University of Southern Queensland

Faculty of Engineering & Surveying

Security Performance of a Wireless LAN Testbed

A dissertation submitted by

Kamarul Faisal Kamaruddin

in fulfilment of the requirements of

ENG4112 Research Project

towards the degree of

Bachelor of Engineering (Computer Systems Engineering)

Submitted: December, 2006

Abstract

Wireless technology is very popular and ever growing because of its convenience. Many enterprises such as the University of Southern Queensland (USQ) are integrating Wireless into their network infrastructure. However, the performance and security of Wireless Local Area Network (WLAN) is a major concern. Wireless devices with different standards would perform differently with each other. Furthermore, different security implementations would have different effects on the network performance. Current Wireless security protocols may provide adequate protection but they may also deteriorate the performance of the network.

In conjunction, Wireless networking is constantly being researched and tested for better performance and security. As evidence, the developments and integration of various Wireless security standards to date. They would complement the emergence of high performing Wireless standards in the future.

University of Southern Queensland

Faculty of Engineering and Surveying

ENG4111/2 *Research Project*

Limitations of Use

The Council of the University of Southern Queensland, its Faculty of Engineering and Surveying, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its Faculty of Engineering and Surveying or the staff of the University of Southern Queensland.

This dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course pair entitled "Research Project" is to contribute to the overall education within the student's chosen degree program. This document, the associated hardware, software, drawings, and other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.

Prof R Smith

Dean

Faculty of Engineering and Surveying

Certification of Dissertation

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.
I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Kamarul Faisal Kamaruddin

0031131598

_____

Signature

_____

Date

Acknowledgments

I would like to express my utmost gratitude to Dr. Hong Zhou for her support and guidance throughout this project. Her assistance have made the completion of this research project possible. I would like to extend this credit to all my educators in university for their advice has made me a competent student.

The comfort and loyalty from my many friends have helped me endure the tough moments in university life especially while accomplishing this research. It is an invaluable memory that will always be remembered.

On a personal note, I would like to sincerely thank my parents, Mr. Kamaruddin Che Lah and Mrs. Fauziah Shaik Emam, who have been very understanding and encouraging towards my success. My gratefulness also stretches to my family as they have greatly influenced me to achieve my ambitions.

Kamarul Faisal Kamaruddin

University of Southern Queensland
December 2006

Contents

List of Figures

List of Tables

Nomenclature

Cryptography	Defines the method of transforming data into a sequence of bits that appears random and meaningless to any unauthorised user.
Cryptanalysis	Attempts to identify weaknesses of cryptographic algorithm and their implementations. Can also be simply defined as an attack on networks.
Cryptology	Looking at the problems in the mathematical properties of encrypting using cryptography and cryptanalysis.
Signatures	Activity or profile that is defined as illegal as used in antivirus programs and signature-based detection systems.
Hacker	A computer user that has the ability to expose or cause vulnerabilities to the security of a system including Wireless network security.
Snooper	A person using a device to intercept and eavesdrop over communication on a network.
Sniffer	Often called 'packet sniffers', software tools used to monitor and intercept unprotected data packets.
Spoofing	Illegally duplicating and masquerading MAC addresses of an authorised device to gain access into a network.
Forging	Much like 'spoofing' but it involves illegally duplicating certificates for use in 802.1x security mechanisms.
War driving	It is much like 'hacking', an activity to identify and exploit security holes in Wireless network. War drivers would often travel to their desired victim network and use their mobile device to gain access illegally without the owner's knowledge.

Attacker	An unauthorised person trying to gain access into a network illegally.
Denial of Service	Denial of Service, also known as DoS, attack is a form of hacking on computer systems to cause service disruptions and computer resources to be unavailable to users.
Firmware	Software that is embedded in the memory of a hardware device defining its functions and workings (i.e. software instructions in microprocessors fitted in a hardware).
On the fly	Forming a connection while on the move
Backbone	Highest level in computer networking hierarchy and provides connectivity between lower-level networks.
TCP/IP or TCP	Transmission Control Protocol/Internet Protocol, used by applications on separate networked hosts to create connections with one another in order to exchange data. It ensures data reliability.
UDP	User Datagram Protocol, connections are established before data transfer occurs.
Network Topology	Determines the physical layout of all the devices in the network.
Hash function	An algorithm to produce a certain *fingerprint* for the data.

Chapter 1

Introduction

Wireless technology in communication and computing has become very popular and growing rapidly over the recent years. Number of telecommunication businesses is escalating and many enterprises are integrating Wireless into their network architecture combining or replacing traditional wired connections.

Wireless network has many advantages and disadvantages. Among the reasons why Wireless network should be implemented (Shaw 2003, Carter & Whitehead 2004, Gast 2002):

1. Wireless network is highly flexible. It can easily be installed, used and expanded when needed. A Wireless enabled computer can connect to a Wireless network almost instantly depending on network configurations. In addition to fast set up, the network can be expanded virtually effortlessly. If connections to an Access Point (AP) exceeds the maximum number of devices that can be connected to it, expanding the network is simply adding more Access Points (APs).

2. Wireless networks help deal with the problem of cluttered wires and cables. The computer connected to the Wireless network can easily be moved around without the hassle of handling too many cables or wires. Many devices need power cables since battery power is highly limited and often provide power in too short of a duration for medium to high usage. Thus, minimising the number of cables used

is a bonus.

3. Wireless network is an ideal solution to implementing and upgrading a wired network in an older building, with limited capacity to perform any wired networking upgrade. This is because the communication technology in Wireless networks are mainly contained in the specific equipments used, rather than through the wired connections that are considered costly and power-draining. Further chapters will explain the fundamentals of Wireless technology and its networking functions.

4. Wireless networks can be used to provide Internet access where standard Internet connection is unavailable. For example in conference rooms, and by having Internet access in such rooms, Internet or Intranet-based training sessions and presentations can be conducted.

5. In addition to upgrading existing wired networks in buildings that has many physical limitations, offices can be remodeled in an efficient and often cheaper manner. Wireless network allows the transition or change without expensive recabling work and Wireless network is not confined to physical obstructions. A computer remains connected to the network while being mobile enough that it does not have to be disconnected from peripheral devices such as scanners or printers when moving. Just simply move and plug in computers at new desks. Furthermore, as long as a computer (namely a portable device) that is connected to the network is in range, the computer can move freely while still having access to all the services and resources made available to the network.

6. Wireless network is a good investment as it offers a quick return according to Shaw (2003). A study conducted in 2002 by Wireless LAN Association (WLANA) found that when WLAN was installed, the average time to pay back the initial cost in full was 8.9 months. This means that profits can be made in less than 12 months. Several case studies have been done on real-world Wireless networking applications. The Sugarcreek Local School District in Bellbrook, Ohio, formed a strategic planning committee to help the district integrate modern-day technology. This committee came up with the challenge to put 25 workstation computer laboratory in every school building and a computer in every classroom. This task would involve networking all the computers. The first consideration was

presented, setting up and operating a T1 phone line with speed up to 1.54 Mbps that would cost $20,000 for the first year and an annual recurring cost of $10,000. Initially setting up Wireless networking was $40,000 and speed of only 64 kbps. A prototype of a new WLAN was developed at the same time based on the latest technology during that period and it had a range of up to 20 miles and capable of transmitting data at 2 Mbps. The set up and operation cost would only be $16,000 and no recurring costs thus the school project became the pilot project integrating the new Wireless networking technology.

7. Corporate information can be accessed by the workforce equipped with Wireless Fidelity (Wi-Fi) enabled computers as well as their e-mails without having to look for a cable connection. This saves time and money as it would increase the productivity of the workforce.

8. Devices that are connected to the network can share resources. This is desirable as companies do not have to spend unnecessarily. For instance, the company may only need to buy a single laser printer to be used by workers in the same office.

9. Synchronising between devices can be done through Wireless and it is a more convenient and faster way. When in range, devices can synchronise with each other automatically instead of waiting to be connected using a cable or sync cradle.

10. WLANs can be bridged and further create a Wireless Wide Area Network (WWAN) so that users in different locations can access the Internet, share files and access network resources without the need for wired connections. This is prevalent in important sectors for example, the health care and education sectors, where there might be operations taking place at the same time in a large area.

11. Although currently, wired connection is faster than Wireless connection, Wireless network speed in the future is predicted to be faster. With the development of better Wireless technologies, fast network speed means that data can be transferred in a shorter time while retaining all the major benefits of a Wireless computing environment.

Nonetheless, the switch from wired to Wireless comes at a cost. WLAN performance and security are the two major issues that manufacturers and users alike face. Users may exploit the full potential of Wireless convenience (mobility and portability) but may still be unable to obtain a full strength signal and really secured network as desired. These problems are rarely an issue in a wired network.

Carter & Whitehead (2004) further exposes the disadvantages of Wireless networking:

Power Consumption It is known that Wireless devices have radio transmitters and receivers. Radio devices require a large amount of power to operate, thus computer devices with Wireless adapters have a significant effect on power consumption. Thus, battery power is often inadequate for medium to long period users and they need power cables for the devices.

Interference The air medium is a common commodity to the public and it is used by many types of radio (Wireless) devices. These devices use radio waves or signals that may cause interference with each other and it is very hard to track down and eliminate this problem.

Network Security This is one of the major problem of Wireless networks. In concurrent to the previous problem (i.e. Interference), Wireless networks and Wireless devices are more susceptible to attacks because of its nature of using air as a medium of operation. As mentioned above, the air medium is readily available to the public and it is hard to impose restrictions without proper knowledge and method.

Inconsistent Connections Unlike cable connections that have a direct and stable connections between devices, Wireless devices depend strongly on the availability and strength of radio signals in order to connect to a network. Therefore, the connections can easily be interrupted and lost due to blocking transmission path. Interference may also be a contributing factor.

Lack of Management Network administrators are unable to pinpoint each Wireless device that is connected to the network. This is very much a contrast to wired networks where administrators have complete control on the physical layout and locations of devices to manage the network topology.

1.1 Project Objectives

This project is focused on the WLAN security fundamentals for an enterprise. The major objective of this research project is to provide proof of instability and vulnerabilities of WLAN security performance over existing technologies. Also, a proposal of suitable improvement on the security performance of the Wireless LAN Testbed will be made based on testing, findings and research of this project.

The minor objectives of this research project are to assess the historical risks, review the current security technologies, and investigate the challenges and potential security solutions. A Wireless LAN Testbed will be built to satisfy the above objectives of this research project and security issues and technologies will be examined from this Testbed.

1.2 Overview of the Dissertation

This dissertation is organised as follows:

Chapter 2 describes some background information on Wireless technology and the various Wireless standards.

Chapter 3 further describes the technology of Wireless networking and details of WLAN.

Chapter 4 discusses the security and the performance of current security technologies being implemented in WLAN. The imperfection of current security technologies are also addressed in this chapter.

Chapter 5 describes the design of a Wireless LAN Testbed. It includes the hardware selection and softwares used for this research project.

Chapter 6 describes the experiments that have been done to proof the instability and vulnerabilities of Wireless LAN Testbed security performance. It includes the procedures and results analysis of tests.

Chapter 7 concludes the dissertation, includes recommendations and further work in the area of WLAN security and its performance.

Chapter 2

Wi-Fi Technology

This chapter reveals the popularity of Wireless technology that leads to the vast development of Wireless standards and Wireless networking technology over the recent years. More Wireless devices are being manufactured to reach the high market demand and Institute of Electrical and Electronics Engineers (IEEE) and Wi-Fi Alliance are mainly responsible for the emergence of various Wireless standards. The IEEE 802.11 standard is also described in this chapter as it is the major Wireless networking standard that has designations across the alphabetical order, 802.11a to 802.11z. The standards are released with different features and functions to dictate how a Wireless device operate. Some are still under development to replace existing and obsolete standards.

2.1 The History

The evolution of technologies that lead to today's modern technology started in the early 1800s century (Wheat, Hiser, Tucker, Neely & McCullough 2001). Electromagnetism was discovered in the year 1820 by a Danish physicist and philosopher, Hans Christian Oersted. He was working as a professor at the University of Copenhagen during that time. Michael Faraday, who was an English scientific lecturer and scholar, also played a role in theorising induction in the year 1831. The discovery of induction has lead to the creation of *galvanometer* and electric generator in which the fundamentals

are still used in generators today. *Self-inductance* is also an important theory based on electromagnetism and it was theorised by an American professor Joseph Henry. Henry helped a man named Samuel Morse that lead to the emergence of telegraph and Morse Code in the year 1832, revolutionising the method of world communication. Samuel Morse was responsible in developing Wireless communication by *conduction*. In the year 1887, a German named Heinrich Hertz became the first person to prove that electricity can travel through the atmosphere in a waveform (Wheat et al. 2001). He proved that electrical conductors can reflect waves and non-conducting materials simply let the waves pass through the medium for example, air. Hertz also proved that the velocity of light is equal to radio waves and it is possible to detach electrical and magnetic waves from wires and be broadcasted. In taking Hertz's findings and results, an Italian inventor called Guglielmo Marconi built a Wireless receiver and intercepted a faint Morse code signalling the letter "S", sent across the Atlantic Ocean from a colleague in England. This signals the worlds first truly long-distance communication. Reginald Fessenden then proceeded to further develop Marconi's achievements and he became the first person to create a radio band wave of human speech. Thus, radio was no longer limited to telegraph codes but also human speech.

Next in 1921, mobile radios were generally used for law enforcement activities with operating band of 2 MHz range. It was developed for police and emergency services personnel only and not for the public since the technology was still under experimental stages and not feasible for mass distribution. However in 1924, Bell Laboratories invented a voice-based Wireless telephone that had the ability to be bi-directional, a two-way communication suitable for the public. In 1935, Edwin Howard Armstrong introduced Frequency Modulation (FM) in radio communication. This technology increased the overall transmission quality of Wireless radio and also drastically reduced the size of the equipments because previously, radio systems require a large space for its sheer size and it was very expensive. When World War II began, the military quickly implemented FM technology to provide two-way mobile radio communication. This lead to companies wanting to develop the FM technology rapidly and companies such as Motorola and AT&T immediately began designing radio equipments that are considerably small. Many of the new inventions from companies became possible due to the major invention of the circuit board.

According to Wheat et al. (2001), the evolution of computers started in 1822 when and Englishman named Charles Babbage created the first calculator called the "Difference Engine". Then in 1887, Herman Hollerith produced a punch card reader to tabulate the American census for the year 1890. Later on, various other developments were produced such as different other punch card technologies, binary representation and the use of vacuum tubes. The first decoding machine called the Colossus was produced in 1940 during the war period, to break German codes. The next significant breakthrough in computing was the creation of the Electronic Numerical Integrator and Computer (ENIAC), created by Americans John Presper Eckert and John W. Mauchley. The ENIAC was the first general-purpose computer that can compute at speeds 1000 times greater than the Colossus, which could only perform each calculation at 3 to 5 seconds. However, similar to the development of radio systems, early computer machines were very big and consumed over 160 kW of power. When it was running, it dimmed lights in an entire section of Philadelphia. The main reason for these machines to consume so much power was because of the use of the vacuum tube technology. Therefore, the invention of the transistor in 1948 was very significant, computers began to shrink in size and started getting faster and smaller. In 1981, the company IBM introduced the personal computer (PC) that can be used in homes, schools and also businesses. When PCs were slowly getting more and more popular, emphasis was needed to harness their true potential power and make them work together. Thus, the emergence of Network technology that consisted of a mainframe. This mainframe stored information and performed processes that were connected to several "dumb terminals"that provided the input. The Ethernet was developed in the early 1970s and was used to link many PCs together within an area to form a Local Area Network (LAN). A LAN connects devices over a short distance. Sometimes businesses are composed of several LANs that are connected together thus, yielding Wide Area Networks (WANs) that can span to a much wider physical distance. The Internet is considered the largest WAN that spans the entire globe. To date, digital subscriber line (DSL) service to provide fast Internet, WLAN and the mobile phone system is the latest and becoming the most developed technologies.

2.2 What is Wi-Fi?

Wireless technology has been prevalent in many countries including Australia for many decades. Technologies such as radio, television and mobile phones have been around for a very long time. In recent years, Wireless technology for telecommunications has been widely used. As affirmation, it is reported that the sales of mobile handsets continued to grow worldwide, going up from 482.5 million in 2003 to as many as 816.5 million in 2005, according to Pyramid Research (2005). Wireless technology in computing has also been increasingly integrated in private homes, small businesses and enterprises. Similarly, Table 2.1 shows the popularity of Wi-Fi around the world and its complementing figures from the year 2004 to 2006, and the prediction for 2009.

Table 2.1: Wi-Fi Adoption. The table shows the trend of Wi-Fi users from the year 2004 to 2006 and showing the estimated figures expected in the year 2009 (adapted from (Pyramid Research 2005)).

Wi-Fi Users by Region (in thousands)				
	Year			
	2004	2005	2006	2009
Asia Pacific	32,937	55,341	81,048	168,193
Western Europe	16,681	24,877	33,546	63,746
Central and Eastern Europe	2,109	3,172	4,383	9,875
Latin America	2,386	3,401	4,528	8,331
Africa/Middle East	287	664	1,096	2,747
North America	20,570	30,235	40,454	74,174
Total	74,969	117,690	165,056	327,066

Wi-Fi Paid Subscribers by Region (in thousands)				
	Year			
	2004	2005	2006	2009
Asia Pacific	765	1,950	3,540	9,585
Western Europe	333	921	1,747	5,118
Central and Eastern Europe	14	45	102	461
Latin America	46	135	262	769
Africa/Middle East	0	12	32	136
North America	700	1,480	2,844	7,219
Total	1,858	4,542	8,528	32,286

Wi-Fi is the term given to indicate Wireless products. It is a brand name originally licensed by the Wi-Fi Alliance, a non-profit organisation formerly known as the Wireless Ethernet Compatibility Alliance (WECA) prior to the year 2003, to describe the underlying technology of Wireless networks based on the worldwide standards and specifications (Carter & Whitehead 2004). Wi-Fi Alliance is a trade group that intended to use the designation for mobile computing devices. It is now very popular that the title has become a generic label commonly describing the technology, mobile computing devices and many applications including internet access, gaming and basic connectivity of consumer electronics for example game consoles and DVD players.

A device that bears the "WI-FI CERTIFIED"brand such as a computer, laptop or Personal Digital Assistant (PDA) has the convenience to connect to a network without the use of cables (mobile or portable) when in proximity of a Wireless AP. This is named infrastructure mode as defined by the Wireless standard mentioned further in Chapter 3.

Wi-Fi also allows connectivity in ad-hoc mode, again defined in Chapter 3, which enables devices to connect directly with each other without the use of any AP. This connectivity mode is useful in consumer electronics and gaming applications within a small area network, much the same way where wired connections has been predominantly integrated.

Since Wi-Fi is broadly used around the world, it is important that a standard of operation is developed for manufacturers and users. The IEEE is a non-profit organisation developing standards for various technology advancements such as electrical and electronics, telecommunication, computing and Wireless technologies inclusive. They work in parallel with Wi-Fi Alliance who in addition, ensures interoperability between different devices. In the beginning, IEEE did not test devices for interoperability thus the emergence of Wi-Fi Alliance (Wi-Fi Alliance 2006). Interoperability is a major aspect to Wireless networking as manufacturers are focused in delivering devices to users from all over the world.

The most established standard for Wi-Fi is the 802.11 authorised and governed by the IEEE. This standard dictates how Wi-Fi devices operate, consisting of modulation

techniques, frequency band as well as security protocols. 802.11a, 802.11b, 802.11g and designations 802.11i and 802.11n (still under development) are examples of different protocols for Wi-Fi standard. A typical Wi-Fi device of standard 802.11a uses 5 GHz frequency band and the 802.11b Wi-Fi standard protocol has a raw data rate of 11Mbps. A device of 802.11g standard has data rates up to 54 Mbps and mainly utilises 2.4 GHz frequency band. These standards are further explained in the next section. Wi-Fi is one of the most developed technology in parallel with other Wireless standards and technologies for example IEEE 802.15 (Working Group for Wireless Personal Area Networks (WPANs)), Bluetooth and Infrared (IR) communication technology developed by Infrared Data Association (IrDa).

2.3 Wireless Standards

As mentioned in the previous section, there are many Wireless standards available in the market and under development today. In the computing world, standards become obsolete when older standards are slowly abandoned for newer standards. More often than not, the latter standards are developed to enhance older standards and success is again short lived by increasing demand for better performance and security. Table 2.2 summarises the most important and relevant Wi-Fi standards to this research project, followed by sections detailing some of these standards.

Table 2.2: Comparison of 802.11 standards. The table shows a brief comparison of various Wireless standards with a typical wired Ethernet specification.

IEEE standard	Speed	Frequency band	Range (Indoor)	Notes
802.11	1Mbps, 2Mbps	2.4GHz		First standard developed in 1997. Featured frequency-hopping and direct-sequence modulation techniques.
802.11a	up to 54Mbps	5GHz	~30 meters	Second standard (1999), but products were not released until late 2000. Can be considered obsolete at this point.
802.11b	5.5Mbps, 11Mbps	2.4GHz	~50 meters	Third standard. The most common Wi-Fi standard. Slowly being replaced by 802.11g.
802.11g	up to 54Mbps	2.4GHz	~30 meters	Another common standard in devices today. Latest standard.
802.11i				Enhanced security standard.
802.11n	200 - 540 Mbps	2.4GHz or 5GHz	~50 meters	Still under development. Upgraded Wireless performance.
10-Base-T Ethernet				
	10Mbps			Fast Ethernets operate up to 100Mbps.

2.3.1 IEEE 802.11

The IEEE 802.11 standard was first released in 1997 (IEEE Computer Society 2003*a*) and it was developed to operate among the ISM bands (for more information, see Appendix B). The IEEE 802.11 belongs to a group of many other IEEE 802 standards, as seen in Figure 2.1. They are also called Working Groups that deal with Local and Metropolitan Area Networks (Gast 2002, Wheat et al. 2001).

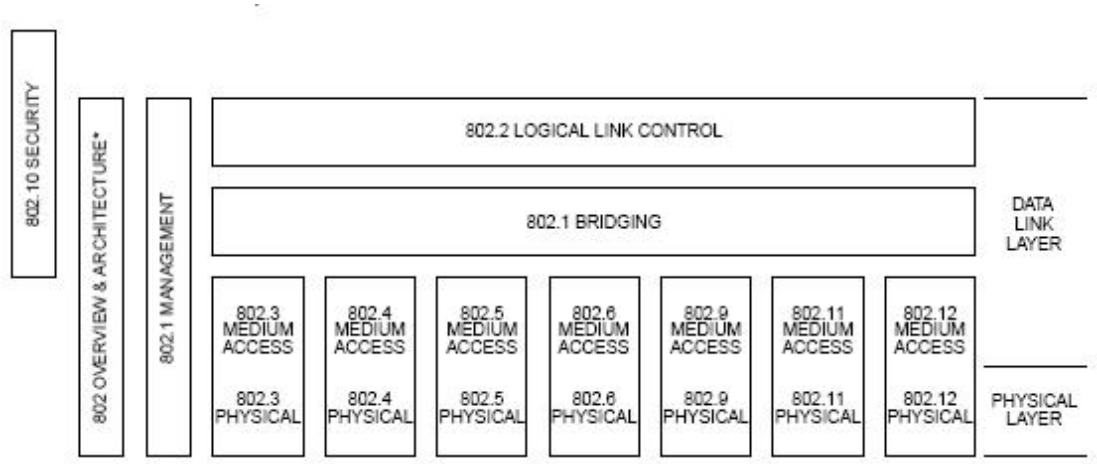

Figure 2.1: The IEEE 802 family. The figure shows the IEEE 802 family and its relations to the Open System Interconnections (OSI) basic reference model. (adapted from (IEEE Computer Society 2003*a*)).

Each of the Working Groups have different functions:

802.1 : Bridging and Management

802.2 : Logical Link Control

802.3 : CSMA/CD Access Method

802.4 : Token-Passing Bus Access Method

802.7 : Broadband LAN

802.11 : Wireless

IEEE 802.11 standard in turn has its own subdivisions that define the different features and functions for Wireless devices to operate by. There are many 802.11 designations

but only a few are most relevant to this research project. Namely the IEEE 802.11a, 802.11b, 802.11g, the newly released 802.11i and finally 802.11n is still under development as of the time this dissertation is written.

2.3.2 IEEE 802.11a

The IEEE 802.11a was developed in 1999 but devices were released in 2001 (IEEE Computer Society 2003*b*). It operates in the 5GHz band range and provides up to 54Mbps data rate. To achieve this, IEEE 802.11a uses a modulation method called orthogonal frequency division multiplexing (OFDM).

2.3.3 IEEE 802.11b

The IEEE 802.11b was first released in 1999 and again in 2001 with amended specifications (IEEE Computer Society 2001*a*). It uses a 2.4GHz bandwidth operating frequency range and transmits data up to 11Mbps. It uses the direct-sequence spread spectrum (DSSS) modulation method.

2.3.4 IEEE 802.11e

The IEEE 802.11e provides enhancements to the 802.11 standard (Wheat et al. 2001). The enhancements include multimedia capability that is made possible with the adoption of Quality of Service (QoS) and other security improvements. QoS provides the functionality required to accommodate time-sensitive applications such as video and audio and it includes queuing, traffic shaping tools and scheduling. These characteristics allow priority for traffic. For example, data traffic that is not time sensitive has a lower priority than applications like streaming video and other real-time applications.

2.3.5 IEEE 802.11g

The IEEE 802.11g standard was released in 2003 (IEEE Computer Society 2003*c*). It is a combination of IEEE 802.11a and IEEE 802.11b specifications. It operates in the 2.4GHz band region as 802.11b and it uses the OFDM modulation technique to achieve up to 54Mbps data transfer rate.

2.4 Protocol layers

The IEEE 802 Working Groups are focused on the two lowest layers of the OSI model that incorporates both physical and data link components (Gast 2002, Wheat et al. 2001). Subsequently, IEEE 802 consists of two major components called the Physical (PHY) layer and Medium Access Control (MAC) layer. The MAC layer acts as a bridge between the link layer and the physical medium and the PHY layer defines modulation techniques, namely the frequency-hopping spread spectrum (FHSS) and DSSS. Previously, the 802.11 PHY layer documented the IR mechanism and to date, the high-rate direct-sequence layer (HR/DSSS) and OFDM has been added. The 802.11 MAC and PHY layers are highly complex compared to the other IEEE 802 MAC and PHY specifications thus much of the complexity details are hidden.

2.4.1 PHY layer

The PHY layer defines the frequency band, data rate and other details for radio transmission (Gast 2002, Wheat et al. 2001). Divided into two sublayers, the *Physical Layer Convergence Procedure* (PLCP) and the *Physical Medium Independent* (PMD). The PLCP is used to map MAC frames onto the radio frequency medium and PMD system is used to transmit those frames. The architecture of PHY layer and the association with the Data Link layer or MAC layer is shown in Figure 2.2.

Modulation of RF signals

Initially, Wireless communication and data transfer used FSK, PSK and QAM for modulation (Olexa 2005, Wheat et al. 2001). However these modulation techniques

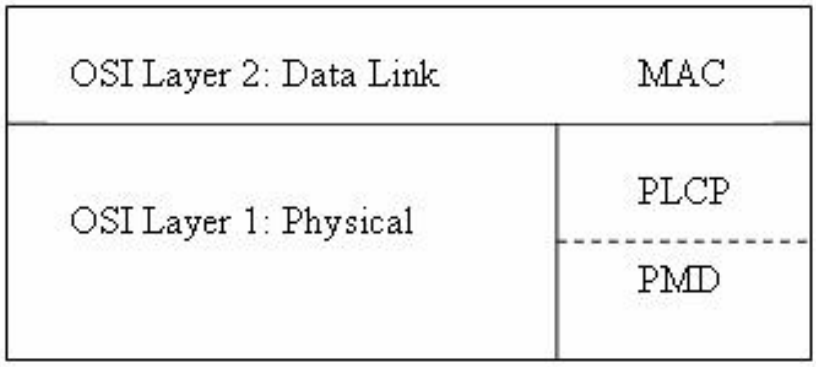

Figure 2.2: Physical layer logical architecture. (adapted from (Gast 2002)).

were for one-way communication thus duplex systems were developed in two forms, Frequency Division Duplexing and Time Division Duplexing. Duplex systems allow receivers to send messages acknowledging that information received is good or contains error. When this has been achieved, Frequency Division Multiple Access (FDMA) and Time Division Multiple Access (TDMA) were needed to address Wireless networks for many users simultaneously using the one common network capacity.

Spread-Spectrum Modulation

TDMA and FDMA systems are unattainable because they do not accommodate interference rejection and high throughput. They are also unable to provide adequate active frequency management and interference control to multiple users. Spread spectrum modulation is the solution to these matters. FHSS and DSSS uses spreading or hopping codes that are unique and works on the principle that collisions would certainly happen. Thus, when two or more users collide on a common channel continually, data might be lost and can only be recovered on the next reception. Since the spreading or hopping code is unique to each user, the receiver on the device may single out the desired channel. Consequently, this reduces the need for frequency management and interference control and allows radio device designs to address coverage and capacity.

OFDM

Since there are only limited numbers of frequency bandwidth available for use in Wireless networking, effectively implementing the right modulation method is very impor-

tant. OFDM is a variation of FDM that uses a large number of channels that overlap each other when transmitting data. In one bandwidth there are sub-channels, also called 'tones', appears as an independent carrier that contains its own modulator/demodulator (modem). Although these sub-channels are overlapped, they are still spaced apart at certain frequencies that provide 'orthogonality'. The center of the modulated carrier is centered on the edge of the next carrier beside it thus preventing the demodulator to see the frequency of that adjacent carrier. OFDM is known for its high spectral efficiency, great flexibility to conform to available channel bandwidth and lower susceptibility to multipath distortion.

2.4.2 MAC layer

MAC layer forms the core framing operations and the interaction between the Wireless network and a wired network backbone. It regulates access from Wireless devices into the shared radio frequency band so that data transmission do not interfere with one another (Gast 2002, Wheat et al. 2001). In order to achieve this, 802.11 uses a Carrier Sense Multiple Access (CSMA) scheme to control access from the Wireless device to the transmission medium (i.e. underlying air medium). Collision Avoidance (CSMA/CA) is used to address the issue of frequency collisions that may waste data transmission capacity. A distributed access scheme with no centralised controller is used, similar to how Ethernet connections operate. It also has two sublayers, the *Distributed Coordination Function* (DCF) and *Point Coordination Function* (PCF).

DCF uses an Ethernet-style contention algorithm that provides access to all traffic. It will first check if the radio link is clear and ready to transmit data. PCF is a centralised MAC algorithm that provides contention-free service by polling stations in turn. Higher priority traffic (traffic with greater time requirements) would utilise this function in order to be allowed to transmit frames after a shorter interval. PCF resides in APs and restricted to infrastructure networks only. The basic order of how network communication is processed in the MAC layer is illustrated in Figure 2.3. Contention-free delivery is passed through the PCF then using DCF before transmission whereas normal data delivery uses DCF directly.

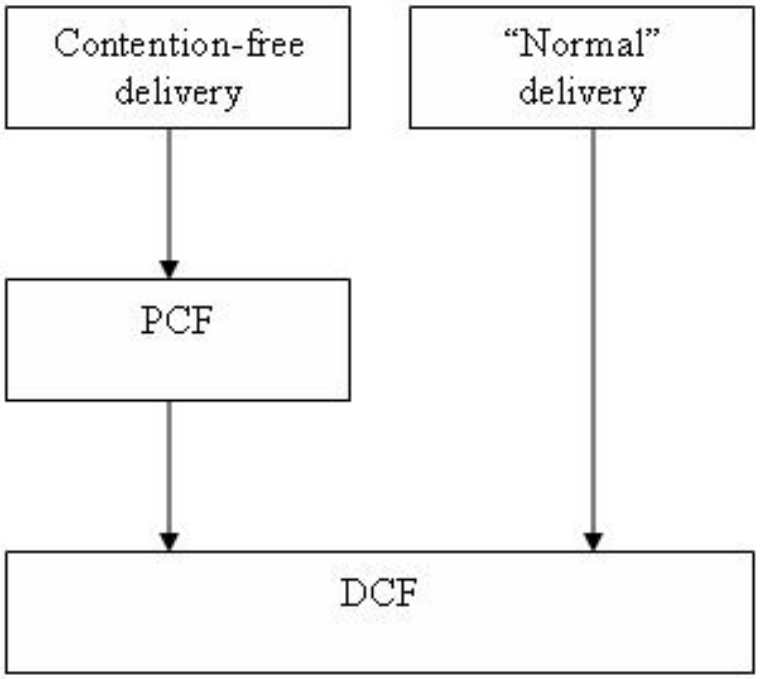

Figure 2.3: MAC coordination functions. (adapted from (Gast 2002)).

802.11 also specified **Scanning** as a primary MAC function (Gast 2002, Wheat et al. 2001). The MAC can perform a *passive* or *active* scanning for Wireless stations to search for APs in a network. Passive scanning is essential as Wireless stations scans each channel for the best AP signal. It will take note of all the information from 'beacons' including signal strength and do a comparison before deciding on which network to connect to. The optional active scanning is similar but it will broadcast a probe frame and retrieve probe responses from all APs in range before making a decision. It is a faster way of locating a network however, it will create an overhead on the network.

2.5 Framing

Framing is a unique feature in Wireless data link adopted in the MAC layer of 802.11 devices (Gast 2002). Framing encapsulates the data for transmission after adding the preamble and PLCP header. It assigns binary bits, 0 or 1, to different fields so that Wireless stations can process the data accordingly.

2.5.1 Frame Control (FC)

Framing utilises four address fields but not all fields may be used and the values assigned to the address fields may vary on the type of frame being transmitted. Figure 2.4 illustrates the fields that are transmitted from left to right with most significant bits appearing last. Table 2.3 further describes each field of the MAC frame and the bits are numbered from right to left.

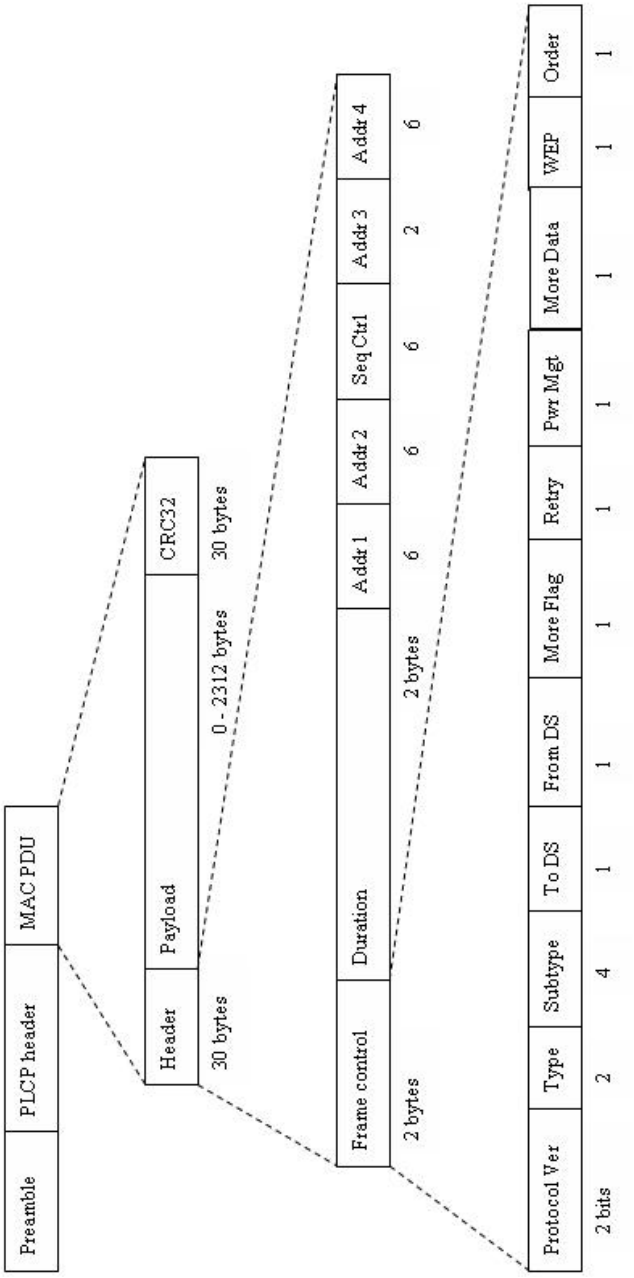

Figure 2.4: 802.11 Frame Format. (adapted from (Yeo 2005)).

Table 2.3: Generic 802.11 MAC Frame. (adapted from (Gast 2002, ZyTrax, Inc. 2006)).

Field	Bits	Notes/Description
Frame Control	15 - 14	Protocol version. 2 bits indicating which version of 802.11 MAC. Currently assigned as 0.
	13 - 12	Type. To identify the type of frame used (i.e. Management Frame(00) or Control Frame(01) or Data Frame(10).
	11 - 8	Subtype. Extension to Type field above. 4 bits long.
	7	To Distribution System (DS). 1 = to the DS.
	6	From DS. 1 = exit from DS.
	5	More Fragment bit. 1 = more fragment frames to follow (0 = last or unfragmented frame).
	4	Retry bit. 1 = frame is a re-transmission. To avoid duplicate frames.
	3	Power Management bit. 1 = station is in power save mode or active mode. Able to power down parts of the network interface (also called atomic frame exchange).
	2	More Data bit. 1 = additional frames buffered by the AP to accommodate destination stations in power save mode.
	1	WEP bit. 1 = data encrypted and frame is changed slightly.
	0	Order bit. 1 = frames are strictly ordered and additional processing by both data sender and receiver is needed.
Duration /ID	15 - 0	If data frames = duration of frame. If Control Frame = ID is of transmitting station.
Address 1	47 - 0	Source address (6 bytes long).
Address 2	47 - 0	Destination address (6 bytes long).
Address 3	47 - 0	Receiving station address (destination Wireless station).
Sequence Control	15 - 0	16-bit field used for defragmentation and discarding duplicate frames. Divided into a 4-bit fragment and 12-bit sequence number fields.
Address 4	47 - 0	Transmitter station address (transmitting Wireless station).
Frame Body		0 - 2312 bytes. Also called the Data field. Maximum payload of 2304 bytes but with WEP overhead, becomes 2312 bytes.
Frame Check Sequence		FCS is a 32-bit Cyclic Redundancy Check (CRC). See Section 4.2 for further description.

Duration

The Duration field carries the value of the Network Allocation Vector (NAV) (Gast 2002). The NAV is simply a timer that indicates the amount of time the frequency is reserved and used for data transmission. If the NAV is nonzero, the medium is indicated as busy by the carrier-sensing function of the MAC. If the NAV is zero, the carrier-sensing function indicates that the medium is idle and ready for use. Four rules for setting the Duration field in the data frames:

1. The Duration field is set to 32,768 if the frame is to be transmitted during the contention-free period.

2. Frames that are not part of an atomic exchange and not acknowledged by receivers are frames transmitted to broadcast or multicast destination. It has a duration of zero. Contention-based access to the radio medium can begin after the completion of such data frame transmission.

3. In parallel to the previous rule above, when there is no more fragments remaining in the frame (i.e. More Fragment field is bit 0), contention-based access can resume operation. Also, at this point the final fragment would only need to reserve the medium for its own ACK.

4. Conversely, if the More Fragment field has a bit 1, that means there are more fragments remain in transmission. The Duration field is then set to the amount of time required to transmit two ACKs, three short interframe spaces and the time required to transmit the next fragment.

Addressing and DS Bits

The number and functions of the address field depends on the type of network deployed (Gast 2002). In other words, it depends on which DS bits are set. For ease of understanding, there are four different functions the address fields can be used for. First function is the Independent Basic Service Set (IBSS), no APs and no distributions systems are used. Address 2 has the source address and Address 1 has the destination address since Address 1 and Address 2 does not necessarily always be source and destination addresses respectively. Address 3 carries the Basic Service Set ID (BSSID) in order for stations to check for broadcasts and multicasts and this ensures that only

stations connected to the same BSS can process the broadcasts and multicasts. This BSSID is simply created by a random-number generator and it is 48-bit long. It is used to distinguish one BSS from the others throughout the same network, much like a filter.

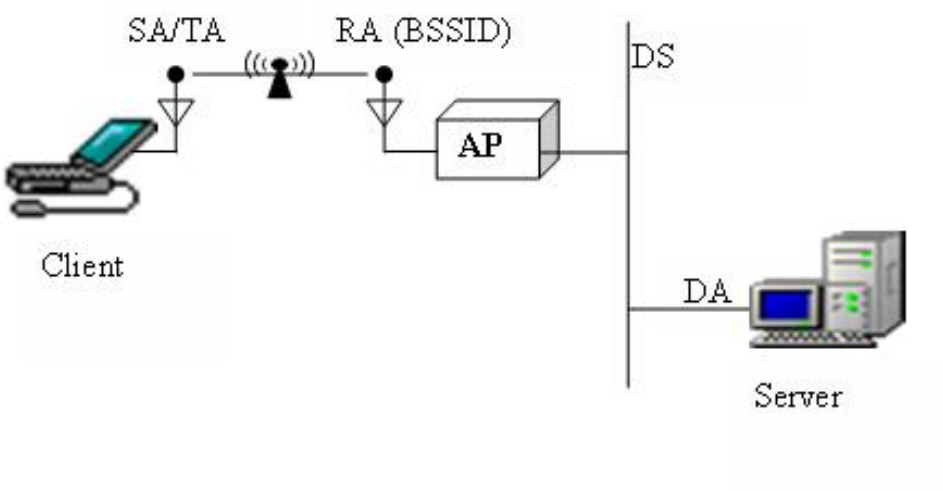

Figure 2.5: Address field usage in frames to the DS. (adapted from (Gast 2002)).

Figure 2.5 shows the features of devices typically used in a network. In an infrastructure network, there is at least one client, an AP and a server connected through Wireless and distribution systems. In the case of communication between a client and the server, frames are sent through the 802.11 network using BSSID, source address and destination address in Address 1, Address 2 and Address 3 fields respectively. Again, Address 4 is not used. When frames are to be transmitted to a device on the DS, the client becomes both the source and transmitter. The AP becomes the receiver but only an intermediate destination because the frame will be relayed to the DS before finally reaching the server. Table 2.4 shows that the BSSID is set to Address 1 because the client is connected to the AP directly and the AP is responsible in creating the BSS from its own Wireless interface. Table 2.4 summarises the use of address fields in data frames transmission.

In the reverse process of the server replying to the client, frames are transmitted through the AP again. This time, the frame is created by the server thus the server's MAC address is the source address in Address 1 field. The AP uses its Wireless interface as the transmitter address and relay then relaying the frames to the client (destination

Table 2.4: Use of address field in data frame transmission. (adapted from (Gast 2002)).

Function	ToDS	FromDS	Address 1	Address 2	Address 3	Address 4
IBSS	0	0	DA	SA	BSSID	not used
To Ap	1	0	BSSID	SA	DA	not used
From AP	0	1	DA	BSSID	SA	not used
WDS	1	1	RA	TA	DA	SA

and receiver). Similar to the previous case, the AP's interface address is the BSSID.

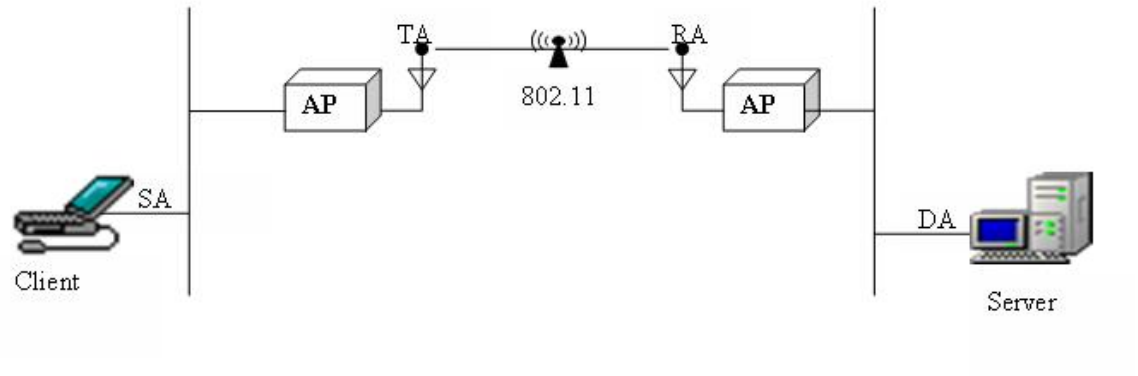

Figure 2.6: Wireless Distribution System. (adapted from (Gast 2002)).

Finally, the address fields are used in a Wireless Distribution System (WDS) or sometimes called a *Wireless bridge* as represented in Figure 2.6. Two wired networks, the client and the server, are joined by APs acting as *Wireless bridges*. The source and destination addresses remain to be the client's and the server's addresses respectively. If the frame is bound from the client to the server, the transmitter is the client's side AP and the receiver is the server's side AP and vice versa for frames bound from server to client.

2.5.2 Control Frames (CF)

Request to Send (RTS)

According to Gast (2002), for large transmission the RTS frames are needed to gain control over the frequency since reservations to access the medium can only be done

for unicast frames; broadcasts and multicasts can be transmitted immediately without reserving access to the medium. RTS frames are optional but it is highly useful in reducing frame collisions. This frame is the first part of a hand-shake communication that has to be performed and validated before sending data frames.

Clear to Send (CTS)

The CTS frame is the respond to the RTS frame request, the second part of the hand-shake communication (Gast 2002). It provides clearance to indicate that data frames are permitted for transmission. The CTS frame has a time value and it is used as reference to pause all other frame transmission whilst the data frame is sent from the requesting station. This again would minimise collisions among frames being transmitted by many different stations on the same network and ultimately result in higher throughput.

Acknowledgements (ACK)

The ACK frame is a positive feedback from receiving stations to tell the sending station that no errors are found in the data frame sent (Gast 2002). The error checking process utilises the FCS field in the Frame Control as shown in Table 2.3. If the sending station does not receive an ACK frame after a period of time, the sending station will simply retransmit the data frame and puts a bit 1 in the Retry bit as seen in Table 2.3.

Power-Save Poll (PS-Poll)

Gast (2002) explains that when a station wakes from a power-saving mode, it will transmit a PS-Poll frame to the AP to retrieve any buffered frames.

2.5.3 Management Frames (MF)

Management frames are required by stations to establish and maintain communication with each other (Gast 2002). 802.11 has provided three simple procedures in managing the Wireless network:

1. Mobile stations must first search and locate a compatible Wireless network to connect to, either using Active or Passive Scanning.

2. Next, the network must authenticate the mobile station before establishing a connection.

3. Lastly, the mobile station must associate with the AP of that Wireless network to gain access to its wired backbone.

The types of management frames are:

Beacon frames are used to announce that a network is available. It is transmitted at a regular intervals to allow mobile stations to locate and identify a network in range. It is also used to match the parameters of a mobile station joining the network. The AP transmits beacon frames in an infrastructure network and it usually defines the BSS of that network.

Probe Request frames are used by mobile stations to scan the area for existing 802.11 networks. As mentioned before that a mobile station must first locate a network, Probe Requests are sent to APs asking permission to join the network in range. When an AP receives the frame, it will use the SSID and support rate information contained in it and if the mobile station supports all data rates of the network and has the SSID of that particular network, access is granted to the mobile station.

Probe Response frame is sent by an AP in an infrastructure network after encountering Probe Request frames above. This frame carries all the information contained in a Beacon frame and it is used by the mobile station to match the parameters and join the network.

IBSS announcement traffic indication map (ATIM) is sent to a station in low-powered mode to notify that it has buffered data. Since there are no APs in an IBSS to buffer frames, ATIM is needed.

Disassociation and Deauthentication Disassociation frames are used to simply end the connection between a mobile station and the network. Deauthentication frames are used to end the authentication relationship between the station and the network.

Association Request Before actually connecting to the network, a mobile station must locate and authenticate itself to the AP and finally sends an Association

Request frame to gain access to the network. This frame has a Capability Information field to indicate the type of network desired by the mobile station and it is also used to verify a match with the parameters of the network.

Reassociation Request frame is simply used to reassociate the mobile station with the network when it moves from one BSS to another in the same extended service area. Stations may also need to reassociate with the AP if they temporarily leave the network coverage area and re-enter at a later time. The Reassociation Request frame differ from Association Request frame because it includes the address of the current AP in order for the new AP to get any association data and buffered frames from the old AP.

Association Response and Reassociation Response When mobile stations send Association or Reassociation Request frames, the AP may send Association Response or Reassociation Response frames respectively. The operations are much like the Probe Request and Probe Response frames exchanges.

Authentication frames are exchanged between the mobile station and the AP for authentication purposes. It may include several authentication algorithms thus the Authentication Algorithm Number field is used for selection. It may also include several steps of authentication process thus it includes a sequence number (Authentication Transaction Sequence Number) field in each authentication frame exchange. The frame body has a Status Code field and a variable Challenge Text field that depends on the authentication implementation method.

2.5.4 Data Frames (DF)

802.11 data frames are carriers of protocols or packets flowing from higher layers in the network (Gast 2002). A data frame is encapsulated in the body of a frame, Frame Control, Management Frame or Control Frame. For example, a beacon frame body contains the SSID, timestamp, and other important information about AP and the network.

2.5.5 Frame Classes

Frames can be divided into different classes (Gast 2002) as seen in Table 2.5:

Table 2.5: Frame Classes. (adapted from (Gast 2002)).

Class 1 frames)		
Control	**Management**	**Data**
RTS	Probe Request	Any frame with bit 0 for ToDS and FromDS.
CTS	Probe Response	
ACK	Beacon	
CF-End	Authentication	
CF-End+CF-ACK	Deauthentication	
	ATIM	
Class 2 frames)		
None	Association Request/Response	None
	Reassociation Request/Response	
	Disassociation	
Class 3 frames)		
PS-Poll	Deauthentication	Any frames including those with either the ToDS or FromDS bits set.

2.6 How Wi-Fi works?

Wi-Fi allows data to be transferred through the air medium using specific frequency bandwidth from the radio spectrum (Geier 2005*a*). The specific bandwidth is allocated by the governing bodies of Wireless telecommunication. Information (electrical or digital) signals are converted into a form (Radio Frequency, RF, or light signals - analogue signals) suitable for transmission from one point to another. This is called modulation, which is done in the transmitter of a Wireless transceiver. There are many methods of modulation:

- Frequency Shift-Keying (FSK)
- Phase Shift-Keying (PSK)
- Quadrature Amplitude Modulation (QAM)
- Spread Spectrum (FHSS and DSSS)
- Orthogonal Frequency Division Multiplexing (ODFM)
- Ultrawideband Modulation

Common Wi-Fi uses both single carrier DSSS radio technology (802.11b) and FHSS. They are part of the larger family of spread spectrum systems and multi-carrier OFDM radio technology (802.11a). Amplifiers are used to increase the magnitude of these signals before departing an antenna in order to achieve better propagation. This amplification does not affect the human hearing thus making it advantageous and appropriate. The analogue (sinusoidal) signals travel from a Wireless workstation to an AP. This can be illustrated as in Figure 2.7, the Wireless workstation may be identified as the computer devices and the AP is also known as base stations. Then the sender device may receive feedback or that data can continue to travel to another Wireless workstation from there. Further at the receiving end, an opposite process occurs, the analogue signals are demodulated into digital signals again. The signals contain useful data for the destination computer. It is within the network interface card (NIC) installed in a computer device that actually provides the interaction between the device and the network. The network card used would have to comply with the international standards of

Wi-Fi, provided in earlier sections. The steps mentioned above involve clients or users.

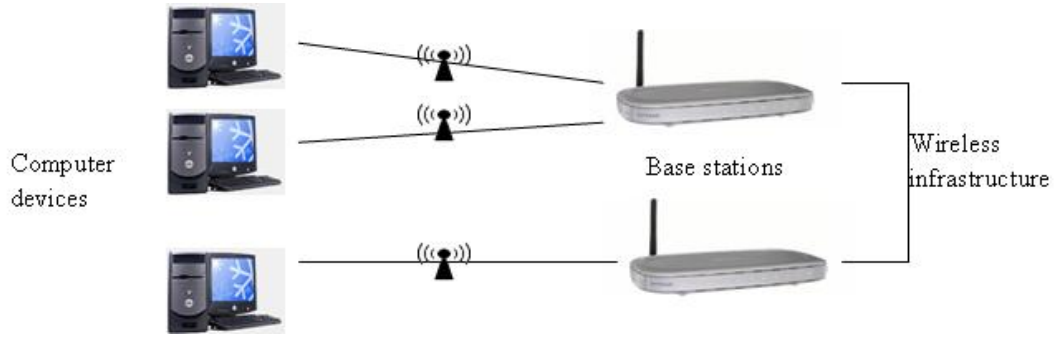

Figure 2.7: Basic Wireless network infrastructure. The figure shows a basic Wireless network infrastructure.

The term client refers to computer devices that operate on a Wireless network. The computer device is specifically designed for the user. A user can be a person directly operating or using the Wireless network, or a machine such as a robot receiving direct instructions over a Wireless network. The user is the component at the receiving end (commonly known as the end-user), thus it is a very important part of the Wireless network. The user may initiate or terminate the Wireless network services through an operating system (OS) of a computer device such as *Windows XP, LINUX or MAC OS*. Some Wireless network infrastructure may have end systems, computer devices that are designed to communicate with other computer devices on the same Wireless network. For example, servers, databases and websites are end-systems.

2.7 Chapter Summary

Wireless networking is constantly evolving. The increasing integration and usage of many Wireless networks are proof of the vast development on Wireless networking technology by IEEE and Wi-Fi Alliance. For example the widely used IEEE 802.11 standards, the 802.11a, 802.11b and the 802.11g. Many more high performing and more secure standards under development today for instance, the 802.11i and 802.11n. Wireless devices manufactured and sold throughout the world would only increase in the coming years thus the need for on-going or advancing Wireless networking projects.

Chapter 3

Wi-Fi Networking Technology

This chapter describes the fundamentals of WLAN alongside other Wireless networks used for various applications. Wireless devices may communicate with each other in two modes, Ad-Hoc or Infrastructure modes to connect to a network. In USQ, users on campus may connect to the network to access the Internet, lecture notes, and many other University resources. The ICT is responsible in implementing and monitoring the Information Technology services. A combination of various security mechanism is enabled for the Wireless network on campus (i.e. Wired Equivalency Privacy (WEP) 40-bit/64-bit authentication and encryption, MAC Address Filtering, SSID hiding and Virtual Private Network (VPN) client for staffs to connect to departmental servers (supports 128 and 168 bit encryption rates).

3.1 Wireless Network

The IEEE 802.11 network standard explained in the previous chapter (Chapter 2) is built around the *basic service set* (BSS), which simply defines how devices communicate with each other within a network (Gast 2002, Wheat et al. 2001). There are two types of BSS namely the independent or ad-hoc mode and infrastructure mode which are explained further in this section and illustrated by Figure 3.2 and Figure 3.1.

3.1.1 Independent networks

In Ad-Hoc mode or Peer-to-Peer mode, Wireless stations communicate with each other directly without the use of an AP (Gast 2002, Wheat et al. 2001). Typically, this mode of communication is set up for specific purposes and applications for a short period of time. For example, this network, also known as Independent BSS (IBSS), can be used by participants of a meeting in a conference room to share data only during the meeting. The IBSS can be terminated when the meeting ends which would be after a short duration. Figure 3.1 is a simple representation of the Independent network.

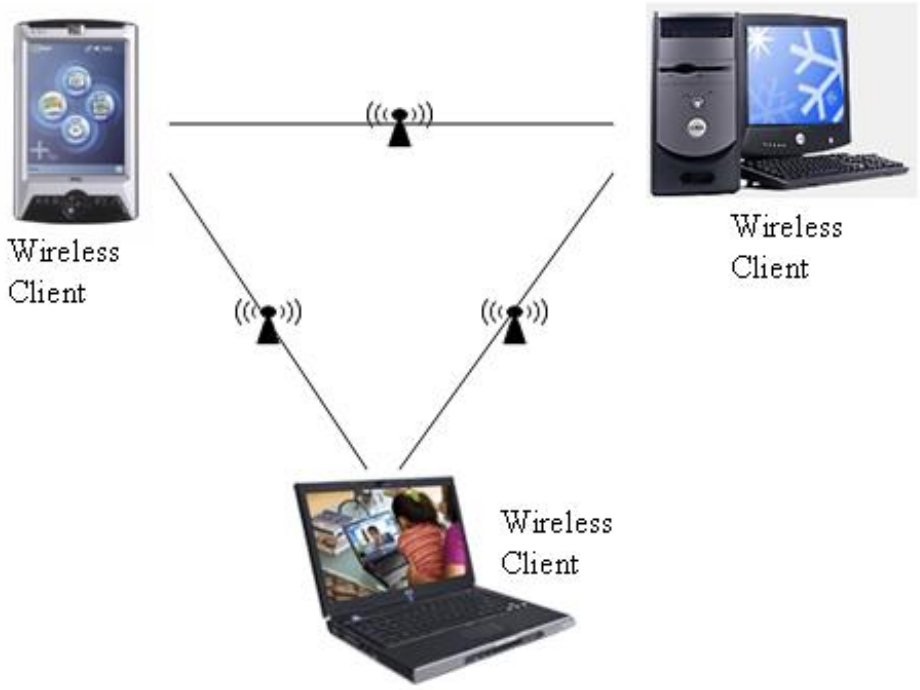

Figure 3.1: Independent network. (adapted from (Microsoft Corporation 2006))

3.1.2 Infrastructure networks

In infrastructure mode, the Wireless network operates with at least one AP. This is a primary aspect in this research project. All communication happens through the AP and it requires two hops for one station to communicate with another station in

the same service area (Gast 2002, Wheat et al. 2001). The first hop comes from the originating mobile station that transfers the frame to the AP. Then the second hop occurs where the AP transfers this frame to the destination station. The simplest configuration of Infrastructure network is shown in Figure 3.2.

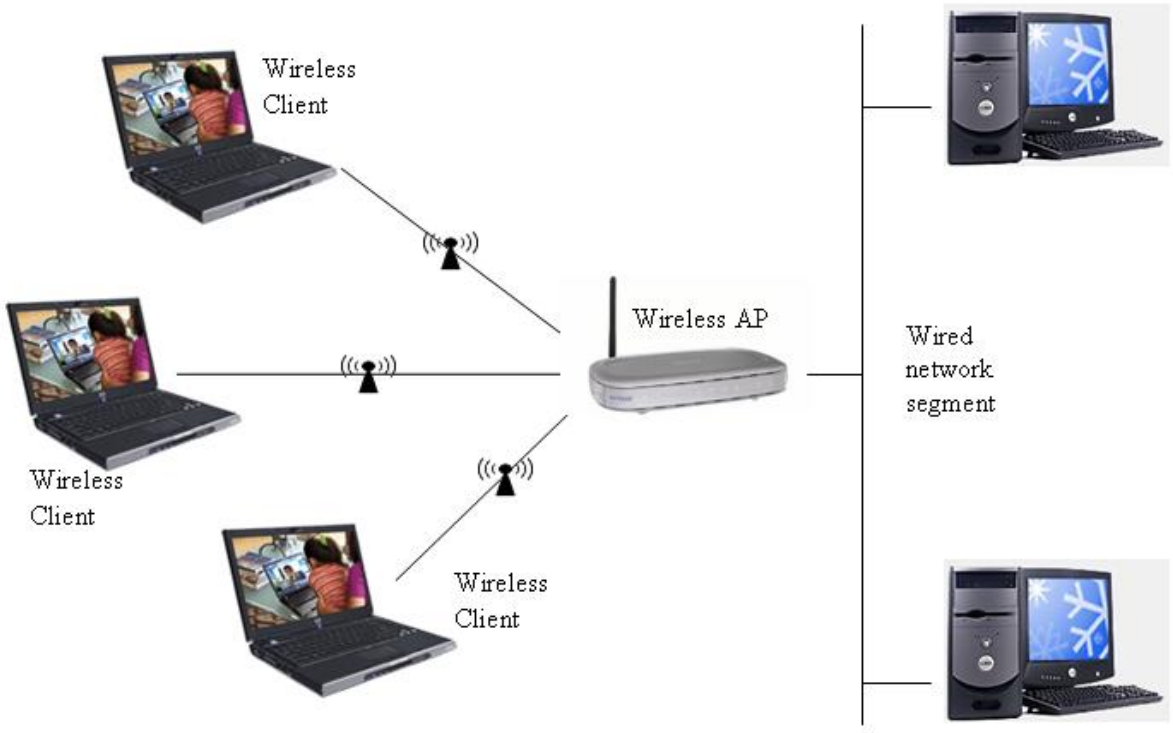

Figure 3.2: Infrastructure network. (adapted from (Microsoft Corporation 2006))

Wireless networking provides convenience for users to be able to freely roam a certain area and still be connected to the network. A Wireless infrastructure network may contain more than one BSS in order to achieve this. According to Wheat et al. (2001), an Extended Service Set (ESS) performs this task where APs can communicate with each other forwarding traffic from one BSS to another. As well as switching the connection of the devices from one BSS to another. ESS utilises a medium called the Distribution System (DS) that forms the spine of WLAN. The DS is responsible in making the decisions whether to transfer traffic from one BSS to a wired network or back out to another AP or BSS as the user is moving. Figure 3.3 shows a Wireless network that consists of several clients that are connected to their respective AP in range, a server,

the DS, BSS and the ESS.

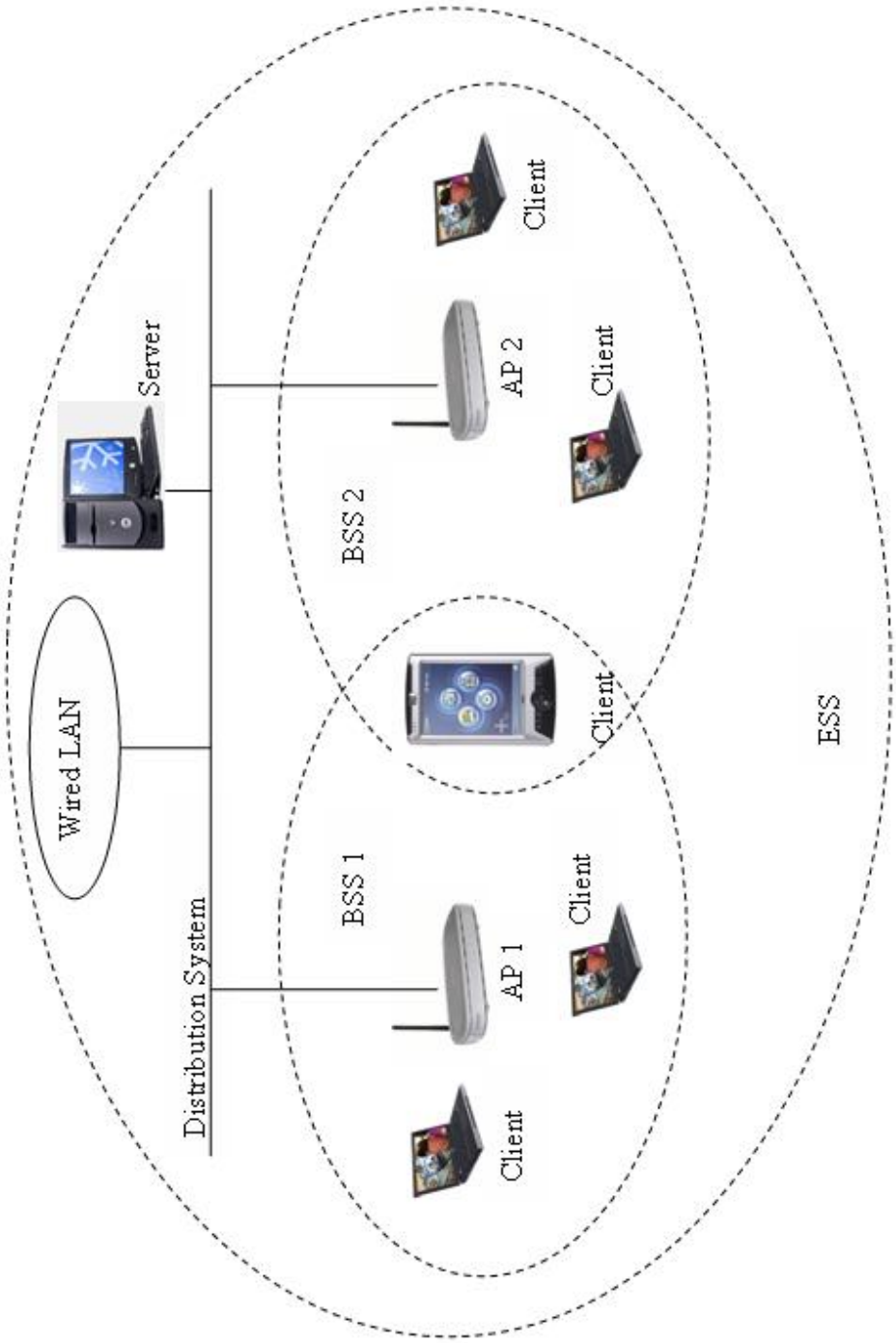

Figure 3.3: Infrastructure network illustrating BSS, DS and ESS.

3.1.3 Network Services

When defining network technology, network services can be defined and Wireless device manufacturers can implement these network services as they see fit (Gast 2002). Nine services are available, three services are defined for data moving and the remaining six services are management control. Following are short description of all network services:

Distribution This service is used for frame delivery to its destination station.

Integration This service is provided by the distribution system for frame delivery to an outside non-IEEE 802.11 network.

Association Used to establish which AP a Wireless device can connect to.

Reassociation As a user moves into the range of another AP, this service is used to change the connection from the previous AP to the adjacent AP or BSS.

Disassociation To terminate the association of a mobile station.

Authentication It is a security service that establishes the identity and credibility of a station before connecting to an AP.

Deauthentication In parallel to disassociation, this service is used to terminate the authentication between the AP and the authenticated station.

Privacy Provides protection against attacks on Wireless networks. This service is further explained in the next chapter, Chapter 4.

MSDU delivery MAC Service Data Unit (MSDU) delivery service is used to ensure data getting to the recipient or end-user.

3.2 Wireless Local Area Network

Similar to a wired connection, a WLAN can enable devices within a building or campus, generally in a small area, to communicate and access applications and information without wires.

The many applications for Wireless networking are (Carter & Whitehead 2004):

- E-mail
- Messaging
- Scheduling
- Data Collection
- Web Browsing

As mentioned before, a typical Wireless network infrastructure would consist of Radio NICs, APs, Routers, Repeaters and Antennas as well as computer systems functioning as servers and clients (Geier 2005*a*). The typical Wi-Fi setup can contain one or more APs and one or more clients. An AP broadcasts its Service Set Identifier (SSID), or generally called the Network name, via packets that are called 'beacons'. Beacons are broadcasted every 100 milliseconds and transmitted at 1 Mbps. The signals are relatively short therefore does not affect the performance of the network. Since 1 Mbps is the lowest rate for Wi-Fi standards, it assures that the client who receives the beacon can communicate at the rate of at least 1 Mbps using any IEEE 802.11 standard. Based on the user settings (i.e. the SSID broadcasting) the client may decide whether to connect to the network of the broadcasted AP. This instruction will further be processed by the firmware of the device.

The firmware running on the client's Wi-Fi NIC is of major influence. For example, when two AP's of the same SSID are in range of the client, the firmware may decide based on signal strength, which of the two AP's it will connect to. On the other hand. the Wi-Fi standard leaves connection criteria and roaming options totally open to the client. This is one of the advantages of Wi-Fi but it also means that one Wireless adapter may perform substantially better than the other. In Windows XP, there is a feature called zero configurations which shows the user any network available or in range and let the end-user connect to it on the fly. In the future, it is expected that Wireless NICs will be more and more controlled by the OS. Microsoft's newest feature called SoftMAC will take over from on-board firmware according to Geier (2005*a*).

The region covered by one or several APs is called hotspot. Hotspots can range from a single room to large metropolitan areas, such as Brisbane and Sydney, where citywide hotspots have been put into operation to allow internet access for the public.

3.3 WLAN implementation in enterprise

Many enterprises are integrating Wireless into their network infrastructures (Geier 2005*a*). Different companies may implement different types of Wireless network. The various types of Wireless networks are as follow:

- Wireless Personal-Area Network (PAN) - close range Wireless application.
  - mostly private users with small devices.
- Wireless Local-Area Network (LAN)- networking in a small area
  - most private or home users and enterprises such as USQ.
- Wireless Metropolitan-Area Network (MAN) - Wireless broadband internet.
  - iPrimus Broadband, IntraPower Pty Ltd Wireless network and Telstra Wireless Hotspots.
- Wireless Wide-Area Network (WAN) - long range connectivity for mobile application.
  - major telecommunication companies such as Telstra, Optus and Vodaphone

Table 3.1 shows the comparison of the above Wireless network types in more detail. It includes the performance, various standards used and their applications.

For clarification purposes, WPAN is typically Wireless networking that links together low powered devices such as laptops, mobile phones and PDAs. WPAN technology includes Infrared and Bluetooth that have a maximum range of around 10 meters and data rates up to only 1Mbps. The 802.15 WPAN standard has been developed to provide interoperability between WPAN devices and WLAN devices of 802.11 standard (Shaw 2003).

Table 3.1: Comparison of Wireless Network Types. The table shows a more detailed comparison of the many Wireless Network Types. (adapted from (Geier 2005*a*)).

Type	Coverage	Performance	Standards	Applications
Wireless PAN	Within reach of a person	Moderate	Bluetooth, IEEE 802.15 and IrDa	Cable replacement for peripherals
Wireless LAN	Within a building or campus	High	IEEE 802.11, Wi-Fi, and HiperLAN	Mobile extension of wired networks
Wireless MAN	Within a city	High	Proprietary, IEEE 802.16 and WIMAX	Fixed Wireless between home and businesses and the Internet
Wireless WAN	Worldwide	Low	CDPD and Cellular 2G, 2.5G and 3G	Mobile access to the Internet from outdoor areas

WLAN is the major focus in this research project and previous chapter (Chapter 1) have outlined the advantages and disadvantages of WLAN. Also, the standards, functions and workings of a WLAN has been provided in Chapters 2, 3 and 4. The USQ has implemented WLAN technology into its network infrastructure. Students may connect to the internet and home drive on the network in wireless mode.

Fixed line networks to homes are still implemented in most areas to provide the ever growing popular broadband Internet access. Especially when there are already existing pathways for telephone cables. Previously, Wireless MAN (WMAN) is implemented to accommodate users or subscribers that do not have access to physical cables in order to gain broadband Internet access. Today, it is implemented for more reasons, particularly for being relatively cheaper and simpler to implement for homes and small enterprises. WMAN can be considered an extension of WLAN since stations communicate with

base stations and connections are in turn routed to a core network.

WWAN is one of the most significant Wireless technology. It is the technology for Wireless telecommunication in wide coverage area and this network has seen the development of vast improvements. To date, WWAN can provide data rates that range from 9.6Kbps to 348 Kbps (2G and 2.5G technologies) and even up to 2Mbps (3G systems).

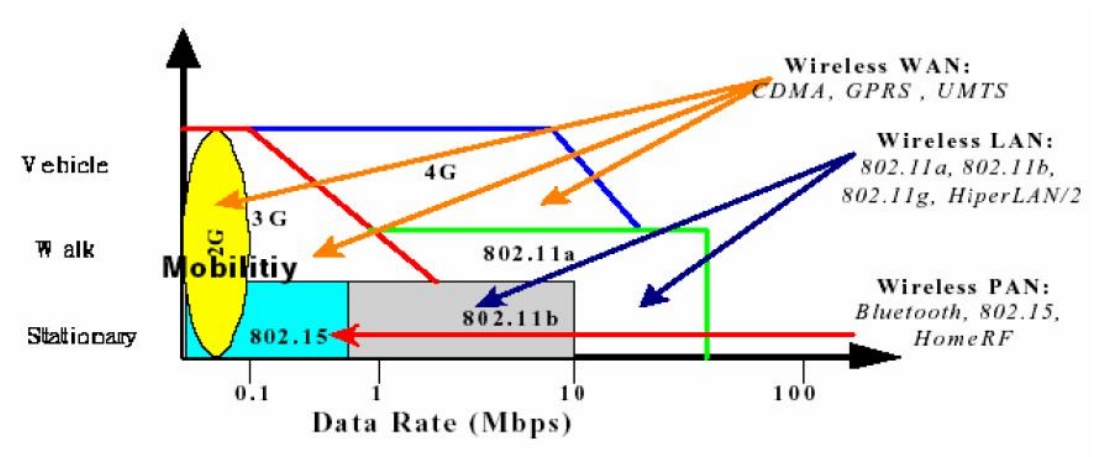

Figure 3.4: Overview of Wireless Networks (adapted from (Baghaei 2003)).

Figure 3.4 shows the comparison between various Wireless networks, WWAN, WLAN and WPAN and their respective technologies. It also includes the comparison between data rates and their mobility level, which indicates a standard to be implemented only in stationary, able to move but in small areas and finally wide area coverage.

3.4 WLAN infrastructure in USQ

USQ has integrated various computing and networking technology on-campus in Toowoomba. This ranges from computer labs for students to computerised video projection equipment in lecture theaters for staffs (University of Southern Queensland 2003). The University has most of the computers in labs on-campus connected to servers and networked using cables. Staff and students can also connect to Wireless network using devices fitted with Wireless NICs.

The Division of Information and Communication Technology (ICT) Services in USQ, Toowoomba provides communication and Information Technology (IT) services and support to students and staff on-campus and off-campus. This department of the University is responsible in implementing and maintaining the networking services in University including Wireless network. The major infrastructure of WLAN on-campus is APs or base stations connected to University's high speed backbone. These APs or base stations are fitted with appropriate antennas and they are installed at strategic locations that provide adequate coverage for users around campus. The University is using Ethernet 802.11b standard in its Wireless network infrastructure which specifies 11 Mbps data transmission rate. This gives about 160 meters range in large, clear spaces and approximately 50 meters range indoors. Although this is said to be largely dependent on building construction and fit-out. Transmission rates decrease as the distance between the AP and the user's device increases. Range and performance are also affected by interference from devices such as microwave ovens and cordless telephones situated around campus. Furthermore, security might be a significant factor to Wireless performance and this research project is objected to proof this theory.

Among the security implemented into the University's Wireless network are:

- Wired Equivalency Privacy (WEP) 40-bit/64-bit authentication and encryption.
- MAC Address Filtering.
- SSID hiding.
- VPN client for staffs to connect to departmental servers (supports 128 and 168 bit encryption rates).

According to Wheat et al. (2001), there are many factors that may have to be considered in implementing Wireless into the existing network infrastructure in an enterprise. Financial constraints may lead to inadequate implementation of security mechanism. This has been confirmed by ITS in implementing 802.11b only.

3.5 Chapter Summary

This chapter had further described the technology of WLAN alongside other Wireless networks used for various applications. Small and low powered devices may communicate with each other without the use of APs (Ad-Hoc mode). This is suitable when transferring small data and in short durations. When using APs (infrastructure mode), devices are able to connect to a network that provides services such as Internet and access to company resources. In USQ, users on campus may connect to the network to access the Internet, lecture notes, and many other University resources. The ICT is responsible in maintaining the computers and online resources for USQ including the Wireless network. ICT has implemented a combination of various security mechanism for the Wireless network on campus (i.e. WEP 40-bit/64-bit authentication and encryption, MAC Address Filtering, SSID hiding and VPN client for staffs to connect to departmental servers (supports 128 and 168 bit encryption rates)).

Chapter 4

WLAN Security

Security is one of the major concerns in networking especially in Wireless networking. This chapter discusses the aspects of today's WLAN security and their known shortcomings. There has not been a security protocol that can be considered perfect but with the efforts of many developers, better security options are being released and replacing current weak security protocols. It also outlines the security performance of WLAN from past projects. Figure 4.1 illustrates the Wireless network environment and some of the problems.

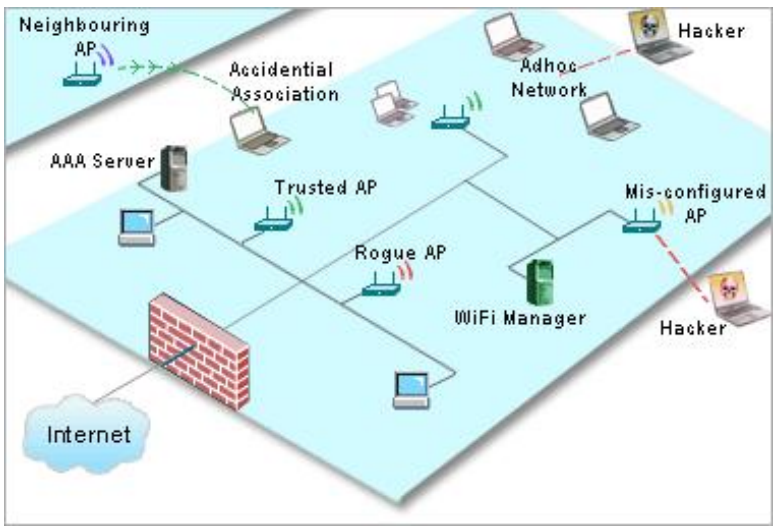

Figure 4.1: Security and Management of Enterprise WLAN (adapted from `http://manageengine.adventnet.com/`).

4.1 WLAN Security Background

Geier (2005*b*) explained that there is constant security threat over Wireless networks. Traffic monitoring and Unauthorised Access pose great threat to users. A 'hacker' may have access into computers to gain vital data. Important data can also be intercepted over the air. A casual 'snooper' can easily monitor and intercept unprotected data packets using software tools, also called 'packet sniffers', available for the public such as AirMagnet and AiroPeek. Usernames, passwords and other private information may be monitored illegally without the users' knowledge. Miller (2003) has classified these threats as Theft.

Campbell, Calvert & Boswell (2003) also defines it as *Identity Threat*. The following threats continue to jeopardise the success of having adequate network security:

Identity theft An unauthorised person may steal the identity and personal information of an authorised person, then impersonate to gain vital resources such as money. Campbell et al. (2003) stated that the research by Identity Theft Resource Center found that each year, over 700,000 Americans have their personal information illegally obtained and used. This is a concern since many applications and activities are done over the Internet and Wireless network that are very vulnerable to attacks.

Privacy concerns Privacy refers to the ability to control personal information that is used to access sensitive information and also used during confidential communication. Privacy according to Campbell et al. (2003), is a major focus for companies since considerable effort is needed in complying to recent privacy laws. Privacy should also be appropriately treated as risk management issue where legal elements are combined with reputation and operations risks.

Wireless access Increasing popularity of WLAN and Wireless Internet access from mobile devices such as mobile phones and PDAs have lead to the need for better security. As mentioned in the previous sections, Wireless connection uses radio frequencies to transmit and receive data. Firewalls does not have the same effect on RF signals compared to wired connections thus the standard approach to

access, authentication and authorisation on Wireless devices have been greatly changed.

Primary factors of Wireless Security can be divided into 5 categories (Geier 2005*b*):

- Theft
- Access Control
- Authentication
- Encryption
- Safeguards

Access Control is characteristically a hardware component that is applied between the AP and the protected side of the network. It simply controls traffic between the open side of the Wireless network and the important resources in the network. Manufacturers now have integrated this form of Security solution inside APs but at the expense of higher cost, lack of open connectivity and short of efficiency in support services. Network administrators who fail to actually implement even this type of security measure would allow any random user to access the network whenever in range. Thus, the network would be vulnerable to 'hacker' attacks and computer virus attacks. Current Access Controllers consists of Subnet Roaming and Bandwidth Management.

Authentication is another low but important form of WLAN Security. It is a method of identifying users that has requested access to the network. It compared the credentials of the user to the database and a match would grant access to the user but if the authentication process fails, the network access is denied to the user. Every Wireless NIC in the market has an internationally unique MAC address to identify itself. This address can be used for Authentication. Routers can be set up to authenticate certain and particular MAC addresses which are specially authorised to use the network. This is also called MAC address filtering. Any random or deliberate unauthorise access to the network can be prevented. However, there is the exception of 'spoofing' where MAC addresses are duplicated and masqueraded to gain authorisation into the network

illegally. Authentication can be further elaborated into two systems. Namely, open system and closed (encrypted) system. Any user is free to roam onto an open or unsecured Wireless network which may still be authenticated by the SSID verification where the SSID acts like a crude password (Wheat et al. 2001). Only authorised users have permission to connect to an encrypted Wireless network. In a practical process, Authentication is followed by Authorisation. When a request is presented to the AP and the user has been approved to gain access to the network, a context within that authentication determines the particular tasks or resources that the user can access. Authentication, Authorisation and Accounting in a structure together is the AAA user access control model. Accounting is the final aspect in AAA and it is a process that measures and records the use of network resources. This monitoring and reporting of events and usage of the network can be used for trend analysis, policy maintenance and capacity planning or upgrading.

Remote Authentication Dial-In User Service (RADIUS) security protocol is a client-server authentication approach that is widely used in many network access server (NAS) environments. It's major function is to authenticate remote users (Gast 2002, Campbell et al. 2003). It is based on the Requests for Comments (RFC) 2138 and RFC 2139 implementations. The key features of RFC 2138 are:

- *Client/server model.*
- *Network security.*
- *Flexible authentication mechanisms.*
- *Extensible protocol.*

At present, two more widely integrated security mechanisms (combined authentication and encryption) in Wi-Fi data privacy are:

- Wired Equivalent Privacy (WEP)
- Wi-Fi Protected Access (WPA)

WEP is an encryption and authentication standard implemented in most NICs. WEP encrypts data before being transmitted over radio waves to protect the data during transmission from one end point to another (Miller 2003). This includes either a 64-bit (also called 40-bit) or 128-bit (also called 104-bit) levels authentication modes with a user specified network key or passkey needed to decode data into usable form.

Moreover, another form of WEP authentication is known as shared key authentication. Arbraugh et al as cited in Bing (2001*b*) explains that this authentication method uses a standard challenge and response along with a shared secret key. A Wireless workstation (initiator) sends an authentication request management frame indicating that "shared key "authentication is to be used in the system in order to gain access to the network. Then, the recipient (responder) responds by sending an authentication management frame containing the challenge text. The challenge text is generated by using WEP pseudo-random number generator (PRNG) and a random initialisation vector (IV).

Encryption is an excellent way to avoid eavesdropping on Wireless network traffic. Encryption scrambles data and makes it unusable thus 'packet sniffers' are unable to view or use the data, also can be defined as *Cryptography* (Vladimirov, Gavrilenko & Mikhailovsky 2004). These data can only be obtained and unscrambled (decrypted) by authorised users in the network. To understand encryption in simpler terms, it is illustrated in Figure 4.2.

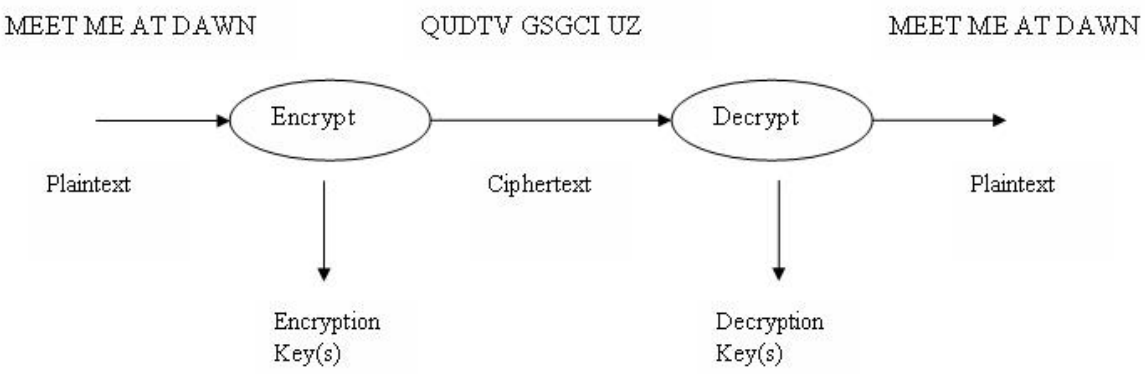

Figure 4.2: Simple encryption system. (adapted from (Nichols & Lekkas 2002))

Data is encrypted using Rivest Cipher (RC4), or sometimes known as Ron's Code 4 named after its developer Ronald L. Rivest. It is a synchronous stream cipher which is the default cipher used by WEP and WPA. It is a variable key-size stream cipher and its operations are byte-oriented (Gast 2002). In more detail, RC4 uses a variable 0 - 256-bit key size. As described in Vladimirov et al. (2004), RC4 encrypts data bit by bit and able to encrypt/decrypt data on the fly. The streaming algorithm designed is advantageous over speed and throughput. It is also designed to generate identical keystreams on both encrypting and decrypting sides thus becoming a reliable tool. Similar to shared key authentication system, RC4 cipher uses PRNG to generate its keystream and perform a Binary Addition to its XOR encryption/decryption algorithms as depicted in Figure 4.3.

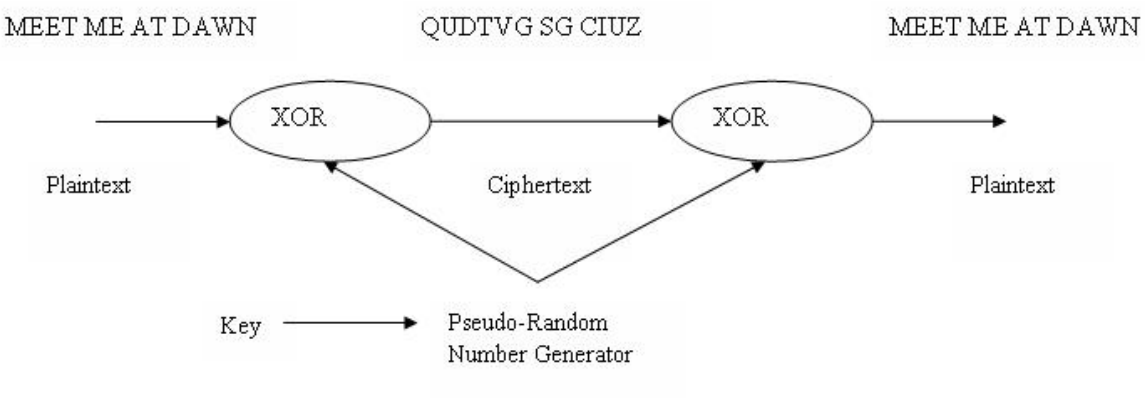

Figure 4.3: Binary Addition encryption. (adapted from (Nichols & Lekkas 2002))

Initially known as WEP2, Temporal Key Integrity Protocol (TKIP) is an upgrade of WEP. It is also from the IEEE 802.11i standard before its final release (Geier 2005*a*, Held 2003). TKIP encrypts data with a key produced by a large 16-octet IV which is added to a combined temporal key and client's MAC address. This ensures that every Wireless station uses different key streams for encryption. The temporal key is changed for every 10000 packets and it still uses the RC4 encryption cipher from WEP.

Attempts to identify weaknesses of cryptographic algorithm and their implementations is called *Cryptanalysis* which is also defined as an attack on the network. Finally, the complexity and importance of data encryption towards network security lead to

Cryptology, where both cryptography and cryptanalysis help look at the problem in the mathematical properties of encrypting.

VPN is an encrypted connection that flows through a shared **public** infrastructure network and two network nodes can have a dedicated and secure link between each other (Campbell et al. 2003). It is a flexible and cost saving security option for organisations to have access to corporate data using the **public** Internet without the risk of unauthorised access to the information, also called a remote access VPN. As mentioned in Chapter 3, USQ has implemented this security mechanism for staff to gain access to information using the same **public** network on-campus without the risk of students, non-staff or unauthorise access to the information. Internet Protocol Security (IPSec) and Point-to-Point Tunneling Protocol (PPTP) are the two key technologies in implementing VPN alongside the common Generic Route Encapsulation (GRE) protocol and Layer Two Tunneling Protocol (L2TP) Tunneling protocol (Vladimirov et al. 2004). The concept of VPN and *tunneling* is encrypting data then enabling one network to send the data through another network. This is performed by implementing the lowest levels of existing Transmission Control Protocol/Internet Protocol (TCP/IP) which is much popular than User Datagram Protocol (UDP) connection. The VPN software or hardware then encrypts the data and wrapping it in another IP packet for delivery through the **public** network. IPSec is an enhanced encryption method for VPNs and it supports both IP version 4 (IPv4) and version 6 (IPv6).

Campbell et al. (2003) further explains the following aspects as designs to address the threats mentioned above:

Integrity refers to the reliability that data is not altered or destroyed when used legally or illegally. It is important that data integrity is maintained as data sent is still identical to data received even for none confidential data. For example, business transactions may be common information but it would be an issue if figures in the transactions can be modified.

Confidentiality is hiding information and protection of data against third party and unauthorised access to sensitive information. Businesses are responsible in maintaining confidentiality on customer data or even internal company data. In main-

taining a trustworthy environment between company and customer, customers have the right to privacy and confidentiality.

Availability is defined as the continuing or continuous service of computer systems such as applications and database servers, storage devices and network providers. It is the opposite of **denial-of-service attacks** that have a significant effect in slowing down or even crashing systems. It is recommended that companies integrate differing availability levels that would address the different business impacts during downtime. This is important when businesses are highly dependent on networked applications, the Internet and various other customer oriented services. System downtime to any of the services may result to serious consequences such as customer dissatisfaction, lack of credibility and even lost revenue.

Moreover, in safeguarding Wireless networks, the most recent and recognised techniques in tackling Wireless Security issue is the Intrusion Detection Systems (IDS), cited in Boukerche (2002). Expert systems or knowledge-based IDS and behaviour-based IDS can detect intrusions into Wireless networks by searching for activities known to be dangerous. Firstly, it will scan a proper traffic behaviour and consider it as normal. The systems' database would include specific attacks and system vulnerabilities to be compared with this previously saved network traffic activities. Seeking in real time, an alarm is immediately triggered if any action is found to be of unacceptable nature. Campbell et al. (2003) further cites the different types of IDS, the software form Computer-based IDS, using hardware device named Network-based IDS and both these IDS perform under two different methods, Anomaly-based Detection and Signature-based Detection. Following are descriptions in more detail of these IDS and Detection methods:

Computer-based IDS is used to secure critical network servers and other systems that contain very important information. Software applications known as agents are installed into each computer on the network. These agents analyses the system and compare it to its database to determine any security breach. These agents are focused to computer-related activities and it is extremely sensitive.

Network-based IDS monitors a specific network segment. It contains a sensor that

passively analyses the network traffic in the monitored segment and compare it to the parameters that have been configured by security personnel. The parameter configuration is done through a management system that displays alarm information from the sensor. An alarm is triggered when illegal packets are detected in the traffic and this causes the router to block all traffic from the source device that sent the packet.

Anomaly-based Detection involves building a database that contains the profiles of user activities. This includes attributes such as time spent by the user when logged on to the network, location of network access and accessed files. Anomaly-based Detection is not a popular detection method. This is because users do not access the network in a static manner and often in enterprises, employees are transferred between other departments or work (accessing the network) at a different location.

Signature-based Detection is similar to antivirus programs in detecting network attacks. It contains a list of 'signatures' that is used to compare against the activity of the network. When a match is found, the IDS would take action in logging the event and triggers an alarm to a management console. Although this is considered the better method of IDS, sometimes it may produce "false positives"according to Campbell et al. (2003), meaning that normal activity is taken as malicious.

4.2 WLAN Security Problems

Four primary causes of network security threats (Campbell et al. 2003):

Technology weaknesses Every network and computing technology today has an issue with security. Especially Wireless network and Wireless computing technologies where Wireless implementations has changed the standard approach of security.

Configuration weaknesses Security problems can be caused by misconfiguration where vulnerabilities are easily exposed if detected.

Policy weaknesses Poor choice of security options can make the best network technology vulnerable to abuse.

Human error This is a major issue where individuals may share their passwords or accidently give out their passwords.

4.2.1 Technology weaknesses

Technology weaknesses in WLAN security may include weaknesses in TCP/IP communication protocol, operating systems and the network equipment (Campbell et al. 2003). As mentioned in the previous section, WEP is one of the first authentication and encryption security mechanism that is still widely implemented in many home and enterprise security policy. However, there are known problems in WEP, outlined by the Internet Security, Applications, Authentication and Cryptography (ISAAC) group at the University of California, Berkeley (Gast 2002):

1. Manual key management causes major concern to network security in very high user population such as in an enterprise. In an ideal secured environment, rekeying would be performed and simultaneously distributed to all systems on the network when any employee leaves the company. However, due to administrative burden this does not happen often enough. Consequently, the key would become public over time. Passive 'sniffing' attacks nowadays only require the WEP keys to gain access to the network then all traffic is readable to the attacker.

2. Even though WEP comes with a long 128-bit cipher key option, standardised WEP offers a shared secret of only 40-bit (or 64-bit) alike USQ Wireless network security implementation. Gast (2002) found that security experts recommend at least 128-bit encryption over sensitive data. This is contradictory as there is no standard being developed for longer keys apart from only 64-bit keystream. Therefore, interoperability between multivendor devices in a single network is not guaranteed.

3. Attackers are able to know if a keystream is reused from WEP's use of IVs. Two frames that appear to have the same IV would most probably be using the same

secret key. Poor implementation of WEP would further degrade this problem as IVs would not be as random as desired. Stream ciphers are vulnerable to *cryptanalysis* when the keystream is reused.

4. Given that network administration staff and typical network systems are overworked, infrequent rekeying of WEP encryption password is common. This in turn enables attackers to gather what is called *decryption dictionaries*. It is a large collection of frames encrypted with the same keystream and a decent amount of frames intercepted means a decent amount of IV can be collated. In the end, the attacker would have enough information available on open frames.

5. WEP uses a Cyclic Redundancy Check (CRC) for the data integrity checking. CRC is a type of hash function that produces a small, fixed number of bits called checksum to protect the file header against corruption such as noise during transmission. This checksum is very useful in detecting errors after transmission or storage especially for compressed data and Wireless data transmissions. When data is transmitted, the sender would calculate a CRC and this is later verified by the receiver that there is no changes on the header during transit. Even though the data may be encrypted by the RC4 cipher, CRCs are not cryptographically secure. Meaning it can only check for errors but not prevent transparently modified frames if done so by attackers.

6. Section 2.5 had briefly explained framing in Wireless data transmission and WEP encryption/decryption in Section 4.1. Frames are particularly decrypted in the AP. Attackers can trick APs that would decrypt any received frames encrypted by WEP and retransmit it to the attacker's station. At this point, the AP would use the attacker's key instead for WEP encryption on the frame.

Although the RC4 is accepted as a strong cryptographic cipher, every security vulnerability and weakness is always the main target of hackers Thus, the need to improve on WEP security mechanism in all aspects.

RADIUS server has also been found to have a set of weaknesses which include both weaknesses in the protocol itself; and possible poor client or security policy implementation (Vladimirov et al. 2004). The UDP protocol in RADIUS is a known weakness

that is highly susceptible to 'forging' and 'spoofing'.

In more detail, the RADIUS server has the following vulnerabilities:

Response Authenticator Attack Response Authenticator is mainly a MD5-based hash. If attackers can observe valid Access-Request, Access-Accept or Access-Reject packet sequences, the attacker can then launch an exhaustive offline attack on the shared secret. When computing this MD5 hash for (Code + ID + Length + RequestAuth + Attributes), the attacker can retrieve the compiling parts of the Authenticator. Finally, the attacker can continue for each shared secret guess.

Password Attribute-Based Shared Secret Attack Attackers can gain information about the shared secret simply by monitoring authentication attempts by users. If an attacker can authenticate with a known password and then capture the Access-Request packet, the attacker can then crack the User-Password attribute. By this time, the Request Authenticator is known and consequently the attacker can launch an **offline brute-force attack** against the shared secret.

User Password-Based Attack This is similar to the attack above. If the attacker knows the shared secret, he or she can modify and replay captured Access-Request packets. Moreover, if the server does not enforce a user-based authentication limit, the attacker can perform an exhaustive online search for the correct user password easily. This however can be rectified by strong data authentication scheme in the Access-Request packet.

Request Authenticator-Based Attacks In RADIUS, the Request Authenticator must be unique and nonpredictable since RADIUS packets rely on the formation of the Request Authenticator field. However, the protocol specifications of RADIUS does not highlight the importance of Authenticator generation and consequently, poor Request Authenticator is generated. PRNG is part of the cryptography in the protocol and when values are repeated or has short cycle, the protocol itself becomes inefficient to secure the network.

Replay of Server Responses By 'sniffing' and intercepting server/client traffic, the attacker can generate his or her own database of Request Authenticators, identifiers and associated server responses. The attacker can then impersonate as the

server when intercept a request that matches information in the database entries. This attacker can then replay any previously observed server response and even a valid-looking Access-Accept server response. Finally, the attacker can actually authenticate the client without proper credentials.

Shared Secret Issues RADIUS permits the use of the same shared secret by multiple clients. Obviously, when sharing passwords, many machines are compromised if an attack happens.

VPN devices are known to be fault intolerable and VPNs have both software or hardware initiated problems (Campbell et al. 2003). Software solutions poses the trouble of processing many connections simultaneously on a large network. On the other hand, hardware solution can solve this issue but at a higher cost as opposed to the much cheaper alternative of software solution. VPN is a complex security mechanism thus it is directly related to the increase of project costs.

4.2.2 Configuration Weaknesses

It is possible to misconfigure network security services. (Campbell et al. 2003). In *unsecured accounts*, critical user account information such as usernames and passwords may be transmitted across the network causing the network to be exposed to 'sniffers'. In addition, *system accounts with easily guessed passwords* may cause the same problem of exposing the network to attacks. 'Hackers' have come up with attacks through hostile Java applets thus *misconfiguring Internet services* for example, turning on Java and JavaScript in web browsers, may leave the network vulnerable. Many computer products have *unsecured default settings* and users may not change the settings upon using the devices or programs regardless of having the knowledge to do so or not. This would also leave the network vulnerable to attacks and unauthorised use of the network.

On the other hand, when changing the *unsecured default settings*, users or network administrators could easily *misconfigure network equipments* and leave large security holes. Dangerous software applications such as **trojan horse** programs, **vandals** and **viruses** can be very destructive to networks as a user that is affected can experience

data being deleted without his or her knowledge. Anonymous mail copies may be sent to email address lists, files and computer systems may be destroyed and finally affecting other computers on the network.

4.2.3 Policy Weaknesses

Security problems can be caused by implementing weak security policy (Campbell et al. 2003). Policies provide clear guidelines that administrators and users can abide by but it cannot be consistently applied with *lack of **written** security policy*. Furthermore, security policies may be ineffective due to *politics* where political battles and staff conflicts occur. In enterprises, *high turnover* may cause vulnerability to the network security because lack of continuity in enforcing security policies to the network. As mentioned repeatedly in this research project, *concise access controls that are not applied* for instance, using default passwords or choosing poor and easily cracked passkeys can allow unauthorised access to the network (Campbell et al. 2003). *Software and hardware installation changes that do not follow policy* can leave large security holes. In addition, *improper security* can cause many more problems to companies as attacks and unauthorised access may waste company resources and expose the company to legal action. When attacks actually occur, it may cause panic and confusion to network administrators and users since *disaster recovery plan is nonexistent*. Policy weakness on the RADIUS server can also cause problems and it has been addressed earlier.

4.2.4 Human Error

Human error or user error is one of the major causes to network breaching, even when there is good intentions in trying to secure a network (Campbell et al. 2003). When errors occur without the knowledge of the well-intentioned administrator or user, it often can cause major harm to the network and as previous sections have revealed, attacks on the network can be highly destructive. Authentication and encryption security protocols can be implemented with hard to guess passwords but security is considered to be breached as soon as the administrator or user tells any confidential information, especially passwords, to an unauthorised person. This kind of act, unsuspecting users

giving away passwords to preying professional 'hackers' and criminals, is becoming common. The different ways of unauthorised access to networks:

- Accidental destruction such as accidently modifying or deleting important data by a network user.
- Ignoring the importance of Wireless network security, includes inadequate awareness, lack of security guidelines and documentation and lack of knowledge.
- Extreme workload as having too many or too few network administrator.
- Dishonesty such as fraud and theft frequently happens and in entreprises, embezzlement and corporate information may be sold.
- A disgruntled employee that has been fired or laid off may have ill feelings towards the company. He or she may attack the network using their knowledge of the network security as an act of revenge.
- Attackers may use impersonation of network users to gain important information from administrators.
- In addition to selling corporate information by current or laid off employees, 'snoopers' may illegally obtain such information and take part in corporate espionage.

4.3 WLAN Security Performance

There have been many Wireless network performance tests that does not involve network security (Shaw 2003). From a single aspect, the Wireless range, tests reveals many factors affecting the quality and performance of the Wireless network. Among the factors are hardware location (i.e. distance between Wireless devices), physical environment (i.e. traveling radio signals reach farther distance when there are less physical obstructions such as walls, metals and concrete) and interference.

When investigating WLAN security performance, it is best to consider many different aspects surrounding the IEEE 802.11 standard since it addresses specific issues relating

to PHY layer optimisations, MAC layer enhancements, security definitions and vendor interoperability as mentioned in earlier chapters. Many existing work on WLAN networks have focussed on the security vulnerabilities and these previous studies have tested and evaluated the security performance of one particular IEEE 802.11b standard Wireless devices. Baghaei & Hunt (2004) has completed a research project into the effect of multiple security mechanisms on the performance of multi-client congested and uncongested networks using different layers of security model. This approach was taken based upon a single server-client and basic traffic model from other similar research projects. It assessed the interaction between different security layers and their effects on performance on the network.

The results, many factors affect network performance and some of these interact with each other to provide overall operating performance. These results can vary depending on the choice of hardware devices, software application and network topology design (Baghaei 2003). Some of the major performance measurements include response time, throughput, coverage area, mobility, bandwidth, latency and radio signal strength. Response time and throughput were measured in depth to provide a comprehensive view of the network security performance. The research referred here also evaluated the performance effect of different communication protocols, TCP and UDP packet size distributions on secure Wireless networks. The paper further revealed the advantages of the Wireless network study conducted focussing on finding ways in which to configure Wireless networks such that it can meet security requirements. From the comprehensive tests conducted by the project, it was reported that in general, the stronger and more complex security mechanism implemented, the poorer the network performance. However, when implementing simple security layers such as MAC authentication and WEP encryption for a single client, it had little performance impact but still provided adequate security. On the other hand, a network with multiple clients and congested traffic degrades the performance significantly even when a simple WEP authentication and encryption method was applied. Following are results obtained from the tests performed by Baghaei (2003).

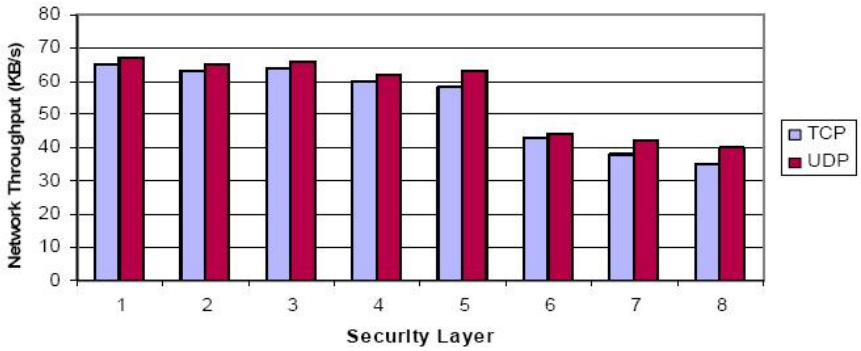

Figure 4.4: Throughput of TCP and UDP traffic in an uncongested Wireless network (adapted from (Baghaei 2003)).

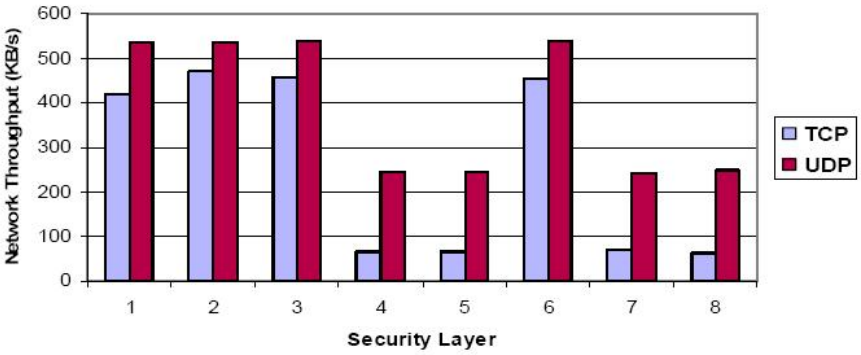

Figure 4.5: Throughput of TCP and UDP traffic in an congested Wireless network (adapted from (Baghaei 2003)).

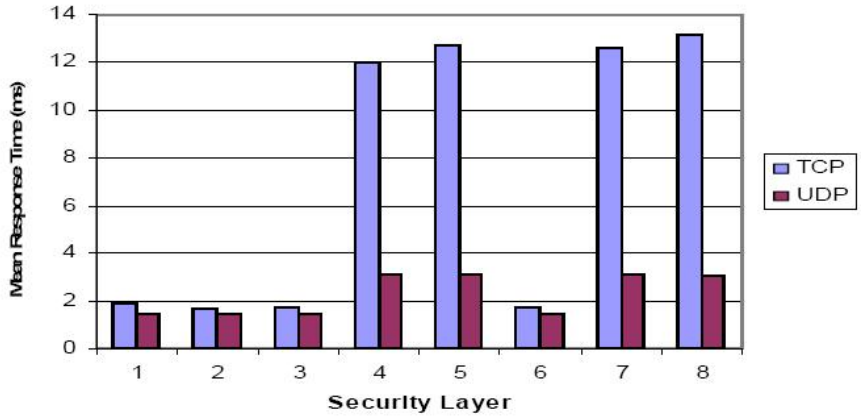

Figure 4.6: Response Time (adapted from (Baghaei 2003)).

4.4 Chapter Summary

Security is one of the major issue in WLAN. This chapter has discussed current WLAN security and their known problems. Much effort has been given into enhancing or replacing weak security mechanisms. Finally, this chapter outlined the security performance of WLAN from past projects.

Chapter 5

Project Methodology

This project is currently sponsored by the Early Career Researcher Program (ECRP) of Faculty of Engineering and Surveying in USQ Toowoomba campus. Most materials have been kindly provided by the project supervisor, Dr. Hong Zhou. The following sections present in more detail the components acquired for the project.

5.1 Hardware Selection

Figure 5.1 shows the AP device to be used extensively in this research project. The

Figure 5.1: *NETGEAR* 54 Mbps Wireless Router with 4-port 10/100 Mbps switch WGR614v2.

key features of this AP are:

- Speeds of 802.11g - up to 5x faster than 802.11b.

- 10/100 Mbps on WAN.
- Double Firewall - Stateful Packet Inspection (SPI) & Network Address Translation (NAT).
- Smart Wizard^{TM}automatically detects Internet Service Provider (ISP) settings and walks you through installation.
- Compatible with 802.11b devices and 802.11g devices.
- Shares broadband Internet access to every PC in your home.
- Wi-Fi Protected Access Pre-Shared Key (WPA-PSK).
- Up to 128-bit WEP encryption.
- VPN pass-through support.
- MAC address authentication.

Figure 5.2 shows the Desktop device to be used in this research project as a server or a Wireless client. Key features of the Desktop:

Figure 5.2: *Dell* Desktop PC.

- Microsoft Windows XP Home Edition OS.
- On-board Ethernet.

Figure 5.3 shows the Wireless network card fitted in the Desktop shown in Figure 5.2. This NIC provides Wireless network capabilities to the Desktop. The key features:

Figure 5.3: *BELKIN* High-Speed Wireless G Desktop Network Card.

- Adds 802.11g Wireless capabilities to any desktop computer, for faster Wireless networking available for home or office.
- Works with all 802.11b Wireless devices.
- Fits any standard 32-bit Peripheral Component Interconnect (PCI) expansion slot.
- Provides 3 times the Wireless range of 802.11a clients.
- Offers interoperability with all 802.11b 2.4GHz Wireless devices.
- Features Wireless 64- and 128-bit WEP encryption.
- Allows you to use Turbo Mode and network at 54Mbps, the highest data rate for all 802.11g clients.
- Works with Windows®2000, Me, and XP.
- Comes with a Belkin Lifetime Warranty.

Figure 5.4 shows the privately owned Laptop mainly used as a Wireless client in the experiments of this research project. The key features:

Figure 5.4: *HP* Pavilion DV1020AP Laptop PC with Wireless capabilities - privately owned.

- approximately 2kg weight.
- Random Access Memory (RAM) 478MB double-data-rate synchronous dynamic random access memory (DDR SDRAM) shared.
- Intel Centrino Mobile Technology, Intel Pentium M processor 1.60GHz, Intel 855GM Chipset, Intel Pro Wireless 2100 WLAN Mini PCI 802.11b.
- Graphics Intel internal/integrated 64MB shared memory.
- Liquid Crystal Display (LCD) Panel size 14.0" Thin-film Transistor Wide eXtended Graphics Array (TFT WXGA) high definition widescreen with 15:9 aspect ratio.
- Hard Drive 60GB.
- Optical drive DVD/CD-RW Combo Drive.
- Integrated 10/100 Base TX LAN RJ-45 with High Speed 56K Modem RJ-11.
- 6 in 1 integrated digital media reader slot.
- 3 USB slots, VGA x 1, TV out x 1, Audio in/out, Firewire 1394 x 1.
- Microsoft Windows XP Home pre-loaded software.

Figure 5.5 shows the privately owned PDA mainly used as a Wireless client for small applications. Among the key features of the PDA:

Figure 5.5: *HP* iPAQ RX3417 Personal Digital Assistant with built-in Wireless - privately owned.

- Operating System Windows Mobile 2003.
- RAM 92MB.
- Screen 240 W x 320 H.
- Storage 4-bit SDIO and 4-bit SD/MMC.
- Battery - removeable/rechargeable 920 mAh Lithium-Ion.
- Connectivity 1.1 compliant Bluetooth, 1-22 pin connector, Infrared Port, Integrated WLAN 802.11b.

5.2 Software Selection

Traffic Generator

IPTraffic that includes *WinPcap 3.1* (installed together with *IPTraffic*).

IPTraffic was chosen as it has beneficial features as pointed out by Baghaei (2003):

- It is suitable for experimenting wired and Wireless networks.

- It is able to generate, receive, capture and replay IP traffic and consequently measure performance and QoS over IP networks.
- It is capable of overloading a network.
- *IPTraffic* has the flexibility for users to change the size and inter-packet delay.
- Users are also allowed to select the traffic generation algorithm.
- It can manage a number of simultaneous IP connections.
- Runs on Windows platforms.

Traffic Capture

Ethereal can capture live network statistics. Thus, giving a depth look into the traffic flow within a network. The results contain information on Wireless transmission Frame Control.

802.1x Security Mechanism Configuration Software

FreeRADIUS

FreeRADIUS is an open source code RADIUS server program that handles the implementation of 802.1x security mechanism for networks.

5.3 Testbed Design

Testbed can be defined as a place with equipments for experimentations under real working conditions. These may include testing engines or machineries or even computer programs and computer devices.

There are numerous steps involved in designing a WLAN (Wheat et al. 2001). The six major phases to a sound design method:

1. Conducting preliminary investigation regarding necessary changes.
2. Performing analysis of existing network environment.
3. Create design.

4. Finalise design.

5. Implement design.

6. Create necessary documentation that will act as crucial tool for troubleshoot.

5.3.1 Preliminary investigation

According to Wheat et al. (2001), the *who*, *what*, *where*, *when* and *why* of the network have to be identified in this preliminary investigation. This Testbed will be used mainly for experiments to address the objectives of this research project. Therefore, it does not include any other users nor the public involvement. The environment of the Testbed is simply, all devices are indoors and confined in the Electronic Lab room. The time experimenting with this Testbed is only limited until the completion of the Research Project Part 1 and Part 2 courses in USQ. There are no current networks exist therefore, it will be a new model implementation.

5.3.2 Analysis of existing network environment

As mentioned in Section 5.3.1, there are no existing network environment. However, the reference network will be the USQ WLAN on-campus. Basic information of USQ WLAN configuration has been provided in Section 3.4.

5.3.3 Create design

The design for the Wireless LAN Testbed for this research project is shown in Figure 5.6.

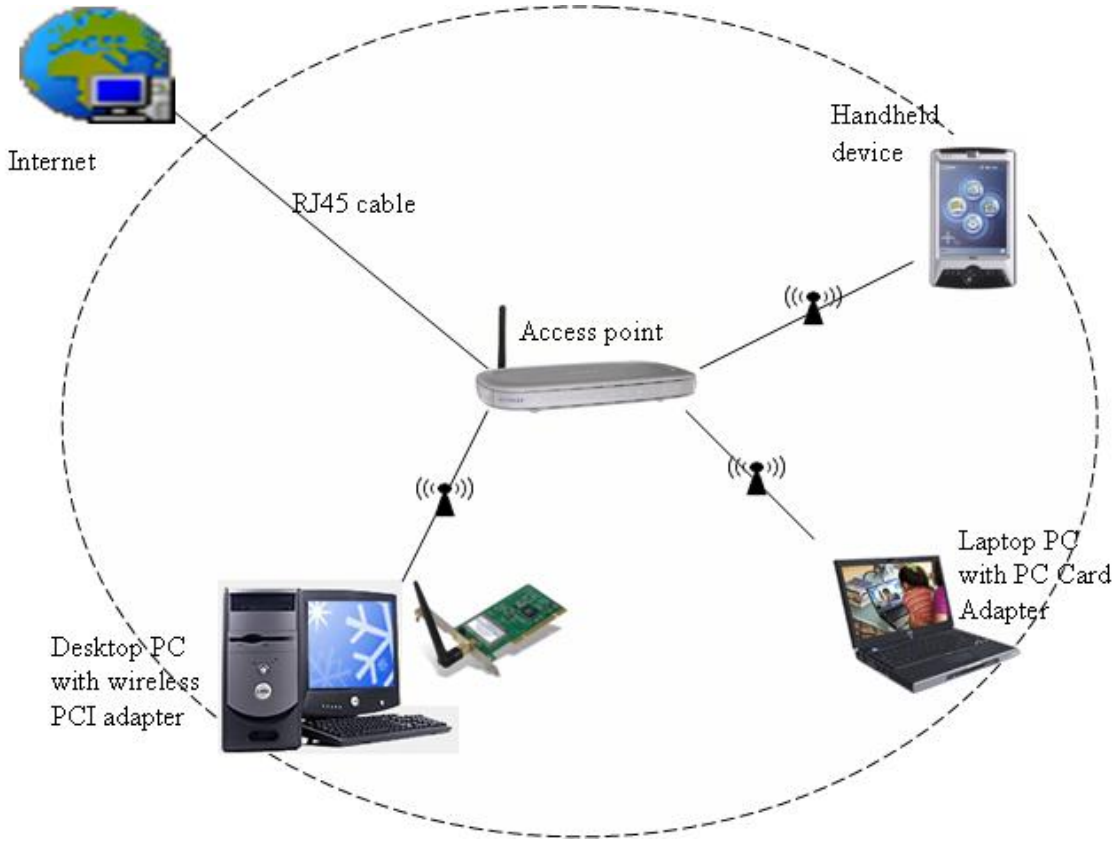

Figure 5.6: Topology design of desired Wireless LAN Testbed.

This design includes all hardware acquired in this research project. It also utilises the common 'Star Topology' which is predominantly an infrastructure network configuration.

5.3.4 Finalise design

The resultant Testbed setup differs from the desired hardware configuration as shown in the Figure 5.6 in Section 5.3.3. This is due to unavailability of main resources (i.e. the Internet) and difficulty to acquire extensive hardware resources for multiple client environment. Figure 5.7 shows the setup used for the experiments of this research project.

Figure 5.7: Experimental setup.

Chapter 6

Experiments

6.1 Aim

The aim of this research project is to proof the instability and vulnerabilities of WLAN security performance over existing technologies. Many factors have been proven to affect the network performance and results vary depending on the choice of hardware, software and network configuration (Wong 2003). Measuring the performance of a network may include Response Time, Throughput, Coverage area, Mobility, Bandwidth, Latency and Radio signal strength. In this research project, **the effects of different security mechanisms over a congested and uncongested Wireless LAN Testbed** and **the impact of security mechanism over different WLAN traffic types** are addressed in particular. In doing so, several measurable and observable factors are considered:

Response time is the total time taken between initiating the data transfer and traffic to actually start flowing. This includes dial-up connection establishment and security negotiation time.

Throughput the total number of data transmitted over the network in a certain time (i.e. Response Time).

Traffic TCP and UDP traffic types, congested and uncongested traffics.

This is very similar to wired connection but, taking into account that Wireless technology or particularly Wireless enabling devices such as Wireless router have to adhere to certain standards that dictates the performance. Many other external factors also have significant effects on the performance of WLAN but can be ignored for the simple experiments described in this chapter.

6.2 Procedures

Defining Security Layers

As the major part of this research project is to compare the performance of the network over different security mechanisms. Therefore, all possible security protocols that are supported by the acquired devices are chosen as outlined below:

1. **no security**
2. **open authentication with 64-bit WEP encryption**
3. **open authentication with 128-bit WEP encryption**
4. **shared key authentication with 64-bit WEP encryption**
5. **shared key authentication with 128-bit WEP encryption**
6. **WPA-PSK (TKIP) Wireless security**
7. **MAC address filtering**

The configuration of these security setups are shown in Appendix C for both AP and the end-user.

Wireless Traffic

As mentioned in Section 6.1, this research project is also interested in looking at the effects of security mechanisms over congested and uncongested traffics. As well as the different types of traffics, particularly TCP and UDP traffics. Figure 6.1 shows the

parameters configured in IPTraffic for experiments over TCP traffic and Figure 6.2 for experiments over UDP traffic. Based on Baghaei (2003), the packet contents chosen are to be random from hex 41-7A and the packet length is varied from 40 bytes to 1500 bytes. The packet numbers are chosen to be 10000 for uncongested traffic, 43000 for congested traffic and 60000 for experiments with very high bandwidth (when 802.11g support is enabled in the AP).

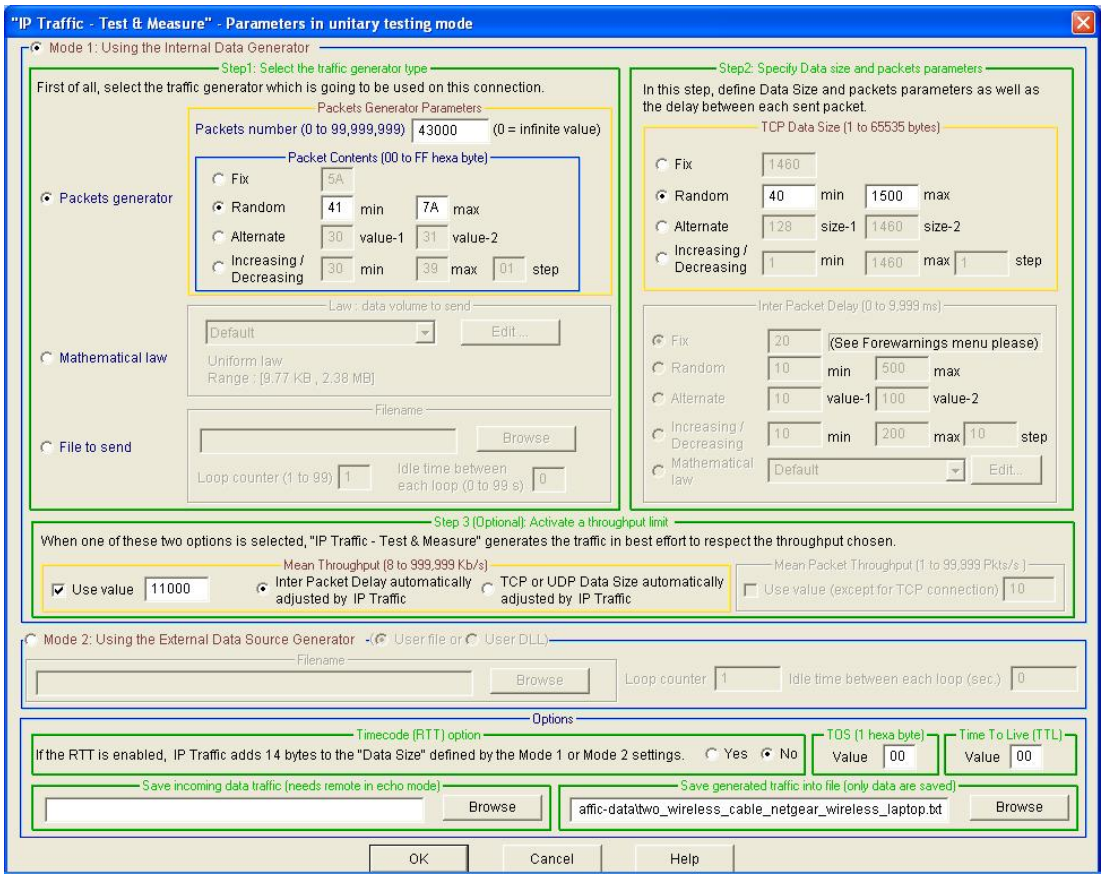

Figure 6.1: IP generator parameters for TCP traffic.

Transmission speed was set to depict congested traffic and uncongested traffic at 12Mbps and 200kbps respectively for TCP traffic. Similarly with UDP traffic, 12Mbps and 100kbps for congested and uncongested traffics respectively. Since the AP and the Desktop are IEEE 802.11g standard devices, experiments with outgoing bandwidth of 54Mbps were also performed. However, it is not entirely relevant to designing the initial Testbed based on the implementation of WLAN in USQ that only supports 802.11b standard transmission. Furthermore, the Laptop device acquired in this re-

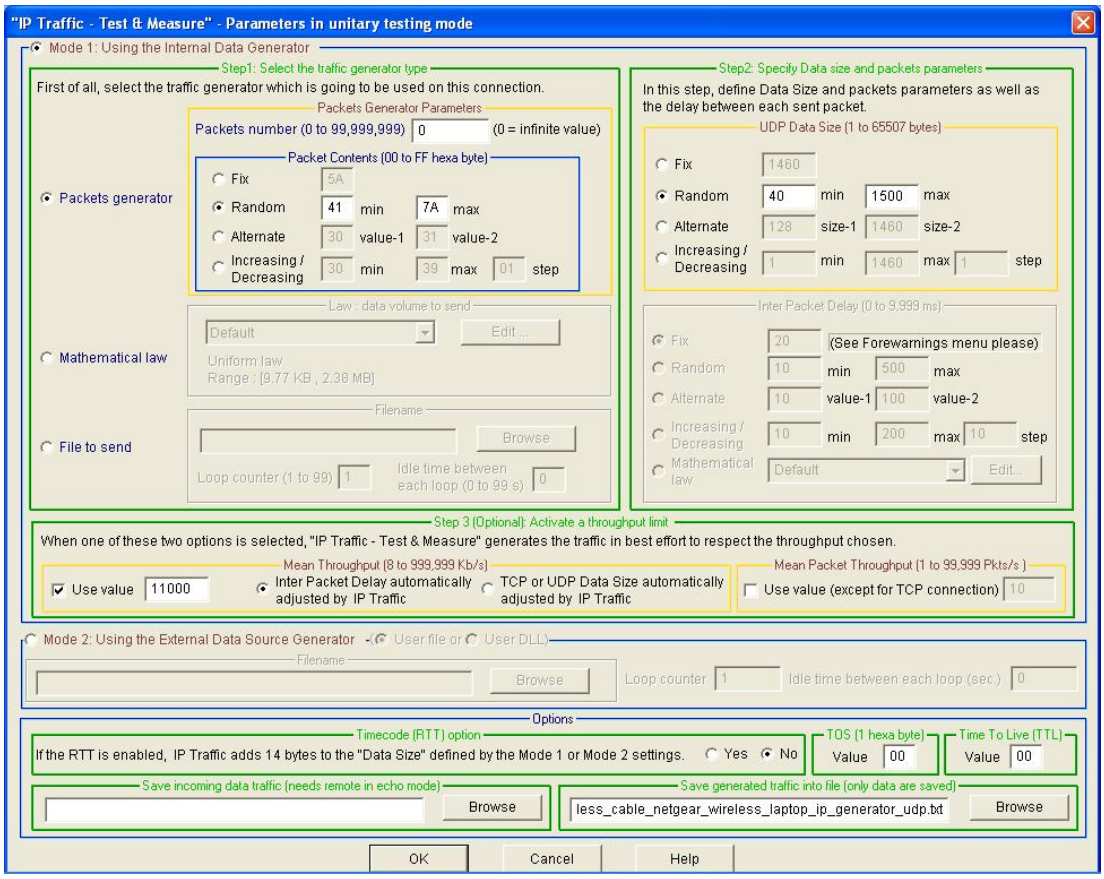

Figure 6.2: IP generator parameters for UDP traffic.

search project only supports 802.11b transmission therefore would significantly affect the performance of a LAN traffic of 54Mbps (Bing 1999).

The *Ethereal* software is used to measure and analyse in more detail of the traffic flowing through the Wireless LAN Testbed. In particular, the measurements were taken from the wired side of the Testbed.

WLAN Configuration

The experiments were conducted using the following configuration (based on the design illustrated in Section 5.3). Figure 6.3 shows the IP addresses of each device used in the experiments.

For further clarification, the Desktop and Laptop are connected directly to the Wireless AP using Wireless and wired modes. Initial experiment is to have data traffic

Figure 6.3: WLAN configuration.

flowing from Laptop and back passing through the AP. This experiment is repeated with the Desktop. Next, the experiment above has been repeated with the Desktop PC connected by cable to the AP and Laptop is connected Wireless to the AP. It was decided that Windows-based operating systems are used in the final experiments of this research project. The reason being is that the softwares acquired only support Windows-based platforms.

6.3 Results

The results follow the experiments on the effects of the seven security layers on the Wireless LAN Testbed performance described in Section 6.2. Performance measures were gathered after running through each security layer experiments and compared after the completion of all experiments. This is simply done by referring to data collected into log files generated by both *IPTraffic* and *Ethereal*. The analysis of the results were taken based on the mean at 95% confidence interval.

Much emphasis has been given to TCP connections since many applications rely on this connection and applications using UDP type connection is minimal. The graphs in Figure 6.4 and Figure 6.5 shows the response time of the network over both uncongested TCP and uncongested UDP traffics. When taking the response time of the Testbed at **no security** as reference, the response time for security mechanism 2(**open authentication with 64-bit WEP encryption**) increased by 4% and 9% for TCP and UDP traffic respectively. The Testbed response times when security mechanisms 3, 4, 5 and 6 were applied increased at a lower margin, by 1% (on average) for TCP

and UDP traffic. Finally, there is only less than 1% increase in the response time of the network when MAC address filtering was applied. It is evident that security mechanisms other than no security layer, takes a substantially longer time to establish a connection and transfer data.

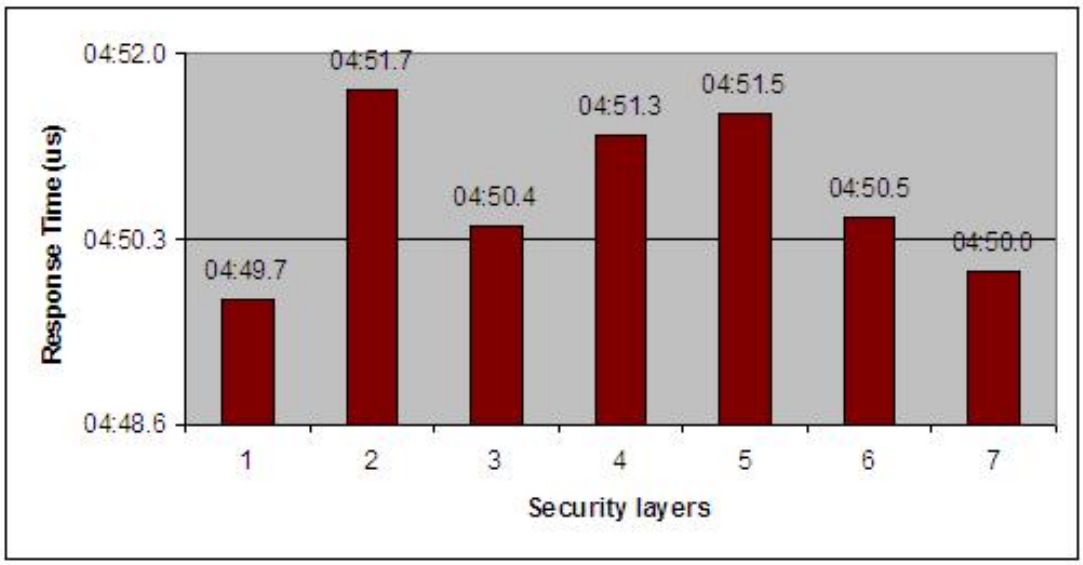

Figure 6.4: Response Time of Wireless LAN Testbed over uncongested TCP traffic.

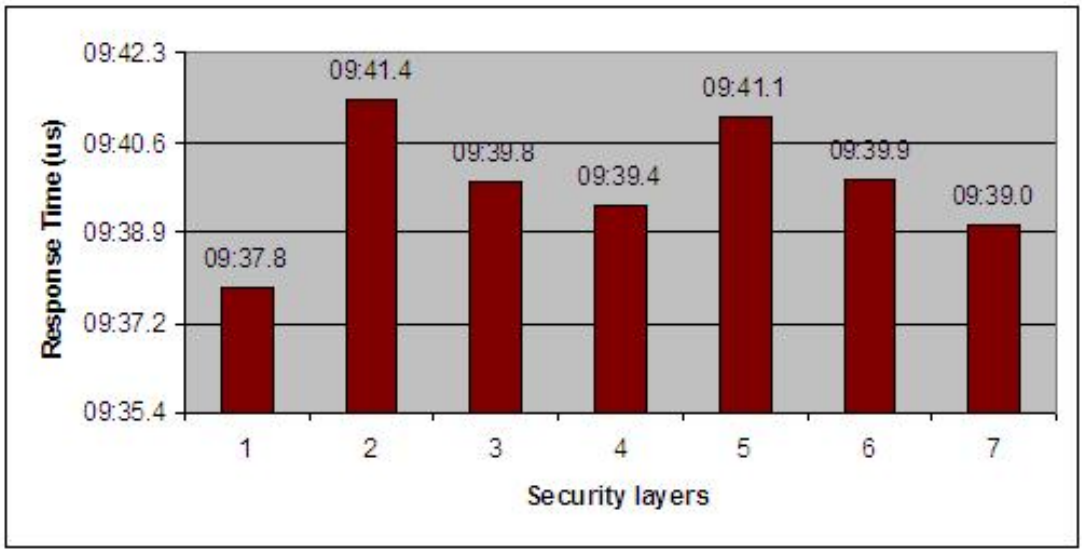

Figure 6.5: Response Time of Wireless LAN Testbed over uncongested UDP traffic.

The graphs in Figure 6.6 and Figure 6.7 shows the traffic throughput of the Testbed, when each chosen security layers were implemented over both uncongested TCP and uncongested UDP traffics. It is clearly shown that the throughput decreases as stronger security is applied. There are approximately 1% difference between the performance of the Testbed with **no security** and the performance of the Testbed with security mechanisms 2 to 7 implementations. This figure has been rounded up since the exact figures were less than $10\mu s$. This also applies to Testbed performance over UDP traffic with decrease of 1% and the standard deviation of 0.20 ms.

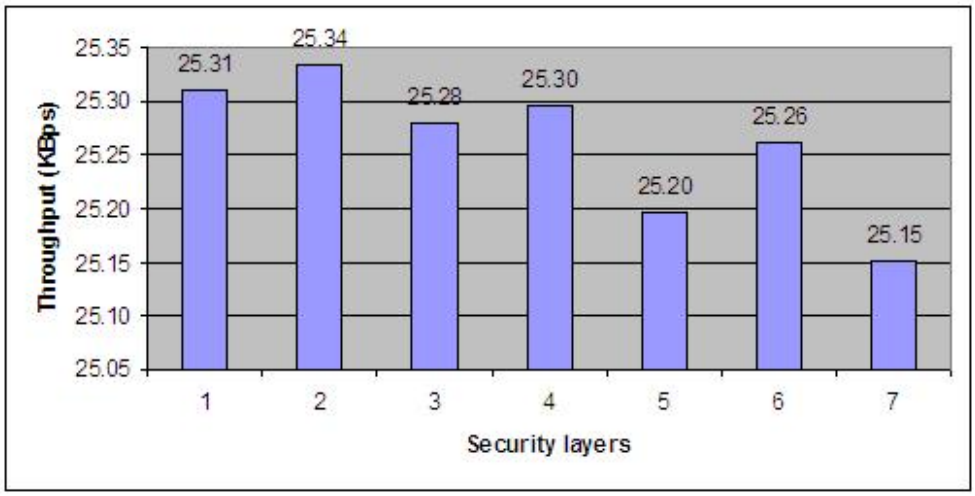

Figure 6.6: Throughput of Wireless LAN Testbed over uncongested TCP traffic.

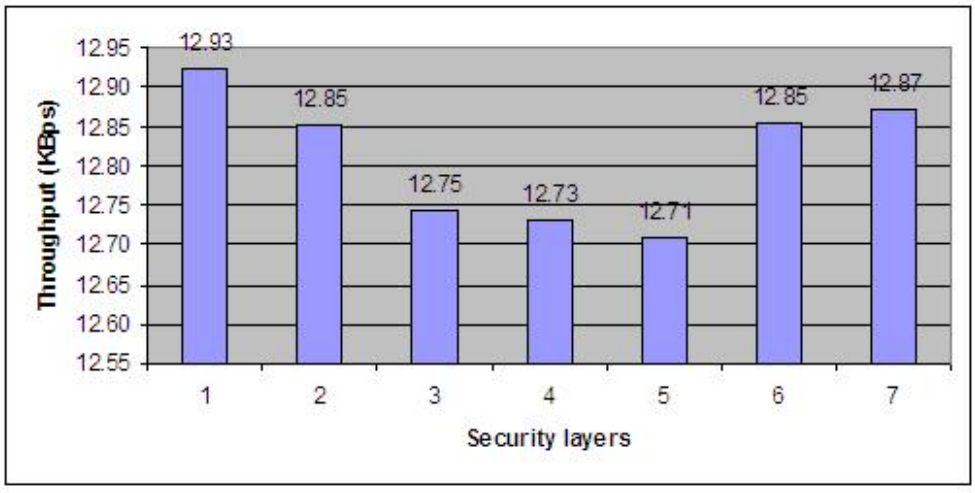

Figure 6.7: Throughput of Wireless LAN Testbed over uncongested UDP traffic.

Figures 6.8 and 6.9 shows the activities of Wireless transmission in a network from *IPTraffic*. It includes the speed, throughput and amount of transmitted packets. These statistics are exported into a simple *.csv* file.

Figure 6.8: IP traffic Generator.

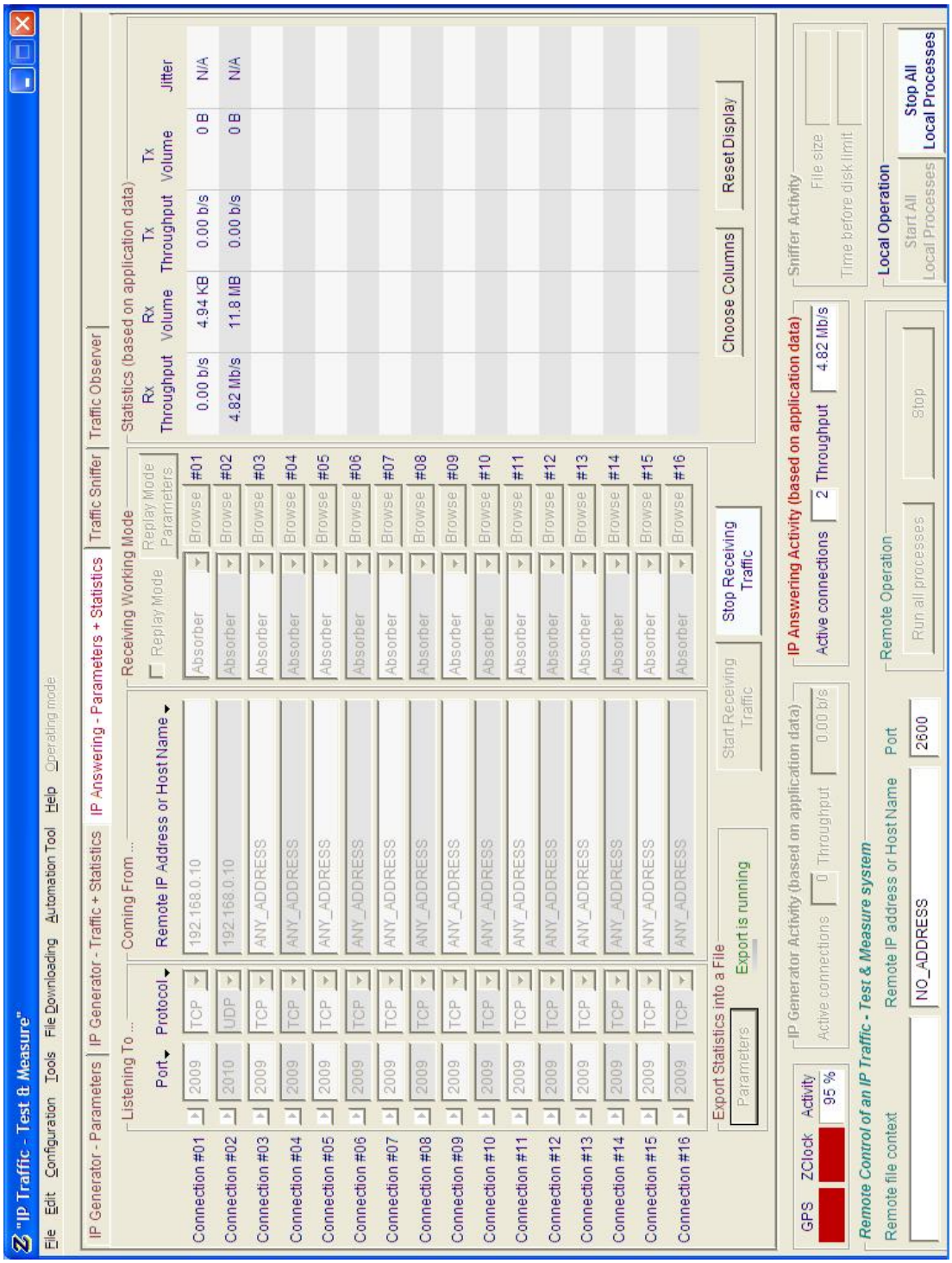

Figure 6.9: IP traffic Answering.

In a congested traffic, the results of the response time and throughput when security mechanisms are implemented over TCP traffic are shown in Figure 6.10 and Figure 6.11 respectively. The response time at security layers 2 to 7 are approximately 1.02% more than the **no security** layer 1. The TCP throughput however has a mean of 441.42 KBps and standard deviation of 4.11 KBps. The performance of the Testbed with implementation of security layers with higher encryption bits, WEP and WPA, display a significant decrease with over 4 KBps to 6 KBps less than the average value.

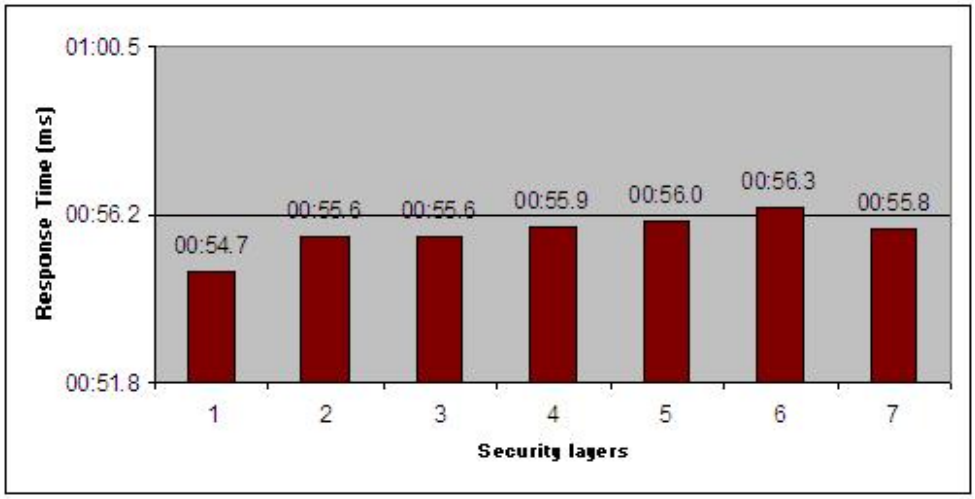

Figure 6.10: Response Time of Wireless LAN Testbed over congested TCP traffic.

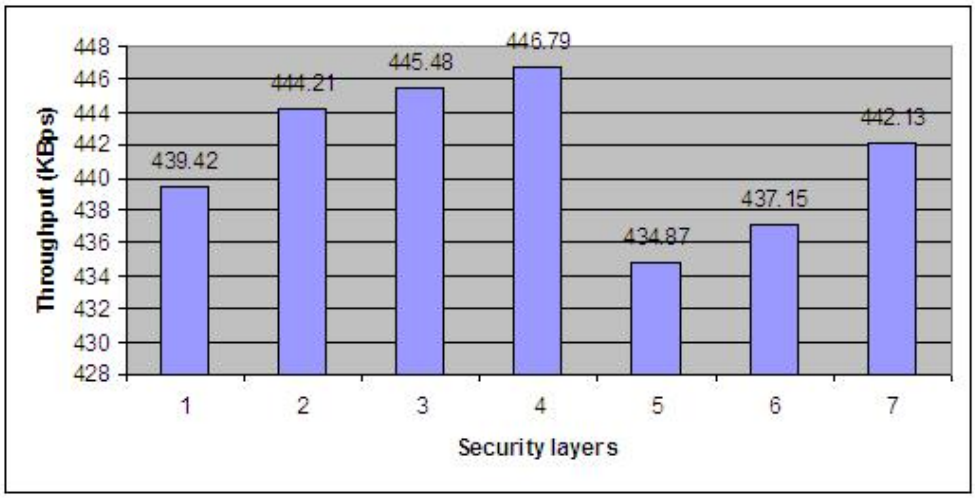

Figure 6.11: Throughput of Wireless LAN Testbed over congested TCP traffic.

Figure 6.12 and Figure 6.13 shows the Wireless LAN Testbed performance when 802.11g mode was activated on the AP. They show the significant decrease in performance since the traffic is to flow from a 802.11g device with 54Mbps bandwidth support into a device with much smaller bandwidth of 11Mbps (802.11b standard). As an analogy, the network traffic flow had a 'bottle-neck' effect.

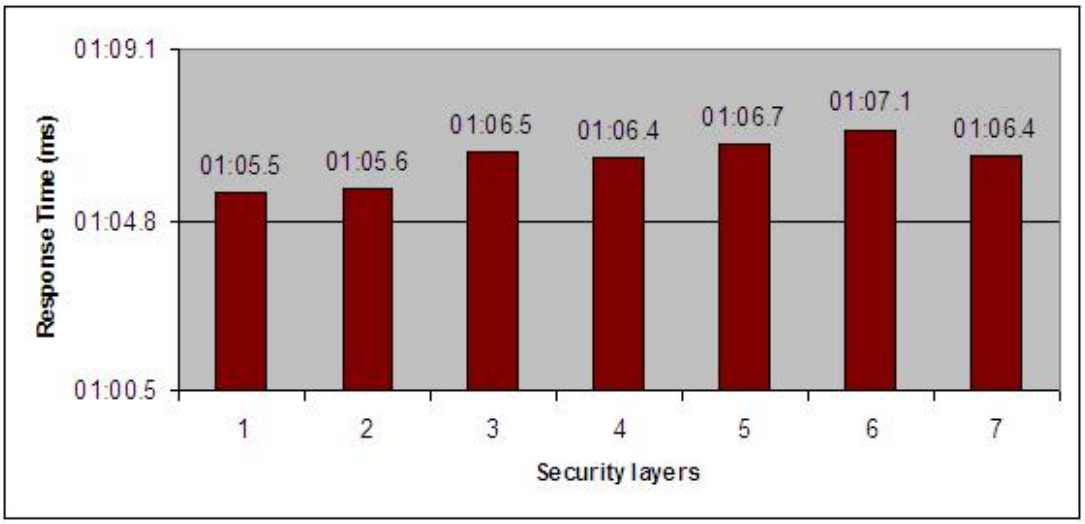

Figure 6.12: Response Time of Wireless LAN Testbed over 802.11g TCP traffic.

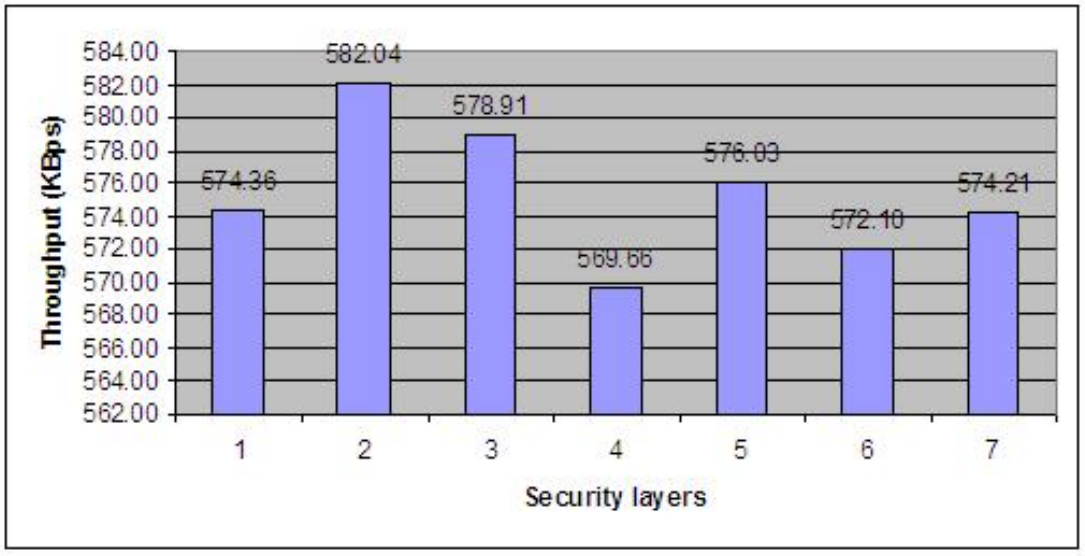

Figure 6.13: Throughput of Wireless LAN Testbed over 802.11g TCP traffic.

A sample of the traffic captured by *Ethereal* is shown in Figure 6.14.

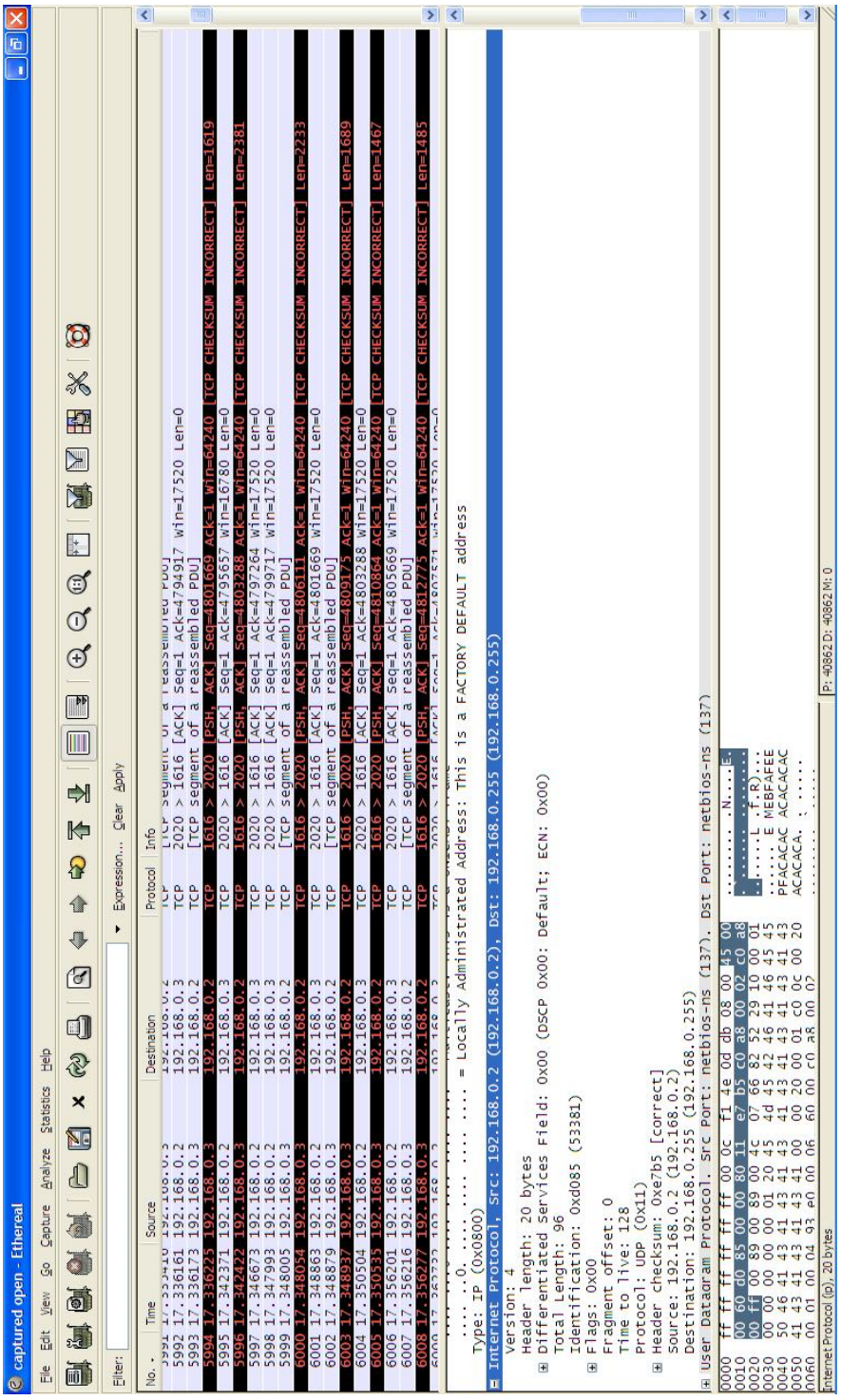

Figure 6.14: TCP traffic captured by *Ethereal*.

As shown in the preceding graphs, the performance of the Wireless LAN Testbed de-

creases when stronger security mechanisms are implemented. This clearly confirms the general trend of Security Performance of a Wireless LAN, as found by Baghaei (2003) and Wong (2003). Moreover, the performance is significantly slower when traffic is congested.

6.4 Discussion

The Wireless LAN Testbed experiment results observed in Section 6.3 showed a significant decrease in performance when WEP authentication and longer bit encryption was applied. A number of factors can be contributing to this as revealed by Gast (2002) and Bing (2001*b*). WEP and WPA uses RC4 stream cipher to encrypt data before transmission as explained in Section 4.1. Chapter 2 earlier mentioned that data transmission with WEP encryption would enable the WEP bit in the Frame Control and increases the data payload by eight bytes. In its frame transmission, the additional bytes consists of four bytes for the IV header and the other four bytes used for the Integrity Check Value (ICV) trailer. The packet will simply be dropped by the AP because of the extra length and there is not enough bandwidth to accommodate it consequently decreases the network performance in a congested traffic.

Furthermore, the comparison between the network performance over TCP and UDP traffics is substantial. In regards to TCP being a connection oriented and UDP is a connection-less environment, the drop rate of packets in a congested traffic leads to poor network performance in both cases. However the performance of TCP should be much lower than UDP since it has a congestion control mechanism. This mechanism detects packet drops in a congested traffic as it will adjust the behaviour of data flow in the network implementing sliding window, a slow-start algorithm, the congestion avoidance algorithm, the fast retransmit or fast recovery algorithms (Geier 2005*a*).

6.4.1 Limitations

There are several limitations that has been detected in the experiments of this research project. The resultant Testbed (as seen in Section 5.3.4) does not simulate a real world

network. Firstly, the experiments have been conducted in a confined space without interruption from external factors such as interference from other radio frequency or weather conditions. Moreover, the devices involved in the experiments are located close to each other. Thus, able to provide maximum network performance in contrast to a real enterprise network implementation, where physical interference exist such as walls.

Next, the Testbed only consist a single client. The purpose of networking is to provide service to multiple clients and in an enterprise, this may be large numbers ranging from hundreds to thousands. Even though the experiments of this research project have been successful in proving the instability of Wireless LAN Security Performance, the results may be far worse in the real world WLAN. Enterprises for example, USQ, tend to implement a combination of many different security mechanisms together in order to have the optimum network security.

Lastly, the experiments do not include all security possibilities in Wireless networks due to hardware implementation. It also does not include security performance testings on ad-hoc networks. The experiments have been mainly focused on 802.11b and the experiments on 802.11g was minimal. There are many other security mechanisms that would be significant in security performance experiment such as VPN.

Chapter 7

Conclusions and Further Work

7.1 Achievement of Project Objectives

The following objectives have been addressed:

WLAN security performance issue Proof of the WLAN security performance shortcomings over existing technologies has been provided in Chapter 6.

Historical risk assessment Chapter 1 presented a summary of the advantages and disadvantages of Wireless technology.

Current security technologies review Chapter 4 showed the many Wireless security technologies, the problems and effects on Wireless performance according to past projects.

7.2 Recommendations

In many research projects that involve experiments and proving that current technologies are insufficient to the needs of users today, potential solutions are highly recommended. In this research project, the recommendations are made based on the experiments and results (see previous Chapter 6) and potential security solution is thought

to be the better, if not the best, option in securing the Wireless network for enterprises.

802.1x has been widely chosen as the security policy by many enterprises as it addresses the weakness of WEP before the release of later security protocols such as WPA and 802.11i (WPA2). 802.1x is a port-based authentication method that uses a **dynamic** key distribution, supplied by randomly generating two sets of keys (Bing 2001*a*, Vladimirov et al. 2004, IEEE Computer Society 2001*b*, Interlink Networks 2002). The first set of keys is called the session or pairwise keys and it is unique for every time a client is connecting to an AP. This is advantageous as opposed to WEP since WEP authentication uses only one key for all sessions. The second set of keys is called group or groupwise keys that is shared among all in the network and used for encryption. Another differing aspect of 802.1x to WEP is that 802.1x has a fixed 128 bits key length. The Pairwise Master Key (PMK) creates the session key for every client, it is distributed from the RADIUS server and 256 bits long. Similarly, the groupwise master key (GMK) creates the groupwise key for participating devices in the network. When deriving the two sets of keys, the PMK and GMK is used in the procedure of four Extensible Authentication Protocol (EAPOL or EAP) handshake keys, also referred to as the pairwise transient key. 802.1x authentication processes are illustrated in Figure 7.1.

There are many different EAP types:

- EAP-MD5 is the first EAP type to be developed and it is the lowest level of EAP support. It is a duplication of Challenge-handshake authentication protocol (CHAP) which is used by Point to Point Protocol (PPP) servers to authenticate a remote user connecting to an Internet access provider. Thus, EAP-MD5 does not support dynamic WEP key distribution and it is very vulnerable to **man-in-the-middle attacks** described in Chapter 4.
- EAP-TLS (Transport Layer Security) is based on the SSLv3 protocol that incorporates a certificate-based authority. It is the most supported and deployed EAP method because of its invincibility against attacks.
- EAP-LEAP (Lightweight EAP or EAP-Cisco Wireless) is a proprietary EAP type by Cisco Systems. It is a password-based authentication scheme and it

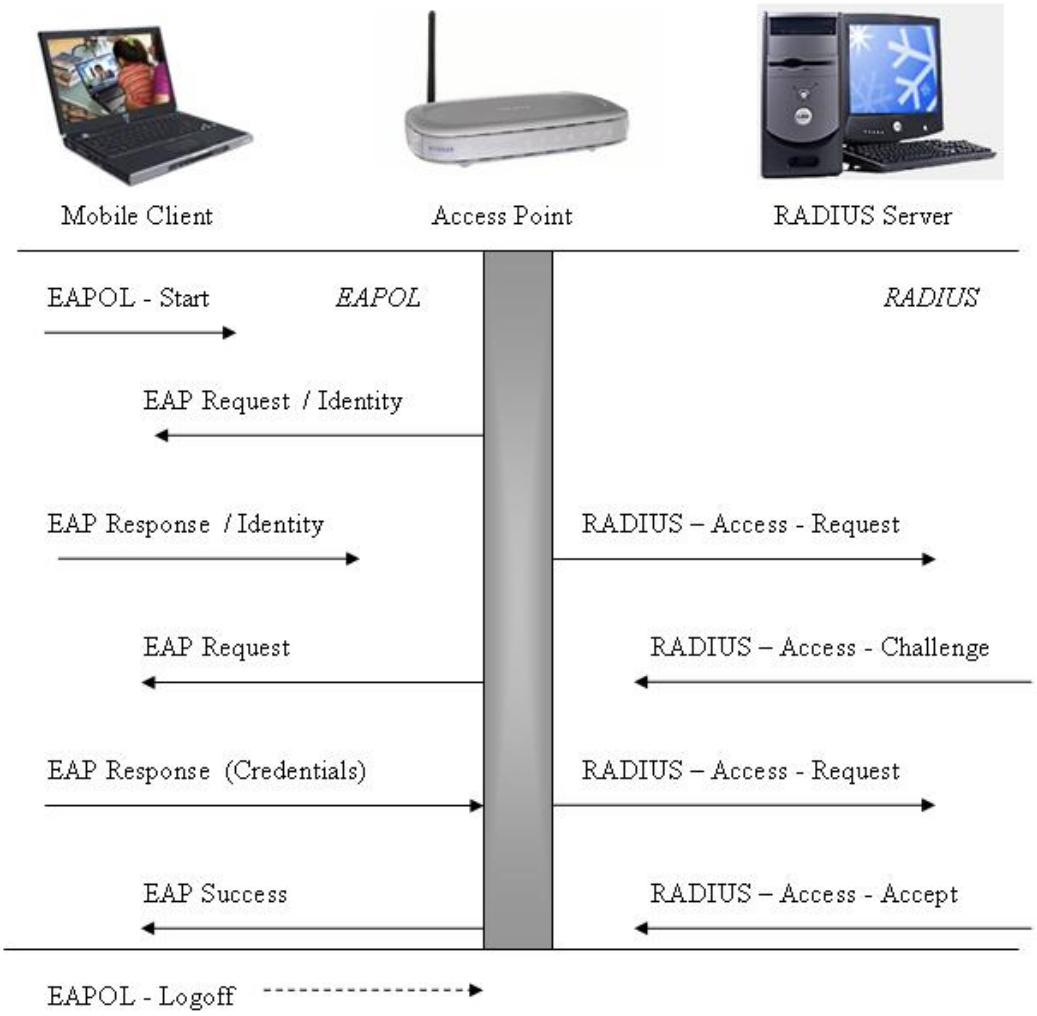

Figure 7.1: EAP Frame Exchange, authentication steps in an EAP security mechanism. (adapted from (ManageEngine 2006)).

gained much popularity because it is the first and only password-based EAP protocol. The functioning of EAP is simply that it is a straightforward challenge-password hash exchange. Firstly, the authentication server sends a challenge text to the client. The client then has to return a hashed password along with the challenge string. Being a password-based authentication method, the strength of this security method lies in the user as the password choice is a major factor and it is vulnerable to **dictionary** and **brute-forcing** attacks that is absent in EAP-TLS, a certificate-based EAP type.

- EAP-PEAP (Protected EAP) and EAP-TTLS (Tunneled Transport Layer Secu-

rity EAP) are the least common but still strong EAP choices. Strong support by manufacturers such as Microsoft and Cisco, might see these protocols to be implemented in more network systems.

Previous Wireless security standards are classified as traditional security network (TDN) but 802.11i is sometimes referred to as Robust Security Network (RSN) (Vladimirov et al. 2004). It is developed to be a new Wireless security standard and completely replace the legacy WEP. This was proposed at the end of 2003 but when 802.11i was not yet finished, WPA emerged that implemented part of 802.11i before its final release that is called WPA2. WPA has similar technical characteristics as 802.11i but with the exclusion of secure ad-hoc networking, secure fast-handoff, secure deauthentication and deassociation and finally the use of Advanced Encryption Standard (AES) encryption algorithm. RC4 cipher encryption method is still a major feature in WPA and this security mechanism only serves infrastructure networks (IEEE Computer Society 2004).

7.3 Further Work

A newly enhanced and secured Testbed would be designed and tested based on the research (see Chapter 4), testings and findings (see Chapter 6) of this research project. This is also the second major objective which has been omitted due to time constraints. The new Testbed would incorporate existing security enhancements used in the industry today, namely the 802.1x standard. Appendix C include the configuration files for implementing the 802.1x security mechanism. It is taken from the *FreeRADIUS* software that administers this 802.1x network security. New prospective security solutions could also be implemented if gained full access to current RC4, RC6 or even other AES encryption algorithms.

Furthermore, Vladimirov et al. (2004) is a main source outlining the methods to 'hack' and 'war drive' Wireless networks. It also includes various methods of 'snooping', 'sniffing' and 'spoofing' Wireless networks and Wireless devices. These activities can satisfy the minor objective of investigating the Wireless security challenges. As this issue is only related to security and not security performance of a Wireless LAN, it was

omitted from the original major objective.

7.4 Conclusions

Wireless technology is becoming very common and the technology is ever growing because of its convenience. Many home users and even enterprises such as the USQ are integrating Wireless into their network infrastructure. It is shown in Chapter 2 and Chapter 3 of the vast development in Wireless standards and Wireless networking.

However, the performance and security of WLAN is often a major concern. Wireless devices with different standards would perform differently with each other. Thus, the design of the Testbed for this research project includes various different devices with different standards. In parallel to the emergence of many IEEE 802.11 Wireless standards, the development of enhancing Wireless security must also be acknowledged. Since many current security mechanisms are known to be unstable and vulnerable, the efforts to rectify this issue has been very positive. They are a range of security protocols in the market today and many devices can implement a combination of security mechanisms, such as MAC address filtering, WEP, WPA and even WPA2 in newer devices.

Furthermore, different security mechanisms would have different effects on the network when applied. It is demonstrated in this research project that current Wireless security protocols may provide good network protection but causes a degradation in the performance. On the other hand, the research into 802.1x and 802.11i found claims that these standards appear to be far better standards than the previous security protocols. In conjunction, they would complement future high performing Wireless standards.

References

Baghaei, N. (2003), *IEEE 802.11 Wireless LAN Security Performance Using Multiple Clients*, University of Canterbury, Christchurch, New Zealand.

Baghaei, N. & Hunt, R. (2004), 'Security performance of loaded ieee802.11b wireless networks', *Security and Performance in Wireless and Mobile Networks* **27**(17), 1746–1756.
`http://www.sciencedirect.com/science/article/B6TYP-4CT4FH7-1/2/68b9fda2137ab807e9031970cbdc43cb`
current June 2006.

Bing, B. (1999), Measured performance of the ieee 802.11 wireless lan, *in* 'Local Computer Networks', Department of Electrical & Computer Engineering, Maryland University, Lowell, MA, pp. 34–42.

Bing, B., ed. (2001*a*), *A User Authentication System For Secure Wireless Communication*, World Scientific, New Jersey.

Bing, B., ed. (2001*b*), *Your 802.11 Wireless Network Has No Clothes*, World Scientific, New Jersey.

Boukerche, A. (2002), Security and fraud detection in mobile and wireless networks, *in* I. Stojmenovic, ed., 'Handbook of Wireless Networks and Mobile Computing', Wiley-Interscience, chapter 14, pp. 309–323.

Campbell, P., Calvert, B. & Boswell, S. (2003), *Security + Guide to Network Fundamentals*, Thomson Course Technology, Boston, Mass.

Carter, T. W. & Whitehead, P. (2004), *Teach Yourself Visually: Wireless Networking*, Wiley Publishing, Hoboken, New Jersey.

Gast, M. S. (2002), *802.11 Wireless Networks: The Definitive Guide*, OŔeilly & Associates, Sebastopol, CA.

Geier, J. (2005*a*), *Wireless Network Security: Protecting Information Resources*, Wireless Networks First-Step, Cisco Press, Indianapolis.

Geier, J. (2005*b*), *Wireless Network Security: Protecting Information Resources*, Wireless Networks First-Step, Cisco Press, Indianapolis, chapter 8, pp. 171–199.

Held, G. (2003), *Securing Wireless LANs: A Practical Guide for Network Managers, LAN Administrators and the Home Office User*, John Wiley & Sons, England, chapter Evolving Encryption, pp. 220–223.

IEEE Computer Society (2001*a*), *IEEE Standards 802.11b*, IEEE.
802.11b-1999 Cor1-2001.pdf.

IEEE Computer Society (2001*b*), *Port-Based Network Access Control*.
802.1X-2001.pdf.

IEEE Computer Society (2003*a*), *IEEE Standards 802.11*, IEEE.
802.11-1999.pdf.

IEEE Computer Society (2003*b*), *IEEE Standards 802.11a*, IEEE.
802.11a-1999.pdf.

IEEE Computer Society (2003*c*), *IEEE Standards 802.11g*, IEEE.
802.11g-2003.pdf.

IEEE Computer Society (2004), *IEEE Standards 802.11i*, IEEE.
802.11i-2004.pdf.

Interlink Networks (2002), *Introduction to 802.1x for Wireless Local Area Networks*.
`http://www.interlinknetworks.com/images/resource/802_1X_for_Wireless_LAN.pdf`.

ManageEngine (2006), 'Eapol-logoff attack'.
`http://www.wi-fi.org/about_overview.php`
current November 2006.

Microsoft Corporation (2006), 'Configuring windows xp ieee 802.11 wireless networks for the home and small business'.
`http://www.microsoft.com/technet/prodtechnol/winxppro/maintain/wifisoho.mspx`
current November 2006.

Miller, S. S. (2003), *WiFi Security*, McGraw Hill.

Nichols, R. K. & Lekkas, P. C. (2002), *Wireless Security: Models, Threats, and Solutions*, McGraw-Hill, New York.

Olexa, R. (2005), *Implementing 802.11, 802.16 and 802.20 Wireless Networks: Planning, Troubleshooting and Operations*, Elsevier, Amsterdam.

Pyramid Research (2005), *Wi-Fi Gaining Traction*, Businessweek Online.
`http://www.businessweek.com/technology/tech_stats/wifi051003.htm`
current May 2006.

Shaw, R. (2003), *Wireless Networking Made Easy: Everything You Need to Know to Build Your Own PANs, LANs, and WANs*, American Management Association, New York.

University of Southern Queensland (2003), 'Division of ict services'.
`http://www.usq.edu.au/ict/default.htm`
current November 2006.

Vladimirov, A. A., Gavrilenko, K. V. & Mikhailovsky, A. A. (2004), *Wi-Foo: The Secrets of Wireless Hacking*, Addison-Wesley, Boston.

Wheat, J., Hiser, R., Tucker, J., Neely, A. & McCullough, A. (2001), *Designing Wireless Network: Understand How Wireless Communication Works*, Syngress Publishing Inc., Rockland, MA.

Wi-Fi Alliance (2006), 'Get to know the alliance', *About the Alliance* .
`http://www.wi-fi.org/about_overview.php`
current June 2006.

Wong, J. (2003), *Performance Investigation of Secure 802.11 Wireless LANs: Raising the Security Bar to Which Level?*, University of Canterbury, Christchurch, NZ.

World Mobile Market (2005), Research & Consultancy Outsourcing Services (RNCOS).

`http://www.marketsearch.com`

current May 2006.

Yeo, V. (2005), '802.11 tutorial'.

`http://www.geocities.com/backgndtest/wlan_tut.html`

current October 2006.

ZyTrax, Inc. (2006), '802.11 mac (media access control)'.

`http://www.zytrax.com/tech/wireless/802_mac.htm`

current October 2006.

Appendix A

Project Specification

University of Southern Queensland
Faculty of Engineering and Surveying

ENG4111/4112 Research Project
PROJECT SPECIFICATION

FOR: KAMARUL FAISAL KAMARUDDIN
TOPIC: SECURITY AND PERFORMANCE OF A WIRELESS LAN TESTBED
SUPERVISOR: Dr. Hong Zhou

ENROLMENT: ENG 4111 – S1, D, 2006;
ENG 4112 – S2, D, 2006

PROJECT AIM: This project is focused on the WLAN security fundamentals for an enterprise. The project will assess the historical risks, review the current security technologies, and investigate the challenges and potential security solutions. A wireless testbed will be built and security issues and technologies will be examined.

SPONSORSHIP: ECRP Project, Faculty of Engineering and Surveying

PROGRAMME:

1. Research the background information relating to Wireless Internet (project direction). Find information on Wireless LAN implementation in an enterprise.
2. Design a Wireless LAN testbed. Also, gather appropriate devices for test. Perform test upon completion of tasks.
3. Analyse the performance of Wireless LAN testbed based on test results.
4. Identify and assess security problems in Wireless LAN testbed.
5. Propose suitable improvement on security and performance of Wireless LAN testbed. This may incorporate existing security enhancement and/or prospective security solution.

As time permits

6. Design and test new Wireless LAN testbed based on suggestions from (5).

AGREED: _____ (student)_____ (supervisor)

(dated) ___ / ___ / ___

Appendix B

Radio Frequency Signals

Radio spectrum is a main resource where Wireless devices have to operate on. Wireless devices are constrained to utilise certain frequency bands only (Gast 2002, Wheat et al. 2001). Each band has particular *bandwidth* and this has been classified to be the measure of data capacity that can flow through. Based on mathematical properties, information theory and signal processing, bigger bandwidth slices can be used to transmit much more data than smaller bandwidth slots. For example, an analog mobile telephony channel only requires 20kHz bandwidth whereas more complex Television signals needs larger bandwidth of about 6MHz for operation.

The radio spectrum is highly controlled by regulatory bodies through *licensing processes*. Consequently, a range of different frequency bands are allocated and significantly, the ISM bands are specified. ISM is the abbreviation of industrial, scientific and medical. The ISM bands are set aside for devices that are related to industrial or scientific processes and also for devices that are used by medical equipments. ISM bands are among a range of frequency bands specified to prevent overlapping between different applications that uses the commonly available radio spectrum in the air. ISM bands are also licence-free. For example, microwave ovens are ISM-band devices which operates in the 2.4GHz ISM band. IEEE 802.11 devices also operates over ISM bands particularly the S-band ISM.

Figure B.1 and Figure B.2 are snipets of the radio spectrum allocated to ensure in-

teroperability of different Wireless devices and prevent overlapping between different applications that uses the radio spectrum.

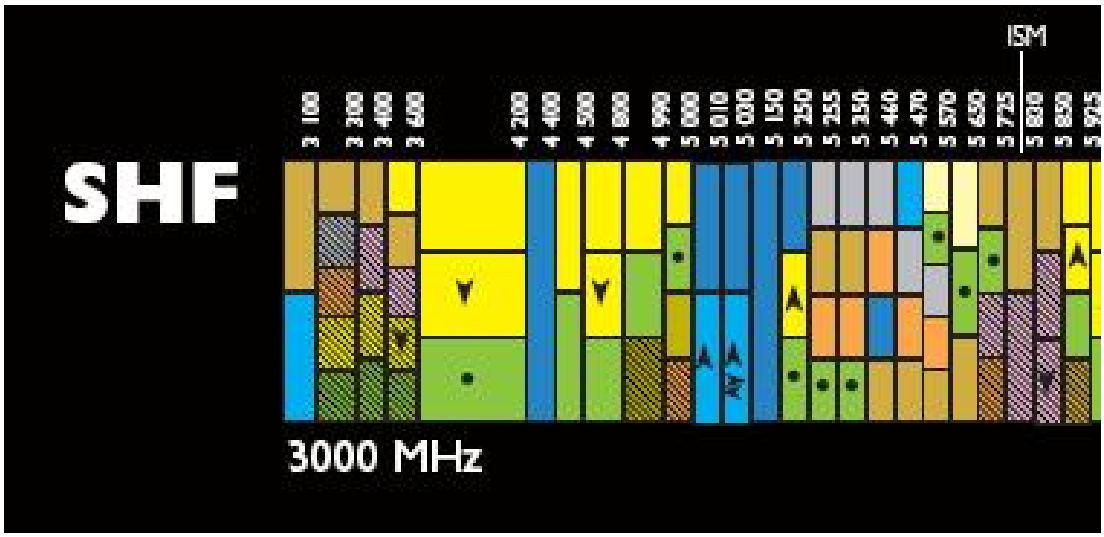

Figure B.1: Super High Frequency (SHF) region. Adapted from Australian radiofrequency spectrum allocations chart

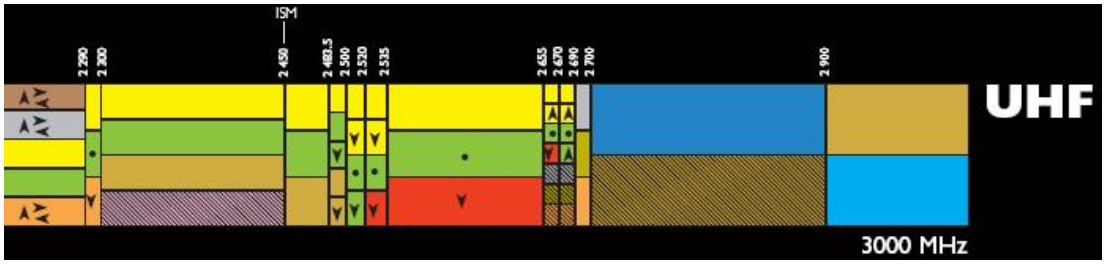

Figure B.2: Ultra High Frequency (UHF) region. Adapted from Australian radiofrequency spectrum allocations chart

In the United States of America, the emission of Radio Frequency (RF) signals from devices is governed by the Federal Communications Commission (FCC) (Olexa 2005, Wheat et al. 2001). Manufacturers have to adhere to the Part 15 rules that administers power output, equipment and antenna configurations that can be used in the unlicensed bands, the ISM bands mentioned earlier.

In more detail, RF data communication involves two main component, the transmitter and receiver, as seen in Figure B.3 and Figure B.4 (Olexa 2005, Wheat et al. 2001). Firstly at the transmitter, RF base signal would be generated at the desired operating frequency. For example, signal with 2.4GHz frequency or 5GHz frequency signal for 802.11a or 802.11b respectively. The oscillator performs this task by using amplification,

feedback and resonance principles. Essentially, the oscillator is an amplifier that has a positive feedback (i.e. some of its output signal transferred back to the input). This would soon make the amplifier stabilise around the resonant frequency of the components in the amplifier and the feedback loop which in turn causes resonance. Resonance is the electrical and magnetic vibration of the AC signal. The synthesiser is used to remove many of the oscillator design complexities and reduces the cost of frequency generation substantially. The next block is the power amplifier and the function in Wireless data transfer has been explained in Chapter 2. The antenna is normally connected to the transmitter and receiver by a coaxial cable or often called *coax*. Antennas come in various shapes and sizes but the perfect antenna is known as the isotropic radiator. Isotropic radiator antennas generate a perfect sphere of energy that surrounds it and of equal intensity in all directions. However, it is only theoretical and does not exist in practical. The dipole and isotrope are omnidirectional antennas that are commonly used in transmitters and receivers next to directional antennas.

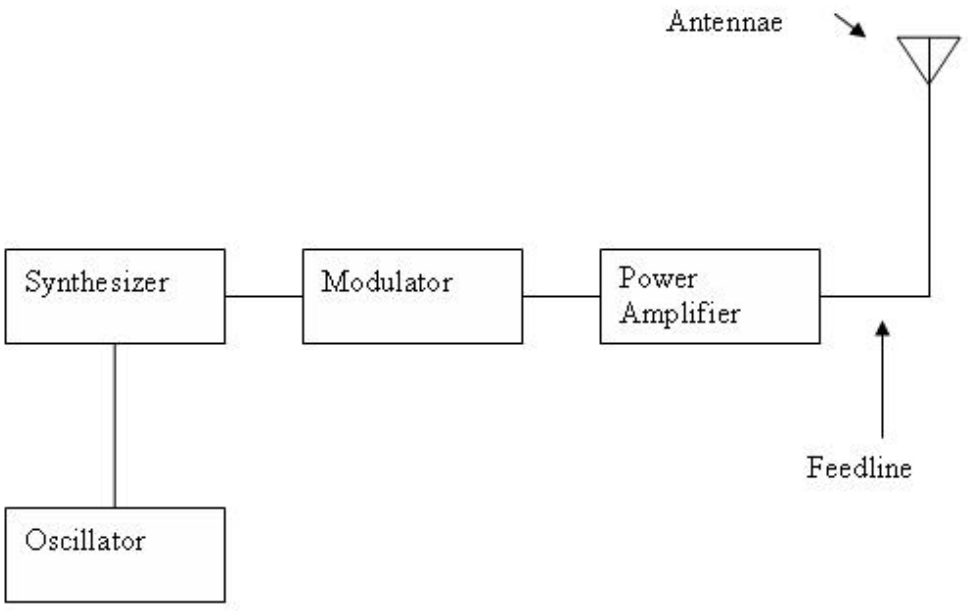

Figure B.3: Transmitter block diagram. Adapted from (Olexa 2005)

In addition to amplifying the transmitted signal, a filter is necessary in a receiver module since RF signal that has traveled through the atmosphere would be attenuated and polluted with noise or unwanted signals from other sources. Bandpass filters are

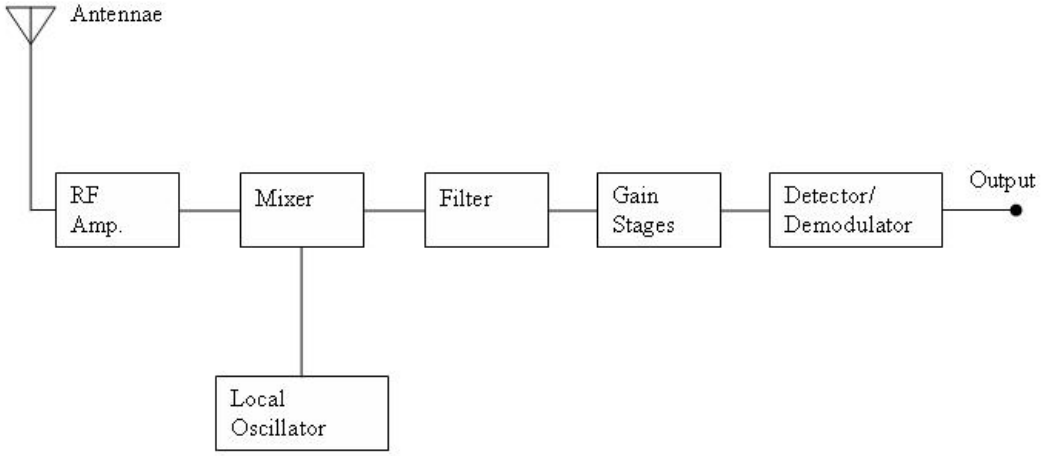

Figure B.4: Receiver block diagram. Adapted from (Olexa 2005)

primarily used for this purpose to remove energy that is outside the bandwidth. Finally, the RF amplifier is a weak-signal amplifier, used to amplify extremely small input signals that arrive at the Receiver.

Appendix C

Configurations

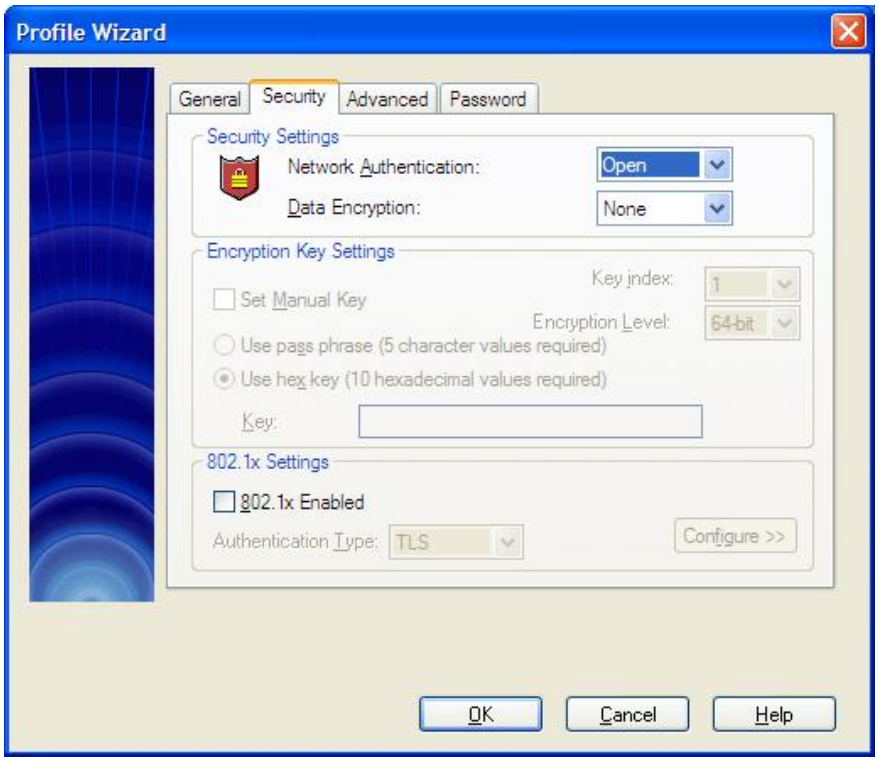

Figure C.1: Client open.

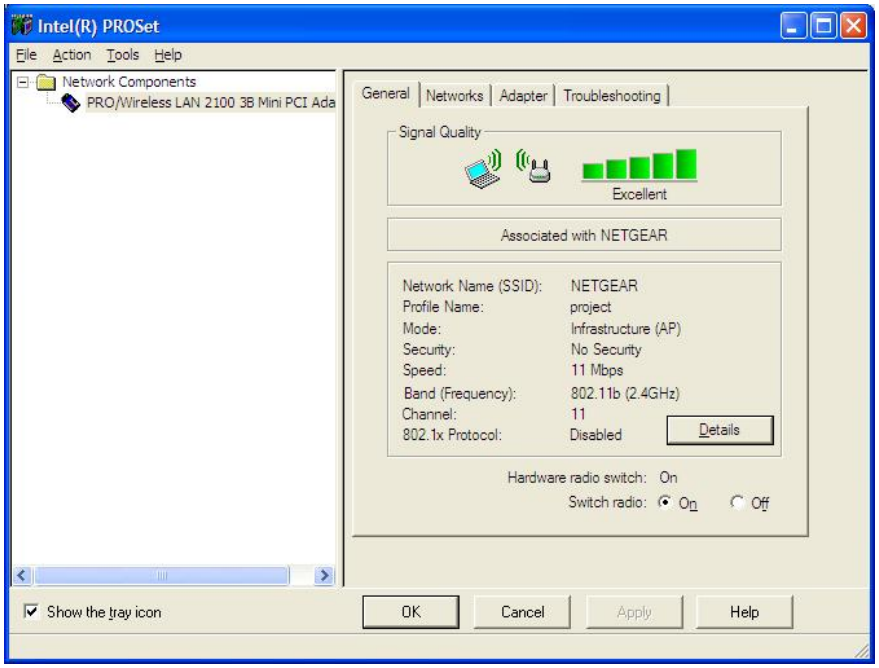

Figure C.2: Laptop open.

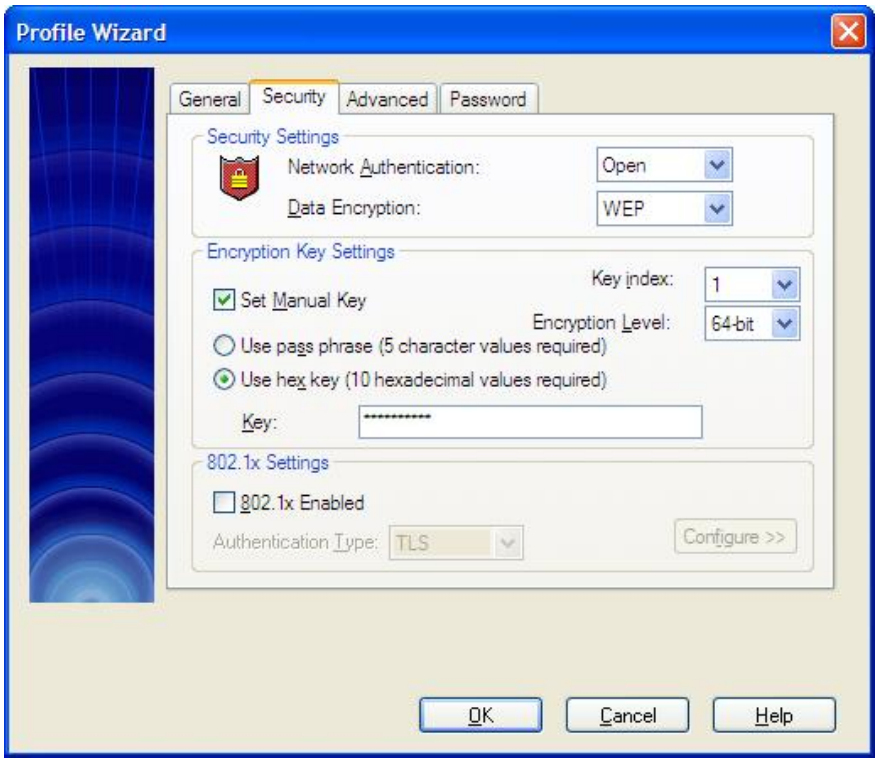

Figure C.3: Laptop open WEP 64bit.

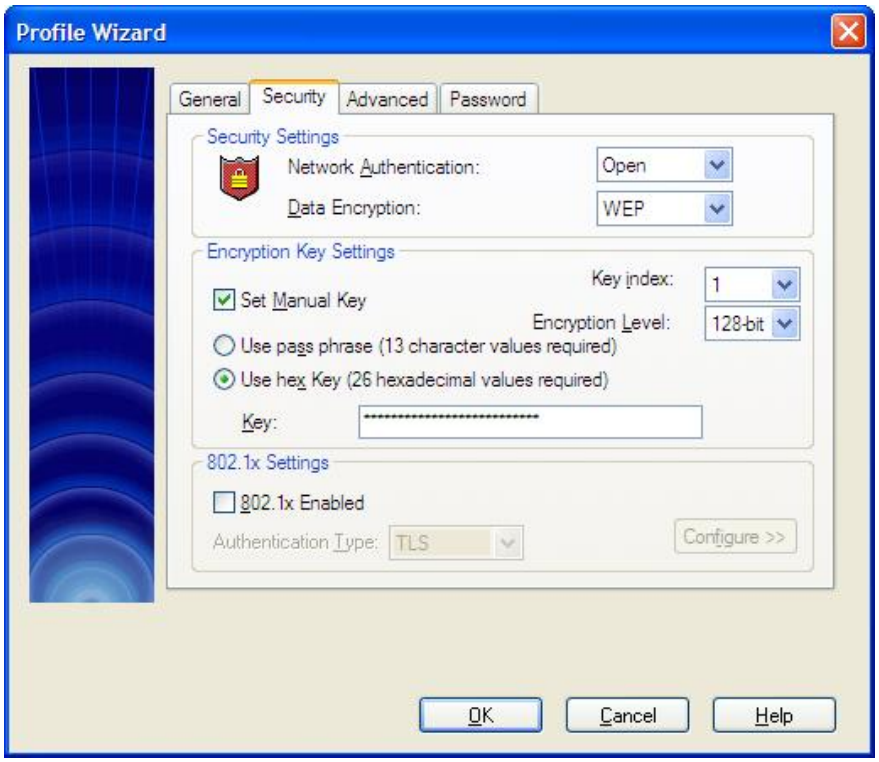

Figure C.4: Laptop open WEP 128bit.

Figure C.5: Laptop shared WEP 64bit.

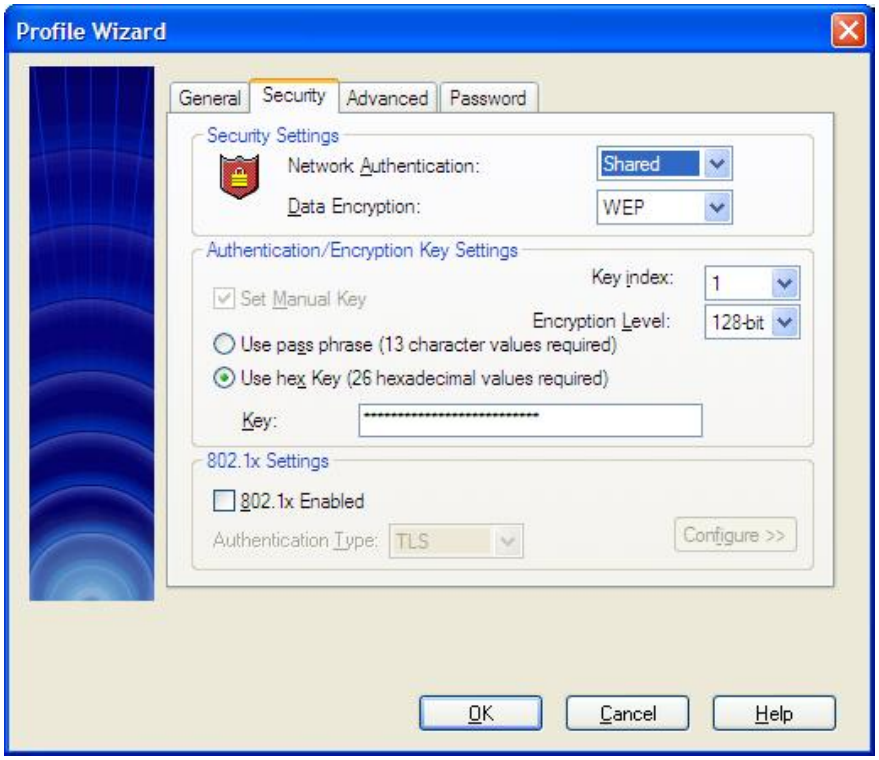

Figure C.6: Laptop shared WEP 128bit.

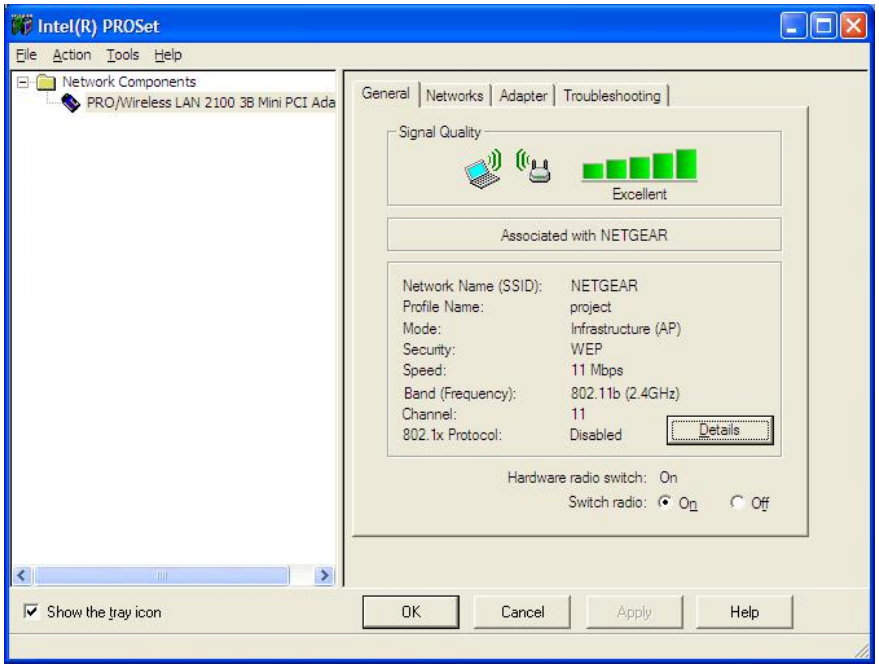

Figure C.7: Laptop WEP.

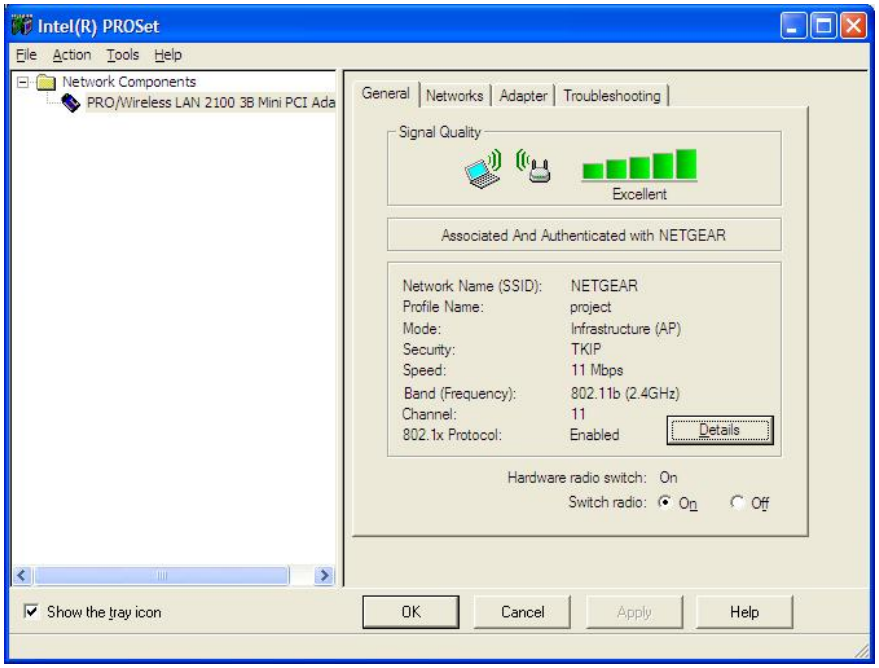

Figure C.8: Laptop WPA.

Figure C.9: Laptop WPA-PSK (TKIP).

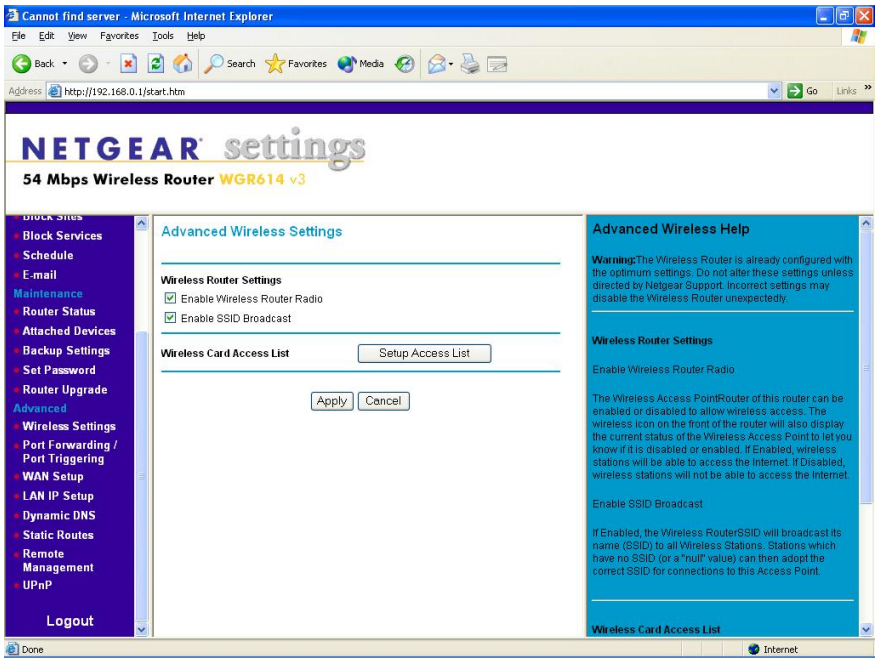

Figure C.10: Netgear Advanced Setting.

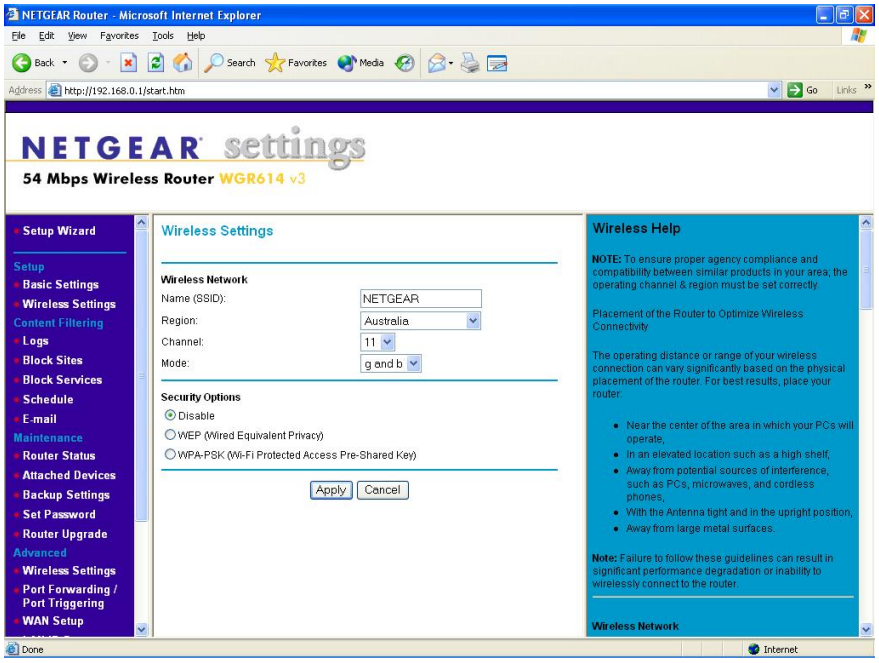

Figure C.11: Netgear AP open.

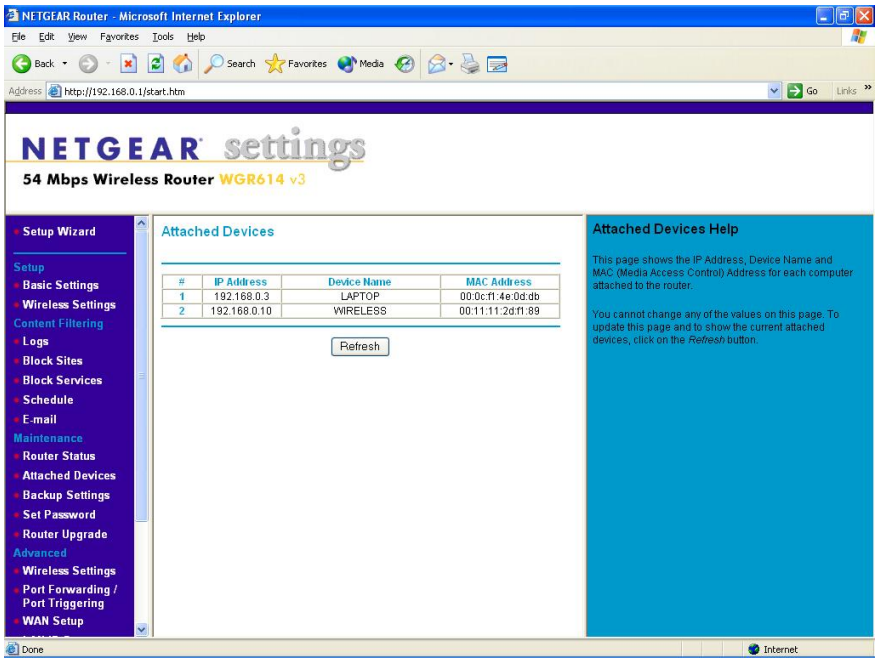

Figure C.12: Netgear Attached Devices Wireless cable Netgear wireless Laptop.

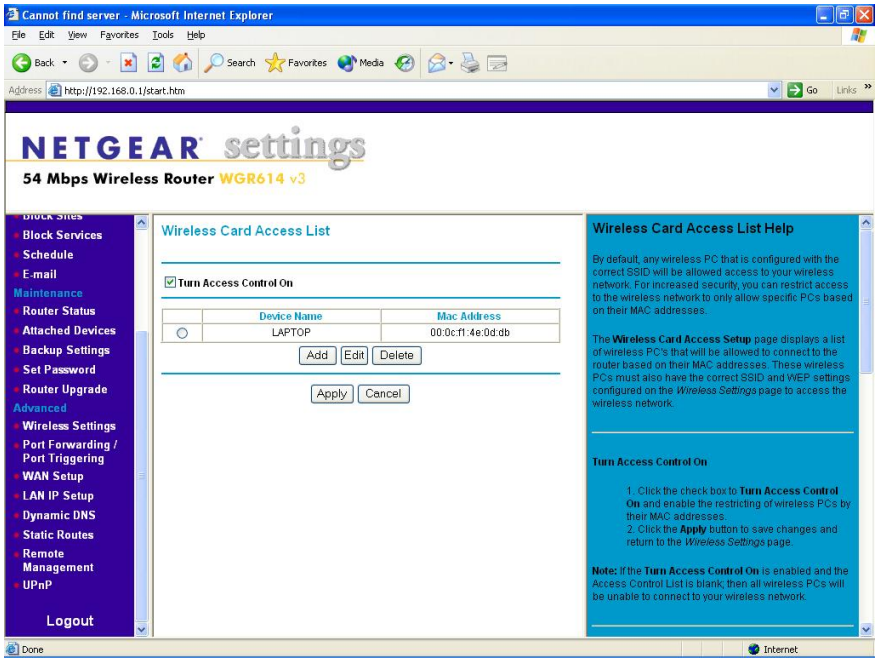

Figure C.13: Netgear MAC access.

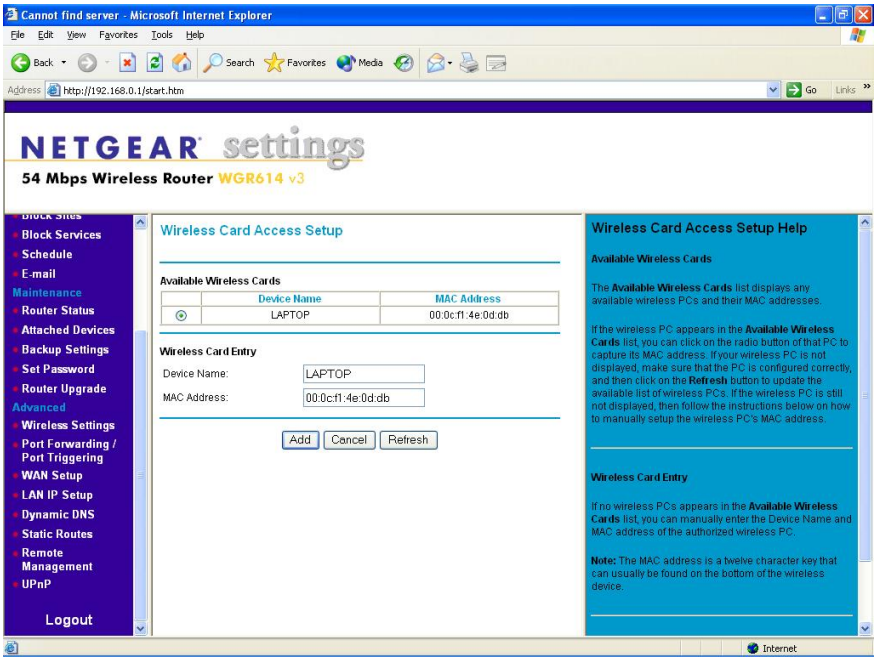

Figure C.14: Netgear MAC access setup.

Appendix D

FreeRADIUS Configuration Files

Includes `clients.conf`, `eap.conf`, `radiusd.conf` and `users` configuration files for implementing 802.1x security mechanism from FreeRADIUS server program.

D.1 The `radiusd.conf` FreeRADIUS configuration file

Listing D.1: FreeRADIUS 802.1x execution codes.

```
##
## radiusd.conf -- FreeRADIUS server configuration file.
##
##       http://www.freeradius.org/
##       $Id: radiusd.conf.in,v 1.188.2.3 2005/02/07 19:52:05 aland Exp $
##

#       The location of other config files and
#       logfiles are declared in this file
#
#       Also general configuration for modules can be done
#       in this file, it is exported through the API to
#       modules that ask for it.
#
#       The configuration variables defined here are of the form ${foo}
#       They are local to this file, and do not change from request to
#       request.
#
#       The per-request variables are of the form %{Attribute-Name}, and
#       are taken from the values of the attribute in the incoming
#       request.  See 'doc/variables.txt' for more information.

prefix = ..
exec_prefix = ${prefix}
sysconfdir = ${prefix}/etc
localstatedir = ${prefix}/var
sbindir = ${exec_prefix}/bin
logdir = ${localstatedir}/log/radius
raddbdir = ${sysconfdir}/raddb
certsdir = ${sysconfdir}/raddb/certs/FreeRADIUS.net/DemoCerts
radacctdir = ${logdir}/radacct

#  Location of config and logfiles.
confdir = ${raddbdir}
run_dir = ${localstatedir}/run/radiusd

#
#  The logging messages for the server are appended to the
#  tail of this file.
#
log_file = ${logdir}/radius.log

#
# libdir: Where to find the rlm_* modules.
#
#   This should be automatically set at configuration time.
```

```
#
#    If the server builds and installs, but fails at execution time
#    with an 'undefined symbol' error, then you can use the libdir
#    directive to work around the problem.
#
#    The cause is usually that a library has been installed on your
#    system in a place where the dynamic linker CANNOT find it.  When
#    executing as root (or another user), your personal environment MAY
#    be set up to allow the dynamic linker to find the library.  When
#    executing as a daemon, FreeRADIUS MAY NOT have the same
#    personalized configuration.
#
#    To work around the problem, find out which library contains that symbol,
#    and add the directory containing that library to the end of 'libdir',
#    with a colon separating the directory names.  NO spaces are allowed.
#
#    e.g. libdir = /usr/local/lib:/opt/package/lib
#
#    You can also try setting the LD_LIBRARY_PATH environment variable
#    in a script which starts the server.
#
#    If that does not work, then you can re-configure and re-build the
#    server to NOT use shared libraries, via:
#
#         ./configure --disable-shared
#         make
#         make install
#
libdir = ${exec_prefix}/lib

#  pidfile: Where to place the PID of the RADIUS server.
#
#  The server may be signalled while it's running by using this
#  file.
#
#  This file is written when ONLY running in daemon mode.
#
#  e.g.:  kill -HUP `cat /var/run/radiusd/radiusd.pid`
#
pidfile = ${run_dir}/radiusd.pid

# user/group: The name (or #number) of the user/group to run radiusd as.
#
#    If these are commented out, the server will run as the user/group
#    that started it.  In order to change to a different user/group, you
#    MUST be root ( or have root privleges ) to start the server.
#
#    We STRONGLY recommend that you run the server with as few permissions
#    as possible.  That is, if you're not using shadow passwords, the
#    user and group items below should be set to 'nobody'.
#
#     On SCO (ODT 3) use "user = nouser" and "group = nogroup".
#
#  NOTE that some kernels refuse to setgid(group) when the value of
#  (unsigned)group is above 60000; don't use group nobody on these systems!
#
#  On systems with shadow passwords, you might have to set 'group = shadow'
#  for the server to be able to read the shadow password file.  If you can
#  authenticate users while in debug mode, but not in daemon mode, it may be
#  that the debugging mode server is running as a user that can read the
#  shadow info, and the user listed below can not.
#
#user = nobody
#group = nobody

#  max_request_time: The maximum time (in seconds) to handle a request.
#
#  Requests which take more time than this to process may be killed, and
#  a REJECT message is returned.
#
#  WARNING: If you notice that requests take a long time to be handled,
#  then this MAY INDICATE a bug in the server, in one of the modules
#  used to handle a request, OR in your local configuration.
#
#  This problem is most often seen when using an SQL database.  If it takes
#  more than a second or two to receive an answer from the SQL database,
#  then it probably means that you haven't indexed the database.  See your
#  SQL server documentation for more information.
#
#  Useful range of values: 5 to 120
#
max_request_time = 30

#  delete_blocked_requests: If the request takes MORE THAN 'max_request_time'
#  to be handled, then maybe the server should delete it.
#
#  If you're running in threaded, or thread pool mode, this setting
#  should probably be 'no'.  Setting it to 'yes' when using a threaded
#  server MAY cause the server to crash!
#
delete_blocked_requests = no

#  cleanup_delay: The time to wait (in seconds) before cleaning up
#  a reply which was sent to the NAS.
#
#  The RADIUS request is normally cached internally for a short period
#  of time, after the reply is sent to the NAS.  The reply packet may be
#  lost in the network, and the NAS will not see it.  The NAS will then
#  re-send the request, and the server will respond quickly with the
#  cached reply.
#
#  If this value is set too low, then duplicate requests from the NAS
#  MAY NOT be detected, and will instead be handled as seperate requests.
#
```

```
#  If this value is set too high, then the server will cache too many
#  requests, and some new requests may get blocked.  (See 'max_requests'.)
#
#  Useful range of values: 2 to 10
#
cleanup_delay = 5

#  max_requests: The maximum number of requests which the server keeps
#  track of.  This should be 256 multiplied by the number of clients.
#  e.g. With 4 clients, this number should be 1024.
#
#  If this number is too low, then when the server becomes busy,
#  it will not respond to any new requests, until the 'cleanup_delay'
#  time has passed, and it has removed the old requests.
#
#  If this number is set too high, then the server will use a bit more
#  memory for no real benefit.
#
#  If you aren't sure what it should be set to, it's better to set it
#  too high than too low.  Setting it to 1000 per client is probably
#  the highest it should be.
#
#  Useful range of values: 256 to infinity
#
max_requests = 1024

#  bind_address:  Make the server listen on a particular IP address, and
#  send replies out from that address.  This directive is most useful
#  for machines with multiple IP addresses on one interface.
#
#  It can either contain "*", or an IP address, or a fully qualified
#  Internet domain name.  The default is "*"
#
#  As of 1.0, you can also use the "listen" directive.  See below for
#  more information.
#
bind_address = *

#  port: Allows you to bind FreeRADIUS to a specific port.
#
#  The default port that most NAS boxes use is 1645, which is historical.
#  RFC 2138 defines 1812 to be the new port.  Many new servers and
#  NAS boxes use 1812, which can create interoperability problems.
#
#  The port is defined here to be 0 so that the server will pick up
#  the machine's local configuration for the radius port, as defined
#  in /etc/services.
#
#  If you want to use the default RADIUS port as defined on your server,
#  (usually through 'grep radius /etc/services') set this to 0 (zero).
#
#  A port given on the command-line via '-p' over-rides this one.
#
#  As of 1.0, you can also use the "listen" directive.  See below for
#  more information.
#
port = 0

#
#  By default, the server uses "bind_address" to listen to all IP's
#  on a machine, or just one IP.  The "port" configuration is used
#  to select the authentication port used when listening on those
#  addresses.
#
#  If you want the server to listen on additional addresses, you can
#  use the "listen" section.  A sample section (commented out) is included
#  below.  This "listen" section duplicates the functionality of the
#  "bind_address" and "port" configuration entries, but it only listens
#  for authentication packets.
#
#  If you comment out the "bind_address" and "port" configuration entries,
#  then it becomes possible to make the server accept only accounting,
#  or authentication packets.  Previously, it always listened for both
#  types of packets, and it was impossible to make it listen for only
#  one type of packet.
#
#listen {
        #  IP address on which to listen.
        #  Allowed values are:
        #        dotted quad (1.2.3.4)
        #        hostname    (radius.example.com)
        #        wildcard    (*)
#       ipaddr = *

        #  Port on which to listen.
        #  Allowed values are:
        #        integer port number (1812)
        #        0 means "use /etc/services for the proper port"
#       port = 0

        #  Type of packets to listen for.
        #  Allowed values are:
        #        auth    listen for authentication packets
        #        acct    listen for accounting packets
        #
#       type = auth
#}

#  hostname_lookups: Log the names of clients or just their IP addresses
#  e.g., www.freeradius.org (on) or 206.47.27.232 (off).
#
#  The default is 'off' because it would be overall better for the net
#  if people had to knowingly turn this feature on, since enabling it
#  means that each client request will result in AT LEAST one lookup
#  request to the nameserver.   Enabling hostname_lookups will also
```

```
#  mean that your server may stop randomly for 30 seconds from time
#  to time, if the DNS requests take too long.
#
#  Turning hostname lookups off also means that the server won't block
#  for 30 seconds, if it sees an IP address which has no name associated
#  with it.
#
#  allowed values: {no, yes}
#
hostname_lookups = no

#  Core dumps are a bad thing.  This should only be set to 'yes'
#  if you're debugging a problem with the server.
#
#  allowed values: {no, yes}
#
allow_core_dumps = no

#  Regular expressions
#
#  These items are set at configure time.  If they're set to "yes",
#  then setting them to "no" turns off regular expression support.
#
#  If they're set to "no" at configure time, then setting them to "yes"
#  WILL NOT WORK.  It will give you an error.
#
regular_expressions     = yes
extended_expressions    = yes

#  Log the full User-Name attribute, as it was found in the request.
#
# allowed values: {no, yes}
#
log_stripped_names = yes

#  Log authentication requests to the log file.
#
#  allowed values: {no, yes}
#
log_auth = yes

#  Log passwords with the authentication requests.
#  log_auth_badpass  - logs password if it's rejected
#  log_auth_goodpass - logs password if it's correct
#
#  allowed values: {no, yes}
#
log_auth_badpass = yes
log_auth_goodpass = yes

# usercollide:  Turn "username collision" code on and off.  See the
# "doc/duplicate-users" file
#
#  WARNING
#  !!!!!!!  Setting this to "yes" may result in the server behaving
#  !!!!!!!  strangely.  The "username collision" code will ONLY work
#  !!!!!!!  with clear-text passwords.  Even then, it may not do what
#  !!!!!!!  you want, or what you expect.
#  !!!!!!!
#  !!!!!!!  We STRONGLY RECOMMEND that you do not use this feature,
#  !!!!!!!  and that you find another way of acheiving the same goal.
#  !!!!!!!
#  !!!!!!!  e,g. module fail-over.  See 'doc/configurable_failover'
#  WARNING
#
usercollide = no

# lower_user / lower_pass:
# Lower case the username/password "before" or "after"
# attempting to authenticate.
#
#  If "before", the server will first modify the request and then try
#  to auth the user.  If "after", the server will first auth using the
#  values provided by the user.  If that fails it will reprocess the
#  request after modifying it as you specify below.
#
#  This is as close as we can get to case insensitivity.  It is the
#  admin's job to ensure that the username on the auth db side is
#  *also* lowercase to make this work
#
# Default is 'no' (don't lowercase values)
# Valid values = "before" / "after" / "no"
#
lower_user = no
lower_pass = no

# nospace_user / nospace_pass:
#
#  Some users like to enter spaces in their username or password
#  incorrectly.  To save yourself the tech support call, you can
#  eliminate those spaces here:
#
# Default is 'no' (don't remove spaces)
# Valid values = "before" / "after" / "no" (explanation above)
#
nospace_user = no
nospace_pass = no

#  The program to execute to do concurrency checks.
checkrad = ${sbindir}/checkrad

# SECURITY CONFIGURATION
#
#  There may be multiple methods of attacking on the server.  This
#  section holds the configuration items which minimize the impact
#  of those attacks
```

```
#
security {
        #
        #  max_attributes: The maximum number of attributes
        #  permitted in a RADIUS packet.  Packets which have MORE
        #  than this number of attributes in them will be dropped.
        #
        #  If this number is set too low, then no RADIUS packets
        #  will be accepted.
        #
        #  If this number is set too high, then an attacker may be
        #  able to send a small number of packets which will cause
        #  the server to use all available memory on the machine.
        #
        #  Setting this number to 0 means "allow any number of attributes"
        max_attributes = 200

        #
        #  delayed_reject: When sending an Access-Reject, it can be
        #  delayed for a few seconds.  This may help slow down a DoS
        #  attack.  It also helps to slow down people trying to brute-force
        #  crack a users password.
        #
        #  Setting this number to 0 means "send rejects immediately"
        #
        #  If this number is set higher than 'cleanup_delay', then the
        #  rejects will be sent at 'cleanup_delay' time, when the request
        #  is deleted from the internal cache of requests.
        #
        #  Useful ranges: 1 to 5
        reject_delay = 1

        #
        #  status_server: Whether or not the server will respond
        #  to Status-Server requests.
        #
        #  Normally this should be set to "no", because they're useless.
        #  See: http://www.freeradius.org/rfc/rfc2865.html#Keep-Alives
        #
        #  However, certain NAS boxes may require them.
        #
        #  When sent a Status-Server message, the server responds with
        #  an Access-Accept packet, containing a Reply-Message attribute,
        #  which is a string describing how long the server has been
        #  running.
        #
        status_server = no
}

# PROXY CONFIGURATION
#
#  proxy_requests: Turns proxying of RADIUS requests on or off.
#
#  The server has proxying turned on by default.  If your system is NOT
#  set up to proxy requests to another server, then you can turn proxying
#  off here.  This will save a small amount of resources on the server.
#
#  If you have proxying turned off, and your configuration files say
#  to proxy a request, then an error message will be logged.
#
#  To disable proxying, change the "yes" to "no", and comment the
#  $INCLUDE line.
#
#  allowed values: {no, yes}
#
proxy_requests  = yes
$INCLUDE  ${confdir}/proxy.conf

# CLIENTS CONFIGURATION
#
#  Client configuration is defined in "clients.conf".
#

#  The 'clients.conf' file contains all of the information from the old
#  'clients' and 'naslist' configuration files.  We recommend that you
#  do NOT use 'client's or 'naslist', although they are still
#  supported.
#
#  Anything listed in 'clients.conf' will take precedence over the
#  information from the old-style configuration files.
#
$INCLUDE  ${confdir}/clients.conf

# SNMP CONFIGURATION
#
#  Snmp configuration is only valid if SNMP support was enabled
#  at compile time.
#
#  To enable SNMP querying of the server, set the value of the
#  'snmp' attribute to 'yes'
#
snmp    = no
$INCLUDE  ${confdir}/snmp.conf

# THREAD POOL CONFIGURATION
#
#  The thread pool is a long-lived group of threads which
#  take turns (round-robin) handling any incoming requests.
#
#  You probably want to have a few spare threads around,
#  so that high-load situations can be handled immediately.  If you
#  don't have any spare threads, then the request handling will
#  be delayed while a new thread is created, and added to the pool.
#
```

```
#  You probably don't want too many spare threads around,
#  otherwise they'll be sitting there taking up resources, and
#  not doing anything productive.
#
#  The numbers given below should be adequate for most situations.
#
thread pool {
        #  Number of servers to start initially --- should be a reasonable
        #  ballpark figure.
        start_servers = 5

        #  Limit on the total number of servers running.
        #
        #  If this limit is ever reached, clients will be LOCKED OUT, so it
        #  should NOT BE SET TOO LOW.  It is intended mainly as a brake to
        #  keep a runaway server from taking the system with it as it spirals
        #  down...
        #
        #  You may find that the server is regularly reaching the
        #  'max_servers' number of threads, and that increasing
        #  'max_servers' doesn't seem to make much difference.
        #
        #  If this is the case, then the problem is MOST LIKELY that
        #  your back-end databases are taking too long to respond, and
        #  are preventing the server from responding in a timely manner.
        #
        #  The solution is NOT do keep increasing the 'max_servers'
        #  value, but instead to fix the underlying cause of the
        #  problem: slow database, or 'hostname_lookups=yes'.
        #
        #  For more information, see 'max_request_time', above.
        #
        max_servers = 32

        #  Server-pool size regulation.  Rather than making you guess
        #  how many servers you need, FreeRADIUS dynamically adapts to
        #  the load it sees, that is, it tries to maintain enough
        #  servers to handle the current load, plus a few spare
        #  servers to handle transient load spikes.
        #
        #  It does this by periodically checking how many servers are
        #  waiting for a request.  If there are fewer than
        #  min_spare_servers, it creates a new spare.  If there are
        #  more than max_spare_servers, some of the spares die off.
        #  The default values are probably OK for most sites.
        #
        min_spare_servers = 3
        max_spare_servers = 10

        #  There may be memory leaks or resource allocation problems with
        #  the server.  If so, set this value to 300 or so, so that the
        #  resources will be cleaned up periodically.
        #
        #  This should only be necessary if there are serious bugs in the
        #  server which have not yet been fixed.
        #
        #  '0' is a special value meaning 'infinity', or 'the servers never
        #  exit'
        max_requests_per_server = 0
}

# MODULE CONFIGURATION
#
#  The names and configuration of each module is located in this section.
#
#  After the modules are defined here, they may be referred to by name,
#  in other sections of this configuration file.
#
modules {
        #
        #  Each module has a configuration as follows:
        #
        #       name [ instance ] {
        #               config_item = value
        #               ...
        #       }
        #
        #  The 'name' is used to load the 'rlm_name' library
        #  which implements the functionality of the module.
        #
        #  The 'instance' is optional.  To have two different instances
        #  of a module, it first must be referred to by 'name'.
        #  The different copies of the module are then created by
        #  inventing two 'instance' names, e.g. 'instance1' and 'instance2'
        #
        #  The instance names can then be used in later configuration
        #  INSTEAD of the original 'name'.  See the 'radutmp' configuration
        #  below for an example.
        #

        # PAP module to authenticate users based on their stored password
        #
        #  Supports multiple encryption schemes
        #  clear: Clear text
        #  crypt: Unix crypt
        #    md5: MD5 ecnryption
        #   sha1: SHA1 encryption.
        #  DEFAULT: crypt
        pap {
                encryption_scheme = crypt
        }

        # CHAP module
        #
        #  To authenticate requests containing a CHAP-Password attribute.
        #
```

```
        chap {
                authtype = CHAP
        }

        # Pluggable Authentication Modules
        #
        #  For Linux, see:
        #       http://www.kernel.org/pub/linux/libs/pam/index.html
        #
        #  WARNING: On many systems, the system PAM libraries have
        #           memory leaks!  We STRONGLY SUGGEST that you do not
        #           use PAM for authentication, due to those memory leaks.
        #
        pam {
                #
                #  The name to use for PAM authentication.
                #  PAM looks in /etc/pam.d/${pam_auth_name}
                #  for it's configuration.  See 'redhat/radiusd-pam'
                #  for a sample PAM configuration file.
                #
                #  Note that any Pam-Auth attribute set in the 'authorize'
                #  section will over-ride this one.
                #
                pam_auth = radiusd
        }

        # Unix /etc/passwd style authentication
        #
        unix {
                #
                #  Cache /etc/passwd, /etc/shadow, and /etc/group
                #
                #  The default is to NOT cache them.
                #
                #  For FreeBSD and NetBSD, you do NOT want to enable
                #  the cache, as it's password lookups are done via a
                #  database, so set this value to 'no'.
                #
                #  Some systems (e.g. RedHat Linux with pam_pwbd) can
                #  take *seconds* to check a password, when th passwd
                #  file containing 1000's of entries.  For those systems,
                #  you should set the cache value to 'yes', and set
                #  the locations of the 'passwd', 'shadow', and 'group'
                #  files, below.
                #
                # allowed values: {no, yes}
                cache = no

                # Reload the cache every 600 seconds (10mins). 0 to disable.
                cache_reload = 600

                #
                #  Define the locations of the normal passwd, shadow, and
                #  group files.
                #
                #  'shadow' is commented out by default, because not all
                #  systems have shadow passwords.
                #
                #  To force the module to use the system password functions,
                #  instead of reading the files, leave the following entries
                #  commented out.
                #
                #  This is required for some systems, like FreeBSD,
                #  and Mac OSX.
                #
                #       passwd = /etc/passwd
                #       shadow = /etc/shadow
                #       group = /etc/group

                #
                #  The location of the "wtmp" file.
                #  This should be moved to it's own module soon.
                #
                #  The only use for 'radlast'.  If you don't use
                #  'radlast', then you can comment out this item.
                #
                radwtmp = ${logdir}/radwtmp
        }

        #  Extensible Authentication Protocol
        #
        #  For all EAP related authentications.
        #  Now in another file, because it is very large.
        #
$INCLUDE ${confdir}/eap.conf

        # Microsoft CHAP authentication
        #
        #  This module supports MS-CHAP and MS-CHAPv2 authentication.
        #  It also enforces the SMB-Account-Ctrl attribute.
        #
        mschap {
                #
                #  As of 0.9, the mschap module does NOT support
                #  reading from /etc/smbpasswd.
                #
                #  If you are using /etc/smbpasswd, see the 'passwd'
                #  module for an example of how to use /etc/smbpasswd

                # authtype value, if present, will be used
                # to overwrite (or add) Auth-Type during
                # authorization. Normally should be MS-CHAP
                authtype = MS-CHAP

                # if use_mppe is not set to no mschap will
```

```
        # add MS-CHAP-MPPE-Keys for MS-CHAPv1 and
        # MS-MPPE-Recv-Key/MS-MPPE-Send-Key for MS-CHAPv2
        #
        #use_mppe = no

        # if mppe is enabled require_encryption makes
        # encryption moderate
        #
        #require_encryption = yes

        # require_strong always requires 128 bit key
        # encryption
        #
        #require_strong = yes

        # Windows sends us a username in the form of
        # DOMAIN\user, but sends the challenge response
        # based on only the user portion.  This hack
        # corrects for that incorrect behavior.
        #
        #with_ntdomain_hack = no

        # The module can perform authentication itself, OR
        # use a Windows Domain Controller.  This configuration
        # directive tells the module to call the ntlm_auth
        # program, which will do the authentication, and return
        # the NT-Key.  Note that you MUST have "winbindd" and
        # "nmbd" running on the local machine for ntlm_auth
        # to work.  See the ntlm_auth program documentation
        # for details.
        #
        # Be VERY careful when editing the following line!
        #
        #ntlm_auth = ...
        #"/path/to/ntlm_auth --request-nt-key ...
        #--username=%{Stripped-User-Name:-%{User-Name:-None}} ...
        #--challenge=%{mschap:Challenge:-00} ...
        #--nt-response=%{mschap:NT-Response:-00}"
}

# Lightweight Directory Access Protocol (LDAP)
#
#  This module definition allows you to use LDAP for
#  authorization and authentication (Auth-Type := LDAP)
#
#  See doc/rlm_ldap for description of configuration options
#  and sample authorize{} and authenticate{} blocks
ldap {
        server = "ldap.your.domain"
        # identity = "cn=admin,o=My Org,c=UA"
        # password = mypass
        basedn = "o=My_Org,c=UA"
        filter = "(uid=%{Stripped-User-Name:-%{User-Name}})"
        # base_filter = "(objectclass=radiusprofile)"

        # set this to 'yes' to use TLS encrypted connections
        # to the LDAP database by using the StartTLS extended
        # operation.
        # The StartTLS operation is supposed to be used with normal
        # ldap connections instead of using ldaps (port 689) connections
        start_tls = no

        # tls_cacertfile         = /path/to/cacert.pem
        # tls_cacertdir          = /path/to/ca/dir/
        # tls_certfile           = /path/to/radius.crt
        # tls_keyfile            = /path/to/radius.key
        # tls_randfile           = /path/to/rnd
        # tls_require_cert       = "demand"

        # default_profile = "cn=radprofile,ou=dialup,o=My Org,c=UA"
        # profile_attribute = "radiusProfileDn"
        access_attr = "dialupAccess"

        # Mapping of RADIUS dictionary attributes to LDAP
        # directory attributes.
        dictionary_mapping = ${raddbdir}/ldap.attrmap

        ldap_connections_number = 5

        #
        # NOTICE: The password_header directive is NOT case insensitive
        #
        # password_header = "{clear}"
        #
        # Set:
        #       password_attribute = nspmPassword
        #
        # to get the user's password from a Novell eDirectory
        # backend. This will work *only if* freeRADIUS is
        # configured to build with --with-edir option.
        #
        #
        #  The server can usually figure this out on its own, and pull
        #  the correct User-Password or NT-Password from the database.
        #
        #  Note that NT-Passwords MUST be stored as a 32-digit hex
        #  string, and MUST start off with "0x", such as:
        #
        #       0x000102030405060708090a0b0c0d0e0f
        #
        #  Without the leading "0x", NT-Passwords will not work.
        #  This goes for NT-Passwords stored in SQL, too.
        #
        # password_attribute = userPassword
        #
```

```
        # Un-comment the following to disable Novell eDirectory account
        # policy check and intruder detection. This will work *only if*
        # FreeRADIUS is configured to build with --with-edir option.
        #
        # edir_account_policy_check=no
        #
        # groupname_attribute = cn
        # groupmembership_filter = "(|(&(objectClass=GroupOfNames) ...
        #(member=%{Ldap-UserDn}))(&(objectClass=GroupOfUniqueNames) ...
        #(uniquemember=%{Ldap-UserDn})))"
        # groupmembership_attribute = radiusGroupName
        timeout = 4
        timelimit = 3
        net_timeout = 1
        # compare_check_items = yes
        # do_xlat = yes
        # access_attr_used_for_allow = yes
}

# passwd module allows to do authorization via any passwd-like
# file and to extract any attributes from these modules
#
# parameters are:
#   filename - path to filename
#   format - format for filename record. This parameters
#            correlates record in the passwd file and RADIUS
#            attributes.
#
#            Field marked as '*' is key field. That is, the parameter
#            with this name from the request is used to search for
#            the record from passwd file
#            Attribute marked as '=' is added to reply_itmes instead
#            of default configure_itmes
#            Attribute marked as '~' is added to request_items
#
#            Field marked as ',' may contain a comma separated list
#            of attributes.
#   authtype - if record found this Auth-Type is used to authenticate
#            user
#   hashsize - hashtable size. If 0 or not specified records are not
#            stored in memory and file is red on every request.
#   allowmultiplekeys - if few records for every key are allowed
#   ignorenislike - ignore NIS-related records
#   delimiter - symbol to use as a field separator in passwd file,
#            for format ':' symbol is always used. '\0', '\n' are
#            not allowed
#

#  An example configuration for using /etc/smbpasswd.
#
#passwd etc_smbpasswd {
#        filename = /etc/smbpasswd
#        format = "*User-Name::LM-Password:NT-Password:SMB-Account-CTRL-TEXT::"
#        authtype = MS-CHAP
#        hashsize = 100
#        ignorenislike = no
#        allowmultiplekeys = no
#}

#  Similar configuration, for the /etc/group file. Adds a Group-Name
#  attribute for every group that the user is member of.
#
#passwd etc_group {
#        filename = /etc/group
#        format = "=Group-Name:::*,User-Name"
#        hashsize = 50
#        ignorenislike = yes
#        allowmultiplekeys = yes
#        delimiter = ":"
#}

# Realm module, for proxying.
#
#  You can have multiple instances of the realm module to
#  support multiple realm syntaxs at the same time.  The
#  search order is defined by the order in the authorize and
#  preacct sections.
#
#  Four config options:
#        format          -  must be 'prefix' or 'suffix'
#        delimiter       -  must be a single character
#        ignore_default  -  set to 'yes' or 'no'
#        ignore_null     -  set to 'yes' or 'no'
#
#  ignore_default and ignore_null can be set to 'yes' to prevent
#  the module from matching against DEFAULT or NULL realms.  This
#  may be useful if you have have multiple instances of the
#  realm module.
#
#  They both default to 'no'.
#

#  'realm/username'
#
#  Using this entry, IPASS users have their realm set to "IPASS".
realm IPASS {
        format = prefix
        delimiter = "/"
        ignore_default = no
        ignore_null = no
}

#  'username@realm'
#
```

```
realm suffix {
        format = suffix
        delimiter = "@"
        ignore_default = no
        ignore_null = no
}

#  'username%realm'
#
realm realmpercent {
        format = suffix
        delimiter = "%"
        ignore_default = no
        ignore_null = no
}

#
#  'domain\user'
#
realm ntdomain {
        format = prefix
        delimiter = "\\"
        ignore_default = no
        ignore_null = no
}

#  A simple value checking module
#
#  It can be used to check if an attribute value in the request
#  matches a (possibly multi valued) attribute in the check
#  items This can be used for example for caller-id
#  authentication.  For the module to run, both the request
#  attribute and the check items attribute must exist
#
#  i.e.
#  A user has an ldap entry with 2 radiusCallingStationId
#  attributes with values "12345678" and "12345679".  If we
#  enable rlm_checkval, then any request which contains a
#  Calling-Station-Id with one of those two values will be
#  accepted.  Requests with other values for
#  Calling-Station-Id will be rejected.
#
#  Regular expressions in the check attribute value are allowed
#  as long as the operator is '=~'
#
checkval {
        # The attribute to look for in the request
        item-name = Calling-Station-Id

        # The attribute to look for in check items. Can be multi valued
        check-name = Calling-Station-Id

        # The data type. Can be
        # string,integer,ipaddr,date,abinary,octets
        data-type = string

        # If set to yes and we dont find the item-name attribute in the
        # request then we send back a reject
        # DEFAULT is no
        #notfound-reject = no
}

#  rewrite arbitrary packets.  Useful in accounting and authorization.
#
#
#  The module can also use the Rewrite-Rule attribute. If it
#  is set and matches the name of the module instance, then
#  that module instance will be the only one which runs.
#
#  Also if new_attribute is set to yes then a new attribute
#  will be created containing the value replacewith and it
#  will be added to searchin (packet, reply, proxy, proxy_reply or config).
# searchfor,ignore_case and max_matches will be ignored in that case.
#
# Backreferences are supported: %{0} will contain the string the whole match
# and %{1} to %{8} will contain the contents of the 1st to the 8th parentheses
#
# If max_matches is greater than one the backreferences will correspond to the
# first match

#
#attr_rewrite sanecallerid {
#        attribute = Called-Station-Id
        # may be "packet", "reply", "proxy", "proxy_reply" or "config"
#        searchin = packet
#        searchfor = "[+ ]"
#        replacewith = ""
#        ignore_case = no
#        new_attribute = no
#        max_matches = 10
### If set to yes then the replace string will be appended to the original string
#        append = no
#}

# Preprocess the incoming RADIUS request, before handing it off
# to other modules.
#
#  This module processes the 'huntgroups' and 'hints' files.
#  In addition, it re-writes some weird attributes created
#  by some NASes, and converts the attributes into a form which
#  is a little more standard.
#
preprocess {
        huntgroups = ${confdir}/huntgroups
        hints = ${confdir}/hints
```

```
        # This hack changes Ascend's wierd port numberings
        # to standard 0-??? port numbers so that the "+" works
        # for IP address assignments.
        with_ascend_hack = no
        ascend_channels_per_line = 23

        # Windows NT machines often authenticate themselves as
        # NT_DOMAIN\username
        #
        # If this is set to 'yes', then the NT_DOMAIN portion
        # of the user-name is silently discarded.
        #
        # This configuration entry SHOULD NOT be used.
        # See the "realms" module for a better way to handle
        # NT domains.
        with_ntdomain_hack = no

        # Specialix Jetstream 8500 24 port access server.
        #
        # If the user name is 10 characters or longer, a "/"
        # and the excess characters after the 10th are
        # appended to the user name.
        #
        # If you're not running that NAS, you don't need
        # this hack.
        with_specialix_jetstream_hack = no

        # Cisco (and Quintum in Cisco mode) sends it's VSA attributes
        # with the attribute name *again* in the string, like:
        #
        #   H323-Attribute = "h323-attribute=value".
        #
        # If this configuration item is set to 'yes', then
        # the redundant data in the the attribute text is stripped
        # out.  The result is:
        #
        #  H323-Attribute = "value"
        #
        # If you're not running a Cisco or Quintum NAS, you don't
        # need this hack.
        with_cisco_vsa_hack = no
}

# Livingston-style 'users' file
#
files {
        usersfile = ${confdir}/users
        acctusersfile = ${confdir}/acct_users

        #  If you want to use the old Cistron 'users' file
        #  with FreeRADIUS, you should change the next line
        #  to 'compat = cistron'.  You can the copy your 'users'
        #  file from Cistron.
        compat = no
}

# Write a detailed log of all accounting records received.
#
detail {
        #  Note that we do NOT use NAS-IP-Address here, as
        #  that attribute MAY BE from the originating NAS, and
        #  NOT from the proxy which actually sent us the
        #  request.  The Client-IP-Address attribute is ALWAYS
        #  the address of the client which sent us the
        #  request.
        #
        #  The following line creates a new detail file for
        #  every radius client (by IP address or hostname).
        #  In addition, a new detail file is created every
        #  day, so that the detail file doesn't have to go
        #  through a 'log rotation'
        #
        #  If your detail files are large, you may also want
        #  to add a ':%H' (see doc/variables.txt) to the end
        #  of it, to create a new detail file every hour, e.g.:
        #
        #    .../detail-%Y%m%d:%H
        #
        #  This will create a new detail file for every hour.
        #
        detailfile = ${radacctdir}/%{Client-IP-Address}/detail-%Y%m%d

        #
        #  The Unix-style permissions on the 'detail' file.
        #
        #  The detail file often contains secret or private
        #  information about users.  So by keeping the file
        #  permissions restrictive, we can prevent unwanted
        #  people from seeing that information.
        detailperm = 0600
}

#
#  Many people want to log authentication requests.
#  Rather than modifying the server core to print out more
#  messages, we can use a different instance of the 'detail'
#  module, to log the authentication requests to a file.
#
#  You will also need to un-comment the 'auth_log' line
#  in the 'authorize' section, below.
#
# detail auth_log {
        # detailfile = ${radacctdir}/%{Client-IP-Address}/auth-detail-%Y%m%d

        #
```

```
        #  This MUST be 0600, otherwise anyone can read
        #  the users passwords!
        # detailperm = 0600
# }

#
#  This module logs authentication reply packets sent
#  to a NAS.  Both Access-Accept and Access-Reject packets
#  are logged.
#
#  You will also need to un-comment the 'reply_log' line
#  in the 'post-auth' section, below.
#
# detail reply_log {
        # detailfile = ${radacctdir}/%{Client-IP-Address}/reply-detail-%Y%m%d

        #
        #  This MUST be 0600, otherwise anyone can read
        #  the users passwords!
        # detailperm = 0600
# }

#
#  This module logs packets proxied to a home server.
#
#  You will also need to un-comment the 'pre_proxy_log' line
#  in the 'pre-proxy' section, below.
#
# detail pre_proxy_log {
        # detailfile = ${radacctdir}/%{Client-IP-Address}/pre-proxy-detail-%Y%m%d

        #
        #  This MUST be 0600, otherwise anyone can read
        #  the users passwords!
        # detailperm = 0600
# }

#
#  This module logs response packets from a home server.
#
#  You will also need to un-comment the 'post_proxy_log' line
#  in the 'post-proxy' section, below.
#
# detail post_proxy_log {
        # detailfile = ${radacctdir}/%{Client-IP-Address}/post-proxy-detail-%Y%m%d

        #
        #  This MUST be 0600, otherwise anyone can read
        #  the users passwords!
        # detailperm = 0600
# }

# Create a unique accounting session Id.  Many NASes re-use or
# repeat values for Acct-Session-Id, causing no end of
# confusion.
#
#  This module will add a (probably) unique session id
#  to an accounting packet based on the attributes listed
#  below found in the packet.  See doc/rlm_acct_unique for
#  more information.
#
acct_unique {
        key = "User-Name, Acct-Session-Id, NAS-IP-Address, Client-IP-Address, NAS-Port"
}

#  Include another file that has the SQL-related configuration.
#  This is another file only because it tends to be big.
#
#  The following configuration file is for use with MySQL.
#
#  For Postgresql, use:          ${confdir}/postgresql.conf
#  For MS-SQL, use:              ${confdir}/mssql.conf
#  For Oracle, use:              ${confdir}/oraclesql.conf
#
$INCLUDE  ${confdir}/sql.conf

#  For Cisco VoIP specific accounting with Postgresql,
#  use:          ${confdir}/pgsql-voip.conf
#
#  You will also need the sql schema from:
#        src/billing/cisco_h323_db_schema-postgres.sql
#  Note: This config can be use AS WELL AS the standard sql
#  config if you need SQL based Auth

#  Write a 'utmp' style file, of which users are currently
#  logged in, and where they've logged in from.
#
#  This file is used mainly for Simultaneous-Use checking,
#  and also 'radwho', to see who's currently logged in.
#
radutmp {
        #  Where the file is stored.  It's not a log file,
        #  so it doesn't need rotating.
        #
        filename = ${logdir}/radutmp

        #  The field in the packet to key on for the
        #  'user' name,  If you have other fields which you want
        #  to use to key on to control Simultaneous-Use,
        #  then you can use them here.
        #
        #  Note, however, that the size of the field in the
        #  'utmp' data structure is small, around 32
        #  characters, so that will limit the possible choices
```

```
        #  of keys.
        #
        #  You may want instead: %{Stripped-User-Name:-%{User-Name}}
        username = %{User-Name}

        #  Whether or not we want to treat "user" the same
        #  as "USER", or "User".  Some systems have problems
        #  with case sensitivity, so this should be set to
        #  'no' to enable the comparisons of the key attribute
        #  to be case insensitive.
        #
        case_sensitive = yes

        #  Accounting information may be lost, so the user MAY
        #  have logged off of the NAS, but we haven't noticed.
        #  If so, we can verify this information with the NAS,
        #
        #  If we want to believe the 'utmp' file, then this
        #  configuration entry can be set to 'no'.
        #
        check_with_nas = yes

        # Set the file permissions, as the contents of this file
        # are usually private.
        perm = 0600

        callerid = "yes"
}

# "Safe" radutmp - does not contain caller ID, so it can be
# world-readable, and radwho can work for normal users, without
# exposing any information that isn't already exposed by who(1).
#
# This is another 'instance' of the radutmp module, but it is given
# then name "sradutmp" to identify it later in the "accounting"
# section.
radutmp sradutmp {
        filename = ${logdir}/sradutmp
        perm = 0644
        callerid = "no"
}

# attr_filter - filters the attributes received in replies from
# proxied servers, to make sure we send back to our RADIUS client
# only allowed attributes.
attr_filter {
        attrsfile = ${confdir}/attrs
}

#  counter module:
#  This module takes an attribute (count-attribute).
#  It also takes a key, and creates a counter for each unique
#  key.  The count is incremented when accounting packets are
#  received by the server.  The value of the increment depends
#  on the attribute type.
#  If the attribute is Acct-Session-Time or of an integer type we add the
#  value of the attribute. If it is anything else we increase the
#  counter by one.
#
#  The 'reset' parameter defines when the counters are all reset to
#  zero.  It can be hourly, daily, weekly, monthly or never.
#
#  hourly: Reset on 00:00 of every hour
#  daily: Reset on 00:00:00 every day
#  weekly: Reset on 00:00:00 on sunday
#  monthly: Reset on 00:00:00 of the first day of each month
#
#  It can also be user defined. It should be of the form:
#  num[hdwm] where:
#  h: hours, d: days, w: weeks, m: months
#  If the letter is ommited days will be assumed. In example:
#  reset = 10h (reset every 10 hours)
#  reset = 12  (reset every 12 days)
#
#
#  The check-name attribute defines an attribute which will be
#  registered by the counter module and can be used to set the
#  maximum allowed value for the counter after which the user
#  is rejected.
#  Something like:
#
#  DEFAULT Max-Daily-Session := 36000
#          Fall-Through = 1
#
#  You should add the counter module in the instantiate
#  section so that it registers check-name before the files
#  module reads the users file.
#
#  If check-name is set and the user is to be rejected then we
#  send back a Reply-Message and we log a Failure-Message in
#  the radius.log
#  If the count attribute is Acct-Session-Time then on each login
#  we send back the remaining online time as a Session-Timeout attribute
#
#  The counter-name can also be used instead of using the check-name
#  like below:
#
#  DEFAULT  Daily-Session-Time > 3600, Auth-Type = Reject
#       Reply-Message = "You've used up more than one hour today"
#
#  The allowed-servicetype attribute can be used to only take
#  into account specific sessions. For example if a user first
#  logs in through a login menu and then selects ppp there will
#  be two sessions. One for Login-User and one for Framed-User
```

```
#  service type. We only need to take into account the second one.
#
#  The module should be added in the instantiate, authorize and
#  accounting sections.  Make sure that in the authorize
#  section it comes after any module which sets the
#  'check-name' attribute.
#
counter daily {
        filename = ${raddbdir}/db.daily
        key = User-Name
        count-attribute = Acct-Session-Time
        reset = daily
        counter-name = Daily-Session-Time
        check-name = Max-Daily-Session
        allowed-servicetype = Framed-User
        cache-size = 5000
}

# The "always" module is here for debugging purposes. Each
# instance simply returns the same result, always, without
# doing anything.
always fail {
        rcode = fail
}
always reject {
        rcode = reject
}
always ok {
        rcode = ok
        simulcount = 0
        mpp = no
}

#
#  The 'expression' module currently has no configuration.
#
#  This module is useful only for 'xlat'.  To use it,
#  put 'exec' into the 'instantiate' section.  You can then
#  do dynamic translation of attributes like:
#
#  Attribute-Name = `%{expr:2 + 3 + %{exec: uid -u}}`
#
#  The value of the attribute will be replaced with the output
#  of the program which is executed.  Due to RADIUS protocol
#  limitations, any output over 253 bytes will be ignored.
expr {
}

#
#  The 'digest' module currently has no configuration.
#
#  "Digest" authentication against a Cisco SIP server.
#  See 'doc/rfc/draft-sterman-aaa-sip-00.txt' for details
#  on performing digest authentication for Cisco SIP servers.
#
digest {
}

#
#  Execute external programs
#
#  This module is useful only for 'xlat'.  To use it,
#  put 'exec' into the 'instantiate' section.  You can then
#  do dynamic translation of attributes like:
#
#  Attribute-Name = `%{exec:/path/to/program args}`
#
#  The value of the attribute will be replaced with the output
#  of the program which is executed.  Due to RADIUS protocol
#  limitations, any output over 253 bytes will be ignored.
#
#  The RADIUS attributes from the user request will be placed
#  into environment variables of the executed program, as
#  described in 'doc/variables.txt'
#
exec {
        wait = yes
        input_pairs = request
}

#
#  This is a more general example of the execute module.
#
#  This one is called "echo".
#
#  Attribute-Name = `%{echo:/path/to/program args}`
#
#  If you wish to execute an external program in more than
#  one section (e.g. 'authorize', 'pre_proxy', etc), then it
#  is probably best to define a different instance of the
#  'exec' module for every section.
#
exec echo {
        #
        #  Wait for the program to finish.
        #
        #  If we do NOT wait, then the program is "fire and
        #  forget", and any output attributes from it are ignored.
        #
        #  If we are looking for the program to output
        #  attributes, and want to add those attributes to the
        #  request, then we MUST wait for the program to
        #  finish, and therefore set 'wait=yes'
        #
```

```
                # allowed values: {no, yes}
                wait = yes

                #
                #  The name of the program to execute, and it's
                #  arguments.  Dynamic translation is done on this
                #  field, so things like the following example will
                #  work.
                #
                program = "/bin/echo %{User-Name}"

                #
                #  The attributes which are placed into the
                #  environment variables for the program.
                #
                #  Allowed values are:
                #
                #       request         attributes from the request
                #       config          attributes from the configuration items list
                #       reply           attributes from the reply
                #       proxy-request   attributes from the proxy request
                #       proxy-reply     attributes from the proxy reply
                #
                #  Note that some attributes may not exist at some
                #  stages.  e.g. There may be no proxy-reply
                #  attributes if this module is used in the
                #  'authorize' section.
                #
                input_pairs = request

                #
                #  Where to place the output attributes (if any) from
                #  the executed program.  The values allowed, and the
                #  restrictions as to availability, are the same as
                #  for the input_pairs.
                #
                output_pairs = reply

                #
                #  When to execute the program.  If the packet
                #  type does NOT match what's listed here, then
                #  the module does NOT execute the program.
                #
                #  For a list of allowed packet types, see
                #  the 'dictionary' file, and look for VALUEs
                #  of the Packet-Type attribute.
                #
                #  By default, the module executes on ANY packet.
                #  Un-comment out the following line to tell the
                #  module to execute only if an Access-Accept is
                #  being sent to the NAS.
                #
                #packet_type = Access-Accept
        }

        #  Do server side ip pool management. Should be added in post-auth and
        #  accounting sections.
        #
        #  The module also requires the existance of the Pool-Name
        #  attribute. That way the administrator can add the Pool-Name
        #  attribute in the user profiles and use different pools
        #  for different users. The Pool-Name attribute is a *check* item not
        #  a reply item.
        #
        # Example:
        # radiusd.conf: ippool students { [...] }
        # users file  : DEFAULT Group == students, Pool-Name := "students"
        #
        # ********* IF YOU CHANGE THE RANGE PARAMETERS YOU MUST *********
        # ********* THEN ERASE THE DB FILES                     *********
        #
        ippool main_pool {

                #  range-start,range-stop: The start and end ip
                #  addresses for the ip pool
                range-start = 192.168.1.1
                range-stop = 192.168.3.254

                #  netmask: The network mask used for the ip's
                netmask = 255.255.255.0

                #  cache-size: The gdbm cache size for the db
                #  files. Should be equal to the number of ip's
                #  available in the ip pool
                cache-size = 800

                # session-db: The main db file used to allocate ip's to clients
                session-db = ${raddbdir}/db.ippool

                # ip-index: Helper db index file used in multilink
                ip-index = ${raddbdir}/db.ipindex

                # override: Will this ippool override a Framed-IP-Address already set
                override = no

                # maximum-timeout: If not zero specifies the maximum time in seconds an
                # entry may be active. Default: 0
                maximum-timeout = 0
        }

        # ANSI X9.9 token support.  Not included by default.
        # $INCLUDE  ${confdir}/x99.conf

}

# Instantiation
#
#  This section orders the loading of the modules.  Modules
```

```
#  listed here will get loaded BEFORE the later sections like
#  authorize, authenticate, etc. get examined.
#
#  This section is not strictly needed.  When a section like
#  authorize refers to a module, it's automatically loaded and
#  initialized.  However, some modules may not be listed in any
#  of the following sections, so they can be listed here.
#
#  Also, listing modules here ensures that you have control over
#  the order in which they are initalized.  If one module needs
#  something defined by another module, you can list them in order
#  here, and ensure that the configuration will be OK.
#
instantiate {
        #
        #  Allows the execution of external scripts.
        #  The entire command line (and output) must fit into 253 bytes.
        #
        #  e.g. Framed-Pool = `%{exec:/bin/echo foo}`
        exec

        #
        #  The expression module doesn't do authorization,
        #  authentication, or accounting.  It only does dynamic
        #  translation, of the form:
        #
        #       Session-Timeout = `%{expr:2 + 3}`
        #
        #  So the module needs to be instantiated, but CANNOT be
        #  listed in any other section.  See 'doc/rlm_expr' for
        #  more information.
        #
        expr

        #
        # We add the counter module here so that it registers
        # the check-name attribute before any module which sets
        # it
#       daily
}

#  Authorization. First preprocess (hints and huntgroups files),
#  then realms, and finally look in the "users" file.
#
#  The order of the realm modules will determine the order that
#  we try to find a matching realm.
#
#  Make *sure* that 'preprocess' comes before any realm if you
#  need to setup hints for the remote radius server
authorize {
        #
        #  The preprocess module takes care of sanitizing some bizarre
        #  attributes in the request, and turning them into attributes
        #  which are more standard.
        #
        #  It takes care of processing the 'raddb/hints' and the
        #  'raddb/huntgroups' files.
        #
        #  It also adds the %{Client-IP-Address} attribute to the request.
        preprocess

        #
        #  If you want to have a log of authentication requests,
        #  un-comment the following line, and the 'detail auth_log'
        #  section, above.
#       auth_log

#       attr_filter

        #
        #  The chap module will set 'Auth-Type := CHAP' if we are
        #  handling a CHAP request and Auth-Type has not already been set
        chap

        #
        #  If the users are logging in with an MS-CHAP-Challenge
        #  attribute for authentication, the mschap module will find
        #  the MS-CHAP-Challenge attribute, and add 'Auth-Type := MS-CHAP'
        #  to the request, which will cause the server to then use
        #  the mschap module for authentication.
        mschap

        #
        #  If you have a Cisco SIP server authenticating against
        #  FreeRADIUS, uncomment the following line, and the 'digest'
        #  line in the 'authenticate' section.
#       digest

        #
        #  Look for IPASS style 'realm/', and if not found, look for
        #  '@realm', and decide whether or not to proxy, based on
        #  that.
#       IPASS

        #
        #  If you are using multiple kinds of realms, you probably
        #  want to set "ignore_null = yes" for all of them.
        #  Otherwise, when the first style of realm doesn't match,
        #  the other styles won't be checked.
        #
        suffix
#       ntdomain

        #
        #  This module takes care of EAP-MD5, EAP-TLS, and EAP-LEAP
        #  authentication.
        #
```

```
        #  It also sets the EAP-Type attribute in the request
        #  attribute list to the EAP type from the packet.
        eap

        #
        #  Read the 'users' file
        files

        #
        #  Look in an SQL database.  The schema of the database
        #  is meant to mirror the "users" file.
        #
        #  See "Authorization Queries" in sql.conf
#       sql

        #
        #  If you are using /etc/smbpasswd, and are also doing
        #  mschap authentication, the un-comment this line, and
        #  configure the 'etc_smbpasswd' module, above.
#       etc_smbpasswd

        #
        #  The ldap module will set Auth-Type to LDAP if it has not
        #  already been set
#       ldap

        #
        #  Enforce daily limits on time spent logged in.
#       daily

        #
        # Use the checkval module
#       checkval
}

#  Authentication.
#
#
#  This section lists which modules are available for authentication.
#  Note that it does NOT mean 'try each module in order'.  It means
#  that a module from the 'authorize' section adds a configuration
#  attribute 'Auth-Type := FOO'.  That authentication type is then
#  used to pick the apropriate module from the list below.
#

#  In general, you SHOULD NOT set the Auth-Type attribute.  The server
#  will figure it out on its own, and will do the right thing.  The
#  most common side effect of erroneously setting the Auth-Type
#  attribute is that one authentication method will work, but the
#  others will not.
#
#  The common reasons to set the Auth-Type attribute by hand
#  is to either forcibly reject the user, or forcibly accept him.
#
authenticate {
        #
        #  PAP authentication, when a back-end database listed
        #  in the 'authorize' section supplies a password.  The
        #  password can be clear-text, or encrypted.
        Auth-Type PAP {
                pap
        }

        #
        #  Most people want CHAP authentication
        #  A back-end database listed in the 'authorize' section
        #  MUST supply a CLEAR TEXT password.  Encrypted passwords
        #  won't work.
        Auth-Type CHAP {
                chap
        }

        #
        #  MSCHAP authentication.
        Auth-Type MS-CHAP {
                mschap
        }

        #
        #  If you have a Cisco SIP server authenticating against
        #  FreeRADIUS, uncomment the following line, and the 'digest'
        #  line in the 'authorize' section.
#       digest

        #
        #  Pluggable Authentication Modules.
#       pam

        #
        #  See 'man getpwent' for information on how the 'unix'
        #  module checks the users password.  Note that packets
        #  containing CHAP-Password attributes CANNOT be authenticated
        #  against /etc/passwd!  See the FAQ for details.
        #
        unix

        # Uncomment it if you want to use ldap for authentication
        #
        # Note that this means "check plain-text password against
        # the ldap database", which means that EAP won't work,
        # as it does not supply a plain-text password.
#       Auth-Type LDAP {
#               ldap
#       }

        #
        #  Allow EAP authentication.
        eap
```

```
}

#
#  Pre-accounting.  Decide which accounting type to use.
#
preacct {
        preprocess

        #
        #  Ensure that we have a semi-unique identifier for every
        #  request, and many NAS boxes are broken.
        acct_unique

        #
        #  Look for IPASS-style 'realm/', and if not found, look for
        #  '@realm', and decide whether or not to proxy, based on
        #  that.
        #
        #  Accounting requests are generally proxied to the same
        #  home server as authentication requests.
#       IPASS
        suffix
#       ntdomain

        #
        #  Read the 'acct_users' file
        files
}

#
#  Accounting.  Log the accounting data.
#
accounting {
        #
        #  Create a 'detail'ed log of the packets.
        #  Note that accounting requests which are proxied
        #  are also logged in the detail file.
        detail
#       daily

        #  Update the wtmp file
        #
        #  If you don't use "radlast", you can delete this line.
        unix

        #
        #  For Simultaneous-Use tracking.
        #
        #  Due to packet losses in the network, the data here
        #  may be incorrect.  There is little we can do about it.
        radutmp
#       sradutmp

        #  Return an address to the IP Pool when we see a stop record.
#       main_pool

        #
        #  Log traffic to an SQL database.
        #
        #  See "Accounting queries" in sql.conf
#       sql

        #  Cisco VoIP specific bulk accounting
#       pgsql-voip

}

#  Session database, used for checking Simultaneous-Use. Either the radutmp
#  or rlm_sql module can handle this.
#  The rlm_sql module is *much* faster
session {
        radutmp

        #
        #  See "Simultaneous Use Checking Querie" in sql.conf
#       sql
}

#  Post-Authentication
#  Once we KNOW that the user has been authenticated, there are
#  additional steps we can take.
post-auth {
        #  Get an address from the IP Pool.
#       main_pool

        #
        #  If you want to have a log of authentication replies,
        #  un-comment the following line, and the 'detail reply_log'
        #  section, above.
#       reply_log

        #
        #  After authenticating the user, do another SQL qeury.
        #
        #  See "Authentication Logging Queries" in sql.conf
#       sql

        #
        #  Un-comment the following if you have set
        #  'edir_account_policy_check = yes' in the ldap module sub-section of
        #  the 'modules' section.
        #
#       ldap
        #
        #  Access-Reject packets are sent through the REJECT sub-section of the
        #  post-auth section.
        #  Uncomment the following and set the module name to the ldap instance
```

```
        #  name if you have set 'edir_account_policy_check = yes' in the ldap
        #  module sub-section of the 'modules' section.
        #
#       Post-Auth-Type REJECT {
#               insert-module-name-here
#       }

}

#
#  When the server decides to proxy a request to a home server,
#  the proxied request is first passed through the pre-proxy
#  stage.  This stage can re-write the request, or decide to
#  cancel the proxy.
#
#  Only a few modules currently have this method.
#
pre-proxy {
#       attr_rewrite

        #  If you want to have a log of packets proxied to a home
        #  server, un-comment the following line, and the
        #  'detail pre_proxy_log' section, above.
#       pre_proxy_log
}

#
#  When the server receives a reply to a request it proxied
#  to a home server, the request may be massaged here, in the
#  post-proxy stage.
#
post-proxy {
        #

        #  If you want to have a log of replies from a home server,
        #  un-comment the following line, and the 'detail post_proxy_log'
        #  section, above.
#       post_proxy_log

#       attr_rewrite

        #  Uncomment the following line if you want to filter replies from
        #  remote proxies based on the rules defined in the 'attrs' file.

#       attr_filter

        #
        #  If you are proxying LEAP, you MUST configure the EAP
        #  module, and you MUST list it here, in the post-proxy
        #  stage.
        #
        #  You MUST also use the 'nostrip' option in the 'realm'
        #  configuration.  Otherwise, the User-Name attribute
        #  in the proxied request will not match the user name
        #  hidden inside of the EAP packet, and the end server will
        #  reject the EAP request.
        #
        eap
}
```

D.2 The eap.conf FreeRADIUS configuration file

Listing D.2: EAP authentication security protocol codes.

```
#
#  Whatever you do, do NOT set 'Auth-Type := EAP'.  The server
#  is smart enough to figure this out on its own.  The most
#  common side effect of setting 'Auth-Type := EAP' is that the
#  users then cannot use ANY other authentication method.
#
#       $Id: eap.conf,v 1.4 2004/04/15 18:34:41 aland Exp $
#
        eap {
                #  Invoke the default supported EAP type when
                #  EAP-Identity response is received.
                #
                #  The incoming EAP messages DO NOT specify which EAP
                #  type they will be using, so it MUST be set here.
                #
                #  For now, only one default EAP type may be used at a time.
                #
                #  If the EAP-Type attribute is set by another module,
                #  then that EAP type takes precedence over the
                #  default type configured here.
                #
                default_eap_type = tls

                #  A list is maintained to correlate EAP-Response
                #  packets with EAP-Request packets.  After a
                #  configurable length of time, entries in the list
                #  expire, and are deleted.
                #
                timer_expire     = 60

                #  There are many EAP types, but the server has support
                #  for only a limited subset.  If the server receives
                #  a request for an EAP type it does not support, then
                #  it normally rejects the request.  By setting this
                #  configuration to "yes", you can tell the server to
                #  instead keep processing the request.  Another module
                #  MUST then be configured to proxy the request to
                #  another RADIUS server which supports that EAP type.
```

```
#
#  If another module is NOT configured to handle the
#  request, then the request will still end up being
#  rejected.
ignore_unknown_eap_types = no

# Cisco AP1230B firmware 12.2(13)JA1 has a bug.  When given
# a User-Name attribute in an Access-Accept, it copies one
# more byte than it should.
#
# We can work around it by configurably adding an extra
# zero byte.
cisco_accounting_username_bug = no

# Supported EAP-types

#
#  We do NOT recommend using EAP-MD5 authentication
#  for wireless connections.  It is insecure, and does
#  not provide for dynamic WEP keys.
#
md5 {
}

# Cisco LEAP
#
#  We do not recommend using LEAP in new deployments.  See:
#  http://www.securiteam.com/tools/5TP012ACKE.html
#
#  Cisco LEAP uses the MS-CHAP algorithm (but not
#  the MS-CHAP attributes) to perform it's authentication.
#
#  As a result, LEAP *requires* access to the plain-text
#  User-Password, or the NT-Password attributes.
#  'System' authentication is impossible with LEAP.
#
leap {
}

#  Generic Token Card.
#
#  Currently, this is only permitted inside of EAP-TTLS,
#  or EAP-PEAP.  The module "challenges" the user with
#  text, and the response from the user is taken to be
#  the User-Password.
#
#  Proxying the tunneled EAP-GTC session is a bad idea,
#  the users password will go over the wire in plain-text,
#  for anyone to see.
#
gtc {
        #  The default challenge, which many clients
        #  ignore..
        #challenge = "Password: "

        #  The plain-text response which comes back
        #  is put into a User-Password attribute,
        #  and passed to another module for
        #  authentication.  This allows the EAP-GTC
        #  response to be checked against plain-text,
        #  or crypt'd passwords.
        #
        #  If you say "Local" instead of "PAP", then
        #  the module will look for a User-Password
        #  configured for the request, and do the
        #  authentication itself.
        #
        auth_type = PAP
}

## EAP-TLS
#
#  To generate ctest certificates, run the script
#
#       ../scripts/certs.sh
#
#  The documents on http://www.freeradius.org/doc
#  are old, but may be helpful.
#
#  See also:
#
#  http://www.dslreports.com/forum/remark,9286052~mode=flat
#
tls {
        private_key_password = demo
        private_key_file = ${certsdir}/FreeRADIUS.net-Server.pem

        #  If Private key & Certificate are located in
        #  the same file, then private_key_file &
        #  certificate_file must contain the same file
        #  name.
        certificate_file = ${certsdir}/FreeRADIUS.net-Server.crt

        #  Trusted Root CA list
        CA_file = ${certsdir}/FreeRADIUS.net-CA.crt

        dh_file = ${certsdir}/dh
        random_file = ${certsdir}/random

        #
        #  This can never exceed the size of a RADIUS
        #  packet (4096 bytes), and is preferably half
        #  that, to accomodate other attributes in
        #  RADIUS packet.  On most APs the MAX packet
        #  length is configured between 1500 - 1600
```

```
                #  In these cases, fragment size should be
                #  1024 or less.
                #
#               fragment_size = 1024

                #  include_length is a flag which is
                #  by default set to yes If set to
                #  yes, Total Length of the message is
                #  included in EVERY packet we send.
                #  If set to no, Total Length of the
                #  message is included ONLY in the
                #  First packet of a fragment series.
                #
#               include_length = yes

                #  Check the Certificate Revocation List
                #
                #  1) Copy CA certificates and CRLs to same directory.
                #  2) Execute 'c_rehash <CA certs&CRLs Directory>'.
                #     'c_rehash' is OpenSSL's command.
                #  3) Add 'CA_path=<CA certs&CRLs directory>'
                #       to radiusd.conf's tls section.
                #  4) uncomment the line below.
                #  5) Restart radiusd
#               check_crl = yes

                #
                #  If check_cert_cn is set, the value will
                #  be xlat'ed and checked against the CN
                #  in the client certificate.  If the values
                #  do not match, the certificate verification
                #  will fail rejecting the user.
                #
                check_cert_cn = %{User-Name}
        }

        #  The TTLS module implements the EAP-TTLS protocol,
        #  which can be described as EAP inside of Diameter,
        #  inside of TLS, inside of EAP, inside of RADIUS...
        #
        #  Surprisingly, it works quite well.
        #
        #  The TTLS module needs the TLS module to be installed
        #  and configured, in order to use the TLS tunnel
        #  inside of the EAP packet.  You will still need to
        #  configure the TLS module, even if you do not want
        #  to deploy EAP-TLS in your network.  Users will not
        #  be able to request EAP-TLS, as it requires them to
        #  have a client certificate.  EAP-TTLS does not
        #  require a client certificate.
        #
        ttls {
                #  The tunneled EAP session needs a default
                #  EAP type which is separate from the one for
                #  the non-tunneled EAP module.  Inside of the
                #  TTLS tunnel, we recommend using EAP-MD5.
                #  If the request does not contain an EAP
                #  conversation, then this configuration entry
                #  is ignored.
                default_eap_type = md5

                #  The tunneled authentication request does
                #  not usually contain useful attributes
                #  like 'Calling-Station-Id', etc.  These
                #  attributes are outside of the tunnel,
                #  and normally unavailable to the tunneled
                #  authentication request.
                #
                #  By setting this configuration entry to
                #  'yes', any attribute which NOT in the
                #  tunneled authentication request, but
                #  which IS available outside of the tunnel,
                #  is copied to the tunneled request.
                #
                # allowed values: {no, yes}
                copy_request_to_tunnel = no

                #  The reply attributes sent to the NAS are
                #  usually based on the name of the user
                #  'outside' of the tunnel (usually
                #  'anonymous').  If you want to send the
                #  reply attributes based on the user name
                #  inside of the tunnel, then set this
                #  configuration entry to 'yes', and the reply
                #  to the NAS will be taken from the reply to
                #  the tunneled request.
                #
                # allowed values: {no, yes}
                use_tunneled_reply = yes

        }

        #
        #  The tunneled EAP session needs a default EAP type
        #  which is separate from the one for the non-tunneled
        #  EAP module.  Inside of the TLS/PEAP tunnel, we
        #  recommend using EAP-MS-CHAPv2.
        #
        #  The PEAP module needs the TLS module to be installed
        #  and configured, in order to use the TLS tunnel
        #  inside of the EAP packet.  You will still need to
        #  configure the TLS module, even if you do not want
        #  to deploy EAP-TLS in your network.  Users will not
        #  be able to request EAP-TLS, as it requires them to
        #  have a client certificate.  EAP-PEAP does not
```

```
                #  require a client certificate.
                #
                 peap {
                        #  The tunneled EAP session needs a default
                        #  EAP type which is separate from the one for
                        #  the non-tunneled EAP module.  Inside of the
                        #  PEAP tunnel, we recommend using MS-CHAPv2,
                        #  as that is the default type supported by
                        #  Windows clients.
                        default_eap_type = mschapv2
                }

                #
                #  This takes no configuration.
                #
                #  Note that it is the EAP MS-CHAPv2 sub-module, not
                #  the main 'mschap' module.
                #
                #  Note also that in order for this sub-module to work,
                #  the main 'mschap' module MUST ALSO be configured.
                #
                #  This module is the *Microsoft* implementation of MS-CHAPv2
                #  in EAP.  There is another (incompatible) implementation
                #  of MS-CHAPv2 in EAP by Cisco, which FreeRADIUS does not
                #  currently support.
                #
                mschapv2 {
                }
        }
```

D.3 The clients.conf FreeRADIUS configuration file

Listing D.3: Defines RADIUS clients .

```
#
# clients.conf - client configuration directives
#
#########################################################################

#########################################################################
#
#  Definition of a RADIUS client (usually a NAS).
#
#  The information given here over rides anything given in the
#  'clients' file, or in the 'naslist' file.  The configuration here
#  contains all of the information from those two files, and allows
#  for more configuration items.
#
#  The "shortname" is be used for logging.  The "nastype", "login" and
#  "password" fields are mainly used for checkrad and are optional.
#

#
#  Defines a RADIUS client.  The format is 'client [hostname|ip-address]'
#
#  '127.0.0.1' is another name for 'localhost'.  It is enabled by default,
#  to allow testing of the server after an initial installation.  If you
#  are not going to be permitting RADIUS queries from localhost, we suggest
#  that you delete, or comment out, this entry.
#
client 127.0.0.1 {
        #
        #  The shared secret use to "encrypt" and "sign" packets between
        #  the NAS and FreeRADIUS.  You MUST change this secret from the
        #  default, otherwise it's not a secret any more!
        #
        #  The secret can be any string, up to 32 characters in length.
        #
        secret          = testing123

        #
        #  The short name is used as an alias for the fully qualified
        #  domain name, or the IP address.
        #
        shortname       = localhost

        #
        # the following three fields are optional, but may be used by
        # checkrad.pl for simultaneous use checks
        #

        #
        # The nastype tells 'checkrad.pl' which NAS-specific method to
        #  use to query the NAS for simultaneous use.
        #
        #  Permitted NAS types are:
        #
        #       cisco
        #       computone
        #       livingston
        #       max40xx
        #       multitech
        #       netserver
        #       pathras
        #       patton
        #       portslave
        #       tc
        #       usrhiper
        #       other           # for all other types
```

```
        #
        nastype     = other     # localhost isn't usually a NAS...

        #
        #  The following two configurations are for future use.
        #  The 'naspasswd' file is currently used to store the NAS
        #  login name and password, which is used by checkrad.pl
        #  when querying the NAS for simultaneous use.
        #
#       login       = !root
#       password    = someadminpas
}

#client some.host.org {
#       secret          = testing123
#       shortname       = localhost
#}

#
#  You can now specify one secret for a network of clients.
#  When a client request comes in, the BEST match is chosen.
#  i.e. The entry from the smallest possible network.
#
#client 192.168.0.0/24 {
#       secret          = testing123-1
#       shortname       = private-network-1
#}
#
client 192.168.0.0/16 {
        secret          = research
        shortname       = network
#       secret          = testing123
#       shortname       = private-network-2
}

#client 10.10.10.10 {
#       # secret and password are mapped through the "secrets" file.
#       secret      = testing123
#       shortname   = liv1
#       # the following three fields are optional, but may be used by
#       # checkrad.pl for simultaneous usage checks
#       nastype     = livingston
#       login       = !root
#       password    = someadminpas
#}
```

D.4 The users FreeRADIUS configuration file

A simple text file containing authentication security and configuration information for each user on the network. It is also associated with an accounting file `acct_users` which is not included in this Appendix since it is not processed.

Listing D.4: Contain information of all authenticated users.

```
#
#       Please read the documentation file ../doc/processing_users_file,
#       or 'man 5 users' (after installing the server) for more information.
#
#       This file contains authentication security and configuration
#       information for each user.  Accounting requests are NOT processed
#       through this file.  Instead, see 'acct_users', in this directory.
#
#       The first field is the user's name and can be up to
#       253 characters in length.  This is followed (on the same line) with
#       the list of authentication requirements for that user.  This can
#       include password, comm server name, comm server port number, protocol
#       type (perhaps set by the "hints" file), and huntgroup name (set by
#       the "huntgroups" file).
#
#       If you are not sure why a particular reply is being sent by the
#       server, then run the server in debugging mode (radiusd -X), and
#       you will see which entries in this file are matched.
#
#       When an authentication request is received from the comm server,
#       these values are tested. Only the first match is used unless the
#       "Fall-Through" variable is set to "Yes".
#
#       A special user named "DEFAULT" matches on all usernames.
#       You can have several DEFAULT entries. All entries are processed
#       in the order they appear in this file. The first entry that
#       matches the login-request will stop processing unless you use
#       the Fall-Through variable.
#
#       If you use the database support to turn this file into a .db or .dbm
#       file, the DEFAULT entries _have_ to be at the end of this file and
#       you can't have multiple entries for one username.
#
#       You don't need to specify a password if you set Auth-Type += System
#       on the list of authentication requirements. The RADIUS server
#       will then check the system password file.
#
```

```
#       Indented (with the tab character) lines following the first
#       line indicate the configuration values to be passed back to
#       the comm server to allow the initiation of a user session.
#       This can include things like the PPP configuration values
#       or the host to log the user onto.
#
#       You can include another `users' file with `$INCLUDE users.other'
#

#
#       For a list of RADIUS attributes, and links to their definitions,
#       see:
#
#       http://www.freeradius.org/rfc/attributes.html
#

#
# Deny access for a specific user.  Note that this entry MUST
# be before any other 'Auth-Type' attribute which results in the user
# being authenticated.
#
# Note that there is NO 'Fall-Through' attribute, so the user will not
# be given any additional resources.
#
#lameuser         Auth-Type := Reject
#                 Reply-Message = "Your account has been disabled."

#
# Deny access for a group of users.
#
# Note that there is NO 'Fall-Through' attribute, so the user will not
# be given any additional resources.
#
#DEFAULT          Group == "disabled", Auth-Type := Reject
#                 Reply-Message = "Your account has been disabled."
#

############### RFC3580 #################
## Also the "eap.conf" MUST be modified to include the follow line:
## "use_tunneled_reply = yes"
## the default is "use_tunneled_reply = no"
## this allow the "Tunnel*" AV's to be passed outside the eap tunnel
## otherwise the switch will NOT see the VLAN to place the port into
#### Comments added by Jeff Reilly ####
testuser         User-Password == "testpw"

desktop  User-Password == "stationary"

laptop   User-Password == "mobile"

pda      User-Password == "small"

FreeRADIUS.net-Client    User-Password == "demo"

rfc3580  User-Password == "demo"
                 Tunnel-Type = "VLAN",
                 Tunnel-Medium-Type = "IEEE-802",
                 Tunnel-Private-Group-Id = "1",
                 Reply-Message = "Hello, %u"

#
# This is a complete entry for "steve". Note that there is no Fall-Through
# entry so that no DEFAULT entry will be used, and the user will NOT
# get any attributes in addition to the ones listed here.
#
#steve   Auth-Type := Local, User-Password == "testing"
#        Service-Type = Framed-User,
#        Framed-Protocol = PPP,
#        Framed-IP-Address = 172.16.3.33,
#        Framed-IP-Netmask = 255.255.255.0,
#        Framed-Routing = Broadcast-Listen,
#        Framed-Filter-Id = "std.ppp",
#        Framed-MTU = 1500,
#        Framed-Compression = Van-Jacobsen-TCP-IP

#
# This is an entry for a user with a space in their name.
# Note the double quotes surrounding the name.
#
#"John Doe"       Auth-Type := Local, User-Password == "hello"
#                 Reply-Message = "Hello, %u"

#
# Dial user back and telnet to the default host for that port
#
#Deg      Auth-Type := Local, User-Password == "ge55ged"
#        Service-Type = Callback-Login-User,
#        Login-IP-Host = 0.0.0.0,
#        Callback-Number = "9,5551212",
#        Login-Service = Telnet,
#        Login-TCP-Port = Telnet

#
# Another complete entry. After the user "dialbk" has logged in, the
# connection will be broken and the user will be dialed back after which
# he will get a connection to the host "timeshare1".
#
#dialbk  Auth-Type := Local, User-Password == "callme"
#        Service-Type = Callback-Login-User,
#        Login-IP-Host = timeshare1,
#        Login-Service = PortMaster,
#        Callback-Number = "9,1-800-555-1212"

#
# user "swilson" will only get a static IP number if he logs in with
# a framed protocol on a terminal server in Alphen (see the huntgroups file).
#
```

```
# Note that by setting "Fall-Through", other attributes will be added from
# the following DEFAULT entries
#
#swilson         Service-Type == Framed-User, Huntgroup-Name == "alphen"
#                Framed-IP-Address = 192.168.1.65,
#                Fall-Through = Yes

#
# If the user logs in as 'username.shell', then authenticate them
# against the system database, give them shell access, and stop processing
# the rest of the file.
#
#DEFAULT         Suffix == ".shell", Auth-Type := System
#                Service-Type = Login-User,
#                Login-Service = Telnet,
#                Login-IP-Host = your.shell.machine


#
# The rest of this file contains the several DEFAULT entries.
# DEFAULT entries match with all login names.
# Note that DEFAULT entries can also Fall-Through (see first entry).
# A name-value pair from a DEFAULT entry will _NEVER_ override
# an already existing name-value pair.
#

#
# First setup all accounts to be checked against the UNIX /etc/passwd.
# (Unless a password was already given earlier in this file).
#
DEFAULT Auth-Type = System
        Fall-Through = 1

#
# Set up different IP address pools for the terminal servers.
# Note that the "+" behind the IP address means that this is the "base"
# IP address. The Port-Id (S0, S1 etc) will be added to it.
#
#DEFAULT         Service-Type == Framed-User, Huntgroup-Name == "alphen"
#                Framed-IP-Address = 192.168.1.32+,
#                Fall-Through = Yes

#DEFAULT         Service-Type == Framed-User, Huntgroup-Name == "delft"
#                Framed-IP-Address = 192.168.2.32+,
#                Fall-Through = Yes

#
# Defaults for all framed connections.
#
DEFAULT Service-Type == Framed-User
        Framed-IP-Address = 255.255.255.254,
        Framed-MTU = 576,
        Service-Type = Framed-User,
        Fall-Through = Yes

#
# Default for PPP: dynamic IP address, PPP mode, VJ-compression.
# NOTE: we do not use Hint = "PPP", since PPP might also be auto-detected
#       by the terminal server in which case there may not be a "P" suffix.
#       The terminal server sends "Framed-Protocol = PPP" for auto PPP.
#
DEFAULT Framed-Protocol == PPP
        Framed-Protocol = PPP,
        Framed-Compression = Van-Jacobson-TCP-IP

#
# Default for CSLIP: dynamic IP address, SLIP mode, VJ-compression.
#
DEFAULT Hint == "CSLIP"
        Framed-Protocol = SLIP,
        Framed-Compression = Van-Jacobson-TCP-IP

#
# Default for SLIP: dynamic IP address, SLIP mode.
#
DEFAULT Hint == "SLIP"
        Framed-Protocol = SLIP

#
# Last default: rlogin to our main server.
#
#DEFAULT
#       Service-Type = Login-User,
#       Login-Service = Rlogin,
#       Login-IP-Host = shellbox.ispdomain.com

# #
# # Last default: shell on the local terminal server.
# #
# DEFAULT
#       Service-Type = Shell-User

# On no match, the user is denied access.
```