


At Device: Router0
Source: PC0
Destination: 192.168.200.200

In Layers	Out Layers
Layer7	Layer7
Layer6	Layer6
Layer5	Layer5
Layer4	Layer4
Layer 3: IP Header Src. IP: 192.168.100.101, Dest. IP: 192.168.200.200 ICMP Message Type: 8	Layer 3: IP Header Src. IP: 192.168.100.101, Dest. IP: 192.168.200.200 ICMP Message Type: 8
Layer 2: Ethernet II Header 0000.0C04.5E5B >> 00E0.A3B6.6601	Layer 2: Ethernet II Header 00E0.A3B6.6602 >> 0003.E45D.8160
Layer 1: Port FastEthernet0/0	Layer 1: Port(s): FastEthernet0/1

1. The routing table finds a routing entry to the destination IP address.
2. The destination network is directly connected. The router sets destination as the next-hop.
3. The router decrements the TTL on the packet.

The window (in the above screenshot) will appear when a user clicks on one of the events from the event list. It shows the OSI (Open Systems Interconnection) model, along with what happens at each layer of the model whist the packet is being processed by the router. If a user clicks on the ‘Inbound PDU’ (Protocol Data Units) tab they will be presented with a screen that shows what the packet would look like in several different frame formats.



PDU Formats

Ethernet II

0 4 8 14 19 Byte

PREAMBLE: 101010...1011 | DEST MAC: 00E0.A3B6.6601 | SRC MAC: 0000.0C04.5E5B

TYPE: 0x800 | DATA (VARIABLE LENGTH) | FCS: 0x0

IP

0 4 8 16 19 31 Bits

4 | IHL | DSCP: 0x0 | TL: 92

ID: 0x9 | 0x0 | 0x0

TTL: 32 | PRO: 0x1 | CHKSUM

SRC IP: 192.168.100.101

DST IP: 192.168.200.200

OPT: 0x0 | 0x0

DATA (VARIABLE LENGTH)

ICMP

0 8 16 31 Bits

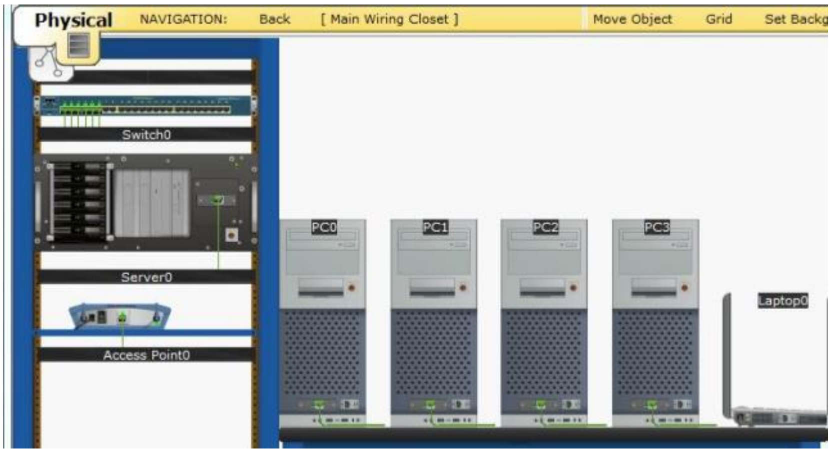
TYPE: 0x8 | CODE: 0x0 | CHECKSUM

ID: 0x3 | SEQ NUMBER: 2

The screenshot above shows how the details of the packet as they would appear in an IP and Ethernet datagram.

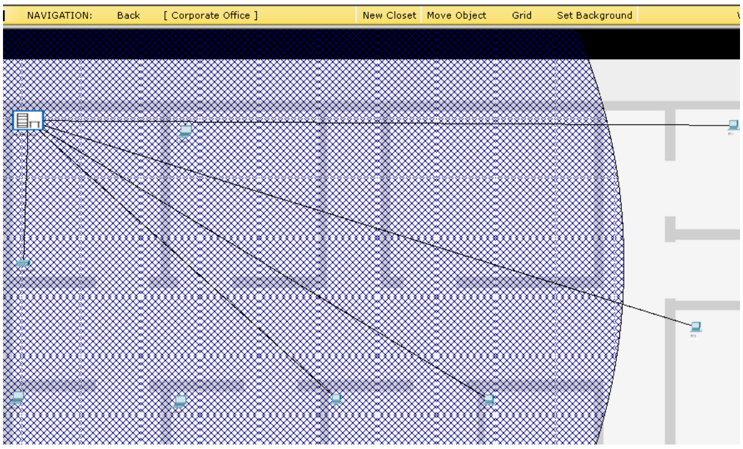
Packet Tracer allows users to plan out the physical layout of a network that has been created in packet tracer's logical view. The physical view allows for various zones to be created (these are cities, buildings, offices and wiring closets (server rooms)) to help plan out where equipment should be located. The network shown in the screenshots will be network 4 from the network specifications.

Initially, packet tracer will put all equipment into the server room (or wiring closet as packet tracer calls it), leaving the user to sort the equipment into the appropriate places.



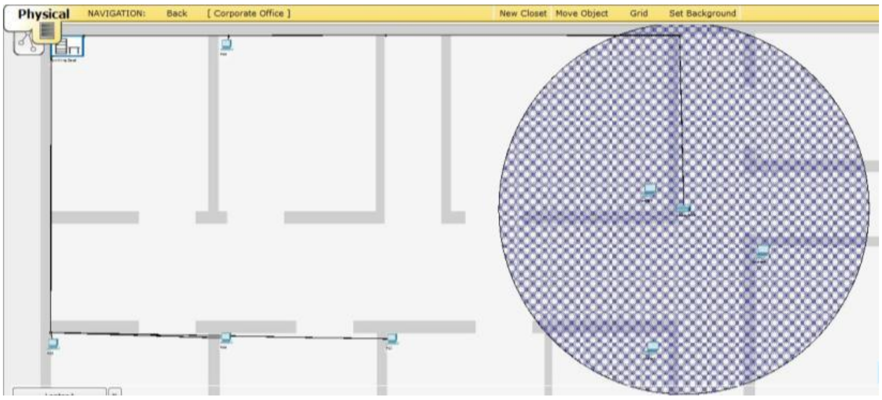
The unsorted wiring closet in packet tracer

As an example, the computers of network 4 will be scattered around the corporate office and the networking equipment (the routers, wireless access points and the server) will be stored in the main wiring closet which is located in the top left corner of the corporate office.



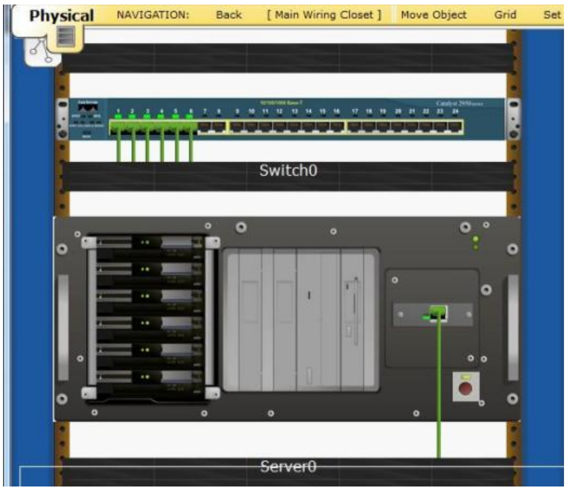
The initial physical view of the office (the shaded circle is the area covered by the wireless access point)

Packet tracer allows users to perform cable management tasks as well. The screenshot above shows unrealistic cable layouts for a building. After performing cable management in packet tracer, the results look more realistic compared with real life cable runs.



The office after performing cable management tasks (shaded area is wireless coverage)

Clicking on the main wiring closet will present the user with a physical view of the wiring closet showing any servers and networking equipment located there:



The wiring closet after moving end user devices out to the office

5.1.6 Troubleshooting networks with Packet Tracer

Packet Tracer would be a good candidate for troubleshooting basic network issues as it provides an easy way to simulate a network. Simple issues such as shut down ports (on switches/routers) and wireless issues (such as incorrect encryption keys) would be able to be simulated by Packet Tracer, but more complex issues such as node failure may be harder to simulate.

6.0 Implementing the networks in Graphical Network Simulator 3 (GNS3)

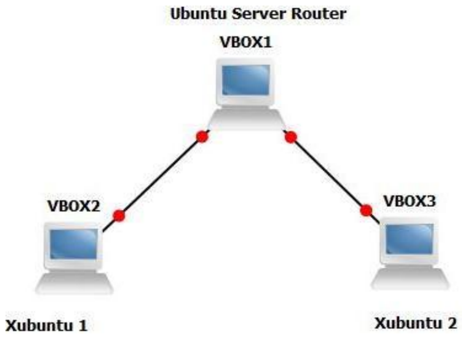
This section will look at the steps to implement each of the networks detailed in the specifications (Chapter 4.1) in the simulation software listed in Chapter 2.

6.1 Graphical Network Simulator 3 (GNS3)

This section will be looking at how each of the networks listed in the specification can be implemented within GNS3. Due to the fact that GNS3 is open source software, it requires the user to provide their own Cisco IOS (Internet Operating System) image or Juniper JunOS image to enable Cisco and Juniper networking equipment to be used within GNS3. To overcome this limitation, a Ubuntu Server 10.04.3 installation was set up in a virtual machine to act as a router.

6.1.1 Network 1

Adding devices to a network in GNS3 is as simple as dragging the device from the ‘Nodes Types’ window to the network topology pane (similar to Cisco Packet Tracer). Like Packet Tracer, GNS3 offers multiple types of connections to allow devices to be connected to a network. The types of connections are limited to what the hosts support (e.g. Ethernet and Serial links).

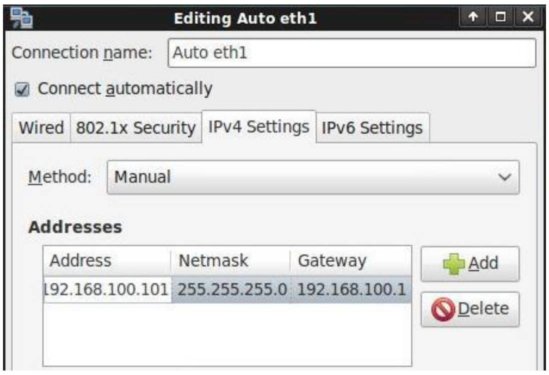


The network topology diagram for network 1

Device Configuration

Hosts

All hosts are virtual machines created within Oracle Virtualbox. All the machines run Ubuntu Linux (Ubuntu server for the router, Xubuntu for the clients) and were installed with the settings detailed in the network specification. The installation options for Ubuntu/Xubuntu can be found in Appendix 1.4.



Editing Auto eth1

Connection name: Auto eth1

☑ Connect automatically

Wired | 802.1x Security | IPv4 Settings | IPv6 Settings

Method: Manual

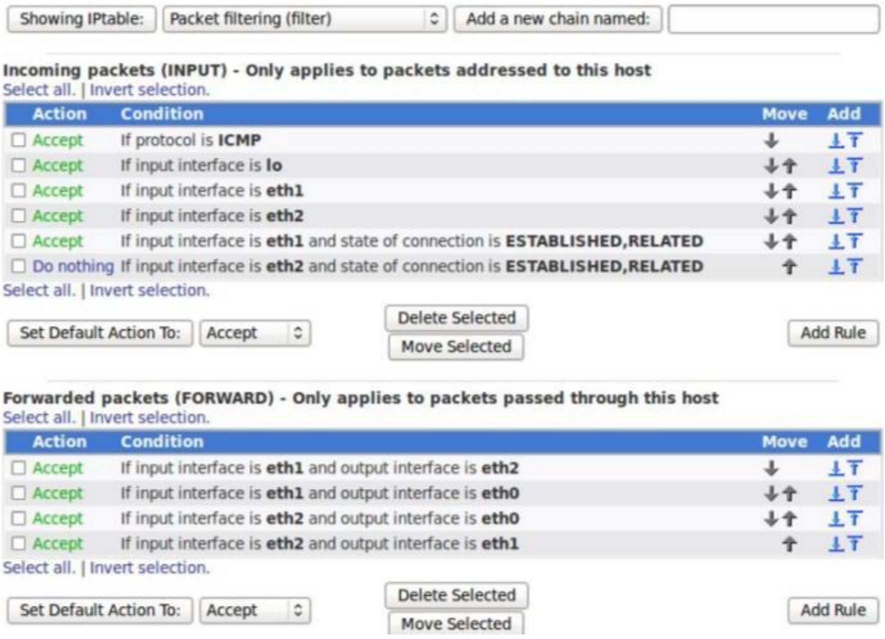
Addresses

Address	Netmask	Gateway
l92.168.100.101	255.255.255.0	192.168.100.1

Setting a static IP address in Xubuntu

Router

Each network interface was given a static IP address (as per the specification), and some additional software (Webmin (Webmin, 2011)) was used to configure the Ubuntu server as a router for use within this network. The IP addresses for eth1 and eth2 were set manually in the network interface configuration file (`/etc/network/interfaces`), along with enabling packet forwarding. The IPTables firewall was configured using Webmin to allow clients on the two networks to communicate. A guide was used to help set up the IPTables rules (UbuntuForums, 2008).



Showing IPtable: Packet filtering (filter)

Incoming packets (INPUT) - Only applies to packets addressed to this host

Select all. | Invert selection.

Action	Condition	Move	Add
☐ Accept	If protocol is **ICMP**	↓	⊥⊤
☐ Accept	If input interface is **lo**	↓↑	⊥⊤
☐ Accept	If input interface is **eth1**	↓↑	⊥⊤
☐ Accept	If input interface is **eth2**	↓↑	⊥⊤
☐ Accept	If input interface is **eth1** and state of connection is **ESTABLISHED,RELATED**	↓↑	⊥⊤
☐ Do nothing	If input interface is **eth2** and state of connection is **ESTABLISHED,RELATED**	↑	⊥⊤

Select all. | Invert selection.

Set Default Action To: Accept

Forwarded packets (FORWARD) - Only applies to packets passed through this host

Select all. | Invert selection.

Action	Condition	Move	Add
☐ Accept	If input interface is **eth1** and output interface is **eth2**	↓	⊥⊤
☐ Accept	If input interface is **eth1** and output interface is **eth0**	↓↑	⊥⊤
☐ Accept	If input interface is **eth2** and output interface is **eth0**	↓↑	⊥⊤
☐ Accept	If input interface is **eth2** and output interface is **eth1**	↑	⊥⊤

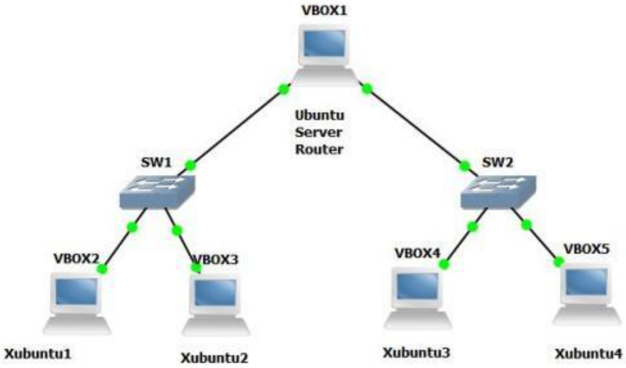
Select all. | Invert selection.

Set Default Action To: Accept

Configuring IPTables via Webmin

6.1.2 Network 2

Two additional Xubuntu hosts (Xubuntu 3 and Xubuntu 4) and 2 network switches have been added to the network:

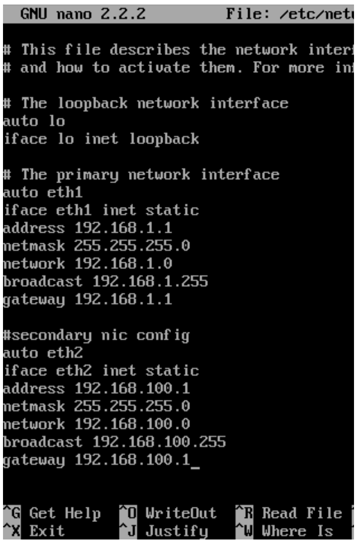


The topology layout for network 2

Device Configuration

Server

First, the servers network cards (NICs) were configured using the text file editor, nano, via the console running on the server. After configuring the IP addresses for the servers NICs, packet forwarding was turned on by altering the ‘`net.ipv4.ip_forward=`’ parameter from 0, to 1, in the file ‘`/etc/sysctl.conf`’.



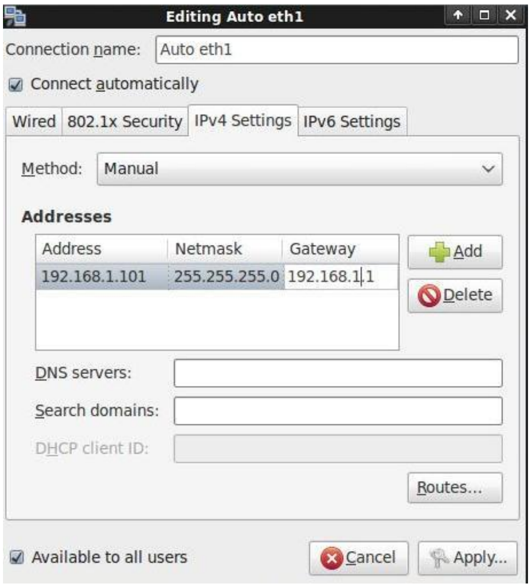
```
  GNU nano 2.2.2              File: /etc/netw

# This file describes the network interf
# and how to activate them. For more inf

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth1
iface eth1 inet static
address 192.168.1.1
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.1

#secondary nic config
auto eth2
iface eth2 inet static
address 192.168.100.1
netmask 255.255.255.0
network 192.168.100.0
broadcast 192.168.100.255
gateway 192.168.100.1_

^G Get Help  ^O WriteOut  ^R Read File
^X Exit      ^J Justify   ^W Where Is
```

Configuring the Static IP addresses for the router

Webmin was then used for configuring IPTables to allow traffic to pass through the Ubuntu Server onto the other networks.

Clients

The clients had a static IP address set, as per the specification, so that they could communicate with the server (and each other).



Connection name: Auto eth1

☑ Connect automatically

Method: Manual

Addresses

Address	Netmask	Gateway
192.168.1.101	255.255.255.0	192.168.1.1

DNS servers:

Search domains:

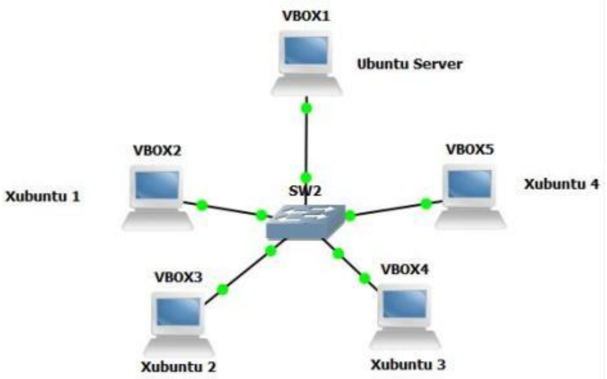
DHCP client ID:

☑ Available to all users

Configuring a Static IP address on a client

6.1.3 Network 3

This network is based around a client/server network, with the server providing infrastructure services such as DHCP and DNS.



The topology diagram for network 3

Device Configuration

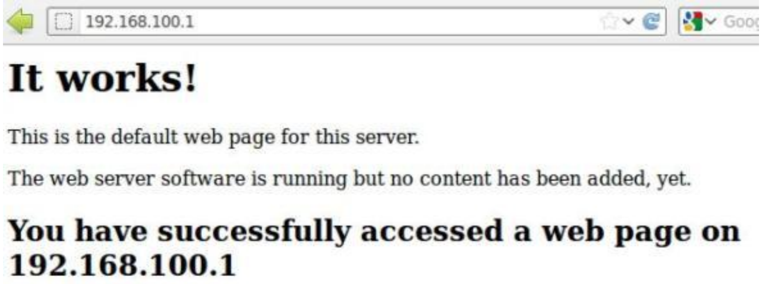
Clients

The clients were Xubuntu virtual machines (installed as per Appendix 1.4), each configured with the network interface to obtain an IP address via DHCP. Each machine had a different visual theme to aid identification of the different machines.

Server

The server was a standard Ubuntu Server virtual machine (installed as per Appendix 1.4).

The apache2 webserver was installed onto the server by using the '`sudo apt-get install apache2`' command. The default apache index.html was then modified to add a message to visitors to say that they had successfully managed to access the HTTP server.



192.168.100.1

It works!

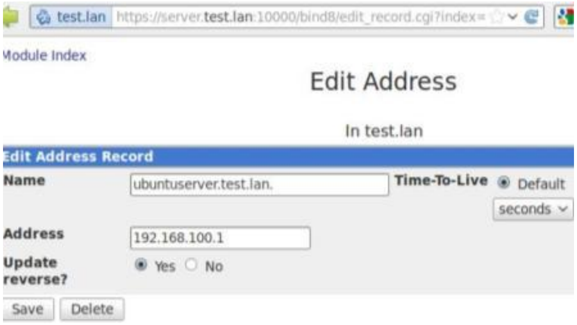
This is the default web page for this server.

The web server software is running but no content has been added, yet.

You have successfully accessed a web page on 192.168.100.1

The modified apache index.html

The DNS server (BIND9) was installed when Ubuntu server was initially installed. Webmin was then used to configure the DNS server with a new zone for the network 'test.lan' to hold forward (DNS Names to IP addresses) and reverse (IP address to DNS name) lookup zones.



test.lan https://server.test.lan:10000/bind8/edit_record.cgi?index=

Module Index

Edit Address

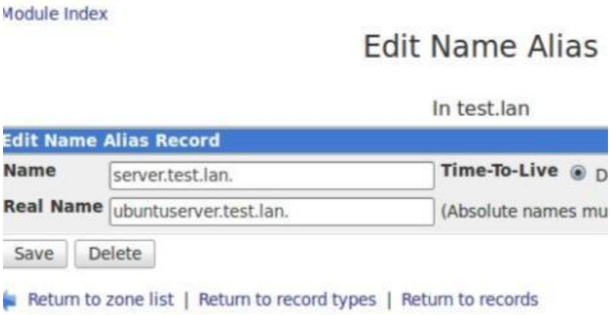
In test.lan

Edit Address Record

Name: ubuntuserver.test.lan.
Time-To-Live: ● Default seconds
Address: 192.168.100.1
Update reverse?: ● Yes ○ No

Save | Delete

Creating a DNS 'A' record for the server



Module Index

Edit Name Alias

In test.lan

Edit Name Alias Record

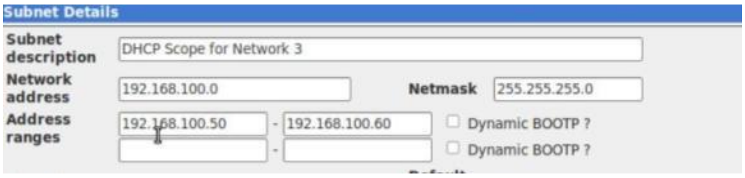
Name: server.test.lan.
Time-To-Live: ● D
Real Name: ubuntuserver.test.lan.
(Absolute names mu

Save | Delete

Return to zone list | Return to record types | Return to records

Creating an alias for the server

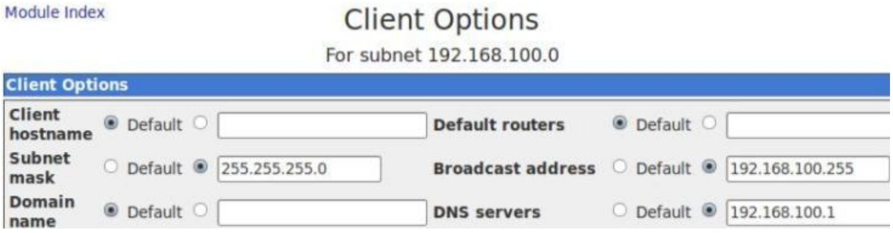
The DHCP server (DHCP3-server package in the Ubuntu repositories) was installed after the initial installation of Ubuntu server had finished. Again, Webmin was used to configure the server with the IP addresses to be handed out, along with the other client options (subnet mask, DNS server and broadcast address).



Subnet Details

Subnet description	DHCP Scope for Network 3		
Network address	192.168.100.0	Netmask	255.255.255.0
Address ranges	192.168.100.50 - 192.168.100.60	□ Dynamic BOOTP ?	
	-	□ Dynamic BOOTP ?	

Configuring the DHCP address range via Webmin



Client Options

For subnet 192.168.100.0

Client Options

Client hostname	● Default ○	Default routers	● Default ○
Subnet mask	○ Default ● 255.255.255.0	Broadcast address	○ Default ● 192.168.100.255
Domain name	● Default ○	DNS servers	○ Default ● 192.168.100.1

Configuring the DHCP options for the DHCP scope

6.1.4 Network 4

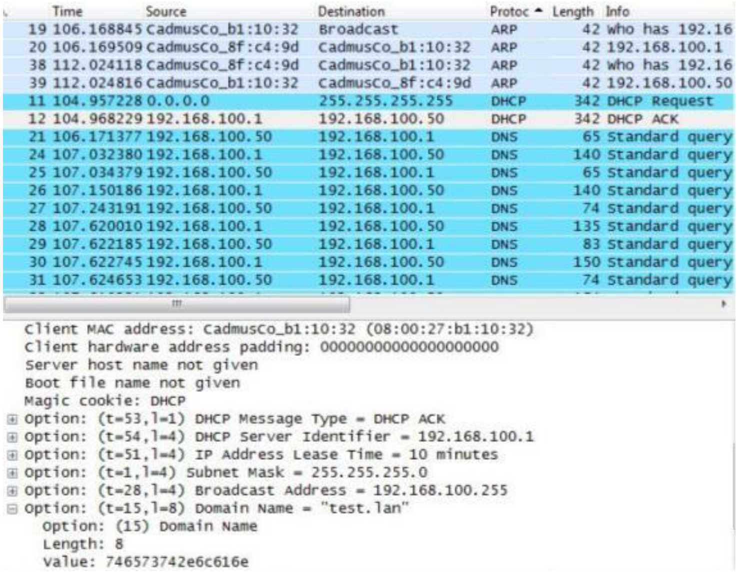
Due to the lack of wireless connectivity features in GNS3, the network has not been implemented as it would be essentially the same as network 3.

6.1.5 Other functionalities in GNS3

Wireshark Integration

Wireshark Integration has been built into GNS3 to allow users to capture packets sent on the virtual LAN. To start a packet capture, a user has to right click on the link (in the network topology pane) and select start capturing. Once a packet capture has been started, Wireshark should show up and provide the user with a view of packets captured from the virtual network in GNS (GNS3 Project, 2007).

The following screenshot shows a capture file showing packets that were transmitted across the link between VBOX1 (The Ubuntu Server machine) and SW2 (Switch 2) of network 3 (see the topology screenshot under network 3 in chapter 6.1.3).



```
   Time       Source               Destination        Protoc ▲ Length Info
19 106.168845 CadmusCo_b1:10:32    Broadcast          ARP          42 Who has 192.16
20 106.169509 CadmusCo_8f:c4:9d    CadmusCo_b1:10:32  ARP          42 192.168.100.1
38 112.024118 CadmusCo_8f:c4:9d    CadmusCo_b1:10:32  ARP          42 Who has 192.16
39 112.024816 CadmusCo_b1:10:32    CadmusCo_8f:c4:9d  ARP          42 192.168.100.50
11 104.957228 0.0.0.0              255.255.255.255    DHCP        342 DHCP Request
12 104.968229 192.168.100.1        192.168.100.50     DHCP        342 DHCP ACK
21 106.171377 192.168.100.50       192.168.100.1      DNS          65 Standard query
24 107.032380 192.168.100.1        192.168.100.50     DNS         140 Standard query
25 107.034379 192.168.100.50       192.168.100.1      DNS          65 Standard query
26 107.150186 192.168.100.1        192.168.100.50     DNS         140 Standard query
27 107.243191 192.168.100.50       192.168.100.1      DNS          74 Standard query
28 107.620010 192.168.100.1        192.168.100.50     DNS         135 Standard query
29 107.622185 192.168.100.50       192.168.100.1      DNS          83 Standard query
30 107.622745 192.168.100.1        192.168.100.50     DNS         150 Standard query
31 107.624653 192.168.100.50       192.168.100.1      DNS          74 Standard query

  Client MAC address: CadmusCo_b1:10:32 (08:00:27:b1:10:32)
  Client hardware address padding: 00000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
⊞ Option: (t=53,l=1) DHCP Message Type = DHCP ACK
⊞ Option: (t=54,l=4) DHCP Server Identifier = 192.168.100.1
⊞ Option: (t=51,l=4) IP Address Lease Time = 10 minutes
⊞ Option: (t=1,l=4) Subnet Mask = 255.255.255.0
⊞ Option: (t=28,l=4) Broadcast Address = 192.168.100.255
⊟ Option: (t=15,l=8) Domain Name = "test.lan"
    Option: (15) Domain Name
    Length: 8
    Value: 746573742e6c616e
```

192.168.100.1 is the Ubuntu Server Machine, and 192.168.100.50 is the IP address leased via DHCP to the Xubuntu client. The top half of the screenshot shows a small selection of the 40 packets captured by wireshark (ordered by protocol), with the bottom half of the screen showing the DHCP ACK (acknowledgement) with the details and options configured in the DHCP server settings like the domain name and subnet mask.

Virtual machine integration

GNS3 supports integration with Virtualbox (amongst others) to allow users to use real world operating systems to interface with various aspects of the simulated network(s). This feature is a benefit as it allows for real world operating systems (unlike the one for PCs in Packet Tracer) to be used in the simulated network (Graphical Network Simulator Project, 2007).

Using real hardware with a simulated network

GNS3 supports using real hardware with a simulated network for cases when a specific item is needed but not implemented within GNS, such as an IP phone. (GNS3, 2007)

6.1.6 Troubleshooting networks with GNS3

GNS3 would be a good candidate for troubleshooting networks, especially as it has integration with virtual machines. With the virtual machine integration, a network administrator could troubleshoot operating system issues, as well as network issues. Getting statistics about the network would be a problem

7.0 Implementing the networks in OPNET IT Guru Academic Edition

This section will look at the steps to implement each of the networks detailed in the specifications (Chapter 4.1) in the simulation software listed in Chapter 2.

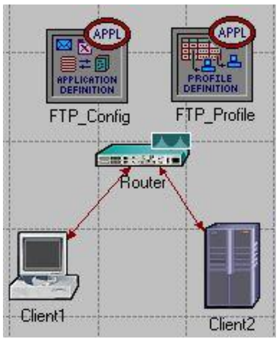
7.1 OPNET IT Guru Academic Edition

This section will deal with the implementation of each of the network specifications in the OPNET IT Guru simulator.

7.1.1 Network 1

When creating a project in IT Guru, it will ask the user to enter a name for the project, and a name for a scenario. The start-up wizard tutorial then guides the user through creating an initial network topology (the ‘Empty Scenario’ was the chosen option here). The ‘Office’ option was chosen for the network location. The office size was set to 20x20M. The technologies used in the network were Cisco and Ethernet.

A Cisco 2621 router, 1 Ethernet workstation and a server were all added to the network map. All hosts were connected together using gigabit Ethernet connections. To help set up the application profiles and configurations, the Lab 09: Mobile WLAN guide from the Network Simulation Experiments Manual was used (Aboelela, 2012).



The topology for network 1

Device Configuration

Client

The client is a standard Ethernet workstation. The only changes were to give the host an IP address/Subnet mask (192.168.100.101/255.255.255.0), and rename it from node_0 to Client1. The FTP service was also added to simulate traffic on the network.

Server

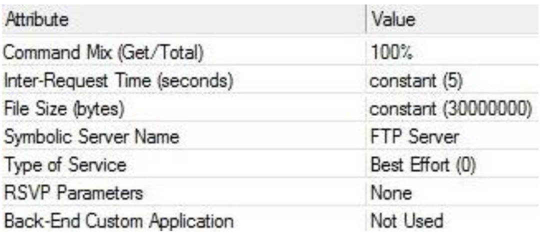
The server is a standard Ethernet server. The server was originally supposed to be another client, but the client workstation didn’t support acting as an FTP server. The Server was configured to accept FTP connections from clients and given an IP address of 192.168.200.200.

Router

The router is a Cisco 2621 router. As per the specification, the router had IF2 (interface 2) configured with an IP address and subnet mask of 192.168.100.1/24 (/24 is the CIDR notation for 255.255.255.0) and IF3 with an IP address and subnet mask of 192.168.200.254/24. It was also renamed from node_3 to Router.

FTP_Config

A new Application definition was created with the name parameter set as 'FTP_Application', with the FTP (File Transfer Protocol) service being configured as per the screenshot below:



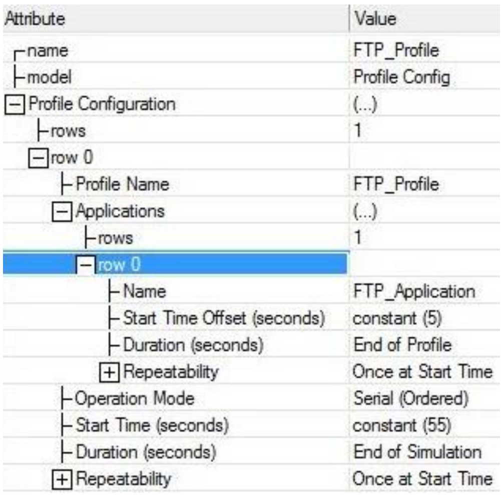
Attribute	Value
Command Mix (Get/Total)	100%
Inter-Request Time (seconds)	constant (5)
File Size (bytes)	constant (30000000)
Symbolic Server Name	FTP Server
Type of Service	Best Effort (0)
RSVP Parameters	None
Back-End Custom Application	Not Used

The settings for the FTP_Application application in network 1

The screenshot above shows that data will be transferred both ways (Client to server, and server to client). This is specified by the command mix variable being set to 100%. The file size is the amount of data to be sent over the network and was set to 30MB.

FTP_Profile

This deals with the main configuration of the profile and holds values such as the name, start time and duration.



Attribute	Value
name	FTP_Profile
model	Profile Config
Profile Configuration	(...)
rows	1
row 0	
Profile Name	FTP_Profile
Applications	(...)
rows	1
row 0	
Name	FTP_Application
Start Time Offset (seconds)	constant (5)
Duration (seconds)	End of Profile
Repeatability	Once at Start Time
Operation Mode	Serial (Ordered)
Start Time (seconds)	constant (55)
Duration (seconds)	End of Simulation
Repeatability	Once at Start Time

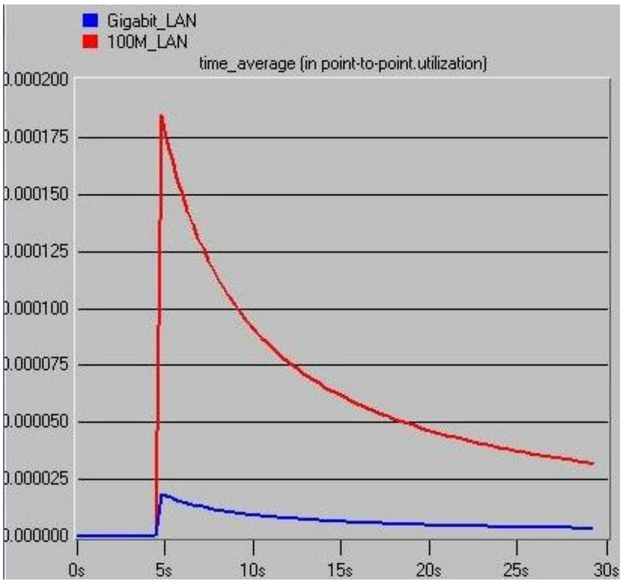
The configuration screen of the FTP_Profile showing the start/end times of the simulation

The screenshot above shows the configuration of the profile. This is where the main configuration of the profile is set. In the example above, the profile has been configured with:

- A name of File_Transfer
- A start time offset of 5 seconds
- A duration that spans the entire length of the simulation

Examining the utilisation of a fast Ethernet link and a gigabit Ethernet link

A test was performed to see how the increased bandwidth/throughput of a gigabit link would affect the total utilisation of the 2 different link speeds.

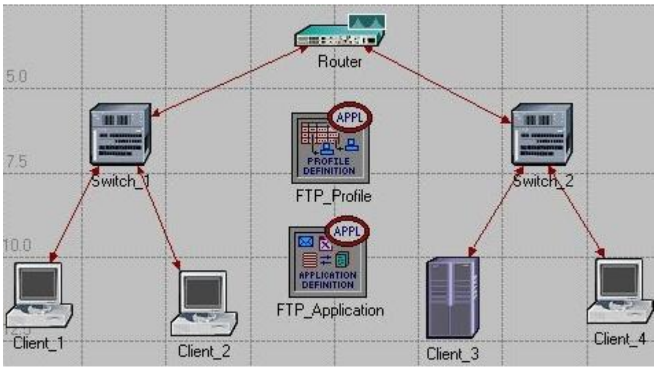


The result of the test showing that the fast Ethernet connection (red line) has a higher overall utilisation than the gigabit link due to the reduced bandwidth/throughput of the fast Ethernet connection

The Ethernet link between client 2 (the FTP Server) was the link that was monitored for this test. When the link was using the slower connection (fast Ethernet), the total utilisation was higher as there was less bandwidth available (when compared to the gigabit link). This also shows that the gigabit link has much more bandwidth available (than a fast Ethernet connection) which in turn reduces the total utilisation of the link.

7.1.2 Network 2

This network consists of 1 Cisco 2621 router, 2 16 port Ethernet switches, 3 Ethernet clients, 1 server and 1 each of a profile definition and application definition. All nodes were connected together via Gigabit Ethernet links.

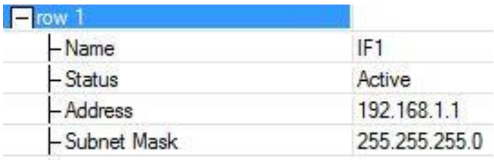


The topology for network 2

Device Configuration

Router

The router used for this network was a Cisco 2621 which had the IP address/subnet mask of 192.168.1.1/24 set for interface 1, and the IP address/subnet mask of 192.168.100.1/24 for interface 2. The node was also renamed to 'Router'.



row 1	
Name	IF1
Status	Active
Address	192.168.1.1
Subnet Mask	255.255.255.0

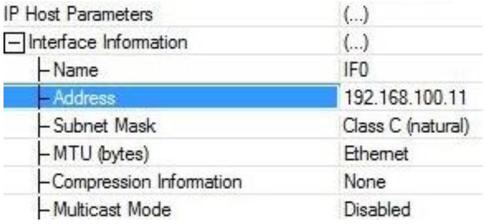
The IP Address/subnet mask for the routers interface 1

Switches

A standard 16 port Ethernet switch was used to provide the switching functionality for the network. No IP addresses were set as the attributes dialogue box didn't provide the option for giving the switches an address.

Clients

3 standard Ethernet workstations were used to simulate end users computers. Each one was renamed from 'node_x' (with x being a number) to 'Client_x' and given an IP address as per the specification.



IP Host Parameters	(...)
⊟ Interface Information	(...)
├ Name	IF0
├ Address	192.168.100.11
├ Subnet Mask	Class C (natural)
├ MTU (bytes)	Ethernet
├ Compression Information	None
├ Multicast Mode	Disabled

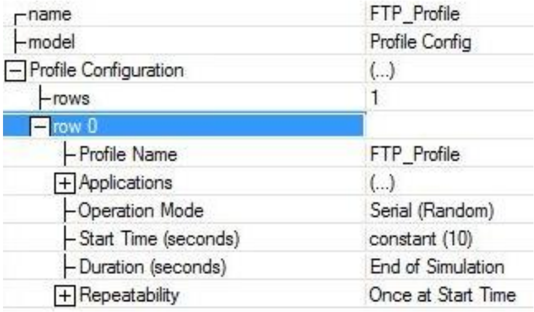
The IP address settings for Client_4

Server

1 standard Ethernet server was added to the network to handle the FTP application during the simulation.

Application Profile

Again, the application profile was configured to run the FTP_App application after 10 seconds of the simulation and run it until the end of the simulation time. It was also set to not repeat the task and to run it only once at the simulation start time.



┌ name	FTP_Profile
├ model	Profile Config
⊟ Profile Configuration	(...)
├ rows	1
⊟ row 0	
├ Profile Name	FTP_Profile
⊞ Applications	(...)
├ Operation Mode	Serial (Random)
├ Start Time (seconds)	constant (10)
├ Duration (seconds)	End of Simulation
⊞ Repeatability	Once at Start Time

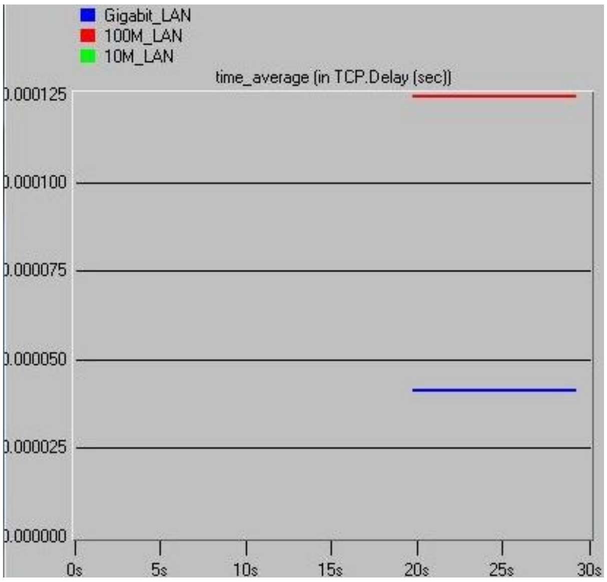
The settings for the FTP_Profile

Application Configuration

Similar to the previous network, the application used to test the network was FTP (File Transfer Protocol). The configuration of this profile is exactly the same as the configuration in the FTP_Config in the previous network (except the name was slightly different – FTP_App).

Comparing the TCP delay on a 100MB/s and 1000MB/s LAN

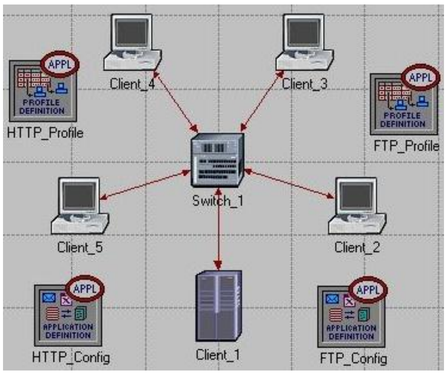
One of the many statistics that IT Guru can capture during simulations is the TCP delay (in seconds). The following screenshot shows the difference in the TCP delay between a 10MB/s, 100MB/s & a gigabit LAN using the network topology for network 2.



The total TCP delay of client 3 (in seconds) on different network speeds. The vertical axis is the time in seconds, and the horizontal axis is the simulation time

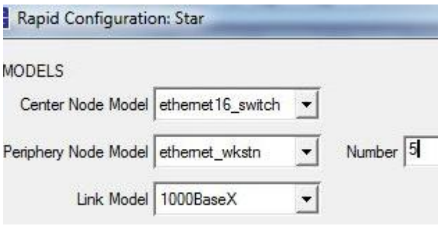
Over a simulated period of 30 seconds, the TCP delay of Client_3 (the FTP server) was a lot shorter on a gigabit (1000MB/s) network compared to the 10MB/s & 100MB/s network. This is due to the increased bandwidth that gigabit networks provide over the other speeds. It is also worth noting that even though a gigabit network states that it can transfer 1GB of data per second it may only max out at around 500MB/s (for example) due to various inefficiencies of implementation (Peterson & Davie, 2012, p. 45). The same also applies to 10MB/s & 100MB/s networks.

7.1.3 Network 3



The topology for network 3

To insert the network devices into the topology , the Rapid Configuration tool was used to insert a star based network topology with 1 Ethernet 16 port switch (set as the centre node) and 5 clients (to be changed to a server later on). All devices were connected together via gigabit Ethernet connections.



Rapid Configuration: Star

MODELS

Center Node Model: ethernet16_switch

Periphery Node Model: ethernet_wkstn Number: 5

Link Model: 1000BaseX

Using the rapid configuration to build a star based topology

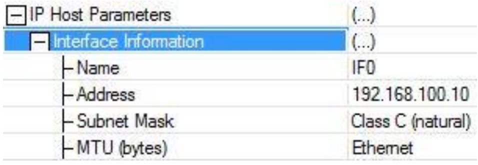
Device Configuration

Switch

One standard Ethernet 16 port switch was added to the network by the rapid configuration tool. Due to the limited configuration options of the standard Ethernet switch, the settings were left at the defaults.

Clients

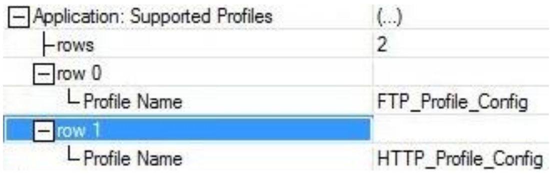
To create the clients, the rapid configuration tool was used to create 5 standard Ethernet clients. Each client had their networking interface 0 (IF0) configured with a static IP address/subnet mask within the 192.168.100.0/255.255.255.0 range.



⊟ IP Host Parameters	(...)
⊟ Interface Information	(...)
├ Name	IF0
├ Address	192.168.100.10
├ Subnet Mask	Class C (natural)
├ MTU (bytes)	Ethernet

The IP address for client 2

Each client was also configured to support the HTTP_Profile_Config and FTP_Profile_Config services to transfer data to/from the server via HTTP/FTP during the simulation.



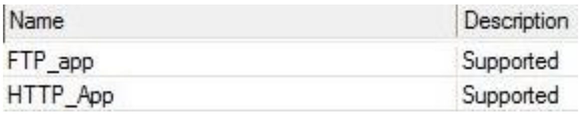
Application: Supported Profiles	(...)
rows	2
row 0	
Profile Name	FTP_Profile_Config
row 1	
Profile Name	HTTP_Profile_Config

The 'Supported Services' configuration of the clients

Server

As IT Guru doesn't support DHCP & DNS, these protocols cannot be configured. The FTP and HTTP applications will be used instead. Instead of using DHCP, each device was allocated a static IP address.

The server was configured to host the FTP service to allow clients to FTP into the server to upload/download data. The server was also configured to accept HTTP connections.

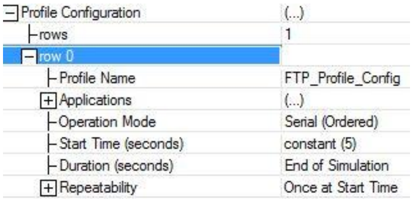


Name	Description
FTP_app	Supported
HTTP_App	Supported

The supported services of the server

Application Profile (FTP)

The application profile was set up to start the simulation after 5 seconds and run until the end of the simulation, as well as the profile not being repeated. The profile name was changed from node_0 to FTP_Profile, and the Profile Configuration name changed from nothing to FTP_Profile_Config.



Profile Configuration	(...)
rows	1
row 0	
Profile Name	FTP_Profile_Config
Applications	(...)
Operation Mode	Serial (Ordered)
Start Time (seconds)	constant (5)
Duration (seconds)	End of Simulation
Repeatability	Once at Start Time

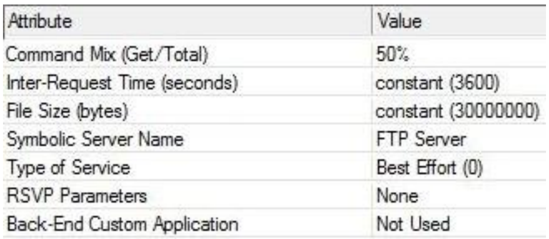
The FTP_Profile configuration for network 3

Application Profile (HTTP)

The application profile for the HTTP service was configured with the same parameters as the FTP application profile, apart from the names (HTTP_profile and HTTP_Profile_Config).

Application Configuration (FTP)

The application configuration profile was renamed from node_1 to FTP_Config. The configuration profile was configured to enable the FTP service with custom settings as detailed below.



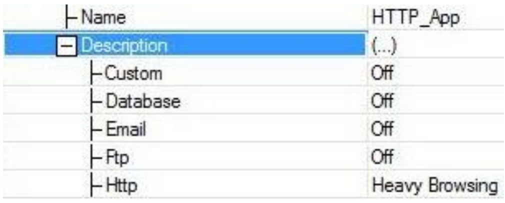
Attribute	Value
Command Mix (Get/Total)	50%
Inter-Request Time (seconds)	constant (3600)
File Size (bytes)	constant (30000000)
Symbolic Server Name	FTP Server
Type of Service	Best Effort (0)
RSVP Parameters	None
Back-End Custom Application	Not Used

The FTP settings for network 3

In the screenshot above, the command mix has been set to 50% to allow clients to transfer data to/from the server. The inter-request time has been set to 3600 seconds (1 hour), along with the file size being set to 30MB (30000000 bytes). The rest of the settings have been left at the defaults.

HTTP (HyperText Transfer Protocol) Application Configuration

The HTTP Application was set to simulate heavy web browsing during the simulation. The configuration had the name changed to ‘HTTP_App’.

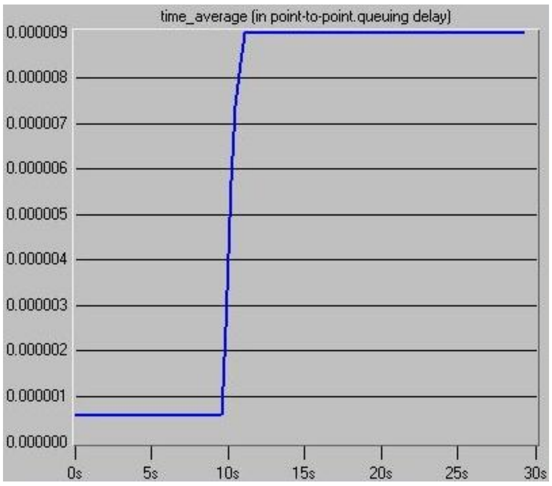


├ Name	HTTP_App
⊟ Description	(...)
├ Custom	Off
├ Database	Off
├ Email	Off
├ Ftp	Off
├ Http	Heavy Browsing

The HTTP_App configuration for network 3

Finding out the queuing delay of an Ethernet link in network 3

One of the many statistics that IT Guru can collect from an Ethernet link is the queuing delay. This statistic is helpful to find out how much data can be transferred over a link before it gets congested and packets get lost/corrupted/dropped.

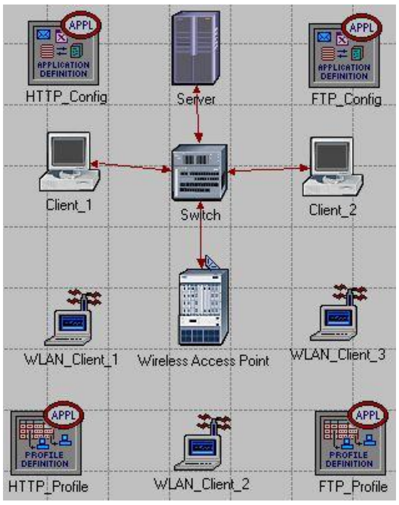


The results of the queuing delay test. The vertical axis shows the delay in seconds, while the horizontal axis shows the simulation time.

The results from the queuing delay test showed that the queuing delay was initially low, but rapidly increased at around 10 seconds before stabilising and remaining the same for the rest of the simulation period.

7.1.4 Network 4

The size of the office was set to 50mx50m for this network. The rapid configuration tool was used to place 8 nodes into the topology (which included 7 Ethernet clients and a 16 port Ethernet switch. Some of the Ethernet clients were then changed to servers and wireless devices). Apart from the wireless devices, all hosts were connected together using Gigabit Ethernet connections.

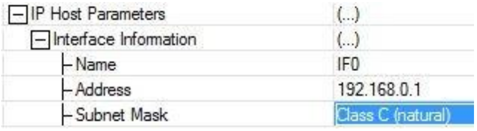


The topology for network 4 showing the network devices and application configurations

Device Configuration

Server

The server was configured to support both the FTP_App and HTTP_App services. The IP address was also set to 192.168.0.1 with a subnet mask of 255.255.255.0, as per the network specification.



⊟ IP Host Parameters	(...)
⊟ Interface Information	(...)
├ Name	IF0
├ Address	192.168.0.1
├ Subnet Mask	Class C (natural)

The server IP address/subnet mask

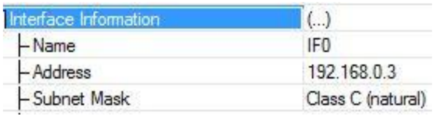
Some research had been carried out to see if IT Guru supported DHCP. This research resulted in one project being found that tried to implement DHCP into OPNET (Shavit, 2003). Unfortunately, the project hadn't reached its goal and looks as though it has been abandoned as the last update was in 2003.

Switch

As per the previous networks, the switch has been left at the default settings.

Clients

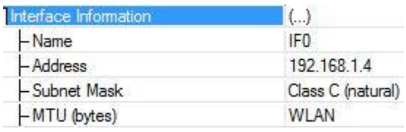
The wired clients were each set with a static IP address within the 192.168.0.0/24 range (rather than DHCP assigned addresses due to the issues mentioned earlier on in this section).



Interface Information	(...)
├ Name	IF0
├ Address	192.168.0.3
├ Subnet Mask	Class C (natural)

Setting the IP address for a wired client

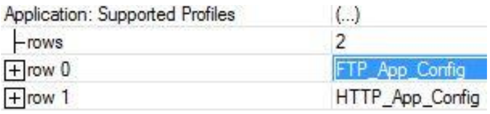
The wireless clients were also given static IP addresses, but on a different range to the wired clients (due to the wireless router). The wireless clients were given an IP address within the 192.168.1.0/24 range.



Interface Information	(...)
├ Name	IF0
├ Address	192.168.1.4
├ Subnet Mask	Class C (natural)
├ MTU (bytes)	WLAN

Setting the IP address for a wireless client

Both the wired and wireless clients were configured to support the HTTP & FTP services during the simulation.



Application: Supported Profiles	(...)
├ rows	2
⊞ row 0	FTP_App_Config
⊞ row 1	HTTP_App_Config

Configuring the profiles that the client can accept

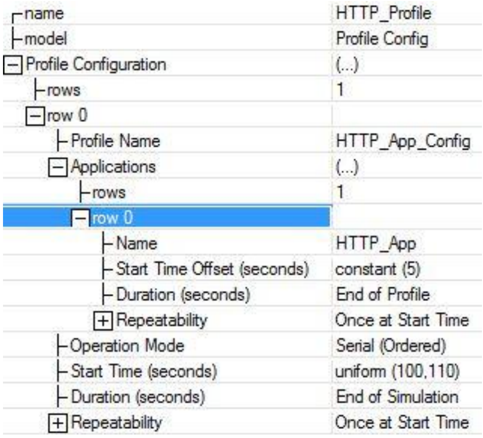
Wireless Networking

A ‘wlan_Ethernet_router’ was used as an access point for wireless clients. The first interface (IF0) has the IP address/subnet mask of 192.168.0.2/255.255.255.0 set, and the second interface (IF1) had the IP address of 192.168.1.1/255.255.255.0 set for the wireless interface (as the ‘access point’ is classed as a router by IT Guru).

Application Profiles

HTTP Profile

The HTTP application profile was configured to run the HTTP_App application after 5 seconds, run until the end of the profile and to be repeated only once at the start time.



┌ name	HTTP_Profile
├ model	Profile Config
⊟ Profile Configuration	(...)
├ rows	1
⊟ row 0	
├ Profile Name	HTTP_App_Config
⊟ Applications	(...)
├ rows	1
⊟ row 0	
├ Name	HTTP_App
├ Start Time Offset (seconds)	constant (5)
├ Duration (seconds)	End of Profile
⊞ Repeatability	Once at Start Time
├ Operation Mode	Serial (Ordered)
├ Start Time (seconds)	uniform (100,110)
├ Duration (seconds)	End of Simulation
⊞ Repeatability	Once at Start Time

The HTTP Profile for network 4

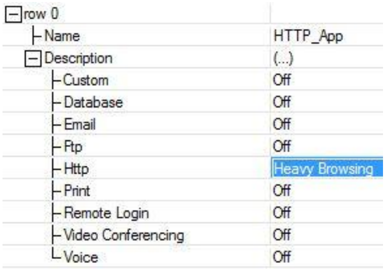
FTP Application Profile

The FTP Profile was configured with the same settings as the HTTP profile, but used the FTP_App application instead of HTTP_App.

Application Configuration

HTTP

The HTTP configuration definition was set to simulate heavy web browsing by the clients. The heavy browsing consisted of 5 medium sized images with 10 pages per server.

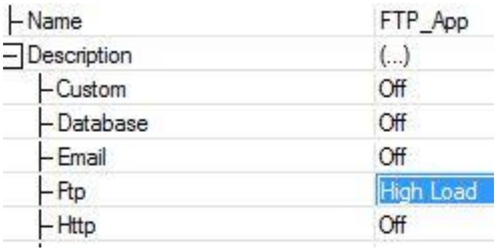


⊟row 0	
├Name	HTTP_App
⊟Description	(...)
├Custom	Off
├Database	Off
├Email	Off
├Ftp	Off
├Http	Heavy Browsing
├Print	Off
├Remote Login	Off
├Video Conferencing	Off
└Voice	Off

The configuration profile for HTTP showing that heavy browsing was simulated

FTP

The FTP configuration definition was also set to produce a heavy load on the server, which consisted of clients transferring 5mb of data to/from the server.



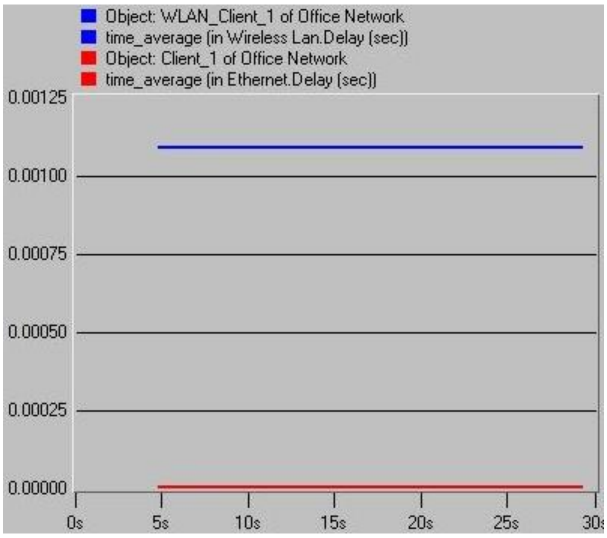
├Name	FTP_App
⊟Description	(...)
├Custom	Off
├Database	Off
├Email	Off
├Ftp	High Load
├Http	Off

The FTP configuration showing that it was set to produce a high load

Comparing the difference of the delay between a wired client and a wireless client

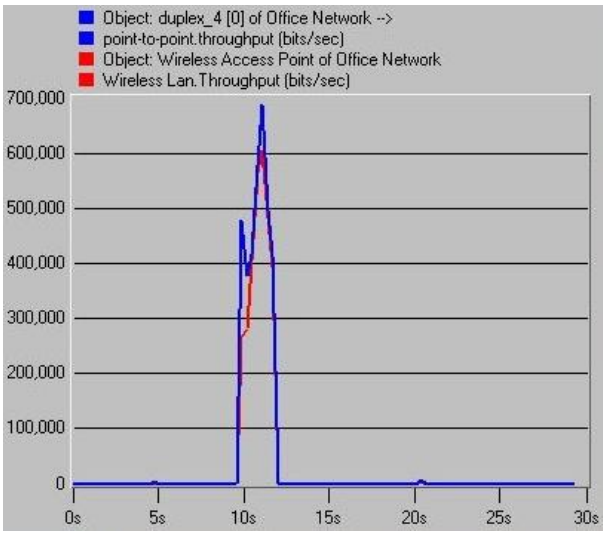
To find how much a wireless network would affect the network delay, a simulation in IT Guru was set up and run to establish the results using the topology from network 4.

The results of the test were that the average delay over the wired LAN was much less than the wireless network. The wired client had an average delay of between 0.025µs and 0.0175µ, whereas the wireless clients delay was much higher averaging at between 0.015µs and 0.025µs.



The difference in delay between a wireless network (blue) and gigabit Ethernet link (red). The vertical axis is the delay time (seconds) and the horizontal axis is the simulation time in minutes.

Another test was also carried out to see how much slower a wireless network was (in terms of throughput) when compared with a gigabit Ethernet connection. This test was measured using the bits per second statistic on both a wireless link and a gigabit Ethernet link. The results from the second test can be found below.



The throughput of a gigabit link (blue) and a wireless link (red). The vertical axis is the bits per seconds measurement, and the horizontal axis the simulation time in minutes.

The graph above shows that although the gigabit link is faster, the wireless link still manages to maintain a similar pattern to the wired link. The wireless network peaks at around 600,000 bits per second and averages around 400,000-500,000 bits per second with 3 wireless clients connected. The gigabit link averages at around 400,000-500,000 bits per second (with 2 wired clients and the 3 wireless clients) and peaks at around 700,000 bits per second.

7.1.5 Other features of IT Guru

Application Characterization Environment (ACE)

The ACE feature of IT Guru allows a user to install a capture agent onto computers connected to the network to allow for network traffic to be captured. If many capture agents are running, they can be centrally managed (Cisco Systems, OPNET, 2005).

The ACE feature will allow network managers to troubleshoot performance issues on a network and allow them to see what/where the bottleneck is. It also allows for a fix to be implemented (in the simulator) and tested before it is rolled out into the real world (University of Maribor, 2006).

7.1.6 Troubleshooting networks with IT Guru

OPNET lists one of the key features of IT Guru as being able to help troubleshooting networks in case of failures of network nodes (OPNET Technologies, 2012). IT Guru would also be a good candidate for troubleshooting networks as it can simulate link or node failures within a network to test the performance of equipment during an outage. The statistics provided by IT Guru would give a good idea of how the network would cope during a period where the network is encountering problems.

8.0 Implementing the networks in Network Simulator 2 (NS2)

This section will look at the steps to implement each of the networks detailed in the specifications (Chapter 4.1) in the simulation software listed in Chapter 2.

8.1 Network Simulator 2 (NS2)

This section will deal with the implementation of the four networks within NS2.

The information from the following guides and tutorials was used in assisting with implementing the networks in NS2: (Xiao, 2011) (Wang, 2004) (Wang, Jianping, Virginia University, 2004)

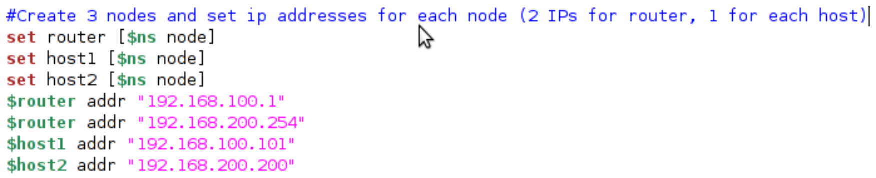
To create a network for use with NS2, the network must be created in a text file (saved with the TCL file extension) using the commands supported by NS2.

The following list is a brief checklist on how to create a network for use with NS2:

1. Open a new text file and create a simulator object by adding '`set ns [new Simulator]`' to the top of the file.
2. Set up the trace files to log details about the network (e.g. packets sent/received)
3. Provide a procedure that NS2 executes after the simulation has completed (e.g. to close the trace files)
4. Create the network hosts and define the link type/speed to connect the hosts together
5. Set up the applications and agents for the needed protocols
6. Set up the timing of when the agents/applications should run
7. Add '`$ns run`' to start the simulation and save the file to disk (with a .tcl file extension)
8. Execute NS from the command line using a command such as '`ns network.tcl`' (where network.tcl is the filename of the file saved in step 7)

8.1.1 Network 1

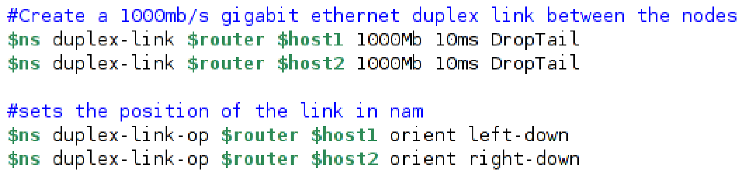
For network 1, there were 3 hosts created. They were named Router, Host1 and Host2. Each of them had an IP address set as per the network specifications.



```
#Create 3 nodes and set ip addresses for each node (2 IPs for router, 1 for each host)
set router [$ns node]
set host1 [$ns node]
set host2 [$ns node]
$router addr "192.168.100.1"
$router addr "192.168.200.254"
$host1 addr "192.168.100.101"
$host2 addr "192.168.200.200"
```

Creating the nodes and setting the IP addresses for each node

The next step was to create and specify the speed and type of the links between hosts. In total, 2 1000mb links (with a 10ms delay) were created to link host1 and host2 to the router. The screenshot on the next page also shows that a user can specify the positioning of a link.



```
#Create a 1000mb/s gigabit ethernet duplex link between the nodes
$ns duplex-link $router $host1 1000Mb 10ms DropTail
$ns duplex-link $router $host2 1000Mb 10ms DropTail

#sets the position of the link in nam
$ns duplex-link-op $router $host1 orient left-down
$ns duplex-link-op $router $host2 orient right-down
```

Creating and specifying the options for the links, as well as specifying the positioning of the links using the orient command

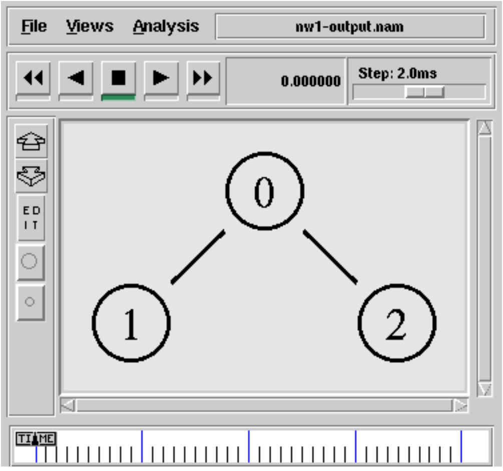
Setting up the TCP and FTP connection was the next process to be completed for this network. TCP (Transmission Control Protocol) was the first to be set up as FTP relies on TCP to establish the connections between hosts before data is sent via FTP.

The TCP connection was created by specifying '`set tcp0 [new Agent/TCP]`' in the configuration file. The TCP agent was then attached to host 1 to allow TCP connections. The next step to allow an FTP transfer (over TCP) was to create a new FTP application and attach it to the existing TCP connection.

A TCP Sink agent was the next item to be configured. The TCP sink receives data from a TCP sender and processes it (Henderson, 2011). The TCP sink was attached to host 2 to allow it to receive the FTP traffic sent from host 1.

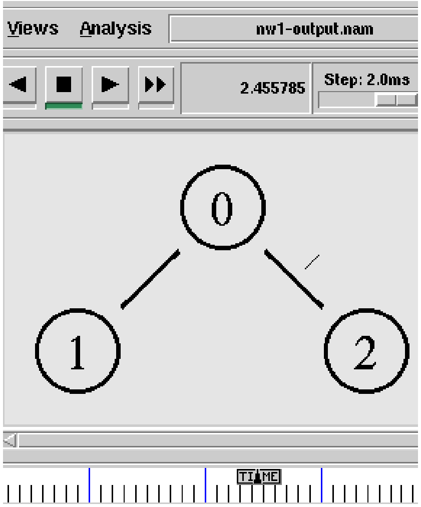
The schedule of events was the next item specified in the file for network 1. This portion of the configuration specified when the FTP service should start/stop sending data over the TCP connection and to call the ending procedure to end the simulation process.

When the network configuration file was loaded into Nam (Network Animator), Nam then produced a visual representation of the network:



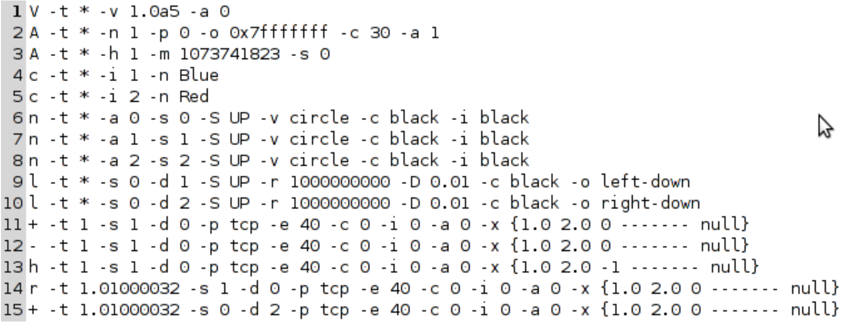
The topology of network 1, as shown by Nam

NS also produced trace files which contained information about the packets travelling across the network.



The FTP simulation in action (shown by the packet being transferred between the router & host 2)

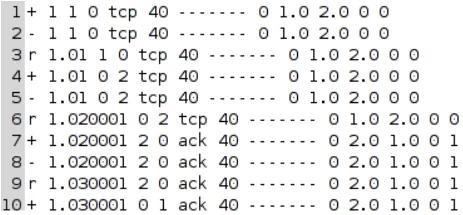
When the output file from Nam (in this case, nw1-output.nam) is opened in a text editor, it should look similar to the following screenshot:



```
V -t * -v 1.0a5 -a 0
A -t * -n 1 -p 0 -o 0x7fffffff -c 30 -a 1
A -t * -h 1 -m 1073741823 -s 0
c -t * -i 1 -n Blue
c -t * -i 2 -n Red
n -t * -a 0 -s 0 -S UP -v circle -c black -i black
n -t * -a 1 -s 1 -S UP -v circle -c black -i black
n -t * -a 2 -s 2 -S UP -v circle -c black -i black
l -t * -s 0 -d 1 -S UP -r 1000000000 -D 0.01 -c black -o left-down
l -t * -s 0 -d 2 -S UP -r 1000000000 -D 0.01 -c black -o right-down
+ -t 1 -s 1 -d 0 -p tcp -e 40 -c 0 -i 0 -a 0 -x {1.0 2.0 0 ------- null}
- -t 1 -s 1 -d 0 -p tcp -e 40 -c 0 -i 0 -a 0 -x {1.0 2.0 0 ------- null}
h -t 1 -s 1 -d 0 -p tcp -e 40 -c 0 -i 0 -a 0 -x {1.0 2.0 -1 ------- null}
r -t 1.01000032 -s 1 -d 0 -p tcp -e 40 -c 0 -i 0 -a 0 -x {1.0 2.0 0 ------- null}
+ -t 1.01000032 -s 0 -d 2 -p tcp -e 40 -c 0 -i 0 -a 0 -x {1.0 2.0 0 ------- null}
```

A small section of the contents of the nw1-output.nam file

There is also another file generated by Nam/NS (nw1-output.tr in this case) and should look similar to the following when opened in a text editor:



```
+ 1 1 0 tcp 40 ------- 0 1.0 2.0 0 0
- 1 1 0 tcp 40 ------- 0 1.0 2.0 0 0
r 1.01 1 0 tcp 40 ------- 0 1.0 2.0 0 0
+ 1.01 0 2 tcp 40 ------- 0 1.0 2.0 0 0
- 1.01 0 2 tcp 40 ------- 0 1.0 2.0 0 0
r 1.020001 0 2 tcp 40 ------- 0 1.0 2.0 0 0
+ 1.020001 2 0 ack 40 ------- 0 2.0 1.0 0 1
- 1.020001 2 0 ack 40 ------- 0 2.0 1.0 0 1
r 1.030001 2 0 ack 40 ------- 0 2.0 1.0 0 1
+ 1.030001 0 1 ack 40 ------- 0 2.0 1.0 0 1
```

A small section of the nw1-output.tr file

The first line from the trace file (in the screenshot above) is:

+ 1 1 0 tcp 40 ------- 0 1.0 2.0 0 0

- The + sign means that the packet has been placed in the queue
- ‘1’ is the time that the even happened (e.g. after 1 seconds of simulation time)
- ‘1’ is the number of the input node where the event happened
- ‘0’ is the number of the output node where the event happened

- ‘tcp’ is the type of packet (could be TCP, SYN or ACK (acknowledgement))
- ’40’ is the size of the packet in bytes
- The dashes are additional flags on the packet
- The ‘0’ following the dashes is the Flow ID for IPv6
- The ‘1.0’ is the source of the packet (in address:port format)
- The ‘2.0’ is the destination of the packet (also in the same form as above)
- The ‘0’ is sequence number of the packet
- The last ‘0’ is the unique ID of the packet

The second line is similar to the first, but has a - symbol instead of a + symbol. The - symbol means that the packet has been removed from the queue.

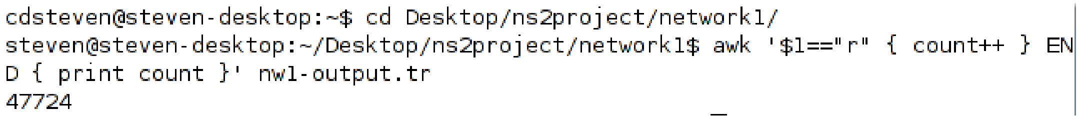
The third line is slightly different as it starts with a letter rather than a symbol:

r 1.01 1 0 tcp 40 ------- 0 1.0 2.0 0 0

- The ‘r’ stands for received (as this line shows a packet received from another host)
- The remaining fields are the same as given in the first line

The following references were used to help understand the trace file format: (Fall & Varadhan, 2011, p. 159) (ManojKumar.A, 2010) (Altman, 2003, p. 27)

Analysing the trace file was accomplished by using AWK scripts. As a starting point, A few basic tasks were performed (such as counting the number of transmitted packets) to learn the basics of writing scripts using AWK.

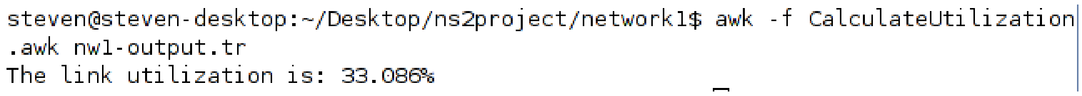


```
cdsteven@steven-desktop:~$ cd Desktop/ns2project/network1/
steven@steven-desktop:~/Desktop/ns2project/network1$ awk '$1=="r" { count++ } EN
D { print count }' nw1-output.tr
47724
```

A simple AWK script that counts the total number of lines starting with ‘r’ (for received packets) within the network1 trace file

Examining the utilisation of a fast Ethernet link and a gigabit Ethernet link

An AWK script was used to calculate the utilisation of the links that connect the network nodes together. An example of the script used to calculate the utilisation can be found in Appendix 1.5.1: Script to calculate the network utilisation. The calculation for finding out the utilisation was found on a forum post (hesam, 2010), and placed into an AWK script.

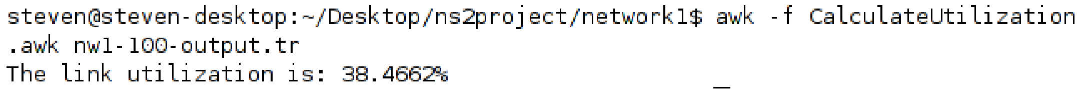


```
steven@steven-desktop:~/Desktop/ns2project/network1$ awk -f CalculateUtilization
.awk nw1-output.tr
The link utilization is: 33.086%
```

The result of calculating the link utilisation

When the script was run on the gigabit network, it calculated the network utilisation as just over 33%. Any network with over 30% utilisation is considered to be under heavy load (Peterson & Davie, 2012, p. 127).

When tested against fast Ethernet speeds, the utilisation jumps to just over 38%, meaning that this is also a heavily loaded network.

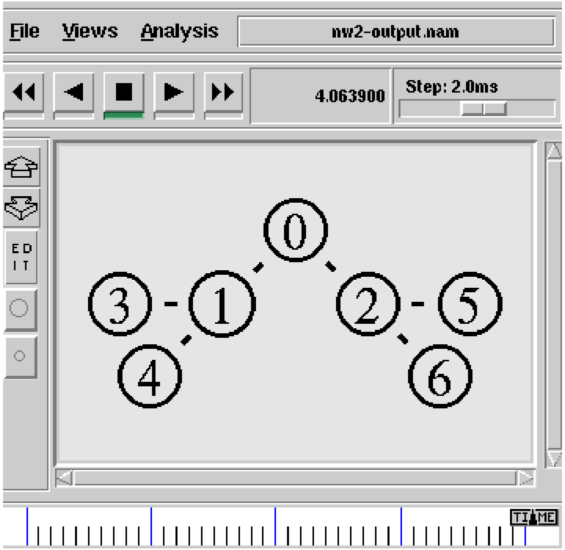


```
steven@steven-desktop:~/Desktop/ns2project/network1$ awk -f CalculateUtilization
.awk nw1-100-output.tr
The link utilization is: 38.4662%
```

The utilisation calculation performed on the network using fast Ethernet

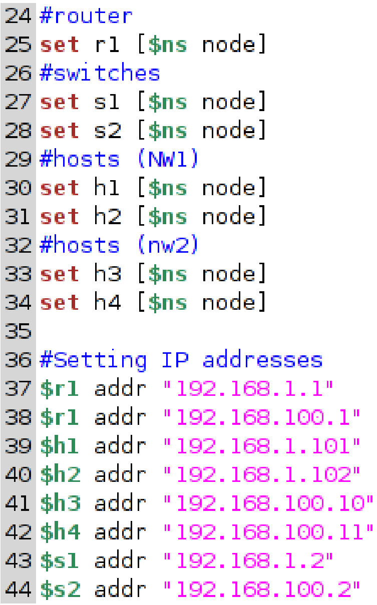
8.1.2 Network 2

For network 2, 7 nodes were created, as well as an FTP transfer also being set up to test the performance of the network.



The topology for network 2

Creating and setting the IP addresses for hosts was done in a similar fashion to network 1:

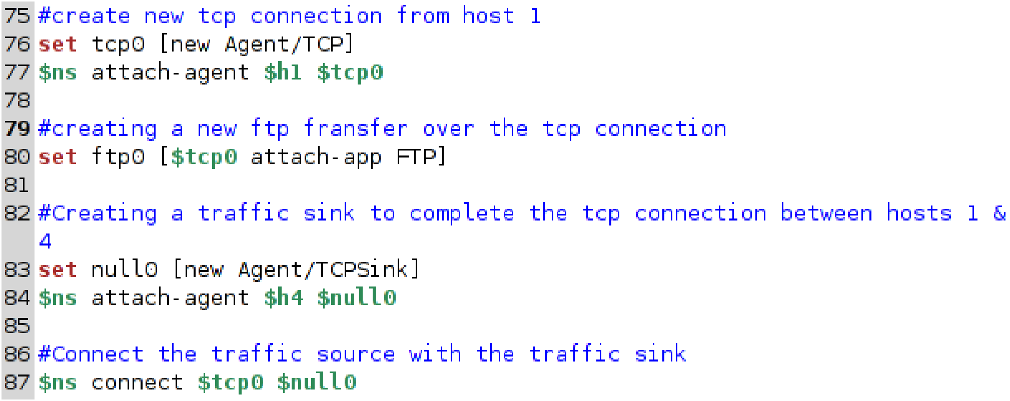


```
#router
set r1 [$ns node]
#switches
set s1 [$ns node]
set s2 [$ns node]
#hosts (NW1)
set h1 [$ns node]
set h2 [$ns node]
#hosts (nw2)
set h3 [$ns node]
set h4 [$ns node]

#Setting IP addresses
$r1 addr "192.168.1.1"
$r1 addr "192.168.100.1"
$h1 addr "192.168.1.101"
$h2 addr "192.168.1.102"
$h3 addr "192.168.100.10"
$h4 addr "192.168.100.11"
$s1 addr "192.168.1.2"
$s2 addr "192.168.100.2"
```

Creating hosts and setting the IP addresses

Creating the links was also similar to the previous network. Each link was configured to allow gigabit speeds with a 2ms (millisecond) delay. The link layouts were also specified to allow for the network topology to be clearly displayed in Nam.

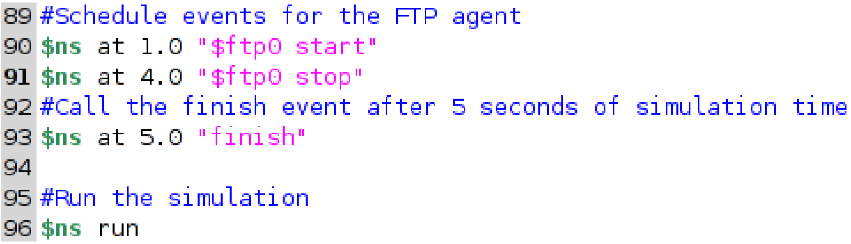
The FTP service configuration was also done in the same way as the previous network. A new TCP connection was created and connected to host 1. The FTP connection was then created and attached to the TCP connection so that FTP could transfer data over the TCP protocol. A traffic sink was created and attached to host 4 to complete the TCP connection.



```
#create new tcp connection from host 1
set tcp0 [new Agent/TCP]
$ns attach-agent $h1 $tcp0

#creating a new ftp fransfer over the tcp connection
set ftp0 [$tcp0 attach-app FTP]

#Creating a traffic sink to complete the tcp connection between hosts 1 & 4
set null0 [new Agent/TCPSink]
$ns attach-agent $h4 $null0

#Connect the traffic source with the traffic sink
$ns connect $tcp0 $null0
```

Setting up the TCP/FTP connections

The next item to be configured was the event scheduler. This looked after when specific event should start and stop. In the screenshot below, the schedule shows that the FTP service will start after 1 second of simulation time and stop after 4 seconds, with 5 seconds being the point where the simulation stops. The final line (`'$ns run'`) allows the simulation to start.



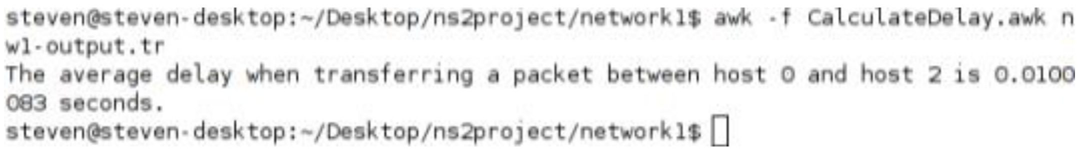
```
#Schedule events for the FTP agent
$ns at 1.0 "$ftp0 start"
$ns at 4.0 "$ftp0 stop"
#Call the finish event after 5 seconds of simulation time
$ns at 5.0 "finish"

#Run the simulation
$ns run
```

Setting up the task scheduler

Comparing the delay on a 100MB/s and 1000MB/s LAN

To help determine the delay of the network, an AWK script was used. The script was adapted from one found on the NS2Ultimate website (Issariyakul, 2011).

When the script was run against the trace file from network 2, it calculated the delay as being 0.01µs over a gigabit network with a 10ms delay. A copy of the script can be found in Appendix 1.5.2.

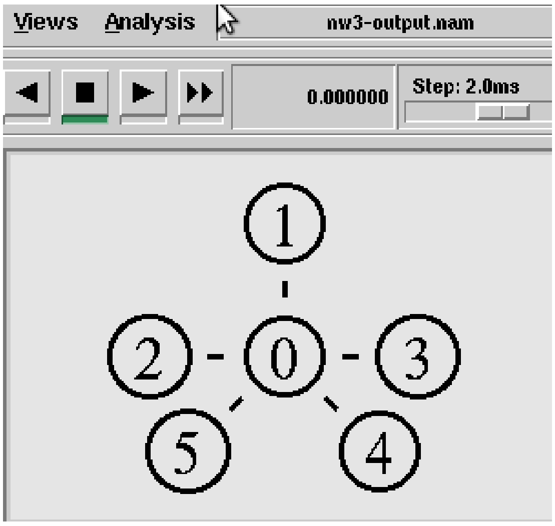


```
steven@steven-desktop:~/Desktop/ns2project/network1$ awk -f CalculateDelay.awk n
w1-output.tr
The average delay when transferring a packet between host 0 and host 2 is 0.0100
083 seconds.
steven@steven-desktop:~/Desktop/ns2project/network1$ 
```

The output of the CalculateDelay AWK script

When the script was run on the same network, using 100mb/s links rather than gigabit links, there was only a marginal difference (with a 0.010µs delay).

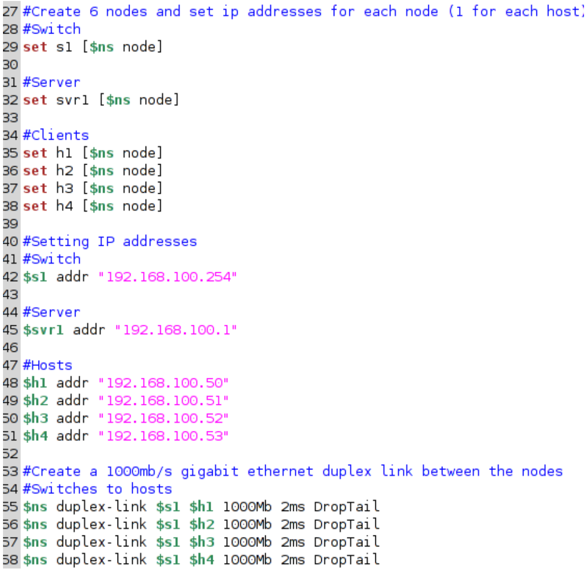
8.1.3 Network 3



The topology of network 3 (0 is the switch, 1 is the server and 2-5 are clients)

6 nodes were created for network 3. The 6 nodes included 1 server, 1 switch and 4 computers. A traffic generator was set up (using UDP rather than TCP) to simulate DNS requests, as well as an FTP transfer using TCP. All devices were given a static IP address instead of using DHCP.

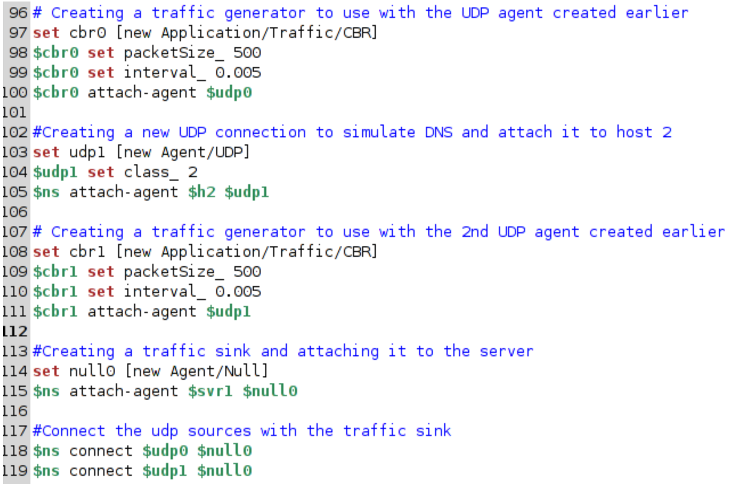
Whilst setting up this network, a segmentation error was encountered when trying to run the simulation using NS2. As the error message wasn't very helpful (NS2 just gave 'Segmentation Fault' as the error message and then quit), the source code for the network had to be checked for errors that contributed towards the segmentation fault error. Once the error had been found and corrected (the fault was that each of the CBR traffic generators was assigned the same variable name), the code ran fine.



```
#Create 6 nodes and set ip addresses for each node (1 for each host)
#Switch
set s1 [$ns node]

#Server
set svr1 [$ns node]

#Clients
set h1 [$ns node]
set h2 [$ns node]
set h3 [$ns node]
set h4 [$ns node]

#Setting IP addresses
#Switch
$s1 addr "192.168.100.254"

#Server
$svr1 addr "192.168.100.1"

#Hosts
$h1 addr "192.168.100.50"
$h2 addr "192.168.100.51"
$h3 addr "192.168.100.52"
$h4 addr "192.168.100.53"

#Create a 1000mb/s gigabit ethernet duplex link between the nodes
#Switches to hosts
$ns duplex-link $s1 $h1 1000Mb 2ms DropTail
$ns duplex-link $s1 $h2 1000Mb 2ms DropTail
$ns duplex-link $s1 $h3 1000Mb 2ms DropTail
$ns duplex-link $s1 $h4 1000Mb 2ms DropTail
```

Creating the nodes and links

The links also had the orientation set to keep links/nodes in order; otherwise Nam would have placed the nodes randomly.



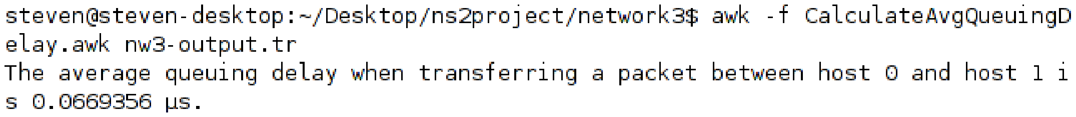
```
# Creating a traffic generator to use with the UDP agent created earlier
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0

#Creating a new UDP connection to simulate DNS and attach it to host 2
set udp1 [new Agent/UDP]
$udp1 set class_ 2
$ns attach-agent $h2 $udp1

# Creating a traffic generator to use with the 2nd UDP agent created earlier
set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 500
$cbr1 set interval_ 0.005
$cbr1 attach-agent $udp1

#Creating a traffic sink and attaching it to the server
set null0 [new Agent/Null]
$ns attach-agent $svr1 $null0

#Connect the udp sources with the traffic sink
$ns connect $udp0 $null0
$ns connect $udp1 $null0
```

Setting up the UDP connections to the server

Once the TCP/FTP and UDP traffic generators were set up, the schedule was the next item to be configured. The event scheduler was set up in a similar way to the previous networks to start/stop the FTP connection, as well as the 2 UDP traffic generators.

Calculating the average queuing delay

An AWK script was created to calculate the average queuing delay between two hosts on the network. A copy of the script can be found in Appendix 1.5.4.



```
steven@steven-desktop:~/Desktop/ns2project/network3$ awk -f CalculateAvgQueuingD
elay.awk nw3-output.tr
The average queuing delay when transferring a packet between host 0 and host 1 i
s 0.0669356 µs.
```

The result from calculating the average queuing delay using the AWK script

The screenshot above shows that the average queuing delay for a packet travelling between host 0 and host 1 on network 3 was calculated as 0.66µs.

8.1.4 Network 4

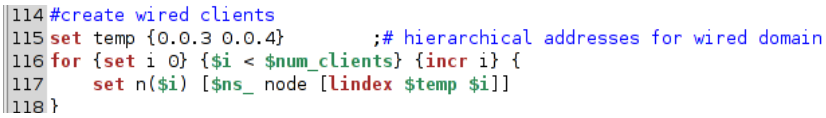
This network introduces wireless networking into the simulation using NS2.

As this network included wireless networking, some research was performed to see how wireless access points are implemented within NS2. The research lead to the following information being found about how to implement a network that contains wired and wireless hosts in NS2 (VINT, 2011).

Network 4 was based on the wireless2.tcl file from the NS2 website (VINT, 2011), by modifying the network file downloaded from the NS website.

Network 4 included 1 server, 1 switch, 1 wireless access point (WAP), 2 wired hosts and 3 wireless hosts, connected together by a gigabit Ethernet connection (for the wired nodes) and a wireless connection (for the wireless nodes).

Creating the network nodes was completed by using the following for statement:

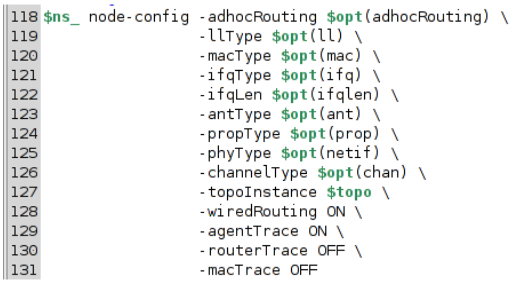


```
#create wired clients
set temp {0.0.3 0.0.4}        ;# hierarchical addresses for wired domain
for {set i 0} {$i < $num_clients} {incr i} {
    set n($i) [$ns_ node [lindex $temp $i]]
}
```

Creating the wired clients for network 4

This statement starts off by assigning the hierarchal addresses for the nodes within the wired network to the temp variable. The number of clients (`num_clients`) is specified at the top of the file in a variable (2 in this case), then the `'$i'` value is incremented by 1 until it hits the number specified in `num_clients`. The node is then assigned its name, along with the number contained in `'$i'`. Finally the hierarchal address is then set for each node.

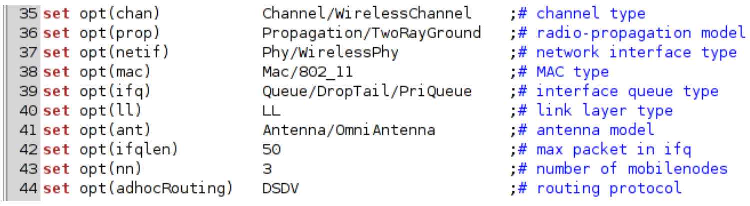
Setting up the configuration of the wireless access point (WAP)/wireless nodes was handled by the following code:



```
$ns_ node-config -adhocRouting $opt(adhocRouting) \
                 -llType $opt(ll) \
                 -macType $opt(mac) \
                 -ifqType $opt(ifq) \
                 -ifqLen $opt(ifqlen) \
                 -antType $opt(ant) \
                 -propType $opt(prop) \
                 -phyType $opt(netif) \
                 -channelType $opt(chan) \
                 -topoInstance $topo \
                 -wiredRouting ON \
                 -agentTrace ON \
                 -routerTrace OFF \
                 -macTrace OFF
```

Setting up the WAP options

All the variables were specified at the top of the file with the following values:



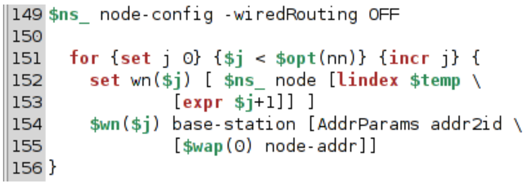
```
set opt(chan)           Channel/WirelessChannel    ;# channel type
set opt(prop)           Propagation/TwoRayGround   ;# radio-propagation model
set opt(netif)          Phy/WirelessPhy            ;# network interface type
set opt(mac)            Mac/802_11                 ;# MAC type
set opt(ifq)            Queue/DropTail/PriQueue    ;# interface queue type
set opt(ll)             LL                         ;# link layer type
set opt(ant)            Antenna/OmniAntenna        ;# antenna model
set opt(ifqlen)         50                         ;# max packet in ifq
set opt(nn)             3                          ;# number of mobilenodes
set opt(adhocRouting)   DSDV                       ;# routing protocol
```

The settings for the wireless network

This section sets out the different settings for the wireless network, such as the type of wireless antenna, the routing protocol used by wireless clients and the number of wireless clients.

The access point was the next item to be created. 5 hierarchal addresses were created to provide an address to the wireless clients and stored under the `temp` variable. The addresses for the wireless network were different to the addresses for the wired network.

Creating and attaching the wireless nodes to the wireless access point used a for loop to create the number of wireless nodes as specified in the `nn` variable at the start of the file, and then to associate the wireless nodes to the wireless access point.



```
$ns_ node-config -wiredRouting OFF

  for {set j 0} {$j < $opt(nn)} {incr j} {
    set wn($j) [ $ns_ node [lindex $temp \
            [expr $j+1]] ]
    $wn($j) base-station [AddrParams addr2id \
            [$wap(0) node-addr]]
}
```

The for loop used to create the wireless nodes and associate nodes with the access point

The links between the wired nodes were the next item to be configured. The configuration of the links was done in the same way as the previous networks built in NS2.

Two FTP connections were then set up to generate network traffic. One of the connections was from a wired client to the server, and the other connection was from a wireless client to the server. These connections were set up in the same way as network 2. When a UDP connection was set up to simulate DNS, NS2 produced a segmentation fault with no other error message indicating where the problem was. Trying other solutions did not work as NS2 produced more unhelpful errors that were hard to understand.

A node movement file was used to hold the details of where the wireless nodes should be placed at certain times during the simulation. This file initially caused some confusion as NS produced an error message about not being able to create links between nodes as the node movement file had not been changed to include the new node names.

Implementing network 4 was a challenge as various errors were encountered that stopped the simulation from running. A few of the errors encountered have been mentioned above.

The error caused by the node names not being updated in the movement file was:

```
ns: [Simulator instance] get-link-head 0 4: can't read "link_(0:4)": no such
element in array
    while executing
"$link_($n1:$n2) head"
    (procedure "_o3" line 3)
    (Simulator get-link-head line 3)
    invoked from within
"[Simulator instance] get-link-head 0 4"
```

In the TCL file for network 4, line 3 is a commented line (if the comments are counted within the line count) or if comments are not counted, then the problematic line was to do with the wireless network interface. The `"$link_($n1:$n2) head"` line gave more clues pointing towards the movement file, as no nodes called $n1 and $n2 existed within the network. However, the node movement file contained the 2 incorrectly named nodes.

As the TCL file for network 4 had so many errors (due to the amount of edits performed), the file was recreated (which managed to get the network working) and both files were compared using the `diff` utility found on most Linux based operating systems.

A small sample of the output from the `diff` utility when comparing the non-working and working network 4 TCL files can be found below:

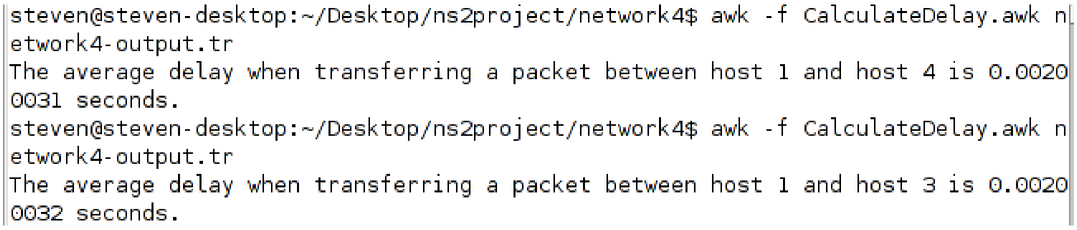
```
< #create wired nodes
< set temp {0.0.0 0.0.1 0.0.2 0.0.3 0.0.4}        ;
< # hierarchical addresses for wired domain  0.1.0
< for {set i 0} {$i < $num_wired_clients} {incr i} {
<     set h($i) [$ns_ node [lindex $temp $i]]
< }
< #creates new server nodes
< # hierarchical addresses for wired domain
< for {set i 0} {$i < $num_wired_servers} {incr i} {
<     set svr($i) [$ns_ node [lindex $temp $i]]
< }
< #creates switches
< # hierarchical addresses for wired domain
< for {set i 0} {$i < $num_switches} {incr i} {
<     set sw($i) [$ns_ node [lindex $temp $i]]
---
> #create server
> set temp {0.0.1}        ;# hierarchical addresses for wired domain
> for {set i 0} {$i < $num_server} {incr i} {
>     set s($i) [$ns_ node [lindex $temp $i]]
> }
>
> #create switch
> set temp {0.0.2}        ;# hierarchical addresses for wired domain
> for {set i 0} {$i < $num_switch} {incr i} {
>     set sw($i) [$ns_ node [lindex $temp $i]]
> }
>
> #create wired clients
> set temp {0.0.3 0.0.4}        ;# hierarchical addresses for wired domain
> for {set i 0} {$i < $num_clients} {incr i} {
>     set n($i) [$ns_ node [lindex $temp $i]]
```

The top half of the sample output is the non-working network, and the bottom half is the working network.

For the first half, all of the hierarchal addresses were assigned on the first for loop but not on each of the other for loops for the servers and switches. This resulted in the array error described earlier on. To solve this problem, unique hierarchal addresses were placed in each for loop for the different devices (e.g. the address 0.0.1 being assigned to the server during the for loop that creates the server node) which allowed the network simulation to take place and complete successfully.

Comparing the difference of the delay between a wired client and a wireless client

Calculating the difference in delay between wired and wireless clients was completed using the AWK script that can be found in Appendix 1.5.2. This script was also adapted from the NS2Ultimate website (Issariyakul, 2011).

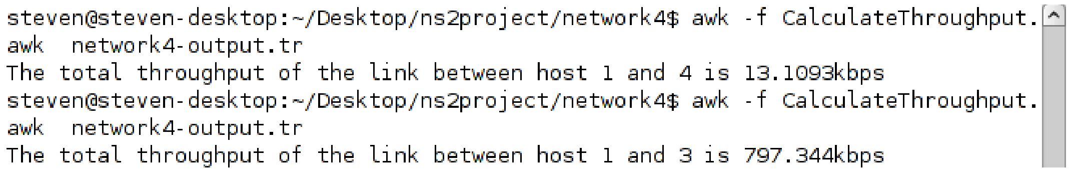


```
steven@steven-desktop:~/Desktop/ns2project/network4$ awk -f CalculateDelay.awk n
etwork4-output.tr
The average delay when transferring a packet between host 1 and host 4 is 0.0020
0031 seconds.
steven@steven-desktop:~/Desktop/ns2project/network4$ awk -f CalculateDelay.awk n
etwork4-output.tr
The average delay when transferring a packet between host 1 and host 3 is 0.0020
0032 seconds.
```

The difference in delay between a wireless and wired client in NS2

As the screenshot above shows, the delay in sending between the wired clients is 0.200032µs seconds, with the wireless clients being slightly quicker at 0.200031µs. This is strange as wireless networking is usually much slower, both in simulations and real life.

A test to calculate the throughput of the links was run on the trace file. A screenshot of the results can be found below. This script was also adapted from the NS2Ultimate website (Issariyakul, 2011).



```
steven@steven-desktop:~/Desktop/ns2project/network4$ awk -f CalculateThroughput.
awk  network4-output.tr
The total throughput of the link between host 1 and 4 is 13.1093kbps
steven@steven-desktop:~/Desktop/ns2project/network4$ awk -f CalculateThroughput.
awk  network4-output.tr
The total throughput of the link between host 1 and 3 is 797.344kbps
```

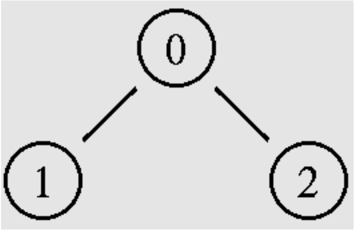
The results obtained from the throughput test.

The wireless link between hosts 1 and 4 averaged at just over 13kbps per second, which is extremely slow, especially when the wired link between hosts 1 & 3 managed an average of just over 797kbps.

8.1.5 Other Features of NS2

Network Animator (NAM)

NAM allows for a network built with NS2 to be visualised in a graphical user interface.



An example of a network displayed in NAM

NAM can display various details of the network, such as:

- The nodes (marked by the circle with the numbers in the above screenshot)
- The links between nodes
- The packets being transferred from host to host
- The queues at hosts within the network

8.1.6 Troubleshooting networks with NS2

NS2 would not be a suitable candidate for troubleshooting networks, primarily due to the complexity of setting up a network. If the network has a large number of hosts, the log files could take up several gigabytes of space to store for analysis. Analysing the log file would have to be done by using a script, which could take time to create (especially if there are multiple statistics that need to be collected).

9.0 Comparing the simulation statistics

This section will compare the statistics produced by each simulator and see how much of a difference there is between the simulators. Packet Tracer and GNS3 will not be included within the comparison as they did not produce any statistics like IT Guru and NS2 did.

9.0.1 Network 1

The statistic that was measured in the first network was the total utilisation of gigabit and fast Ethernet connections. Both simulators indicated that the total utilisation was higher on the fast Ethernet link, when compared with a gigabit link.NS2 produced 33% for the gigabit Ethernet utilisation, and 38% for the Fast Ethernet utilisation. IT Guru provided a peak of 0.000025% for the gigabit LAN utilisation, and 0.000175% for the fast Ethernet network. Although both simulators has the same result (the gigabit LAN having lower utilisation), the statistics produced by the two simulators were very different.

9.0.2 Network 2

For the second network, the monitored statistic was comparing the TCP delay between a fast Ethernet and Gigabit Ethernet link. IT Guru calculated the TCP Delay at just under 0.50µs for a gigabit connection and 0.125µs for a fast Ethernet connection. This shows that there is a difference between the two speeds, but it is unlikely that a human would notice the difference.NS2 put the delay at 0.01µs for a gigabit network and 0.01µs for a fast Ethernet connection, which shows there is little difference between the two speeds according to NS2.

When comparing the two results from NS2 & IT Guru, this shows that there is a considerable difference between the two simulators which is likely related to the higher delay time specified in the network file for NS2.

9.0.3 Network 3

The statistic monitored in the third network was the queuing delay on an Ethernet link. IT Guru initially calculated the queuing delay as being very low (under 0.01µs) which then increased to 0.09µs around 10 seconds into the simulation. The AWK script for NS2 provided a result of 0.66µs. Again, this is a low delay but is still higher than IT Guru’s result. The two different results show that both simulators are not consistent.

9.0.4 Network 4

Comparing the difference in delay between the wired and wireless clients was the monitored statistic for network 4. IT Guru calculated the delay of the wired clients as being between 0.25µs and 17.5µ, whereas the delay from the wireless clients was much higher averaging at between 0.015µs and 0.025µs. IT Guru calculated the throughput to peak at 700kbps and averaged around 500kbps over Ethernet. The wireless statistics were calculated to be around the same as the Ethernet network. NS2 calculated the delay between a wireless and wired host to be 0.200031µs, with the wired network being slightly slower at 0.200032µs. The throughput of the wireless network in NS2 was around 13kbps, with the wired network averaging around 797kbps. The sets of statistics produced by each program vary, showing that, again, there is a considerable difference between the two simulators.

9.1 The difference between the results

The results obtained from the tests above indicate that the simulators each provided different results, of which some were very different and others being similar.

There are many reasons why the statistics produced could differ. Some of the main factors are discussed below. In NS2, the delay has to be set manually, whereas IT Guru did not ask for a delay to be manually set. This alone could be the main reason why the results were not the same in the simulators. The delays in NS2 were higher due to the fact that a higher network delay was set in the TCL file.

Another factor could be that NS2 and IT Guru were run on different operating systems (Windows for IT Guru, and Debian for NS2). As each simulator was built for a different environment, using different programming languages, this could have an effect as different programming languages may do the same thing differently to each other (such as accessing files on a file system).

NS2 requires a user to use a scripting language to find results which could also have a big effect on results as it relies on the calculations within the script to be correct (the calculations being correct also applies to IT Guru as well) to produce the desired result or calculation.

10.0 Comparing the network simulators

This section will compare the simulators against a set of criteria to see which simulator is suited to specific purposes, such as for learning about networks or for testing planned network upgrades before the upgrades go live.

10.1 The criteria that the simulators will be tested against

Each simulator will be tested against the following criteria:

- Ease of Use, including:

  How easy it is to use for a novice and how easy is it to build a simple network within the program.

- Range of equipment supported, including:

  Wired and wireless network support, range of hardware simulated (e.g. switches, servers and routers), and the range of vendor products supported (e.g. Cisco and Juniper).

- Program features, including:

  Supported protocols and logging/reporting features.

Each category will be given 10 points, with the simulators being given marks (out of 10) for each category.

10.2 Comparing the simulators against the criteria

10.2.1 Cisco Packet Tracer

Ease of use

Cisco Packet Tracer was easy to understand and use due to the fact that it had a well-designed user interface that provided quick access to the main features of the program. Building a simple network could be accomplished easily as many of the more basic functions (such as configuring routers with IP addresses and setting up a webserver) have an option within the device settings GUI (graphical user interface) to configure the settings. When using the GUI to set something such as an IP address, a command is given at the bottom of the window that shows the command to perform the same action within the command line interface (CLI).

For most devices (such as PCs, Servers, switches and routers), a CLI was provided to configure settings that were not present in the GUI (such as setting a default gateway on a switch) or to provide utilities to troubleshoot the connection (such as ping).

For a novice user to build a simple network in Packet Tracer, it would be easy as they would need to drag and drop devices onto the workspace, then connect them together using the appropriate type of cable (which could automatically be selected by the program itself) and finally configure the devices using the GUI interface (or CLI if they wish).

Overall, Packet Tracer scores highly in the ease of use category with 9/10.

Range of equipment supported

Packet tracer supports a wide variety of Cisco equipment, such as routers, switches, wireless networking, as well as general equipment such as servers, PCs/Laptops, tablets and printers.

Packet Tracer also supports a variety of different connection types, such as copper cable, serial, console and fibre, to connect network devices together.

Packet tracer, scores highly again, but loses a few marks here as the equipment is based around Cisco (due to the software being designed for use in Cisco certifications). Packet Tracer scores 8/10 in this category.

Program Features

Packet Tracer has several builds available for Windows and Linux (Ubuntu/Fedora based) machine. This means that it can be used on multiple different operating systems.

Packet Tracer supports a wide variety of networking protocols, such as RIP (Routing Information Protocol), DHCP and TCP/UDP. The simulation mode allows a user to see the types of packet (e.g. TCP/UDP or DNS/DHCP) flowing across the network, as well as how the packets are processed at each level of the OSI model at each host the packet passes through.

Logging and reporting functionalities are not currently features of Packet Tracer.

In the functionality section, Packet Tracer scores 8/10.

10.2.2 Graphical Network Simulator 3 (GNS3)

Ease of Use

GNS3 was easy to use, as it had a similar interface to Packet Tracer. Implementing the networks in GNS3 was simple, but initially configuring the virtual machine integration was confusing (e.g. having to have 2 network interfaces (one for GNS3 and the other for the Virtualbox manager) on machines that only needed one interface).

The most challenging part of setting up the networks in GNS3 was configuring the Ubuntu Server virtual machine as a router due to the fact that IPTables needed to be configured to allow packet forwarding. A standard Cisco router was not used due to the lack of a supported Cisco IOS image required to simulate the equipment.

Setting up a network in GNS3 would be more challenging for the novice user (especially if they have no access to a Cisco/Juniper OS image) the network would have to use virtual machines to simulate hosts and network devices

In this category, GNS3 scores 7/10.

Range of equipment supported

GNS3 supports a wide variety of equipment from multiple vendors, such as Cisco and Juniper. Using Cisco and/or Juniper equipment requires the user to source an image of the operating system (e.g. IOS for Cisco or JunOS for Juniper), but it has to be an image that Dynamips (the emulator for GNS3) can support. GNS3 loses a few points as it does not support any wireless networking like other simulators.

GNS3 also has integration with VirtualBox and Qemu to allow virtual machines to connect to the network.

As GNS3 supports a variety of equipment, it scores well in this category, with 8/10.

Program Features

GNS3 has lots of useful features, including integration with virtual machines and connecting virtual networks with physical equipment.

It also has integration with wireshark (a packet sniffer) so that the packets flowing between nodes in a network simulated by GNS3 can be captured and inspected using wireshark.

Like Packet Tracer, GNS3 supports a variety of networking protocols that can be used within a virtual network built within GNS3.

Again, GNS3 scores highly due to the range of features (such as the VM integration) it offers. The score for this category for GNS3 is 9/10.

10.2.3 OPNET IT Guru Academic Edition

Ease of Use

The user interface of IT Guru was very confusing at first, as one of the initial wizard steps asks the user to select the types of technologies for use within the network. As there are over 30 different technologies, this can become confusing for a novice user (especially as some have an advanced category as well). After the wizard has been completed, it then allows for devices to be placed onto the network topology and to be connected together.

One of the harder parts of IT Guru was setting up the configuration of the nodes due to the fact the IT Guru presents the user with a huge number of configuration options, of which some have confusing names.

Setting up the applications and services that run during the simulation was another confusing part of IT Guru. Setting the application definition could be accomplished by choosing a pre-set setting or providing a custom value. Configuring the profile definition was easier as it just needed a name, start time, duration and how often it should be repeated to be set.

Setting the statistics to be collected during the simulation was easy, but IT Guru provided lots of statistics (of which some were very similar, such as the Background traffic delay having a separate entry for outgoing and incoming traffic).

Getting the results from the simulation was fairly easy, but sometimes the initial graphs shown in the preview window were not the most suitable to show the results.

Building a simple network could be very difficult in IT Guru, especially if the user hasn’t attempted to follow one of the tutorials written by OPNET which provide a good insight on how to set up simulations.

In the ease of use category, IT Guru did get a reasonable score of 6/10 as the interface was initially tricky to use without consulting one of the tutorial sheets.

Types of equipment supported

IT Guru supports a very wide range of equipment (from Cisco, 3Com and HP to Ethernet, ATM and wireless networks) for use within simulations. However, the academic edition has limited wireless networking features compared to the commercially available version.

If a specific device is not provided by OPNET, then the user has the option to create a custom device using the device creator function.

IT Guru scored very well in this category due to the wide range of supported equipment. The score for IT Guru in this category is 9/10.

Program Features

IT Guru supports collecting a wide variety of statistics from any of the network nodes during a simulation to help analyse the performance of the network and the individual nodes.

Just like the wireless networking features, the academic edition of IT Guru has limited tools to help import and export network models.

IT Guru also has the Application Characterisation Environment (ACE) to collect data from a real network to be used within a network simulation.

Protocol support is good in IT Guru as it supports lots of the main protocols such as FTP and HTTP (but it did not support some protocols such as DHCP which had to be substituted by other services/protocols in some of the networks).

In the program features category, IT Guru scored well, with 7/10, as it did not support some of the protocols like DHCP that were specified in the network specifications.

10.2.3 Network Simulator 2

Ease of use

NS2 was incredibly difficult to get working as there is no graphical user interface to help create a network. Creating a network in NS2 relies on the user knowing how to create a network file in a text editor, and knowing how to use the terminal (or command prompt on Windows machines) to execute the simulation in NS2.

Creating and linking the nodes to each other can be hard, especially if something such as a for loop was used to create the nodes. Setting up data transfers can also be complex as it requires a user to attach an agent to the sender, a traffic sink to the receiver and set the type of transfer (TCP/UDP) then attach the application protocol to the TCP/UDP connection.

Networks with wired nodes are much easier to set up than scenarios with both wired and wireless devices

Fixing any errors within a network's TCL file takes trial and error as some of the error messages given by NS2 can be cryptic and not contain any useful information (such as line numbers) to help pinpoint the problem.

For a novice user to build a network from scratch in NS2 would be very hard. Most novice users start off by modifying existing NS2 network files to suit their scenario, and then progress onto building networks from scratch.

NS2 did not score well in the ease of use category, due to the fact that it is incredibly hard to get a network working properly. NS2 scored only 3/10 in this category.

Types of equipment supported

NS2 can simulate a variety of different networking equipment, such as routers and wireless networks. However, it has no vendor specific equipment, so just simulates standard networking equipment like computers and generic routers.

NS2 scored better in this category, with 5/10.

Program Features

To get statistics from NS2, the easiest way would be to analyse the trace file using a script (such as an AWK script to calculate the delay). However, for someone new to NS2 and scripting, this could present a huge challenge as the user will first have to get to grips with NS2, and then start to learn about scripting in AWK or ruby (for example).

NS2’s protocol support is good, but like IT Guru, some of the services such as DHCP and DNS had to be substituted by other protocols (e.g. DHCP being substituted by FTP).

NAM allows for networks to be presented to a user as a diagram as well as letting the user see the packets that flow over the network. One downside of NAM is that you have to create the network by hand, then run NS to output the trace and NAM files before the network can be loaded into NAM.

NS2 did not do particularly well in this category either, with it only getting 5/10 due to the complexity of getting statistics and the fact that it didn’t support protocols like DHCP and DNS (as required by the network specification).

10.3 Results

Out of all the simulators, Packet Tracer scored the highest with 25/30, followed closely behind by GNS3 with 24/30. IT Guru scored highly with 22/30, with NS2 being the worst of the 4 simulators with only 13/30.

Ultimately, the best simulator depends on the task at hand, but for this scenario Packet Tracer is the winner as it provided the best all round experience. GNS3 comes a close second as it provided a similar experience to Packet Tracer. IT Guru and NS2 were suitable for this scenario, but they were slightly harder and didn’t support protocols (e.g. DHCP) as needed in the networks specified in the network specifications.

11.0 Discussion & Evaluation

11.1 Achievements

In this project, all the aims (listed in chapter 1.2) were met. Some of the aims were harder to meet due to the fact that some of the programs lacked features (e.g. the aim of securing the networks was done where possible, like in the packet tracer wireless network simulation) and was not helped by the fact that certain parts of the project overran the allocated time (see chapter 11.2).

Working on this project has allowed me to gain experience with using different network simulators and how challenging some programs can be to get a network simulation running. Whilst implementing the networks, I faced a variety of problems with the programs. With GNS3, it was the fact that each virtual machine has to have two network interfaces (one for the Virtualbox console, and the other for GNS3) that initially confused me, but it was easy to fix.

With IT Guru, I found the interface confusing and had to use one of the tutorials to get the networks set up. I also had an issue where a statistic was gathered, even though it was unselected in the window where the monitored statistics are shown.

With NS2, I found it incredibly complex to get a network working successfully. During the course of implementing the networks, I encountered lots of errors (detailed in Chapter 8.1) including segmentation faults.

Each of the problems encountered was successfully overcome, either by starting fresh or debugging the error messages to reveal where the problem lied.

I have learned about the AWK scripting language and how it can be used to analyse files (in this case, NS2 trace files) to get statistics quickly from large log files. I started by using small AWK scripts to calculate the total number of lines starting with ‘r’, and then moved onto creating/editing scripts that performed more complicated tasks like calculating the queuing delay between hosts. Other scripts were successfully modified from performing one action to gathering the statistics from NS2 that were also gathered using IT Guru.

Using four network simulators also showed me that different simulators are made for different purposes (e.g. packet tracer for teaching about Cisco) and how complex the simulators can get. Real networks can also be very complex, so having the simulators being able to simulate complex networks would be beneficial to network administrators that manage large, complex networks.

11.2 Time Management

My time management during this project was well managed, although some parts overran the allocated time scales and other sections needed less time than allocated.

Writing the sections on project introduction, the simulators used, and the other programs investigated were all completed ahead of schedule (as 4 weeks were allocated, but only 2 were used to complete the section).

Implementing the networks and documenting the process took the longest amount of time. Only 5 weeks were allocated at the start, but overran to 9 weeks due to being unfamiliar with some of the simulators and also resolving any problems encountered during the implementation phase. Due to the overrun, the section on security features of the simulators was limited to simple items such as securing wireless connections with a password (where possible). The section on looking into virtualisation products for network simulations was limited to Virtualbox due to the time constraints and the fact that GNS3 required Virtualbox to simulate hosts to connect to a network (as a Cisco IOS image was not available).

Comparing the programs, writing the section on recommendations and writing the conclusion were allocated 6 weeks, but only needed 1 week to complete as I overestimated the time needed to complete these sections.

Even though some parts of the project were completed within the time scales and others needed more or less time to complete, the project was completed within the overall time allocated.

11.3 Future Work

Looking into how hypervisors/virtual machines could be used to simulate networks and expanding on network security would be two possible tasks for future work to be carried out on the project.

11.4 Conclusion

This project has shown that there are a variety of different network simulators available for different purposes, such as for learning about networks or for testing out planned network upgrades. It also shows that people can learn about configuring/implementing networks without having to purchase real networking equipment that could become expensive depending on the equipment required.

Using the different simulators showed that the each simulator was built for different purposes, such as packet tracer being built for education and IT Guru being built for performance monitoring and network planning. For this scenario, Packet Tracer was the most suitable package as it provided a good set of features with an easy to use user interface.

Bibliography

Aboelela, E., 2012. Lab 09: Mobile WLAN. In: R. Adams, ed. *Network Simulation Experiments Manual.* Burlington: Elsvier, p. 183.

Altman, E., 2003. *NS Simulator for beginners.* [Online]
Available at: http://www-sop.inria.fr/members/Eitan.Altman/COURS-NS/n3.pdf
[Accessed 12 March 2012].

Cisco Systems, OPNET, 2005. *ACE User Guide for IT Guru.* [Online]
Available at:
http://www.cisco.com/en/US/docs/net_mgmt/application_analysis_solution/1.1/tutorials_and_examples/ace/aasace.pdf
[Accessed 6 March 2012].

Cisco Systems, 2010. *Cisco Packet Tracer Datasheet.* [Online]
Available at:
http://www.cisco.com/web/learning/netacad/course_catalog/docs/Cisco_PacketTracer_DS.pdf
[Accessed 23 January 2012].

Cisco Systems, 2011. *Cisco Guide to Harden Cisco IOS Devices.* [Online]
Available at:
http://www.cisco.com/en/US/tech/tk648/tk361/technologies_tech_note09186a0080120f48.shtml
[Accessed 27 January 2012].

Convery, S., 2004. *General Design Considerations for Secure Networks.* [Online]
Available at: http://www.ciscopress.com/articles/article.asp?p=174313
[Accessed 27 January 2012].

Cornell University Law School, 2012. *18 USC § 1030 - Fraud and related activity in connection with computers.* [Online]
Available at: http://www.law.cornell.edu/uscode/text/18/1030
[Accessed 5 April 2012].

Davis, D., 2008. *Five things you should know about configuring a Cisco IOS switch.* [Online]
Available at: http://www.techrepublic.com/blog/networking/five-things-you-should-know-about-configuring-a-cisco-ios-switch/428
[Accessed 2 February 2012].

Dunaytsev, R., 2010. *Network Simulators: OPNET Overview and Examples.* [Online]
Available at: http://www.cs.tut.fi/kurssit/TLT-2707/lecture12.pdf
[Accessed 25 January 2012].

Espiner, T., 2007. *Burglars plunder Verizon's London data center.* [Online]
Available at: http://www.zdnetasia.com/burglars-plunder-verizons-london-data-center-62035440.htm
[Accessed 27 January 2012].

Fall, K. & Varadhan, K., 2011. *The ns Manual.* [Online]
Available at: http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf
[Accessed 12 March 2012].

Fewer, S., 2007. *ARP Poisoning.* [Online]
Available at: www.harmonysecurity.com/files/HS-P004_ARPPoisoning.pdf
[Accessed 3 February 2012].

GNS Project, 2007. *What is GNS3?.* [Online]
Available at: http://www.gns3.net/
[Accessed 23 January 2012].

GNS3 Project, 2007. *Packet Capture.* [Online]
Available at: http://www.gns3.net/gns3-packet-capture/
[Accessed 10 February 2012].

GNS3 Project, n.d. *Packet Capture.* [Online]
Available at: http://www.gns3.net/gns3-packet-capture/
[Accessed 10 February 2012].

GNS3, 2007. *Connecting GNS3 to Real Networks.* [Online]
Available at: http://www.gns3.net/gns3-connecting-real-networks/
[Accessed 12 February 2012].

Google, 2012. *Basic Guide to DNS.* [Online]
Available at: http://support.google.com/a/bin/answer.py?hl=en&hlrm=en&answer=48090#H
[Accessed 3 February 2012].

Graphical Network Simulator Project, 2007. *GNS3.* [Online]
Available at: www.gns3.net
[Accessed 1 February 2012].

Henderson, T., 2011. *10.5 Examples: Tcp, TCP Sink Agents.* [Online]
Available at: http://www.isi.edu/nsnam/ns/doc/node107.html
[Accessed 8 March 2012].

hesam, 2010. *calculating link utilization using ns2.* [Online]
Available at: http://stackoverflow.com/questions/4212756/calculating-link-utilization-using-ns2
[Accessed 16 March 2012].

Issariyakul, T., 2011. *Post processing NS2 Result using NS2 Trace — Ex1 [Link throughput calculation].* [Online]
Available at: http://www.ns2ultimate.com/post/3442965938/post-processing-ns2-result-using-ns2-trace-ex1-link
[Accessed 17 March 2012].

Issariyakul, T., 2011. *Post processing NS2 Result using NS2 Trace — Ex3 [Average delay calculation].* [Online]
Available at: http://www.ns2ultimate.com/post/5240359082/post-processing-ns2-result-

using-ns2-trace-ex3
[Accessed 17 March 2012].

IWS, 2000. *Recommendations for the Protection against Distributed Denial-of-Service Attacks in the Internet.* [Online]
Available at: http://www.iwar.org.uk/comsec/resources/dos/ddos_en.htm
[Accessed 3 February 2012].

Kelion, L., 2012. *Hackers retaliate over Megaupload website shutdown.* [Online]
Available at: http://www.bbc.co.uk/news/technology-16646023
[Accessed 27 January 2012].

King, J. & Lauerman, K., 2010. *ARP Poisoning Attack and Mitigation Techniques.* [Online]
Available at:
http://www.cisco.com/en/US/prod/collateral/switches/ps5718/ps708/white_paper_c11_603839.html
[Accessed 27 January 2012].

Legislation.gov.uk, 2012. *Computer Misuse Act.* [Online]
Available at: http://www.legislation.gov.uk/ukpga/1990/18/data.pdf
[Accessed 5 April 2012].

ManojKumar.A, 2010. *Manual interpretation of ns2 trace file.* [Online]
Available at: https://getch.wordpress.com/2010/11/20/manual-interpretation-of-ns2-trace-file/
[Accessed 12 March 2012].

McDowell, M., 2004. *Understanding Denial-of-Service Attacks.* [Online]
Available at: http://www.us-cert.gov/cas/tips/ST04-015.html
[Accessed 27th January 2012].

Microsoft Help & Support, 2007. *How to prevent DNS cache pollution.* [Online]
Available at: http://support.microsoft.com/kb/241352
[Accessed 3 February 2012].

Microsoft Help and Support, 2007. *The Structure of a DNS SOA Record.* [Online]
Available at: http://support.microsoft.com/kb/163971
[Accessed 2 February 2012].

Microsoft, 2009. *Understanding networking with Hyper-V.* [Online]
Available at: http://www.microsoft.com/download/en/details.aspx?id=9843
[Accessed 23 January 2012].

National Security Agency, 2011. *Best Practices for Keeping Your Home Network Secure.* [Online]
Available at: http://www.nsa.gov/ia/_files/factsheets/Best_Practices_Datasheets.pdf
[Accessed 24 January 2012].

NIST, 1998. *How to Secure a Domain Name Server (DNS).* [Online]
Available at:
http://csrc.nist.gov/groups/SMA/fasp/documents/network_security/NISTSecuringDNS/NIST

SecuringDNS.htm
[Accessed 3 February 2012].

OPNET Technologies, 2012. *Network Planning.* [Online]
Available at:
http://www.opnet.com/solutions/network_performance/itguru_network_planner/index.html
[Accessed 6 April 2012].

Oracle, 2012. *Oracle VM Virtualbox.* [Online]
Available at: https://www.virtualbox.org/
[Accessed 27 January 2012].

Perry, R. S., 2011. *Oversimplified DNS - Start of Authority.* [Online]
Available at: http://rscott.org/dns/soa.html
[Accessed 3 February 2012].

Peterson, L. & Davie, B., 2012. In: *Computer Networks: A Systems Approach.* London: Morgan Kaufmann, p. 884.

Sanders, C., 2010. *Understanding Man-in-the-Middle Attacks – ARP Cache Poisoning (Part 1).* [Online]
Available at: http://www.windowsecurity.com/articles/Understanding-Man-in-the-Middle-Attacks-ARP-Part1.html
[Accessed 27 January 2012].

Sanders, C., 2010. *Understanding Man-In-The-Middle Attacks – Part2: DNS Spoofing.* [Online]
Available at: http://www.windowsecurity.com/articles/Understanding-Man-in-the-Middle-Attacks-ARP-Part2.html
[Accessed 27 January 2012].

SANS, 2009. *Top Cyber Security Risks - Executive Summary.* [Online]
Available at: http://www.sans.org/top-cyber-security-risks/summary.php
[Accessed 27 January 2012].

Shavit, E., 2003. *TechDHCP.* [Online]
Available at: http://webee.technion.ac.il/labs/comnet/projects/winter02/cn16w02/about.htm
[Accessed 26 February 2012].

Stanford University, 2001. *Basic Network Security Threats.* [Online]
Available at: http://www.scs.stanford.edu/nyu/01fa/notes/l13.pdf
[Accessed 26 January 2012].

Stewart, K., Adams, A., Reid, A. & Lorenz, J., 2008. Chapter 1: Introducing Network Design Concepts. In: *Designing and Supporting Computer Networks, CCNA Discovery Learning Guide.* s.l.:Cisco Press, p. 864.

UbuntuForums, 2008. *Howto: Set up Ubuntu as a firewall/gateway router with webmin.* [Online]
Available at: http://ubuntuforums.org/showthread.php?t=926001
[Accessed 6 February 2012].

University of Maribor, 2006. *Lab 7: Troubleshooting and Predicting the Performance of an Oracle Application.* [Online]
Available at: http://www.sparc.uni-mb.si/VrednotenjetksUNI/vaje/07%20ACE%20Oracle%202%20Tier.pdf
[Accessed 6 March 2012].

University of Southern California, 2011. *The Network Simulator - ns-2.* [Online]
Available at: http://www.isi.edu/nsnam/ns/
[Accessed 23 January 2012].

University of Tennessee, 2008. *Secure Network Infrastructure Best Practice.* [Online]
Available at: http://security.tennessee.edu/pdfs/SNIBP.pdf
[Accessed 24 January 2012].

VINT, 2011. *X. Creating Wired-cum-Wireless and MobileIP Simulations in ns.* [Online]
Available at: http://www.isi.edu/nsnam/ns/tutorial/nsscript6.html
[Accessed 29 March 2012].

VMware White Paper, 2011. *The Importance of Patching Non-Microsoft Applications.* [Online]
Available at: http://www.vmware.com/files/pdf/solutions/VMware-Importance-of-Patching-Non-Microsoft-Applications-WP-EN.pdf
[Accessed 27 January 2012].

VMware, 2011. *ESXi Configuration Guide - ESXi 4.1.* [Online]
Available at: http://www.vmware.com/pdf/vsphere4/r41/vsp_41_esxi_server_config.pdf
[Accessed 23 January 2012].

Wang, Jianping, Virginia University, 2004. *ns-2 Tutorial (1).* [Online]
Available at: www.cs.virginia.edu/~cs757/slidespdf/cs757-ns2-tutorial1.pdf
[Accessed 5 March 2012].

Wang, J., 2004. *ns-2 Tutorial Exercise.* [Online]
Available at: http://www.cs.virginia.edu/~cs757/slidespdf/cs757-ns2-tutorial-exercise.pdf
[Accessed 5 March 2012].

Webmin, 2011. *Webmin.* [Online]
Available at: http://webmin.com/
[Accessed 6 February 2012].

Xiao, H., 2011. *7COM0184 Wireless Mobile & Ad Hoc Networking: ns2 Practical (part 1).* [Online]
Available at: http://www.studynet2.herts.ac.uk/crs/11/7COM0184-0901.nsf/Homepage?ReadForm
[Accessed 22 February 2012].

Zacker, C., 2006. *Academic Learning Series: Network+ Certifcation.* 4th ed. Redmond: Microsoft Press.

Appendix

Appendix 1.1: Packet Tracer network 1 router configuration

The following code was automatically generated by Cisco Packet Tracer (with input prompts being entered by hand).

```
         --- System Configuration Dialog ---

Continue with configuration dialog? [yes/no]: yes

At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.

Basic management setup configures only enough connectivity
for management of the system, extended setup will ask you
to configure each interface on the system

Would you like to enter basic management setup? [yes/no]: yes
Configuring global parameters:

  Enter host name [Router]: NetworkRouter

  The enable secret is a password used to protect access to
  privileged EXEC and configuration modes. This password, after
  entered, becomes encrypted in the configuration.
  Enter enable secret: 86TsiJnr

  The enable password is used when you do not specify an
  enable secret password, with some older software versions, and
  some boot images.
  Enter enable password: hSs0DYJp

  The virtual terminal password is used to protect
  access to the router over a network interface.
  Enter virtual terminal password: oZKfJZXI
Configure SNMP Network Management? [no]:n

Current interface summary

Interface              IP-Address      OK? Method Status                Protocol

FastEthernet0/0        unassigned      YES manual administratively down down

FastEthernet0/1        unassigned      YES manual administratively down down

Vlan1                  unassigned      YES manual administratively down down

Enter interface name used to connect to the
management network from the above interface summary: FastEthernet0/0

Configuring interface FastEthernet0/0:
  Configure IP on this interface? [yes]:
    IP address for this interface: 192.168.100.1
    Subnet mask for this interface [255.255.255.0] :

The following configuration command script was created:

!
hostname NetworkRouter
enable secret 5 $1$mERr$uNleVDUcMTRbe0BARakNs/
enable password hSs0DYJp
line vty 0 4
password oZKfJZXI
!
interface Vlan1
 shutdown
 no ip address
!
interface FastEthernet0/0
 no shutdown
 ip address 192.168.100.1 255.255.255.0
```

```
!
interface FastEthernet0/1
 shutdown
 no ip address
!
end

[0] Go to the IOS command prompt without saving this config.
[1] Return back to the setup without saving this config.
[2] Save this configuration to nvram and exit.

Enter your selection [2]:
Building configuration...

%LINK-5-CHANGED: Interface Vlan1, changed state to up

%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up

%LINK-5-CHANGED: Interface Vlan1, changed state to administratively down

%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to
down

%SYS-5-CONFIG_I: Configured from console by console
[OK]
Use the enabled mode 'configure' command to modify this configuration.

Press RETURN to get started!

NetworkRouter>enable
Password:
NetworkRouter#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
NetworkRouter(config)#interface FastEthernet0/1
NetworkRouter(config-if)#ip addr 192.168.200.254 255.255.255.0
NetworkRouter(config-if)#no shut

%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up

NetworkRouter(config-if)#^Z
NetworkRouter#
%SYS-5-CONFIG_I: Configured from console by console
^Z
NetworkRouter#show interface fastethernet0/1
FastEthernet0/1 is up, line protocol is up (connected)
  Hardware is Lance, address is 00e0.a3b6.6602 (bia 00e0.a3b6.6602)
  Internet address is 192.168.200.254/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  ARP type: ARPA, ARP Timeout 04:00:00,
  Last input 00:00:08, output 00:00:05, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0 (size/max/drops); Total output drops: 0
  Queueing strategy: fifo
  Output queue :0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 input packets with dribble condition detected
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
NetworkRouter#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
NetworkRouter#exit

NetworkRouter con0 is now available
Press RETURN to get started.
```

Appendix 1.2: 2nd Network Router and Switch Configuration

The following code in appendix 1.2 was automatically generated by Cisco Packet Tracer (with input prompts being entered by hand).

Router

```
         --- System Configuration Dialog ---

Continue with configuration dialog? [yes/no]: yes

At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.

Basic management setup configures only enough connectivity
for management of the system, extended setup will ask you
to configure each interface on the system

Would you like to enter basic management setup? [yes/no]: yes
Configuring global parameters:

  Enter host name [Router]: Network2Router

  Enter enable secret: 86TsiJnr

  Enter enable password: hSs0DYJZXI

  The virtual terminal password is used to protect
  access to the router over a network interface.
  Enter virtual terminal password: oZKfJZXI
Configure SNMP Network Management? [no]:

Current interface summary

Interface              IP-Address      OK? Method Status                Protocol

FastEthernet0/0        unassigned      YES manual administratively down down

FastEthernet0/1        unassigned      YES manual administratively down down

Vlan1                  unassigned      YES manual administratively down down

Enter interface name used to connect to the
management network from the above interface summary: FastEthernet0/0

Configuring interface FastEthernet0/0:
  Configure IP on this interface? [yes]:
    IP address for this interface: 192.168.1.1
    Subnet mask for this interface [255.255.255.0] :

The following configuration command script was created:

!
hostname Network2Router
enable secret 5 $1$mERr$uNleVDUcMTRbe0BARakNs/
enable password hSs0DYJZXI
line vty 0 4
password oZKfJZXI
!
interface Vlan1
 shutdown
 no ip address
!
interface FastEthernet0/0
 no shutdown
 ip address 192.168.1.1 255.255.255.0
!
interface FastEthernet0/1
 shutdown
 no ip address
!
end

[0] Go to the IOS command prompt without saving this config.
[1] Return back to the setup without saving this config.
[2] Save this configuration to nvram and exit.

Enter your selection [2]: 2
Building configuration...

%LINK-5-CHANGED: Interface Vlan1, changed state to up

%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up

%LINK-5-CHANGED: Interface Vlan1, changed state to administratively down

%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to administratively down

%SYS-5-CONFIG_I: Configured from console by console
[OK]
Use the enabled mode 'configure' command to modify this configuration.

Press RETURN to get started!

Network2Router>enable
Password:
Network2Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Network2Router(config)#interface FastEthernet0/1
Network2Router(config-if)#ip addr 10.0.0.1 255.0.0.0
Network2Router(config-if)#no shut

%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up

Network2Router(config-if)#^Z
Network2Router#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Network2Router#exit
```

Switches

Although 2 switches are used in network 2, only one configuration log can be found here as all that differed were the IP addresses.

```
Switch>enable
Switch#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#enable password SQk6rD5Q
Switch(config)#interface vlan1
Switch(config-if)#ip address 192.168.1.2 255.255.255.0
Switch(config-if)#ip default-gateway 192.168.1.1
Switch(config-if)#no shut
Switch(config)#exit
Switch#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Switch#
%SYS-5-CONFIG_I: Configured from console by console
Switch#exit
```

Appendix 1.3: Network 3 & Network 4 Switch configuration

The following code was automatically generated by Cisco Packet Tracer (with input prompts being entered by hand).

```
Switch>enable
Switch#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#interface vlan1
Switch(config-if)#ip addr 192.168.100.254 255.255.255.0
Switch(config-if)#no shut
%LINK-5-CHANGED: Interface Vlan1, changed state to up

Switch(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to up

Switch(config-if)#exit
Switch(config)#exit
Switch#exit
```

Appendix 1.4: Ubuntu/Xubuntu Installation Settings

Ubuntu Server was installed from the 32bit Ubuntu 10.04.3 LTS Server CD, using the minimal virtual machine mode. Mostly the default settings were used for installation as the Language and system locale was changed from United States to United Kingdom. The Username and password were configured as ‘user’ and ‘password’.

The packages installed during the installation of Ubuntu Server were the ‘Basic Ubuntu Server’, ‘OpenSSH Server’ and ‘DNS Server’ Packages. After the installation had completed, the ‘build-essential’, kernel headers, apache2, Virtualbox Guest additions and webmin packages were installed.

The Xubuntu Clients were installed using the default settings (as per the Ubuntu server installations) with no extra packages other than the kernel headers required for the Virtualbox guest additions. The username and password are the same as the Ubuntu Server virtual machine.

Appendix 1.5: NS2 AWK scripts

Any italic text contained in appendix 1.5 was created by another programmer. A reference to the code will be provided.

Appendix 1.5.1: Script to calculate the network utilisation

This AWK script was used to calculate the network utilisation. The text in bold was found in the following reference (hesam, 2010)

```
#Edited by Steven Collings (UH ID: 12002053)
#Script Calculates Link Utilization
#Set up for Network 1
BEGIN {
        #node that sent packet
        SendingNode=1;

        #node that received packet
        ReceivingNode=0;

        #variable to hold total number of bytes received
        TotalRecBytes=0;
        #variable to hold the total simulation time. NOTE: MUST BE FILLED IN BEFORE
SCRIPT IS RUN OTHERWISE SCRIPT WILL NOT WORK PROPERLY
        TotalSimTime=30;

        #Variable to hold the value of the link capacity (e.g. 100mb/s or 1000mb/s
links). NOTE: MUST ALSO BE SET BEFORE SCRIPT IS RUN
        LinkBW=1000;

        #Variable to hold the link utilization
        LinkUtil=0;
}

#Perform pattern matching to select relevant info from trace file
#selects column 3 (the sending node) and the 4th column (the destination node) and
calculates the total bytes received
/^\+/&&$3==SendingNode&&$4==ReceivingNode {

        if ($3 == 1 && $4 == 0)
          {
           #adds up the total of the total received bytes
            TotalRecBytes=TotalRecBytes+$6;
           }
};

END {
        #calculates the link utilization
        #edited to add divide by 100 on end of calculation
        LinkUtil=TotalRecBytes*8/TotalSimTime/LinkBW/1000;

        #Prints out the results to the terminal
        print "The link utilization is: " LinkUtil "%";
}
```

Appendix 1.5.2: Script to calculate the delay

The second script is to calculate the delay. The text in italic font was found in the following reference (Issariyakul, 2011)

```
#Edited by Steven Collings (UH ID: 12002053)
#Set up for Network 1
BEGIN {
        #node that sent packet
        SendingNode=0;

        #node that received packet
        ReceivingNode=2;

        #variable to hold total number of records
        TotalRec=0;
        #variable to hold the running total of the delay
        TotalDelay=0;
}

#Perform pattern matching to select relevant info from trace file
#selects column 3 (the sending node) and the 4th column (the destination node) and
logs the time and id of the packet in the PacketDetails variable
/^\+/&&$3==SendingNode&&$4==ReceivingNode {
        PacketDetails[$12] = $2;
};

# searches for lines that begin with r (received) between the sender/receiver
/^r/&&$3==SendingNode&&$4==ReceivingNode {
        # if the packet id is over 0, then the packet is added to the statistics
        if (PacketDetails[$12] > 0)
           {
             #increments the TotalRec variable by 1
             TotalRec++;

             #Calculates the delay
             Delay = $2 - PacketDetails[$12];
             TotalDelay += Delay;
            };
};

END {
        #Calculates the average delay on the network
        AverageDelay = TotalDelay / TotalRec

        #Prints out the results to the terminal
        print "The average delay when transferring a packet between host " SendingNode
" and host " ReceivingNode " is " AverageDelay " seconds.";
}
```

Appendix 1.5.3: Script to calculate the throughput

The second script is to calculate the delay. The text in italic font was found in the following reference (Issariyakul, 2011).

```
#Edited By Steven Collings (UH ID: 12002053)
#script to calculate the throughput on a network
BEGIN {

#sets the sending and receiving nodes
SendingNode=1;
ReceivingNode=4;

#sets the simulation time
TotalSimTime = 30;

#sets the TotalBitsTransferred to 0
TotalBitsTransferred = 0;
}

# Matches lines starting with r, and also the hosts that match the numbers stated in
sendingnode/receivingnode
/^r/&&$3==SendingNode&&$4==ReceivingNode {

#adds the TotalBitsTransferred up and multiplies by 8
    TotalBitsTransferred += 8*$6;


};
END{
# prints the total throughput of the network to the terminal
print "The total throughput of the link between host " SendingNode " and "
ReceivingNode " is " TotalBitsTransferred/TotalSimTime/1e3 "kbps" ;
```

Appendix 1.5.4: Script to calculate the queuing delay

This script calculates the average delay between two hosts on a network. The regular expressions here are modified from the following reference (Issariyakul, 2011).

```
#Created by Steven Collings (UH ID: 12002053)
#Script calculates the average queuing delay between two hosts
#Set up for Network 3
BEGIN {
        #node that sent packet
        SendingNode=0;

        #node that received packet
        ReceivingNode=1;

        #variable to hold total number of records
        TotalRec=0;
        #variable to hold the running total of the queuing delay
        QueuingDelay=0;
}

#Perform pattern matching to select relevant info from trace file
#selects column 3 (the sending node) and the 4th column (the destination node) and
logs the time and id of the packet in the PacketDetails variable
/^\+/&&$3==SendingNode&&$4==ReceivingNode {
        PacketDetails[$12] = $2;
};

# searches for lines that begin with + (packet Added to the queue) between the
sender/receiver
/^/+//&&$3==SendingNode&&$4==ReceivingNode {
        # if the packet id is over 0, then the packet is added to the statistics
        if (PacketDetails[$12] > 0)
           {
             #increments the TotalRec variable by 1
             TotalRec++;

             #Calculates the queuing delay
             Delay = $2 - PacketDetails[$12];
             QueuingDelay += Delay;
            };
};

# searches for lines that begin with - (packet removed from the queue) between the
sender/receiver
/^/-//&&$3==SendingNode&&$4==ReceivingNode {
        # if the packet id is over 0, then the packet is added to the statistics
        if (PacketDetails[$12] > 0)
           {
             #increments the TotalRec variable by 1
             TotalRec++;

             #Calculates the queuing delay
             Delay = $2 - PacketDetails[$12];
             QueuingDelay += Delay;
            };
};

END {
        #Calculates the average queuing delay between the 2 hosts
        AvgQueuingDelay = QueuingDelay / TotalRec *100

        #Prints out the results to the terminal
        print "The average queuing delay when transferring a packet between host "
SendingNode " and host " ReceivingNode " is " AvgQueuingDelay " µs.";
}


};
```