



# **Experimental Study of a Network Access Server for a public WLAN access network**

M.Sc. Thesis by  
Juan Caballero and Daniel Malmkvist

Department of Microelectronics and Information Technology  
Royal Institute of Technology, KTH  
Stockholm, January 2002

## Abstract

Wireless access networks have gained popularity due to the flexibility they allow to the users, who is able to move away from his or her desk while still being able to access information. Among the different Wireless LAN standards, the most widespread, by far, is IEEE 802.11.

Public WLAN access networks are being set up in hotspots, i.e. areas expected to have high demand for bandwidth. Access to the Internet and to corporate networks is provided at these hotspots with limited coverage but high available bandwidth. Airports and hotels have often been the first targeted locations for these hotspots, but conference centres, cafes, and train stations will then follow. In the near future, any person who owns at least one access point and has a connection to the Internet can become a small operator and offer access to the Internet using these resources.

Existing solutions for such WLAN access networks lack support for security, flexible accounting, mobility, multiaccess, roaming and user-friendly login. The aim of this Master's Thesis was to study if it was possible to solve these problems and how to integrate all this new functionality into existing public WLAN access networks by building and evaluating a prototype of a public WLAN access network.

System requirements were defined, currently available solutions analysed and a prototype was built. Most of the functionality of the prototype is placed in the Network Access Server, which is the main element providing security, mobility, and accounting.

Flexible accounting, improvements in security, an easy and fast way to login the user, a feedback module to provide information about the current session and integration of the RADIUS architecture with the Mobile IP distribution have all been implemented.

## Acknowledgements

We would like to thank our examiner, Professor Gerald Maguire Jr. for his invaluable answers to our questions and his assistance in the preparation of this manuscript. Special thanks, to our advisor George Liu for his advice and guidance during all the time working in the thesis.

We would like to express also our most sincere thanks to Enrico Pelleta, for providing us with his IP-Login code, which saved us a lot of time. We also thank all the GNU community for creating great open software, especially the Dynamics, Netfilter, FreeSwan, Freeradius and MySQL teams; without their software this thesis would not have been possible.

Thanks to R2M for providing us the opportunity to perform this thesis and to all the Elektrosmoggers for helpful input and interesting presentations.

Finally and most specially thanks to our families and friends for supporting us all this time.

## Contact Information

### Examiner

Professor Gerald Maguire Jr.  
IMIT, KTH  
[maguire@it.kth.se](mailto:maguire@it.kth.se)

### Advisor

George Liu, PhD  
IMIT, KTH  
[yuliu@it.kth.se](mailto:yuliu@it.kth.se)

### Authors

Juan Caballero Bayerri  
[x00\\_jcb@e.kth.se](mailto:x00_jcb@e.kth.se)

Daniel Malmkvist  
[e97\\_dma@e.kth.se](mailto:e97_dma@e.kth.se)

This thesis was performed as part of the requirements for the degree of Master of Science in Electrical Engineering at:

Royal Institute of Technology, KTH  
Department of Microelectronics and Information Technology, IMIT  
KTH Electrum 229  
164 40 Kista, Sweden

This Master of Science thesis was performed at:

R2M Meton AB  
Norgegatan 2  
164 32 Kista, Sweden

The thesis presentation was performed on the 22<sup>nd</sup> of January 2002.

## Table of Contents

<b>1 Introduction .....</b>	<b>1</b>
1.1 Background .....	1
1.2 Problem Statement .....	2
1.3 Target Market.....	2
1.4 Outline of the thesis .....	3
<b>2 Background.....</b>	<b>4</b>
2.1 Protocols.....	4
2.1.1 IEEE 802.11 .....	4
2.1.1.1 Architecture Components.....	5
2.1.1.2 Physical layers.....	6
2.1.1.3 Data Rates.....	7
2.1.1.4 MAC layer.....	8
2.1.1.5 Communication exchange .....	8
2.1.2 AAA.....	9
2.1.2.1 RADIUS .....	10
2.1.2.2 DIAMETER .....	12
2.1.3 IP Security .....	13
2.1.3.1 Security protocols.....	13
2.1.3.2 Modes of operation.....	15
2.1.3.3 Key management in the IPSec framework.....	15
2.1.3.4 Internet Key Exchange .....	15
2.1.4 Private addresses and Network Address Translation (NAT).....	15
2.2 Related work .....	16
2.2.1 Research work .....	16
2.2.2 Commercial work .....	17
<b>3 Method.....</b>	<b>19</b>
3.1 Goal.....	19
3.2 Scope.....	19
3.3 Limitations .....	20
3.4 System Requirements.....	20
3.4.1 General.....	20
3.4.2 Login.....	21
3.4.3 Security.....	21
3.4.4 Accounting.....	22
<b>4 System Architecture .....</b>	<b>24</b>
4.1 System Overview .....	24
4.2 User .....	25
4.3 Client.....	25
4.4 NAS.....	25
4.5 Authentication Server.....	26
4.6 Accounting Database .....	26
<b>5 Analysis and Implementation .....</b>	<b>27</b>
5.1 Login.....	27
5.1.1 Wireless access overview .....	28
5.1.1.1 Basic user .....	28
5.1.1.2 Advanced user using Mobile IP with FA support .....	28
5.2 IP-login.....	29
5.2.1 Shared memory .....	30
5.2.2 The authentication daemon (authd) .....	30
5.2.3 The control daemon (ctrlld) .....	31
5.2.4 The login-cgi script.....	32
5.2.5 The logout-cgi script.....	33

5.2.6 The update-cgi script .....	33
5.2.7 The reauthentication-cgi script .....	33
5.2.8 Web Server .....	34
5.2.9 Iptables.....	34
5.2.10 Keep-Alive.....	34
5.3 RADIUS authentication & Accounting.....	36
5.3.1 RADIUS authentication.....	36
5.3.1.1 Duplicate logins.....	37
5.4 Accounting.....	38
5.4.1 RADIUS Accounting.....	39
5.4.2 SQL database.....	41
5.5 Security .....	42
5.5.1 Assumptions about the system.....	43
5.5.2 Risk Analysis.....	43
5.5.3 Security services .....	44
5.5.4 WLAN security.....	45
5.5.4.1 WEP Security .....	46
5.5.4.2 Agere Systems Extra Security provisions .....	47
5.5.4.3 Physical security of the base station .....	48
5.5.5 Network Access Authentication .....	48
5.5.6 Credentials .....	49
5.5.6.1 NAI.....	50
5.5.6.2 Password.....	50
5.5.6.3 MAC address.....	51
5.5.7 Access Control.....	51
5.5.7.1 Access Point versus Gateway Access Control .....	51
5.5.7.2 IEEE 802.1x .....	53
5.5.7.3 802.1x with 802.11 .....	55
5.5.7.4 Firewall.....	56
5.5.7.5 DHCP Security .....	56
5.5.8 Some security threats to a WLAN .....	58
5.5.8.1 Spoofing .....	58
5.5.8.2 Rogue Access Point.....	59
5.5.8.3 Fake NAS Servers .....	59
5.5.8.4 Denial of Service .....	59
5.5.8.5 Monitoring of traffic.....	59
5.5.8.6 Abuse of resources .....	59
5.5.8.7 AP Configuration.....	60
5.5.9 Intrusion Tolerance.....	60
5.5.10 IP Security (IPSec).....	60
5.5.10.1 Key distribution.....	62
5.5.10.2 Ericsson's WLAN security solution.....	63
5.5.11 VPN .....	63
5.5.12 Cookie security .....	63
5.5.12.1 Solution .....	64
5.5.13 Keep-Alive time window security problem.....	64
5.5.14 Mobile IP security risks .....	66
5.5.15 Applet security .....	67
5.6 Roaming.....	67
5.6.1 Roaming with RADIUS.....	69
5.7 Mobility.....	69
5.7.1 Mobile IP integration with AAA infrastructure.....	69
5.7.1.1 Mobile IP-AAA architecture .....	70
5.7.1.2 Challenge-Response .....	71
5.7.1.3 AAA keys.....	71

5.7.1.4 Security Associations .....	72
5.7.1.5 Implementation.....	72
5.7.1.6 Further issues.....	77
5.7.2 Using cookies for mobility .....	77
5.8 User Feedback.....	78
5.8.1 Requirements .....	78
5.8.2 Implementation alternatives.....	78
5.8.3 Applet .....	80
5.8.4 Password change.....	81
5.8.5 Manual .....	82
<b>6 Analysis.....</b>	<b>83</b>
6.1 Analysis of the solution.....	83
6.1.1 Security .....	83
6.1.2 AAA.....	83
6.1.3 Database.....	84
6.1.4 NAS .....	84
6.1.5 User experience .....	85
6.1.6 Network Management.....	85
6.2 Comparison with other solutions .....	85
6.2.1 Homerun .....	85
6.2.2 IT-University 2001-2002 .....	86
6.2.3 LAN Roamer [15].....	86
6.2.4 NOCAT [16].....	86
6.2.5 IP Unplugged [145] .....	87
6.3 Evaluation of the capacity of our system .....	87
6.3.1 Client.....	88
6.3.2 NAS .....	88
6.3.3 RADIUS server.....	89
6.3.4 SQL server.....	90
<b>7 Conclusions .....</b>	<b>91</b>
7.1 Meeting our goals.....	91
7.1.1 Fulfilling the system requirements.....	91
7.1.1.1 General .....	92
7.1.1.2 Login .....	92
7.1.1.3 Security.....	93
7.1.1.4 Accounting .....	93
7.2 Suggestions & Lessons Learned .....	94
<b>8 Future work .....</b>	<b>95</b>
8.1 System upgrades.....	95
8.2 Other areas of investigation .....	96
8.3 Related research groups .....	96
<b>9 References .....</b>	<b>97</b>
<b>Appendix A: Glossary .....</b>	<b>105</b>
<b>Appendix B: NAS Architecture.....</b>	<b>107</b>
Hardware specifications .....	107
Software specifications .....	107
NAS main modules.....	107
Other modules.....	108
<b>Appendix C: 802.1x support.....</b>	<b>109</b>
802.1X OS support.....	109
Microsoft.....	109
Cisco .....	109
RADIUS servers supporting 802.1x .....	109
Access Points vendors supporting 802.1x.....	109
Agere Systems .....	109

Cisco .....	109
3Com.....	109
Enterasys.....	109
Compaq.....	109
HP .....	110
Symbol.....	110
Intel.....	110
Dell .....	110
<b>Appendix D: RADIUS Server modifications.....</b>	<b>111</b>
<b>Appendix E: SQL queries .....</b>	<b>112</b>
<b>Appendix F: Testbed setup.....</b>	<b>114</b>
General testbed.....	114
RADIUS roaming testbed .....	114
Mobile IP testbed .....	115

## **Table of Figures**

Figure 2.1 IEEE 802.11 Protocol Stack.....	4
Figure 2.2 Ad-Hoc Network.....	5
Figure 2.3 Extended Service Set (ESS).....	6
Figure 2.4 Spread Spectrum techniques .....	7
Figure 2.5 Performance of 802.11 versus distance.....	7
Figure 2.6 Authentication & Association in 802.11 .....	9
Figure 2.7 RADIUS communication.....	10
Figure 2.8 RADIUS packet .....	11
Figure 2.9 Attribute Type-Length-Value format.....	12
Figure 2.10 Authentication Header .....	14
Figure 2.11 Encapsulation Security Payload.....	14
Figure 2.12 IPSec tunnel and transport modes .....	15
Figure 2.13 Basic NAT and NAPT description.....	16
Figure 4.1 System overview .....	24
Figure 5.1 Login Page .....	27
Figure 5.2 Login Page for credit card users.....	29
Figure 5.3 System process communication .....	30
Figure 5.4 authd functional diagram.....	31
Figure 5.5 Protocol Stack .....	42
Figure 5.6 Security Services Importance.....	44
Figure 5.7 WEP Mechanism.....	46
Figure 5.8 802.1x Topology .....	53
Figure 5.9 802.1x Authentication Exchange .....	54
Figure 5.10 802.1x Authentication .....	55
Figure 5.11 Iptables Firewall.....	56
Figure 5.12 IPSec usage in the System.....	60
Figure 5.13 Time window for keep-alive attack.....	65
Figure 5.14 Roaming example .....	68
Figure 5.15 Example of a Normal and a Broker model.....	69
Figure 5.16 MIP-AAA Security Associations.....	70
Figure 5.17 RADIUS-Mobile IP integration alternatives.....	74
Figure 5.18 Mobile IP communication.....	75
Figure 5.19 Hierarchical Mobile IP.....	76
Figure 5.20 Capture of the feedback Applet.....	80
Figure 5.21 Password change webpage.....	81

# 1 Introduction

## 1.1 Background

In recent years a multitude of different access networks have come into existence. Among them, the wireless access networks have gained popularity due to the flexibility they allow the user, who is able to move away from his or her desk while still being able to access information.

Wireless access networks can be split in two groups: those providing limited geographical coverage and high bandwidth, like Wireless LANs or Bluetooth, and those providing wide geographical coverage but limited bandwidth, like GSM [140], GPRS [138], UMTS [139], or CDMA2000 [142].

Among the different Wireless LAN standards, the most widespread, by far, is IEEE 802.11 [37]. Others such as RadioLAN [22] and Home RF [129] have not become popular and have lost support. Hiperlan2 [21] still is in the development phase and might arrive too late to rival various IEEE 802.11 standards. Another member of this group, but with more limited bandwidth is Bluetooth [39], which due to several delays in the technology is losing momentum, but still has strong backing.

Wireless LANs are becoming common access networks in several environments. Corporations were the first to install them, in order to allow freedom of movement to their employees. Although this is still by far the most common use of WLANs, new environments are opening up for WLAN deployment such as Small Offices and Homes (SOHO) or public access networks, such as Telia's HomeRun [13].

Public WLAN access networks are being set up in hotspots, i.e. areas with high population density (e.g. train stations or coffee shops) expected to have high demand for bandwidth. As the price for WLAN hardware continues to go down, as WLAN interfaces are increasingly built in notebooks (and other) computers, and user demand increases with the increasing availability of terminals, such public WLAN access networks are expected to become more and more common.

Although complete coverage by such public WLAN access networks is not likely, because of the limited coverage of each access point (of the order of several hundred meters), in city areas where the density of potential users is high the collection of "hotspot" coverage could result in extensive coverage.

Wide area networks (WAN) are built by telecom operators. These WAN networks are usually cellular networks, among which GSM has been the most widespread. However, because GSM was not originally designed to handle packets so it needs upgrades. GPRS and EDGE [141] are upgrades to the GSM network in order to allow better packet transmission (GPRS) with higher bit rates (EDGE). A complete redesign of the GSM network was needed for the third generation networks, like UMTS and CDMA2000. One big advantage of all these packet-oriented networks is that the user does not need to establish a connection in order to send packets.

The goal for these enhanced wide area systems was to provide high bit rates at any place. These solutions have proved far from ideal. The real bit rates provided by these systems and the high infrastructure and spectrum costs are fundamental issues that manufacturers underestimated when designing these systems. In fact operators are becoming interested in



installing WLAN hotspots to provide more capacity to wide-area cellular networks while reducing costs compared to micro cells. It remains a question if WLAN is a complement or a competitor to such cellular networks.

The high development cost of the wide area systems will most certainly lead to high prices for the user, at least at the beginning. A better solution in terms of total cost can be provided by a combination of different access networks. This is becoming known as the fourth generation. It is based on the idea that one access technology cannot be optimised for all different scenarios and that the best solution is to use the most suitable network from a set of technologies that a mobile device supports.

Combining GPRS with IEEE 802.11 access points will allow the user to be always connected and have high bandwidth at hotspots. The price for using the WLAN network in these hotspots will most probably be much less than using the GPRS network at the same location, it might even be part of the basic subscription.

## **1.2 Problem Statement**

Wireless public WLAN access networks provide access to the Internet and to corporate networks at so called hotspots. These are areas with limited coverage but high available bandwidth. Airports and hotels have often been the first targeted locations for these hotspots, but conference centres, cafes, and train stations follow.

Existing solutions for such WLAN access networks lack support for security, flexible accounting, mobility, multiaccess, and user-friendly login. The aim of this Master's Thesis was to study if it is possible to solve these problems and how to integrate all this new functionality into existing public WLAN access networks.

In order to show the results a prototype was built to integrate, modify, and test Authentication, Authorization and Accounting (AAA) [104], Network Address Translation (NAT) [130], IPSec [103], Multiple Packet Delivery Methods [6], Multi-access [6] and Mobile IP [34] existing solutions as adapted to the goal of integrating WLAN and GPRS.

## **1.3 Target Market**

Currently the name "network operator" is associated with large telecom companies such as Telia, Deutsche Telecom, BT, or Telefonica, which have invested huge amounts of money in their infrastructures and have tight control of their users.

In the near future, any person who owns at least one WLAN access point and has a connection to the Internet can become a small operator and offer access to the Internet using these resources. At the present time such a new operator needs to control the access into this network in order to avoid being held responsible for the behaviour of any clients which accesses the Internet through this operator's network. In most countries they would also have to register as a public carrier in order to avoid this liability. They might also need accounting if they plan to charge for the usage of their network.

These small operators are the target market for the system designed in this thesis. Thus neither large operators nor corporations are part of this target market. Examples of small operators that might be interested in our system include airports, hotels, cafes, Internet centres, conference centres, shopping malls, supermarkets, and restaurants.

Corporations are excluded from the targeted market because they do not generally wish to provide public network access and therefore their requirements are quite different, while large operators usually provide their own custom solutions to meet their requirements.

In the case of a normal business, which does not already have a WLAN in their location, if they wish to become small wireless access network operator, R2M (the sponsor of this thesis work) could provide a wireless solution, using hardware from appropriate vendors and the software developed in this thesis.

## **1.4 Outline of the thesis**

Chapter 2 provides an introduction to the main protocols used during the implementation of the system and the previous work examined during the design phase. A reader who is familiar with IEEE 802.11, IPSec, AAA protocols, Private addresses and previous work on public WLAN access networks could skip it.

Chapter 3 explains the goals that drove this Master's Thesis, the derived requirements, and the way that the different problems were tackled. Chapter 4 provides a brief overview of the system architecture in order to make it easier for a reader to understand the following sections.

Chapter 5 provides a more detailed look at the implementation of the system and the problems found and hopefully solved. The different topics such as security, mobility or accounting are each explained in detail.

Chapter 6 analyses the system and compares it to other solutions to establish its utility. It also provides some basic tests of the capacity of the system. Finally, Chapter 7 summarizes our results and Chapter 8 indicates future work.

Appendix A provides a glossary of the terms used in this report. The architecture of the server used in the thesis is presented in Appendix B. Appendix C describes the IEEE 802.x standard for AAA. Modifications made to the Radius server implementation are documented in Appendix D and the SQL queries used are presented in Appendix E. Finally, Appendix F presents the testbed setup used in the implementation phase.

## 2 Background

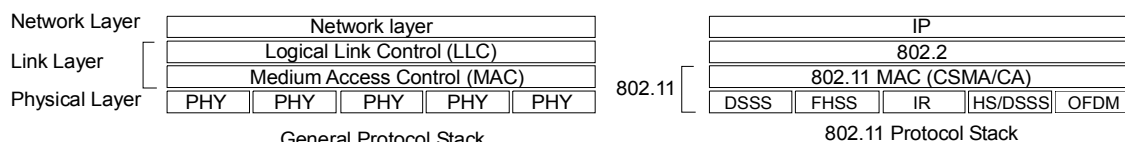
This chapter introduces some of the main protocols used in this thesis and the motivation for using them. In addition, some background to relevant projects is presented.

### 2.1 Protocols

#### 2.1.1 IEEE 802.11

The Institute of Electrical and Electronics Engineers (IEEE) 802.11 [37] standard was ratified in 1997 and revised in March 1999. It defines the Physical (PHY) and Medium Access layers (MAC) for wireless Local Area Networks (LANs)

In fact the standard defined three physical layers and one MAC layer for implementing wireless communications in localized areas. Any 802.11 wireless LAN appears to the Logical Link Control (LLC / IEEE 802.2) layer just as any other 802 LAN, which means, for example, that for the upper layers there is no difference between a IEEE 802.3 (Ethernet) and a IEEE 802.11 wireless network.



**Figure 2.1 IEEE 802.11 Protocol Stack**

The standard originally described two radio physical layers in the 2.4 GHz band based on different techniques: Frequency Hopping (FHSS) and Direct Sequence Spread Spectrum (DSSS). It also defined a third physical layer in the 300-428.000 GHz band for baseband wireless infrared communication.

IEEE 802.11b [79] is a supplement, approved in September 1999, which adds a fourth physical layer to the 802.11 standard. It provides for higher bit rates in the 2.4 GHz band. This is the physical layer mainly considered in this thesis.

IEEE 802.11a [80] is another supplement, which adds yet another physical layer with even higher data bit rates than 802.11b, but located in the 5 GHz band. Commercial hardware supporting it is just becoming available.

The 2.4 GHz (2.4 - 2.4835 GHz) and the 5 GHz frequency bands defined in the standard (and its supplements) are respectively part of the Industrial, Scientific, and Medical (ISM) and the Unlicensed National Information Infrastructure (UNII) bands both defined by the International Telecommunications Union (ITU) [81]. These bands can be used without the need for an end-user license, which is needed in most other bands, but the equipment vendor must certify that the equipment meets the relevant regulations.

Interoperability of IEEE 802.11 products from different hardware vendors is certified by the Wireless Ethernet Compatibility Alliance (WECA) [17], an independent organization which brands compliant devices as “Wi-Fi” compliant, a term which has become synonymous with IEEE 802.11b for many people.

### 2.1.1.1 Architecture Components

The 802.11 architecture is composed of cells, which might overlap. The basic service set (BSS) represents the coverage area of an individual cell and outside the BSS a station cannot communicate with stations in this cell. There exist two different types of BSS providing two different wireless architectures: (1) the individual BSS (IBSS) or Ad-hoc network and (2) the infrastructure BSS.

An independent BSS or Ad-hoc network is composed of a single cell without an Access Point (i.e. no base station). In such a network there is no central point to manage the communication and connect to other networks. Stations can communicate with each other inside the BSS, provided that their signal is strong enough. Such a network is usually created spontaneously since it has no need for external infrastructure.

An example scenario for an Ad-hoc network is to provide a wireless network in an emergency situation (e.g. following an earthquake or a flood). Although no infrastructure is available, teams moving into the affected area need to establish communications. In this case there is no need to install an infrastructure since wireless communication with IEEE 802.11 in Ad-Hoc mode can be quickly established.

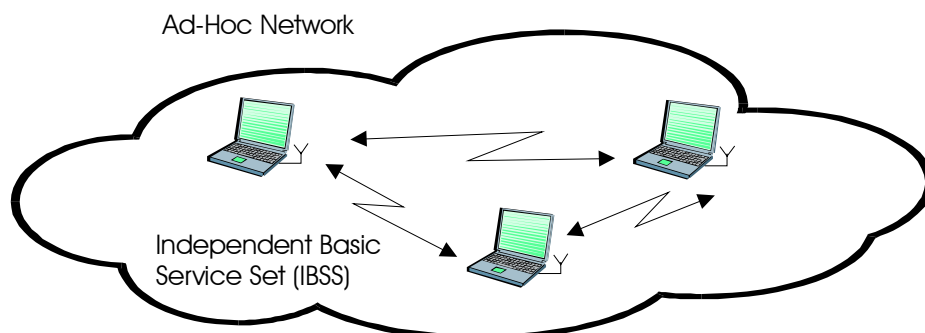
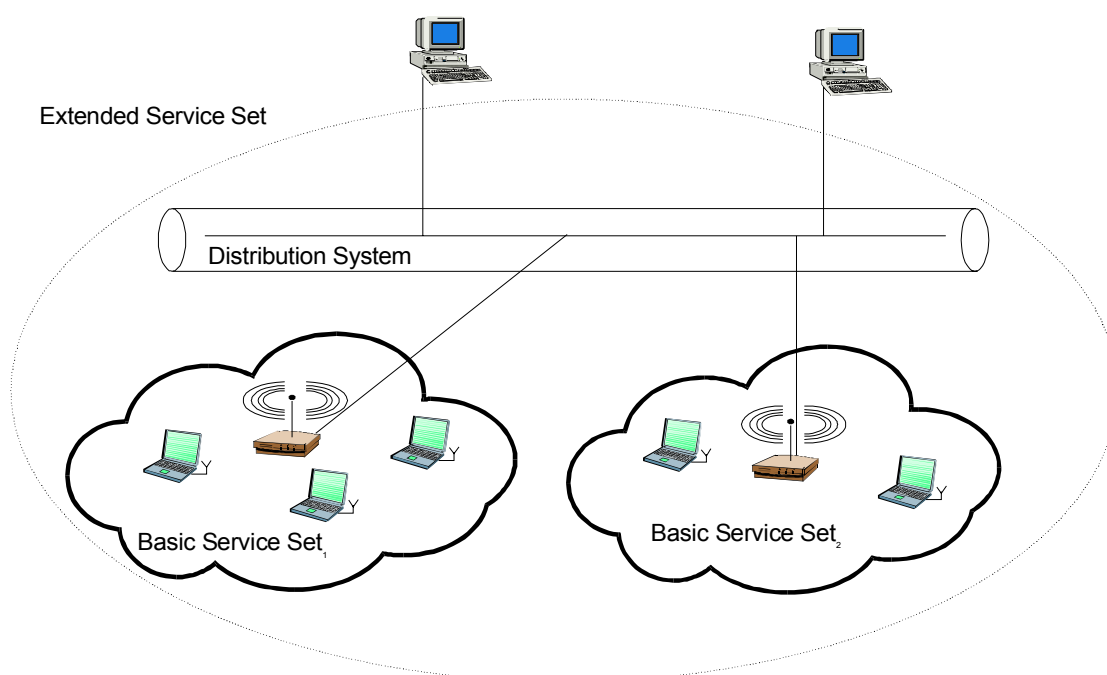


Figure 2.2 Ad-Hoc Network

An infrastructure BSS is composed of a central station called an Access Point (AP) and a variable number of mobile stations. The Access Point provides functionality similar to that provided by a base station in other cellular networks as it acts as a bridge between the wireless segment and the wired segment. This kind of architecture allows for the interconnection of several infrastructure BSS, in order to form what is known as an extended service set (ESS). The ESS is built by connecting several AP through a backbone network called a Distribution System (DS). The most common DS is an 802.3 or Ethernet segment.

Another common name given to the ESS is an infrastructure network. In an infrastructure network the coverage area is defined by the AP, while in an ad-hoc network the coverage area is composed of the overlapping coverage area of each station.

Extending the coverage area of an ad-hoc network is straightforward, much easier and more quickly done than extending the coverage area of an infrastructure network, since no infrastructure needs to be deployed, you simply add new stations. However this is theoretical and it is not clear that it is scalable.



**Figure 2.3** Extended Service Set (ESS)

### 2.1.1.2 Physical layers

The physical layer handles the transmission of the frames via the air interface. In other words it handles the conversion from bits to and from radio signals that can travel through the air. Including both supplements to the original standard, there exist five different physical layers:

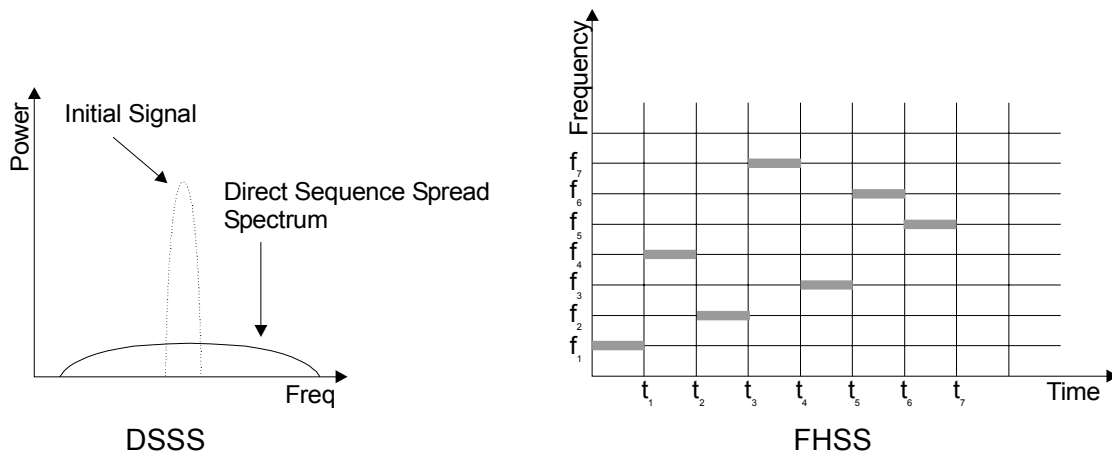
- ❑ Direct Sequence Spread Spectrum (DSSS).
- ❑ Frequency Hoping Spread Spectrum (FHSS).
- ❑ High Rate Direct Sequence Spread Spectrum (HR/DSSS)
- ❑ Orthogonal Frequency Division Multiplexing (OFDM)
- ❑ Infrared (IR)

The first three work in the 2.4 GHz band, OFDM is used in the 5 GHz band, and IR in the 300-428.000 GHz band. The 2.4 GHz band is part of the ISM band while the 5 GHz band belongs to the more recently allocated Unlicensed National Information Infrastructure (UNII) band.

The first commercial APs were usually shipped with either FHSS or DSSS, but since IEEE 802.11b was approved, most APs are shipped with HR/DSSS. This is convenient since interoperability is only possible among those APs using the same physical layer, although all HR/DSSS system can also support DSSS. Access Points supporting OFDM are just beginning to be shipped.

Spread Spectrum technology is a wideband Radio Frequency (RF) technique designed for reliability, integrity, and security. The narrow band RF signal is spread over frequency to make it more robust against interference and make it appear as random noise to an unintended listener, which does not know the transmission parameters.

Frequency Hopping Spread Spectrum uses a narrowband carrier that changes frequency at each time slot. The movement between different frequencies is defined by a hopping pattern. Several hopping patterns (three sets) are defined in the standard.



**Figure 2.4 Spread Spectrum techniques**

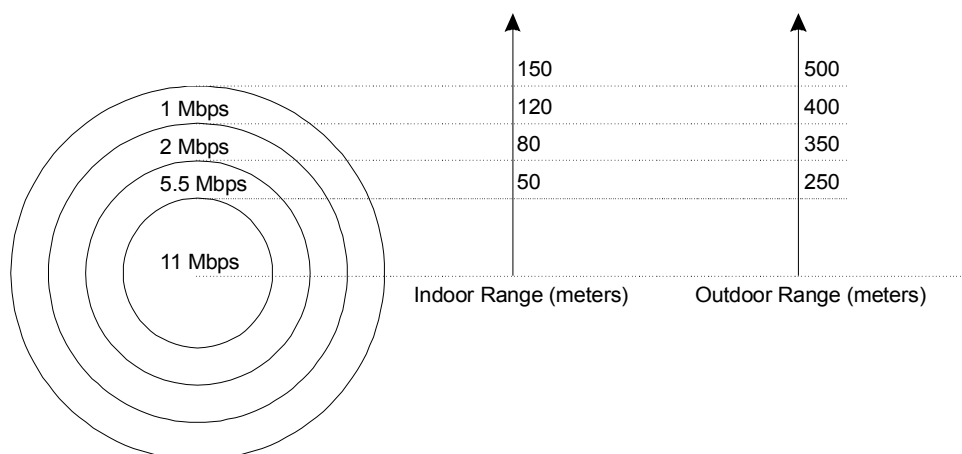
Direct Sequence Spread Spectrum convolves a pseudo random bit pattern with each bit or group of bits that needs to be transmitted over the air. The bit pattern is called the chipping code. The longer the chipping code, the lower the symbol rate but the more protection to the signal also.

Infrared communication allows line-of-sight connections or diffuse communications since infrared light, cannot go through opaque objects. The range is quite limited even for diffuse communication; hence it will not be considered further in this thesis.

**2.1.1.3 Data Rates**

The three original physical layers supported 1 Mbit/s and 2 Mbit/s data rates. For FHSS and IR, the 2 Mbit/s data rate was optional. For DSSS both data rates were mandated.

Another two data rates were defined in 802.11b: 5.5 Mbit/s and 11 Mbit/s. To provide the higher rates, 8-chip complementary code keying (CCK) was defined as the modulation scheme. Thus, an AP that supports 802.11b can provide four different data rates: 1, 2, 5.5, or 11 Mbps. An even higher data rate is being defined as part of 802.11g. The OFDM physical layer, defined in 802.11a provides a wireless LAN with data rates of 6, 9, 12, 18, 24, 36, 48, and 54 Mbit/s.



**Figure 2.5 Performance of 802.11 versus distance**

The data rates step down over distance, that means that several users connected to the same AP might be connected at different data rates depending on the strength of the received signal.

When the signal received at the client is not strong enough to connect at 1 Mbps, the user cannot connect to the AP (see Figure 2.5).

The data in the previous figure was taken from the Linksys WAP 11 Access Point [131] and might differ for APs from other vendors. The coverage area can be extended (reshaped) using directional antennas, although this changes the propagation pattern.

#### **2.1.1.4 MAC layer**

All the physical layers support a common MAC layer. The protocol used is Carrier Sense Multiple Access with collision avoidance (CSMA/CA). A station, which wishes to transmit some data, first listens to the channel to be sure no other station is transmitting and then sends its data.

Collision detection (CSMA/CD) as used in Ethernet cannot be properly used since in a wireless segment it might happen that not all stations can hear each other and it is hard to listen when transmitting due to the relative signal strengths, which is one of the basic assumptions for CSMA/CD. If two stations do not hear each other (Hidden Terminal problem) they might try to transmit at the same time, thus creating collisions.

An optional procedure (RTS-CTS) is provided to help to reduce the Hidden Terminal problem. When a station wants to transmit a frame, it first sends a short Ready To Send (RTS) packet with the length of the frame that will follow. The receiver then acknowledges that it is ready by sending a short Clear-To-Send (CTS) packet and then the transmitter sends the frame. The receiver checks the Cyclic Redundancy Check (CRC) of the frame and sends an ACK packet to the transmitter if no errors were found. Any other station that listens to this exchange uses the length in the RTS packet to estimate how long until the medium is again available.

Fragmentation and reassembly are usually addressed at the network layer, but 802.11 also considers them. Due to higher bit error rates (BER) in wireless segments, compared to wired segments, it is useful to reduce the size of the frames transmitted to avoid retransmissions even at the cost of increased overhead. To deal with the same frame sizes that 802.3 supports (1518 bytes), a simple fragmentation and reassembly method was added to the MAC layer.

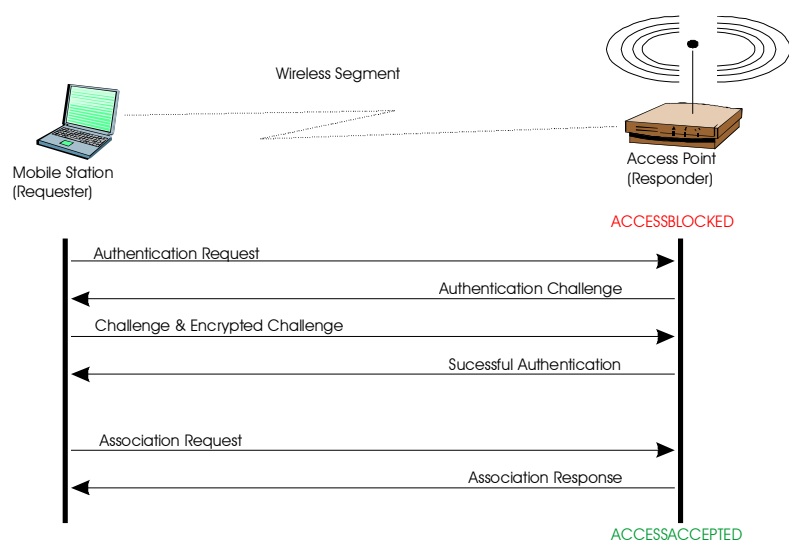
For a deeper knowledge of the physical and MAC layers in 802.11 the reader should refer to [37].

#### **2.1.1.5 Communication exchange**

There are nine services specified by the IEEE 802.11 standard. Six of these services are used to support frame delivery between stations. Three of the services are used to provide access control and privacy. The nine services are: Authentication, Association, Deauthentication, Dissociation, Distribution, Integration, Privacy, Reassociation, and frame delivery.

In an infrastructure network, the only architecture considered in this thesis work, before starting to send and receive data across the wireless segment, a station has to locate and synchronize with the AP. This can be done in two ways: passive scanning and active scanning.

With active scanning a station transmits a Probe Request frame and waits for a Probe Response from the AP, while with passive scanning the station waits until it receives a Beacon frame from the AP.



**Figure 2.6 Authentication & Association in 802.11**

Beacon frames are sent out periodically by the AP in order to inform stations of its presence. They contain information about the chipping code (for DSSS) or the hopping sequence (for FHSS), synchronization info, and the Service Set Id (SSID). The SSID is a name given to the BSS, it is used in order to be able to differentiate between different BSS when cells overlap.

Once the AP is located and the station is synchronized with it, the station starts a process called Association. A station, which is not associated with an AP, cannot send or receive data using that AP. The association is needed in order for the Distribution System to know where to send data for that station. Each station can only be associated with one AP, but an AP can be associated with several stations. An authentication process has to be carried out before association. The combined process is shown in Figure 2.6.

## 2.1.2 AAA

Several Authentication, Authorization, and Accounting (AAA) protocols exist today; the most common are the Remote Access Dial In User Service (RADIUS) [25, 26, 27] and TACACS+ [24]. Both RADIUS and TACACS+ were primary developed and used for remote dial-in services, and are based on a client server model. The requirements in remote dial-in services can be a bit different from the requirements in a wireless access network. Unfortunately neither RADIUS nor TACACS+ were designed to support wireless access networks.

The Internet Engineering Task Force (IETF) [105] has been working for several years on the specification of a new AAA protocol called DIAMETER [28, 29, 30], which takes wireless networks into consideration and integrates other protocols such as Mobile IP with the AAA functionality.

TACACS+ has some good features that RADIUS doesn't: not only can the password be encrypted, but also other information sent in the TACACS+ packet can be encrypted. In addition, authentication and authorization are separated which is not the case in RADIUS. These two features are perhaps not that important in the system described in this thesis, since a separation between authentication and authorization will not be necessary and because we plan to use IPSec between client and server the extra encryption of TACACS+ will not be necessary.



According to [32], TACACS+ does only support time based accounting but does not support accounting based on the number of packets or number of bytes sent, however, this is essential for us, as an operator may want to use this information in computing their charges.

RADIUS on the other hand supports sending a variety of information between client and server, and it allows you to add your own vendor specific attributes. More information about RADIUS is presented in 2.1.2.1. There are a lot of implementations of the RADIUS protocol; one of the best is FreeRadius [31], which is available under GNU General Public License (GPL) [33] license. This is the implementation that we will use throughout this thesis. RADIUS also has support for proxy servers; they can easily redirect authentication requests to another RADIUS server, this makes it easy to implement roaming between different network providers.

Another reason for choosing RADIUS is to make the transition to DIAMETER smoother, since DIAMETER is backwards compatible with RADIUS, thus making it possible to integrate both infrastructures. DIAMETER solves most of the weaknesses of the RADIUS protocol and adds some valuable extra features for mobility. More about DIAMETER can be found in Section 2.1.2.2 .

Another good reason to support RADIUS is that most ISPs already own a RADIUS server with a large user database, which they are already using for their dial-up connections. In order to provide roaming for such networks, RADIUS becomes the best protocol choice for a Wireless Internet Service Provider (WISP).

In conclusion, RADIUS is the best-suited AAA protocol to be used by a WISP today, although DIAMETER is likely to become a better choice in the near future.

### 2.1.2.1 RADIUS

RADIUS uses a client server model. The Network Access Server (NAS) operates as a RADIUS client. Figure 2.7 illustrates how the communication is done.

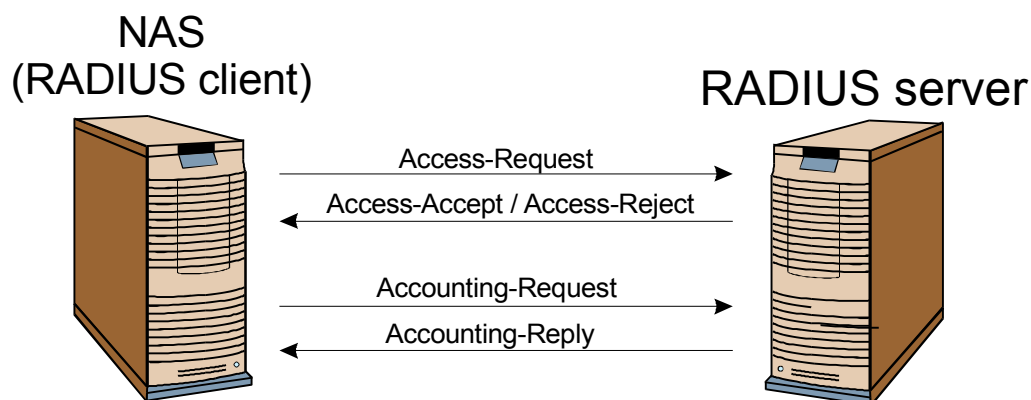


Figure 2.7 RADIUS communication

RADIUS servers are responsible for receiving a user connection request, authenticating the user, and then return all configuration information necessary for the client (NAS) to deliver service to the user. If accounting is used, then the RADIUS server is also used to receive and store accounting information in a suitable way.

A RADIUS server can act as a proxy client to other RADIUS servers or other kinds of authentication servers. This makes it possible to implement roaming between two or more different administrative entities by allowing cross authentication. One forwarding proxy

server can forward to another forwarding server to create a chain of proxies, although care must be taken to avoid introducing loops.

Transactions between the client and RADIUS server are authenticated through a shared secret, which is never transmitted over the network. If the server gets a packet that the server can't authenticate then the server silently discards the packet, the same goes for the client.

The RADIUS packets are sent in clear text format, except for the user password, which is hidden using a method based on the RSA Message Digest Algorithm MD5 [106]. However, a problem with RADIUS is that there is no end-to-end security when a proxy server is used, as security can only be assured for each hop.

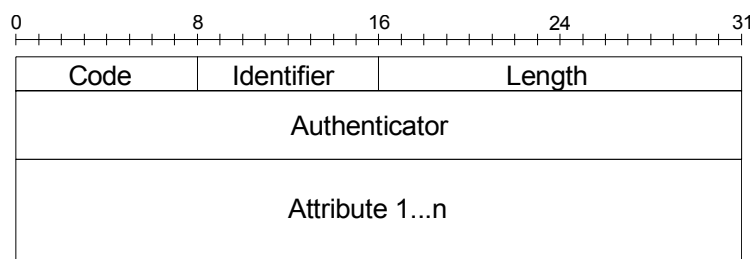
RADIUS use UDP as its transmission protocol, there are several technical reasons for this. Retransmission to secondary authentication server is done when the primary server is not available, this require timers that differ from retransmission timers defined in the TCP protocol. Additional motivations can be read in RFC 2865 [25]. Exactly one RADIUS packet is encapsulated in every UDP packet, the destination port should be set to 1812 for authentication packets and port 1813 for accounting packets. Early deployment of RADIUS used ports 1645 respective 1646. When a reply is generated, the source and destination ports are reversed.

Authentication of the user can be done with different mechanisms, such as:

- Cleartext Password Authentication Protocol (PAP)
- Challenge Handshake Authentication Protocol (CHAP)
- Challenge / Response procedures
- Extensible Authentication Protocol (EAP)

### Packet format

The data format of a RADIUS packet is shown in Figure 2.8.7



**Figure 2.8 RADIUS packet**

The code field identifies the type of RADIUS packet. If an invalid code is used, then the packet is silently discarded. The available codes are:

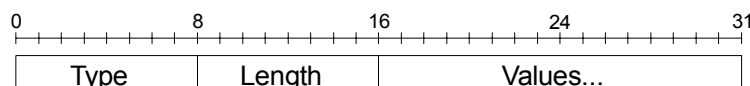
<u>Code</u>	<u>Type</u>
1	Access-Request
2	Access-Accept
3	Access-Reject
4	Accounting-Request
5	Accounting-Reply
11	Access-Challenge
12	Status-Server (experimental)
13	Status-Client (experimental)
255	Reserved

For example, when the RADIUS client wants to authenticate a user it will send an Access-Request and the RADIUS server will reply with either an Access-Accept or with an Access-Reject packet.

The identifier field is used to match requests and replies. The RADIUS server can detect a duplicate request, if it has the same client source IP address and source UDP port and Identifier within a short span of time.

The Length field indicates the total length of the packet, including the RADIUS header. The Authenticator field is 16 octets long, and is used to provide authentication for the requests and replies. The authenticator is based on a MD5 digest value of a random number and the secret shared between the server and client.

The Attribute field specifies what kind of information that is sent in the packet. There are over 60 specified types of attributes in RFC 2865 and new attributes can be added by a vendor, although if new attributes are added there are no guarantees that RADIUS communication between different vendors will work the way it should. Each packet can contain several attributes. The data format of the attribute can be seen in Figure 2.9.



**Figure 2.9 Attribute Type-Length-Value format**

The Type specifies what attribute is sent, the size is 1 octet this mean that a total number of 256 different attributes can be specified. The maximum size of an attribute is 253 bytes since two bytes are used for the Type and the Length fields and since the length field is 8 bits, the total length is  $2^8$  max bytes. Some example of attributes can be seen below.

<u>Type</u>	<u>Name</u>
1	User-Name
2	User-Password
6	Service-Type
12	Framed-MTU
27	Session-Timeout
33	Proxy-State
40	Acct-Status-Type

How the communication and what exact information that is being sent will be presented later in Section 5.3.

### 2.1.2.2 DIAMETER

DIAMETER is the successor to the RADIUS protocol. It addresses many of the flaws in the RADIUS protocol, such as security weaknesses and mobility support. An important change is that DIAMETER works in a peer-to-peer fashion, and not in a client/server way as does RADIUS. This allows the server to send unsolicited messages to the NAS, this could be used for instance to allow the server to inform the NAS that a session should be terminated.

The base protocol defines the basic functionality that must be provided to all services supported in DIAMETER. Application specific functionality is provided through extension mechanisms, e.g. Mobile IP, Accounting, etc.

The base protocol also defines a windowing scheme, which requires each peer to advertise a receiver window. This allows the peer to control the flow of packets, and can be used to distribute the load across multiple servers.

With the strong security extension, DIAMETER can ensure end-to-end integrity and authenticity of the entire packet or part of it. This corrects one of the weaknesses of RADIUS, where only the password could be encrypted. This means that some of the attributes in the packet can be secured and some non-secured (unprotected). This can be useful when the packet goes via a third party proxy, and the proxy needs to know some Attribute Value Pair (AVP), but other AVPs need to be protected.

RADIUS limited the length of each AVP to 255 octets. DIAMETER removes this limitation, making the protocol much more flexible and efficient. For compatibility reasons all RADIUS AVPs are also supported in DIAMETER, plus a lot of new application specific extensions can be added.

### 2.1.3 IP Security

Internet Protocol Security [88] is a standard developed by the IETF. It defines an architecture for securing IP networks. The goal of the architecture is to provide various security services at the IP layer for both the IPv4 and IPv6 environments.

The whole architecture is quite complex and extensive. It contains over 12 documents [93]. The IPsec architecture is composed of several components, the fundamental ones being:

- ❑ Security Protocols: Authentication Header (AH) and Encapsulating Security Payload (ESP)
- ❑ Security Associations: what they are, how they work, and how they are managed.
- ❑ Key Management: manual and automatic (Internet Key Exchange (IKE))
- ❑ Algorithms for authentication and encryption (ciphers).

In general IP Security defines a framework containing several protocols, not only AH and ESP but also IKE [92], ISAKMP [91], OAKLEY [94], and how to use different ciphers with them. However, it does not specify all the possible algorithms that can be used although it mandates a basic set.

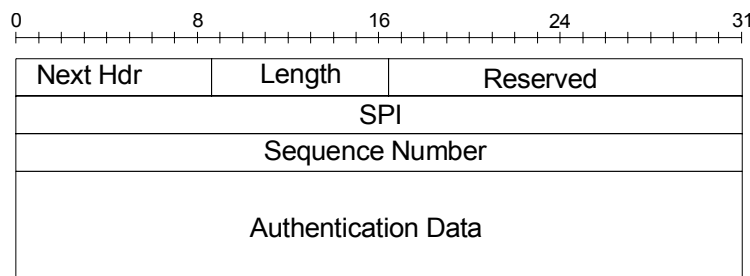
IPsec provides connectionless security over the IP layer, securing transparently all the upper layer protocols including transport protocols, such as TCP and UDP, and applications. No modifications are needed to protocols above the network layer (ISO layer 3) in order to incorporate IPsec into an existing application.

#### 2.1.3.1 Security protocols

IPsec provides security between two entities (point to point) and both of them need to support the IPsec protocols although no changes are needed in the rest of entities in the network. Security services can be provided between a pair of communicating hosts, between a pair of communicating security gateways, or between a security gateway and a host.

Two security protocols are used to provide these security services: the Authentication Header [89] and the Encapsulation Security Payload [90]. The AH provides integrity, data origin authentication, and protection against replay attacks to IP packets. It protects the IP header and the upper layer protocol data. The ESP provides confidentiality, data origin authentication, integrity, limited traffic flow analysis protection, and replay protection to IP packets.

Both structures can be used together or independently, although it is more common to use only one of them because AH plus ESP generates much overhead and little extra service since ESP provides also authentication and integrity of the data. The primary difference between the authentication provided by ESP and AH is that in transport mode no IP header fields are protected by ESP. In tunnel mode ESP provides authentication and integrity of the whole inner IP header (see Figure 2.12).



**Figure 2.10 Authentication Header**

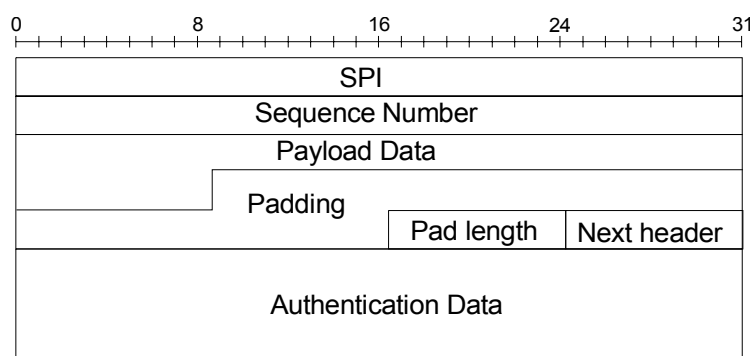
### Authentication header

As mentioned before, the AH provides connectionless integrity, data origin authentication, and protection against replays for IP datagrams. However, some fields in the IP header may change during the network traversal and cannot be protected by the AH. Thus the IP header is only partially protected.

The AH provides a general Integrity Check Value (ICV), which is a keyed hash of the data payload. The standard does not specify the specific hashing algorithm (e.g. HMAC-MD5 or HMAC-SHA-1).

### Encapsulation Security Payload

As mentioned before, ESP provides confidentiality, data origin authentication, connectionless integrity, replay protection, and limited traffic flow confidentiality (only in tunnel mode, see below). The set of services provided depends on the options selected at the time of establishing the Security Association.



**Figure 2.11 Encapsulation Security Payload**

Both headers are inserted between the IP header and the IP payload data. See the next Section for details.

### 2.1.3.2 Modes of operation

The IPSec architecture defines two modes of usage: transport mode and tunnel mode. The transport mode is used to provide host-to-host security. It protects the entire IP payload data, but it does not protect the IP header.

The tunnel mode can be used in all host-to-host, host-to-gateway, gateway-to-gateway configurations but the usual configurations have at least one gateway as an end point. It encapsulates and protects the whole original IP packet, including the original IP header, and attaches a new IP header describing the tunnel between both end points. It is ideal for building a Virtual Private Network (VPN).

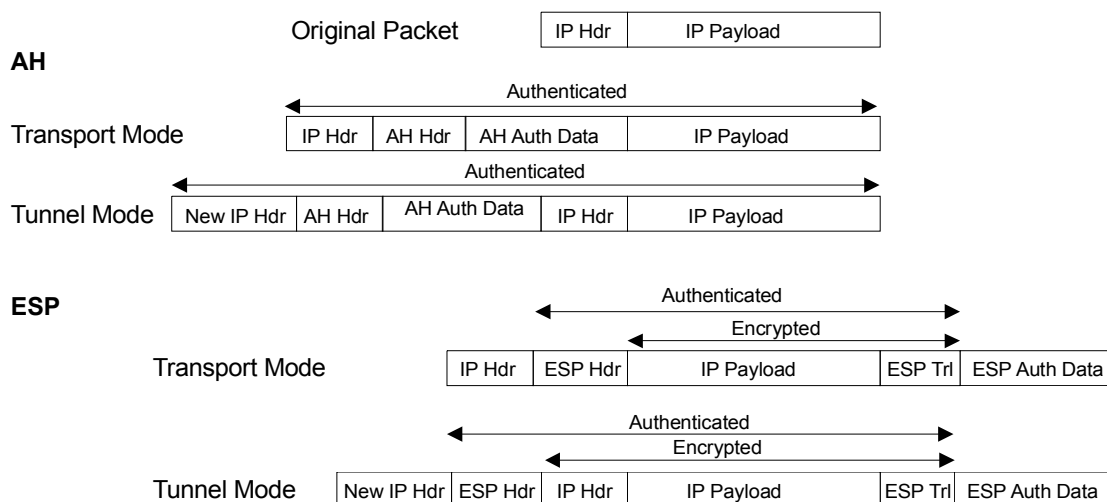


Figure 2.12 IPSec tunnel and transport modes

### 2.1.3.3 Key management in the IPSec framework

The Internet Security Association and Key Management Protocol (ISAKMP) [91] is also part of the general IPSec architecture. It defines a framework for security association management and secure key generation and distribution, but does not specify any key exchange protocols.

### 2.1.3.4 Internet Key Exchange

Usage of the AH and ESP structures assumes that a security association defining the specific algorithms to be used is in place, but does not provide a mechanism for creating it. Another protocol for managing security associations is needed. After discussing several protocols the IETF chose the Internet Key Exchange (IKE) as the standard mechanism for configuration of security associations.

In other words, the purpose of IKE is to negotiate and generate in a secure manner authenticated keying material for security associations. IKE is a complex protocol and an extensive description of it is considered out of the scope of this thesis. A reader might refer to [92] for further information.

## 2.1.4 Private addresses and Network Address Translation (NAT)

Private addresses are blocks of IP addresses that the Internet Assigned Numbers Authority (IANA) has reserved for private networks. These are isolated networks, not connected to other networks or to the Internet. They can therefore do their own assignment of addresses without risk of colliding with other domains [107]. The three following blocks have been reserved:

10.0.0.0 - 10.255.255.255 (10/8 prefix)  
 172.16.0.0 - 172.31.255.255 (172.16/12 prefix)  
 192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

Private addresses were reserved to slow down the consumption of IP addresses since several different private networks can reuse the same allocation range without needing address allocation by the IANA or other organization. Unfortunately they have become widespread and partly lost their original intent, by being used in many networks that are not isolated. In order for this to be possible a translation mechanism was needed.

Traditional Network Address Translation (Traditional NAT) [130] allows hosts in a private network to transparently access the external network and enables access to selected local hosts from the outside. It is a mechanism by which IP addresses are mapped from one group to another, transparent to end-users.

IP addresses need to be translated when the internal addresses cannot be used outside because they are private addresses (and not routable outside the network) or because of privacy reasons. NAT has helped to slow down the use of IP addresses, but created many problems in return because some protocol use IP addresses inside their data.

There exist two different mechanisms inside Traditional NAT: Basic NAT and NAPT. In Basic NAT multiple internal addresses map to multiple external addresses without any other changes needed. In NAPT many IP addresses and their TCP/UDP ports are translated into a single network address and new TCP/UDP port numbers. A host implementing either of these mechanisms is called the NAT gateway.

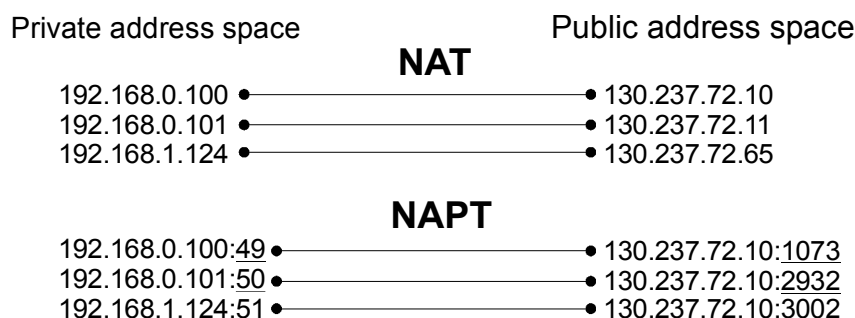


Figure 2.13 Basic NAT and NAPT description

## 2.2 Related work

Since this thesis covers many different topics, thus a large amount of previous work can be found. It is convenient to separate the academic research work done in the different areas: mobility, multiaccess, and security, from the commercial approaches to these problems.

### 2.2.1 Research work

Several alternatives for the integration of WLAN and GPRS were studied in [1] where the authors found that Mobile IP is probably the most suitable solution because it means minimal changes to the existing infrastructures and standards. It is also the only solution based on standardization (Internet Society through the Internet Engineering Task Force). The paper also describes several different handover algorithms and provides some handover measurements with Mobile IP.

During a study of roaming between GPRS and WLAN using Mobile IP [2] it was found that private addresses constitute a major problem for the deployment of this kind of roaming

scenarios, and several solutions were proposed. Also recently a new solution to this problem has been published as an Internet draft [3] based on the insertion of an extra UDP header in the IP over IP encapsulation method used in Mobile IP. Another solution has been proposed in [133] and both have been recently blended together in [132].

Also in the context of mobility, the usage of simultaneous bindings in Mobile IP has been studied in order to provide seamless handovers [4].

The Mosquitonet group [5] from Stanford University has developed a Mobile IPv4 implementation based on the use of co-located care of addresses, which also includes some interesting flow-oriented mechanisms. In [6], two different mechanisms were described: multiple packet delivery methods and simultaneous use of different network interfaces. Based on these two mechanisms, this thesis will establish the concept of multiaccess (use of multiple access networks) and its usefulness in the context of WLAN/GPRS roaming.

The security in WLAN networks has been studied in numerous papers, but most concern only the link layer. A good analysis of the general security requirements for Public WLAN Access Networks can be found in [7] where security in DHCP is also examined. From the different proposals to secure DHCP leases, one has recently become standardized [8].

There has also been some study as to how to introduce basic accounting in such public Wireless LAN scenarios [9]. This paper describes a scenario similar to the one used in this thesis, but with much simpler requirements. A comparison of AAA protocols has also been recently published [143].

There have been several academic implementations of related systems. Two of them are specially worth mentioning. The first one by the Stanford University was called Spinach [35] and provided access control over both public wireless access points and public Ethernet ports on the university campus.

The second one at the Royal Institute of Technology (KTH, Stockholm) for use with information technology students is described in [11] and introduced in a more general project called StockholmOpen.net [12], which is an approach to a city wide open wireless access network with a freedom of choice of service operators. An upgrade of the system has been deployed during autumn 2001 to address some of the former system's weaknesses.

Both systems have been thoroughly used by several hundred people and are still being used at the time of our writing this report. There exist several other universities that provide similar solutions, but not all of them are directly related to this thesis or they are derived from the previously mentioned systems, examples are Carnegie Mellon University, Uppsala University, and UC Berkeley.

## 2.2.2 Commercial work

Today several large operators provide public WLAN access, for example Telia in Sweden, Sonera in Finland, and Telenor in Norway. Of all these, Telia has the most extensive and widely used solution, with their commercial service called Homerun, providing access in numerous locations in Sweden. This service has recently been updated with two new options focused on SOHO environments and corporations.

Smaller operators are also providing public WLAN networks on a regional basis mainly in California in the United States, and in Europe's northern countries. In Stockholm at least two of these operators have put up access points in the city centre [108, 134].



Several companies are working towards the integration of GPRS/WLAN networks. Most of them are focusing on the corporate market, implementing Virtual Private Network (VPN) solutions. Two different approaches to the problem can be identified. One is to provide roaming at the network layer and the other one provides roaming at the transport/session layer. A one-sided comparison of IP mobility to Session Mobility is provided in [14].

Companies listed below are trying to provide roaming at the network layer. They use Mobile IPv4 as the mobility protocol and IP Security (IPSec) for the security needed in the VPN. In this way both security and roaming are transparent to the transport layer.

- ❑ IPUnplugged (Sweden) [www.ipunplugged.com](http://www.ipunplugged.com)
- ❑ Ecutel (USA) [www.ecutel.com](http://www.ecutel.com)
- ❑ Birdstep (Norway) [www.birdstep.com](http://www.birdstep.com)
- ❑ Lifix (Finland) [www.lifix.fi](http://www.lifix.fi)
- ❑ IKV++ (Germany) [www.ikv.de/en/products/roamin/index.html](http://www.ikv.de/en/products/roamin/index.html)
- ❑ Commitment (Norway) [www.commitment.no](http://www.commitment.no)

Some companies are trying to provide roaming at the Transport/Session Layer [135]. They do not care if the transport layer connection is broken. Instead they rely on recovery mechanisms at the session layer for fast transport connection re-establishment. The security is provided by Wireless Transport Layer Security (WTLS), which is a version of TLS adapted for wireless use.

Finally there exist several groups trying to bring together access points owners in order to add value for all the locations, by allowing the roaming between them. SOHO Wireless [15] is one of these groups and quoting from its webpage:

*“The vision of SOHO Wireless is to help popularise and proliferate the use of inexpensive 802.11 standardized radios so as to establish a ‘Fabric’ of radio access to the Internet at all hot zone locations where there are high concentrations of users. In order to expand SOHO wireless coverage rapidly and without requiring a lot of additional capital, the Company plans to distribute its client Linux access control software for free as part of the Linux distribution open source code”.*

SOHO Wireless’ software is called LANRoamer. It allows any user with an IEEE 802.11b interface to act as a base station for their LANRoamer subscribers. The user gets a bootable CD-ROM from LANRoamer and puts it in his or her computer. Little configuration has to be done by the user; all information is stored on the CD-ROM. It is still in early development and much remains to be done. The software for the NAS is released under GNU GPL license. However, the owner of the Access Point loses the control of his or her network, which we believe is not acceptable.

Another similar project is NoCat [16], quoting from their web page:

*“We are working to build a community supported 802.11b wireless network in Sonoma County, CA. We are actively developing WRP (a Linux distribution-on-a-floppy that provides wireless support) and NoCat, the centralized authentication code to make shared Internet services possible.”*

Both projects use a similar architecture to the one used in this thesis (see Chapter 4), but they provide software to their users/associates in order to implement the Network Access Server (NAS).

## 3 Method

### 3.1 Goal

The main goal was to build and analyze a prototype of a public WLAN access network that could serve as a proof of concept for a general public WLAN solution. The following sub-goals were identified at an early stage of the work:

- ❑ Specify the requirements of a public WLAN access network from both the user and the operator point of view
- ❑ Study different currently available systems and how they could be improved
- ❑ Design the system based on the previously stated requirements and the study of other solutions
- ❑ Set up a testbed to support implementation and testing of the system
- ❑ Implement the solution
- ❑ Analyse the system and specify future improvements

### 3.2 Scope

The following topics limit the scope of the work: mobility, accounting, security, and multi-access.

Mobility is one of the functions that the system should offer to the clients. Many users of these public access networks would like to maintain their access when leaving the hotspot, even at the expense of greatly reduced bandwidth. For this a new solution is required that provides roaming to the GPRS network when the client moves out of the coverage area provided by hotspots.

Accounting should be provided in a flexible way, providing support for different payment methods and allowing future methods. Scalable support for roaming between different operators' networks integrates with accounting in the AAA requirements. Integration between the AAA and mobility (e.g. RADIUS and Mobile IP) should also be analyzed.

Security has been heavily criticized in WLAN systems. User friendliness has to be balanced with the security requirements to provide a good solution.

Multi-access focuses on the use of overlay networks. When a user is at a hotspot it will probably also have GPRS coverage. At such a time packets can go via two different access networks. The user would usually prefer to send packets through the WLAN network due to both greater bandwidth and lower price than via GPRS. However, for certain applications, such as those with a required Quality of Service, the user might want to send packets through the GPRS network. The ability to choose which packets go through which network is the focus of multi-access.

Providing a frontend that gives feedback to the user about the resources they have used, and the user perspective when using the system are considered also part of the scope of this thesis.

The following topics are considered out of the scope of this thesis:

- ❑ Ad-Hoc networks
- ❑ Security in GPRS
- ❑ Mobility solutions such as HAWAI , Cellular IP, TeleMip and so on.
- ❑ Network Management

### 3.3 Limitations

- ❑ The client needs support for Java and JavaScript in its browser.
- ❑ The interface with the banking system needed to allow real credit card payment has not been implemented.
- ❑ No suitable implementation of DIAMETER has been found.  
Unfortunately, DIAMETER is still under development, and not many implementations even exist. SUN Microsystems has one early beta available for testing, but it is only available in binary form and only runs under Solaris 8 on the SPARC architecture, such a machine was not available in the company.
- ❑ Basic users should not need any extra hardware or software.

### 3.4 System Requirements

The system requirements were specified during the design phase of the thesis. Most of the public wireless systems share the same main requirements: low cost, easy administration, user-friendliness, flexibility, and scalability and these have been used as an starting point to define more specific requirements.

Flexibility, meaning that different scenarios should be taken into account is an important requirement as is scalability, meaning that the system should be easy to upgrade. The system should also allow for the incorporation of other services later, such as location-specific services.

The key words "**must**", "**should**", and "**may**" in this thesis are to be interpreted as described in RFC 2119. These key words mean the same thing whether capitalized or not.

#### 3.4.1 General

- ❑ It **must** be possible to run different combinations of the NAS modules.
- ❑ Connectivity to outside networks (e.g. Internet) **must** only be provided through the NAS.
- ❑ The system **must** at least support two kinds of clients:
  1. The basic user is only required to have:
    - a laptop or handheld device with an IEEE 802.11b compliant interface,
    - an OS with TCP/IP support and a DHCP client, and
    - a browser with HTTPS support (SSL).
  2. The advanced user, needs to have or to be provided with:
    - IPsec client software,
    - Mobile IP client, and
    - Kerberos V client.
- ❑ The user **should** be able to choose their own WLAN adapter. The user **MAY** need to notify the system about a change in adapters.
- ❑ Redundancy requirements **should** be studied and **might** be implemented in order to allow a fault-protected system.

- ❑ An analysis of the capacity of the system **should** be performed after the implementation is finished. That means being able to measure how many users the system can handle.
- ❑ Roaming with another operators **should** be included in the design of the system.
- ❑ The advanced user **may** optionally utilize a GPRS subscription in order to use wide area mobility, although the wireless operator could provide this subscription.
- ❑ Each user **may** have a User Profile, which contains all the information about this user and it should be retrievable by some external means (such as HTTP).
- ❑ The system **may** provide a method for users to securely update their User Profiles, at least the Password and the MAC address.
- ❑ The system **may** provide support for local services.

### 3.4.2 Login

- ❑ The user **must** be identified by a unique user-name.
- ❑ The user **must** not be able to start several sessions simultaneously.
- ❑ After authentication, the user **must** be redirected to the webpage first requested.
- ❑ The system **should** not require the user to log in more than once for a session, to ensure convenience.
- ❑ The system **should** provide feedback to the user about the status of his connection.
- ❑ Unsuccessful logins **should** be recorded for security purposes.
- ❑ A User Manual **may** be implemented and be accessible from the applet.
- ❑ After a fixed number (e.g. 5) of unsuccessful logins the system **may** block that account for several minutes (e.g. 10 minutes).

### 3.4.3 Security

- ❑ The user **must** be able to authenticate the system before revealing his username and password. In other words, *mutual* authentication is needed.
- ❑ Filtering at the firewall **must** be done on both IP and MAC to avoid IP address spoofing.
- ❑ It **must** NOT be possible for any non-root user to change firewall rules.
- ❑ The system **should** be able to close the session when the user stops using it. A manual logout **must** exist, but **should** not be the only procedure. A keep-alive procedure **should** be implemented.
- ❑ Protection from IP and MAC spoofing **should** be provided.

- ❑ Wire Equivalent Privacy (WEP) **should** only be used if keys are dynamically changed quite often. Even so it **may** not be the primary protection of the wireless segment.
- ❑ The system **should** have the possibility to check MAC addresses (possibly at access points) in authorization, but this is disabled by default.
- ❑ The status of the NAS **should** be logged.
- ❑ Passwords **should** never be stored in clear text. They **should** be hashed when provided and stored in hashed form in the database. The incoming requests **should** only compare hashed passwords.
- ❑ Internal traffic between clients on the wireless network **may** need to be controlled.
- ❑ All unnecessary accounts **may** be removed and all unused ports closed at the NAS.

### 3.4.4 Accounting

- ❑ The system **must** support different payments methods.
- ❑ The system **should** allow later additions of new payment methods.
- ❑ At least the following payment methods **must** be supported by the system:

**Prepaid** - The user pays in advance for a limited amount of time or data to be transferred. When the time or data paid for is up, the user is blocked.

- Information the user has to provide: username and password
- Information stored in the database: username, time connected, time left, bytes left to send, and bytes left to receive
- Authentication based on: username, password, and MAC address

**Credit card** - The user pays with its credit card.

- Information the user has to provide: name, credit card number, expiration date, and type of card (e.g. VISA or MasterCard)
- Information stored in the database: credit card number, time connected, bytes sent, and bytes received
- Authentication based on: Credit Card information, and MAC address

**Test** - The user gets a chance to test the service for five minutes, but only once for each MAC address.

- Information the user has to provide: none
- Information stored in the database: MAC address, and time connected
- Authentication based on: MAC address

**Subscription** - The user gets billed every fixed amount of time (e.g. at the end of each month) for the time connected or amount data transferred during that period.

- Information the user has to provide: username and password
- Information stored in the database: username, time connected, bytes sent, bytes received, bank account, bank name, name, and person number
- Authentication based on: username, password, and MAC address

**Invoice** - Like a temporary subscription, e.g. the user stays at a hotel and just wants to use the service during that visit. The user can get the amount to be paid in the hotel's bill.

- Information the user has to provide: username and password
- Information stored in the database: username, time connected, bytes sent, bytes received, address, name, and person number
- Authentication based on: username, password, and MAC address

## 4 System Architecture

This chapter provides a brief description of the system.

### 4.1 System Overview

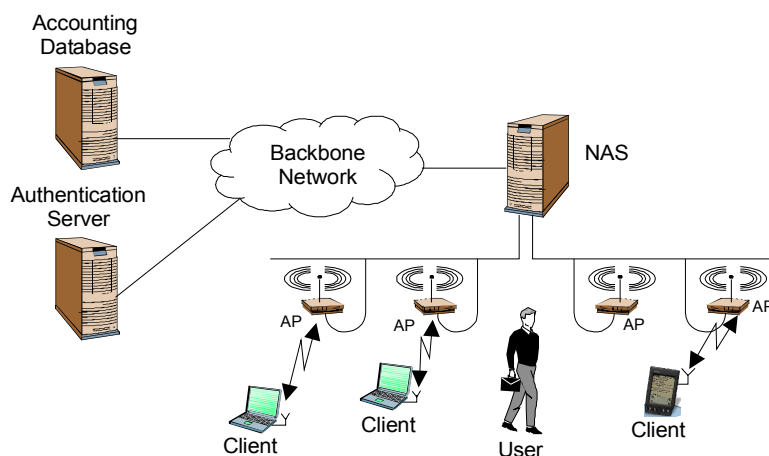


Figure 4.1 System overview

An overview of the different elements that constitute the system is provided in Figure 4.1. Five entities are shown: Client, Access Point (AP), Network Access Server (NAS), Accounting Database, and Authentication Server (AAA). Throughout the thesis, the system consists of the following entities: the Authentication Server, the Accounting Database, the NAS, and the different Access Points.

The system is quite independent of the access technology although it has been designed for used with IEEE 802.11b, which has become the common wireless access network protocol at public hotspots. More specifically, the wireless hardware should be IEEE 802.11b compliant and WECA certified [17] although this is not a requirement.

The system could be implemented with any IEEE 802.11b compliant hardware from vendors such as Agere Systems, Nokia, Ericsson, or 3Com. However, it might also be implemented with Hiperlan2 or Bluetooth Access Points. The system can be built from scratch or can be added to an existing wireless infrastructure at the desired location.

As specified by the requirements, the user will be able to use any IEEE 802.11b compliant network interface. This means that no vendor specific security solution can be used. Recent studies [18, 19, 20] have shown that link layer security provisions included in the IEEE 802.11 standard provide almost no security, therefore the system must rely on higher layer security solutions like IPSec, WTLS, or Kerberos [42].

The commercial GPRS networks could provide a large coverage area. In the autumn of 2001 all three Swedish operators had GPRS networks up and running.

Most of the functionality of the proposed system is placed in the NAS. It provides network access to users and it is the main element providing security, mobility, and accounting.

It should be noted that actually each AP provides 11 Mbps bandwidth on the wireless segment and that the wired network segment to which all those AP are connected has to be able to deal with the *aggregated* bandwidth coming from all these APs. It is a common practice in test

systems to connect the AP to 10 Mbps Ethernet segments, which would become a bottleneck in a large network. It has yet to be established what is the bandwidth needed, but at least a 100 Mbps Ethernet segment will be usually needed if there is more than one AP.

The following sections give a little more detail for each entity.

## **4.2 User**

The user is the person who pays for the wireless service, the one to whom the bill is addressed. They use a client to access the service. The type of user is defined by the payment method they have selected, so there exist four different types of users: prepaid user, test user, credit card user, and subscriber. Users paying via invoice are considered subscribers to the system.

## **4.3 Client**

The client is the device used by user in order to access the Internet or a corporate network. Two different types of clients are identified for use with public WLAN access networks: laptops and handheld devices such as Palms or PDAs. Nowadays laptops are the most common clients in public wireless access networks, but it is expected that this will change in a near future with the introduction of Compact Flash form factor WLAN cards, thus handheld devices will take the lead.

In the design of the system two different types of clients have been specified: basic and advanced. The basic user wants to have a simple, easy to use, secure system to be used in specific locations. The advanced user has stronger security requirements and wishes to stay connected all the time.

## **4.4 NAS**

The NAS plays the main role in access control, security, mobility, and accounting. It provides network access to clients who wish to access the system.

The NAS is modular and can be configured to have an Accounting Database and an Authentication Server integrated in it, thus providing all the necessary system functionality.

The NAS has the following components:

- Firewall
- Secure web server (supporting https)
- Radius client
- Access Control daemon/scripts
- DHCP server
- NTP server
- DNS server
- Mobile IP Foreign Agent (FA)

Other optional components are:

- Radius Server
- SQL database
- Kerberos KDC
- Mobile IP Home Agent



### **4.5 Authentication Server**

The authentication server might hold the access control information such as usernames and passwords, but no accounting information. In our system, the access control information is placed together with the accounting information in the accounting database to allow an easier management. But, this could be changed based on an operator's decision. It also has a main actor in roaming providing an interface to communicate with other domains which might have roaming agreements with its.

There are in fact two different authentication servers in the system. One is a RADIUS Server, which needs to be present in the system and the other one is a Kerberos KDC, which is optional. They could be integrated in the NAS in a small system or replicated for larger configuration scenarios.

### **4.6 Accounting Database**

The accounting database is a SQL database which holds the accounting information for the users and optionally it might hold the access control information such as usernames and passwords. It could be integrated in the NAS in a small system or replicated for larger configuration scenarios.

## 5 Analysis and Implementation

### 5.1 Login

To make it as simple as possible for the user, we want to avoid the user having to explicitly type in the address of the login page. To be able to do this we have added some rules to iptables. These rules will redirect every HTTP request that tries to go through the NAS to the outside networks (e.g. the Internet). The HTTP requests will be redirected to the local IP address and will be taken care of by the Apache web server in the NAS. The local web server will get the original HTTP request, where the original URL requested will be present, e.g. `www.google.com`. This URL will be logged in the file: `/var/log/httpd/access_log`.

Since the URL requested is not available at the local web server, a default page will be sent back to the user. This page will contain Javascript, the script will first see if the user has any cookie (see Section 5.7.2) previously set by the web server. If a cookie is present the script will call the `reauthentication-cgi` script described in Section 5.2.7. If no cookies could be found, the user will be redirected to the login web page, as shown in Figure 5.1.

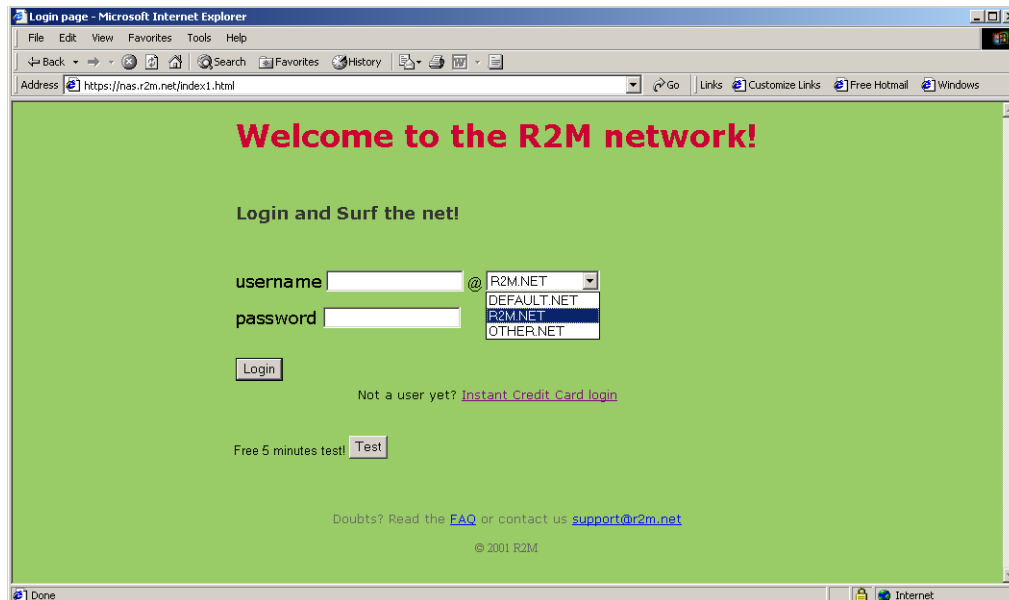


Figure 5.1 Login Page

The user will fill in the information requested in the login page, and then press the login button. How the authentication is done is described in Section 5.2. If the authentication is successful the user will once again be redirected, this time to the original requested URL. This URL is recovered from the apache log file where it was earlier logged. After the authentication is done a new set of rules have been added to the NAS's iptables, these rules will let all the traffic from the user through the NAS and there will be no further redirection as long as the user is logged in.

At the same time the user is redirected to the original requested webpage, an applet will start. The applet will show information about the time connected and the number of byte sent and received. The applet has also two buttons, a logout button for letting the user manually logout and an update button for updating the information shown by the applet. More information about the functionality and how the applet works can be read in Section 5.8.

## 5.1.1 Wireless access overview

### 5.1.1.1 Basic user

The following are the normal steps for a basic user (without Mobile IP, Kerberos, or IPSec support) to start surfing the Internet:

#### 1. Association

When the user is located in the area covered by the access point, the client associates with the corresponding IEEE 802.11b AP. Open System authentication is used by the AP, which means that any user can be associated without having to share a secret with the AP.

#### 2. Network configuration

The AP now bridges the data frames from the client to the NAS. A DHCP client is run to request network configuration information from the DHCP server located at the NAS. An IP address, network mask, gateway address, DNS server, and DNS domain name are provided to the client.

#### 3. Authentication required

The next step is for the user to start their browser in order to surf the Internet. At that moment the login page is automatically showed in the browser. The user is required to provide their credentials in order to log into the system. Possible credentials are username and password, credit card information, or prepaid information. A five minute test trial is also provided.

#### 4. Authentication process

Once the user provides his or her credentials in the login web page, they are checked with the authentication server and if found valid, authorization is granted to the client.

#### 5. Accounting starts

The authentication server also starts the accounting process for the authenticated user.

#### 6. Access control

The firewall located at the NAS stops all outgoing traffic until the user has been authenticated, granted authorization by the authentication server and accounting has started. Once the firewall is open the user can send or receive any IP traffic to the outside.

#### 7. Redirection

The user is redirected to the original page requested, usually the home page configured in the browser.

#### 8. Feedback

An applet is loaded from the NAS in order to provide feedback about the session to the user. A cookie is installed in the client to allow quicker re-login to the system.

### 5.1.1.2 Advanced user using Mobile IP with FA support

The access procedure is far simpler for a user who has a Mobile IP client installed.

#### 1. Association

Same process than for a basic user.

#### 2. Registration Request

The mobile node detects the agent advertisements by the FA located at the NAS and tries to register by sending a registration request to the HA.

### 3. Authentication

The FA forwards the authentication information from the registration request to the authentication server.

### 4. Accounting starts

If the authentication server authorizes the user, the registration process is finished and accounting is started for the user.

### 5. Access Control

The NAS opens the firewall in order to allow traffic from the mobile node to the outside.

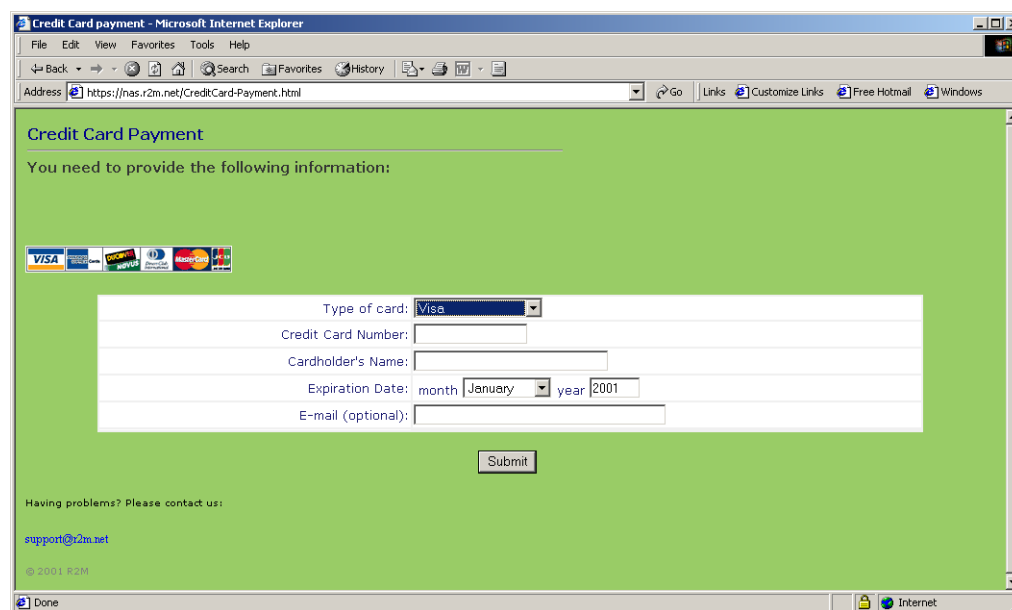
## 5.2 IP-login

The IP-login program has the same structure and shares some code with the program with the same name written by Enrico Pelleta, that is in use at the IT-University in Kista [109]. The author provided us with the code under GPL license. The IP-login program and its sets of Common Gateway Interface (CGI) scripts are all located in the NAS. The main tasks of the program are:

- ❑ Handle user login/logout/update requests
- ❑ Authenticate the user through RADIUS or Kerberos
- ❑ Open and close the firewall
- ❑ Keep track of who is logged in
- ❑ Automatically logout a user if the user leaves the network or if the users prepaid card has expired
- ❑ See that accounting data is sent to the RADIUS server

The authentication procedure as implemented, allows authentication against both a RADIUS server and a Kerberos KDC. The authentication server used depends on what realm was selected on the login page (see Figure 5.1). The mapping from the realm selected to the authentication scheme used is done in the login-cgi script.

New authentication methods can easily be added, e.g. credit card payment where the authentication scheme then depends on what authentication system used by the credit card company. Figure 5.2 shows what a credit card login page might look like.



The screenshot shows a Microsoft Internet Explorer browser window titled "Credit Card payment - Microsoft Internet Explorer". The address bar displays "https://nas.r2m.net/CreditCard-Payment.html". The main content area has a green background and contains the following elements:

- Header: "Credit Card Payment"
- Text: "You need to provide the following information:"
- Logos: A row of logos for VISA, MasterCard, American Express, Discover, and EuroCard.
- Form fields:
  - Type of card: A dropdown menu with "Visa" selected.
  - Credit Card Number: A text input field.
  - Cardholder's Name: A text input field.
  - Expiration Date: Two dropdown menus for "month" (set to "January") and "year" (set to "2001").
  - E-mail (optional): A text input field.
- Submit button: A button labeled "Submit".
- Footer: "Having problems? Please contact us: support@r2m.net" and "© 2001 R2M".

Figure 5.2 Login Page for credit card users

The IP-login program is divided into two main processes, an authentication daemon (authd) and a control daemon (ctrlld). In addition to these main daemons there are also some CGI scripts that send messages to the main daemons. In Figure 5.3 you can see a schematic illustration of how the different parts of the system interact. The RADIUS server could be replaced by a different authentication server such as a Kerberos KDC, but only for authentication purposes since it should still receive accounting information. The authentication scheme described here is RADIUS, although another authentication scheme should look similar.

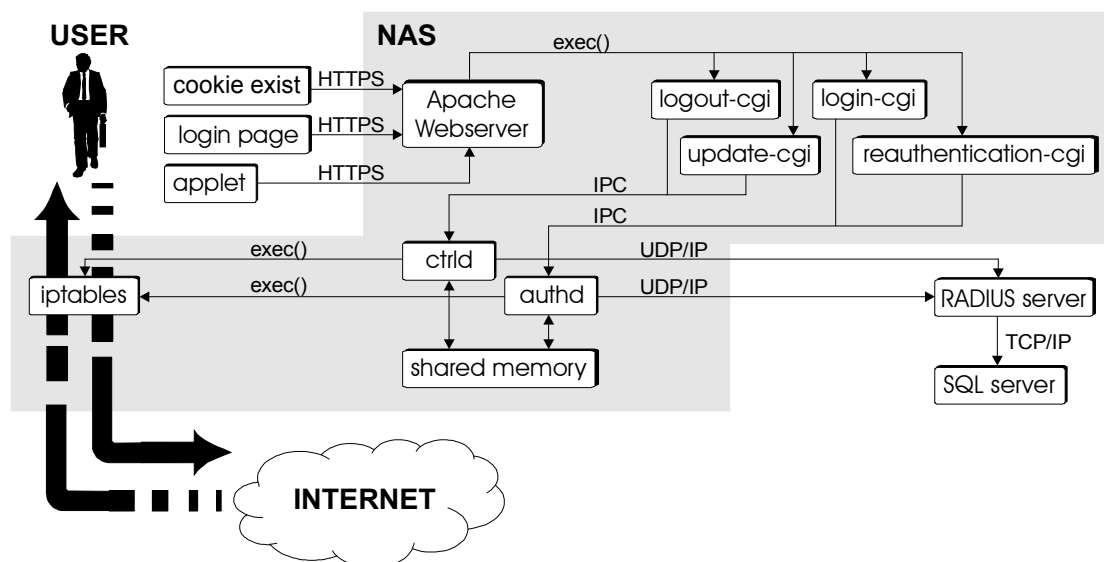


Figure 5.3 System process communication

### 5.2.1 Shared memory

The shared memory contains a table of which users that are currently logged in and information about each user. The memory is shared by the control daemon and the authentication daemon. This way both daemons can have access to the same current information about the users that are logged in. The information stored in the table is:

- Username
- IP address
- MAC address
- Session start time
- Flag that signals if the user is using Mobile IP or not
- Flag that signals what kind of account the user has (prepaid, normal, etc.)
- Session timeout if prepaid is used
- Unique session ID, generated by the RADIUS server
- What interface the user is connected to (e.g. eth0 or eth1)

### 5.2.2 The authentication daemon (authd)

The authentication daemon listens to incoming authentication requests. They can either come from one of the two CGI-scripts (login-cgi or reauthentication-cgi) launched by the web server, or from the Mobile IP Foreign Agent. The later case is more thoroughly explained in Section 5.7.1.

In both cases a message is sent to the authentication daemon. The message sent to the daemon contains IP-address of the user, username, password, and some flags that specifies what kind of user it is, e.g. Mobile IP user. When the daemon receives the message, a new process is

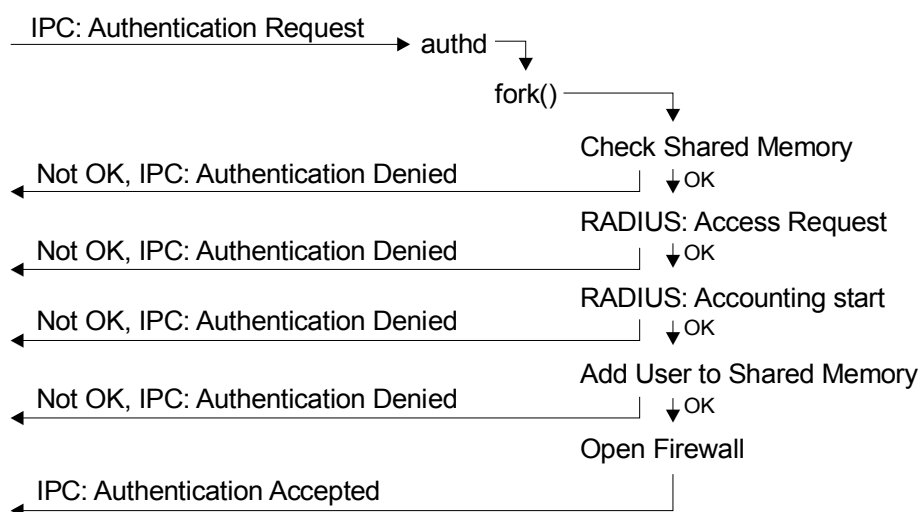
created (through fork) to deal with that request. This way the system can receive a second request while processing the first one.

When a message is received, the shared memory is first checked to see that the user is already logged in, if so the request is denied. If the user does not appear in the shared memory, an Authentication-Request is sent to the RADIUS server. If the reply is positive an Accounting-Request, with the Acct-Status-Type attribute set to START, will be sent to the RADIUS server (see Section 5.3.1 for more info). Otherwise an access denied would be sent back to the requesting process (e.g. login-cgi).

A unique session ID will be created in the RADIUS server when processing the accounting request. The unique session ID is a MD5 digest of a concatenated list of attributes in the accounting request, User-Name, NAS-IP-Address, Acct-Session-ID (used for timestamp) and Framed-IP-Address. This session id will be sent back to the daemon in the accounting reply. The session ID will be stored in the shared memory together with other information described in Section 5.2.1, the use of this session ID will be explained in the following sections.

After the accounting has started, the user is added to the table of current users in the shared memory. Next a number of rules are added to iptables to let the users traffic go trough the firewall.

Finally an accept message will be sent back to the process that sent the original request. Together with an accept message, the unique session ID (originally created in the RADIUS server) will be sent.



**Figure 5.4 authd functional diagram**

If anything should go wrong in any of the above steps (e.g. authentication rejected or out of resources) the request is aborted and an error message will be sent back. Every request, both accepted and denied, is also written to a log file.

### 5.2.3 The control daemon (ctrlid)

The tasks of the control daemon are:

- Detect when a user leaves the network, without manually logging out.
- Deal with incoming messages, i.e. logout requests and update requests.
- Stop the accounting and close the firewall when a user is logged out.

The control daemon contains two threads. The first thread will listen to incoming messages in the same way the authentication process does. The second thread will go into an infinite while loop and check if the users logged in are still attached to the network. This is done by the use of ARP requests, other "keep-alive" methods are described in Section 5.2.10. Two types of messages can be received by the daemon, a logout request and an update request. Both are originally sent by the applet that provides the user feedback module (see Figure 5.3)

One of the main tasks for the daemon was to stop the accounting and close the firewall when the user is logged out. There are three ways a user can be logged out:

1. User leaves the network without manually logging out.

If the user does not reply to three consecutive ARP messages, the user is considered to have left the network and should be logged out.

2. Manually log out.

The user can manually request to be logged out by pressing the logout button in the applet. The applet will call the `logout-cgi` script through the web server. The `logout-cgi` script, described later, will send a logout request to the control daemon.

In the message received by the control process, the unique session ID will be present together with user name and IP address. This information will be compared with the information that is stored in the shared memory table. By comparing this information one can avoid spoofed logout requests, even though SSL is used for the connection between the applet and the NAS this gives an extra level of security. If the compared information is the same, the user will be logged out. A reply is sent back to the `logout-cgi` script with information about the amount of data sent and received.

3. The user uses a prepaid account, and the time is up.

If the user has a prepaid account, then the session timeout is checked at the same time the control process sends ARP request to the user. Depending on how often the ARP request are sent, the session time out could be exceeded by maximum the time between the ARPs.

In all of the above three cases the same logout procedure will occur. Information about the amount of data sent and received by the user will be obtained from iptables. After this the rules will be removed from iptables, thus the firewall is closed for that particular user. An Accounting-request with the `Acct-Status-Type` attribute set to `STOP` is sent together with some more information (see Section 5.4.1 for more info). Finally, the user is removed from the shared memory.

If the control process receives an update request, the unique session ID will be compared in the same way as for a logout request. If it matches, then information about the amount currently sent and received is obtained from iptables, and is sent back to the `update-cgi` script (described in Section 5.2.6).

## 5.2.4 The login-cgi script

When a user first accesses the network and tries to access a web page, the user will then be redirected to a login page (Figure 5.1). On the login page the user is requested to fill in their user name and password, when the user then presses the login button, the form will call the `login-cgi` script. The `login-cgi` script will retrieve whatever information the user entered, then it will parse the information and send them together with the IP address the user currently is using to the authentication process described above. The realm one can see in Figure 5.1 is put together with the username, so that the username is of the form `user@realm`.

If the reply from the authentication process is negative, the user will be redirected to an error page, which will state possible causes for this error.

If the reply says “accept”, the login-cgi script will parse the Apache access log file to retrieve the URL for the original web page the user requested, before he was redirected to the login page. The match is based on the user’s IP address. Before the last redirection to the original page, a cookie will be placed in the users web browser and the java applet will be started. When starting the applet a unique session ID will be passed as an argument, the very same session ID that was created in the RADIUS server (see Section 5.2.2).

### **5.2.5 The logout-cgi script**

In the Java applet described in Section 5.8 there is a logout button the user can press, if the user wants to manually logout. When the button is pressed it will call the logout-cgi script; as an argument the applet will pass the unique session ID and the user name.

The logout-cgi script will send a message to the control process, telling it to logout the user with corresponding IP address, user name, and unique session ID.

In the reply from the control process, there will be information about the amount of data sent and received during the session. This information will be sent to the applet, where it will be presented to the user.

### **5.2.6 The update-cgi script**

Next to the logout button in the java applet (described in 5.8) there is an update button. Upon pressing this button the update-cgi script is called. The update-cgi script works a similar fashion to the logout-cgi script. It sends a message to the control process, requesting an update. The user name, IP address and unique session ID will be sent in response in just the same way as the logout-cgi script does.

In the reply from the control process, there will be information about the amount of data sent and received during the session; this information will be passed on to the applet.

### **5.2.7 The reauthentication-cgi script**

As described in Section 5.7.2 a cookie is used to let the user automatically be logged in if they were logged in earlier. If there is a cookie present the reauthentication-cgi script is called, with the cookie data as an argument if they were logged in earlier. The cookie is encrypted so the first thing the reauthentication-cgi script does is decrypt the cookie. If the expiration time that is present in the decrypted cookie is not yet passed, then the reauthentication-cgi script will try to login the user in the same way the login-cgi script does.

If some reason the user for is not accepted or the cookie is not valid any longer, the user will be redirected to the normal login page.

If the user is accepted, the applet will be started and the user will be redirected to the original requested URL. No new cookie will be put in the user’s browser, the old one will still remain. The reason for this is the security aspects discussed in Section 5.5. If someone should steal a user’s valid cookie he could in theory use it forever if the cookie’s expiration time was renewed every time the reauthentication-cgi script is called. However, with our method the cookie is only valid until the originally specified expiration time.



## 5.2.8 Web Server

The web server is used as an interface towards the user. More about the web server used can be found in appendix B.

## 5.2.9 Iptables

Iptables in Linux kernel 2.4.x are the successor to the ipchains in kernel 2.2.x. Iptables filter the network traffic according to the rules specified. Rules can be inserted and removed dynamically.

In this thesis the rules are added and removed by executing system commands such as:

```
iptables -t nat -I PREROUTING -s 192.168.10.5 -m mac --
mac-source 00:50:04:46:D4:57 -j ACCEPT
```

By default all HTTP requests to outside networks are redirected to the NAS but once the user has been authenticated, the previous rule prevents the following requests to be redirected. A few other rules are added to let the traffic go through. Each rule will keep track of how many packets and the number of bytes the rule have been applied to. It is this information that is obtained by the control daemon to report the number of bytes sent and received.

Instead of using system calls to communicate with iptables, there is a library called libiptc [144] that could be used. This library was discovered too late to be used in this thesis.

## 5.2.10 Keep-Alive

Once the user has logged into the system, accounting of the resources consumed by that user has to be started. The user wants to be charged only for the exact amount of resources consumed, nothing else. The system needs to be designed to be able to finalize the accounting of the consumed resources whenever the user stops using them.

In order to do that, the user is presented with a Logout button in the feedback module, provided by the applet. This is the best way for the user to control when he wants to stop using the system. Whenever he pushes the Logout button the user is assured by the system that his session has been closed and that no further resources are available to him nor will any further usage be charged to him.

But, since the feedback module is based on an applet and the applet is executed using a Java Virtual Machine in the browser, it might happen that the user closes this browser without having previously logged out, therefore killing the applet and disabling the manual logout method. The result of this action is that the user loses the feedback module.

This action has different implications depending on the intention of the user. If the user's intention was not to stop using the network resources, but just to close the browser (because it was no longer used), then the accounting for the resources should not stop. This might happen for example when downloading a file from a remote network. The user would probably want to finish that download even though the browser was closed. Another possible reason for this is that the browser crashes and terminates. In this case the user would like to be able to send and receive traffic even though the browser had been closed.

On the other hand if the user's intention was really to logout and he just forgot to press the logout button, then the user will not want to be charged for further resources. This will not be a problem if the user is charged based on data transfer since no more traffic will be sent from that specific user. But if the user is charged by the time connected to the network, and forgot

to logout, then charging is still being done for resources they are not really consuming. Besides, the user could be charged for unwanted data sent to them.

From the system point of view there is a security risk related to the above actions. The user might leave the hotspot without logging out and thus their session remains open and so does the firewall. That means that an attacker lurking in the network could try to steal that session. The attacker will have to do both IP spoofing and MAC address spoofing in order to fake both addresses of the real user and be able to send and receive traffic through the network.

Faking an IP address is quite simple in most operating systems. Faking a MAC address is more complicated and involves the flashing of a new MAC to the card address or an ioctl call to the WLAN driver, but it is quite feasible for an experienced attacker.

The system should be able to detect that the user left the hotspot and closed the session, in order to solve both problems. A decision had to be made between several different design and implementation alternatives.

1. The traffic that a user is sending through the NAS can be checked. Whenever the user has not sent traffic for a fixed amount of time (for example 3 minutes), then you can consider that the user left the hotspot and therefore that the session for that user needs to be closed. This approach is used in the NetLogon system at the Linköpings University [110] and also in some commercial systems.

2. The system could continuously probe for the user and wait for the replies to check if the user is still available. A typical implementation for this idea probes the user at a fixed interval of time and waits for a reply. If there is no reply, then the user can be logged out. Of course, since the radio link can have areas of bad coverage, you cannot rely on a single probe to log out the user. The usual implementation is to log out the user if there is no response for 3 consecutive probes. There are two easy ways to probe the user, using ARP packets or using ICMP echo request (“ping”) packets. ARP packets are limited to a subnet, are smaller, and faster being therefore usually preferred. This approach was used in the IT-University system at KTH during year 2000.

3. Another possibility is to link the DHCP service present in the NAS with the Keep-alive. The DHCP server can lease addresses for a limited time to the mobile node. Whenever the mobile node fails to renew this lease, it means that he wishes to stop using the system, since no communication is possible without an IP address. When the lease expires, the DHCP server should communicate with the Access Control module in order to block any further communication from the MAC/IP address pair.

There exist two possible ways to implement this third approach. The first requires a modification to the code of the DHCP server. Whenever the DHCP fails to renew a lease, it communicates this to the corresponding logout module. Modifying the DHCP server is usually not a neat approach, especially when you are using commercial software. Whenever an upgrade of the server needs to be done, either to add new functionality, to solve a newly found bug or to close a detected security hole, the changes that had been made to the server have to be ported to the new version. It might happen that the new version is quite different from the used one and that those changes are no longer valid.

The other approach is based on the *dhcpd.leases* file. This file keeps a real-time account of the leased addresses. Therefore you could have a daemon that checks the file every now and then and identifies any changes in the leases. If the expiration time of a lease is past, it means that it has not been renewed and thus the user is no longer using the system. This approach might also help in the event of a user roaming between two subnets that are connected to the same NAS. The problem that we detected in this approach was that in order to keep tight control,

you had to be parsing the file for changes every few seconds. Parsing a file can be a slow process. In fact using a Perl script that extracts the information of the leases from the file and prints the status requires approximately 1 second depending on the number of leases in the file. Using C code in the control daemon for this purpose could be faster, but might still be too slow when the number of users connected is high.

4. In this method we use a security association between the client and the server. The feedback module interacts with the keep-alive module. When the applet is started a shared secret is passed to it from the login-cgi script. The server can ask from time to time for that shared secret in order to make sure the client is still in the hotspot. This could be done signing a challenge to avoid transmitting the shared secret unnecessarily over the network. The applet should start the communication at loading time or a signed applet can be used with special privileges.

The last alternative is probably the best one. The system could perform an automated check every minute for example to see if the security association was still valid and at the same time update the information about the data sent and received by the user.

### **5.3 RADIUS authentication & Accounting**

The RADIUS server will get two types of RADIUS packets, Access-Request and Accounting-Request. They will be treated slightly different, but both will generate SQL queries. Below we describe in detail how the requests are handed.

#### **5.3.1 RADIUS authentication**

To be authenticated through RADIUS an Access-Request is sent. PAP is used as the authentication method. The reason for using PAP is that with CHAP the passwords have to be stored in clear text in the database (see Section 5.5.14). The main disadvantage is that the encryption used by PAP is not very strong, but since an IPSec link is established between the NAS and the Authentication Server, this should pose no problem. The Access-Request in our case contains the following attributes:

- ❑ User-Name
- ❑ Password
- ❑ Acct-Multi-Session-Id

CHAP is used instead of PAP when the end user uses Mobile IP as described in Section 5.7.1. The Password attribute will then be replaced by a CHAP-Password and CHAP-Challenge attributes.

If a RADIUS proxy is used, then the first thing that is done is to see if the packet should be sent on to another RADIUS server. This is done by looking in the proxy configuration file for the realm in the User-Name attribute (user@realm).

In the Acct-Multi-Session-Id attribute, information about the user's MAC address is stored. This is not what this attribute was intended to be used for, but since no other suitable attributes were to be found this one was used. Using the Acct-Multi-Session-Id attribute would present a problem when communicating with a RADIUS server that uses this attribute for its original purpose. A vendor-specific attribute is considered a better implementation solution and it is left as future work.

The MAC address is linked to the User-Name in the database the first time the account is used. This information is used to authenticate the user together with the User-Name and

Password attributes. This way not only the correct User-Name and Password are needed to authenticate a user, but also the correct MAC address is necessary. This prevents IP spoofing (faking of IP address) and even if the MAC address can be spoofed, this gives still some extra security. When a user wishes to change WLAN adapter it needs to notify the system administrator.

When an Access-Request is received, the RADIUS server sends a query to the radcheck (see Section 5.4.2) table in the SQL database. The query tries to find the entry that corresponds to the username and the MAC address, if no MAC address is present then the MAC address is set. If there were no entry that corresponds to this username, then authentication fails and an Access-Reject is sent back by the server. If an entry that corresponds to the username and MAC address exists, that entry is sent to the RADIUS server.

The RADIUS server compares the password from the Access-Request packet with the password from the SQL database, if they do not match, then an Access-Reject is sent. If they match another SQL query is made, this time to the radreply (see Section 5.4.2) table. Different replies for different users can be sent back to the NAS in this way, the replies used in this system include the attribute Session-Timeout. The Session-Timeout attribute specifies the maximum length of the session; this is used for the prepaid accounts. If there are any attributes for this user they are sent back to the NAS in an Access-Accept packet.

### 5.3.1.1 Duplicate logins

Depending upon what kind of payment method is used, duplicate logins at the same time can be a problem. By duplicate login we mean that a user's account is used by several clients at the same time, this means that several people could share the same account or that the user may have multiple clients. If the price model were flat rate, this would decrease the revenue. In a price model based on the time connected or the amount of data sent and received, this would not affect the income in the same way. In this thesis we have decided that duplicated login should not be allowed.

The function that connects the user name to the MAC address of the adapter is one way of preventing duplicate logins. But the MAC address can be spoofed or the operator may not want to connect usernames to a MAC address. To solve this extra functionality has to be added. In the NAS there is a table of all currently logged in users, which can be used to prevent duplicate logins, but this only applies for that particular NAS. The solution has to be centralized to be efficient.

In this thesis we have solved this problem by changing the authentication parameters for the user in the authentication table in the database (radcheck) when the user logs in. The SQL query is "change\_login" described in Appendix E. By doing this, the account appears to be invisible to the RADIUS server, and no other user can use that account. When the user logs out, an accounting stop request is always sent. At the same time the accounting is stopped, the SQL query "change\_logout" is issued (Appendix E). This makes the account visible again.

With this solution, problems could occur e.g. if the NAS reboots, and no accounting stop message is sent to the RADIUS server, then the account will remain invisible, until someone manually changes the account to make it visible. To solve this problem some kind of control mechanism should check with the NAS, to see if the user is still logged in. This has not yet been implemented, but Freeradius has some support for Simple Network Management Protocol (SNMP) and this could be used.

## 5.4 Accounting

By accounting we mean the collection of resource consumption data for the purposes of capacity and trend analysis, cost allocation, auditing, and billing. In this thesis, accounting is one of the main concerns. There are many issues that have to be taken into consideration, when using a modern accounting system. Some of them will be described below.

### Fault resilience

Depending on what the accounting data is used for, the importance of fault resilient is different. Faults that may be encountered include:

- ❑ Packet loss
- ❑ Accounting server failures
- ❑ Network failures
- ❑ NAS reboots

How to deal with packet loss is closely connected with the choice of transport protocol. UDP based transport is frequently used in accounting applications, e.g. RADIUS. To limit the consequences of packet loss when using UDP, a retransmission mechanism has to be implemented on top of the transport layer. TCP guarantees reliable transport, but in case of failure the TCP timeout might be too long. A more aggressive retransmission mechanisms on top of the transport layer can be implemented, this should make it faster to change to a backup server. Things to keep in mind when implementing a retransmitting mechanism for UDP are, retransmitting behaviour, congestion control, and timeout behaviour.

By providing a secondary accounting server much of the risk of accounting server failures can be avoided. Network failures may result in partial or complete loss of connection to the accounting server. In the case of partial connectivity loss, it may be possible to reach the secondary accounting server. In a complete loss of connectivity it may be necessary to store the accounting events in the NAS's memory until connectivity can be re-established.

In the case where the NAS reboots, it's desirable to minimize the loss of data concerning ongoing sessions. This is especially significant in long-lived sessions. To reduce the loss of data interim accounting data can be transmitted, or the NAS can store the data in a non-volatile memory and send the data after booting back up.

The importance of fault resilience depends on what the accounting data is used for, e.g. for capacity analysis packet loss may not be that important, but when it is used for billing, it directly translates to revenue loss.

### Resource consumptions

When scaling the accounting system, it's important to understand the limitations caused by resource consumption, including:

- ❑ Network bandwidth
- ❑ Memory
- ❑ CPU usage

In large systems bandwidth can be a limiting factor. Compression could be used in the accounting protocol to reduce the bandwidth. If the accounting protocol doesn't support this, compression could be done at the IP-layer or the transport layer. To increase the efficiency of the transportation of the accounting data, data from several sessions can be put in the same packet or accounting data could be transferred as a single file.

Data should be written to non-volatile storage as soon as possible, storing data in memory for long periods of time is not a good idea, as a system reboot would lose all data. Excessive use of memory should be avoided; buffers could get full and could result in loss of data.

CPU consumption by the accounting server is proportional to the complexity of the required account processing. Tasks like encryption/decryption and compression/decompression can consume considerable resources. Compression reduces the use of bandwidth and encryption increases the security, but both could overload the CPU and that could affect the overall reliability of the system.

### Data collection models

There are different models of how the data should be collected and transmitted to the accounting server, including:

- Polling model
- Event-driven model

In the polling-model, an accounting manager will poll devices for accounting information at regular intervals. When polling data from the devices, data is typically transferred in a single file, resulting in an efficient transfer process. The polling model scales poorly for the use of roaming services, this is because the accounting manager would have to periodically poll all devices in all domains, most of which would not contain any relevant data. Considerable amounts of bandwidth will be consumed when the accounting manager tries to pull data from devices when there is no data to pull.

In event driven-model, a device will contact the accounting server whenever it is ready to transfer accounting data. Most event-driven accounting systems, e.g. RADIUS, only transfer one accounting event in each packet, which is inefficient. Since an event-driven model sends data to the recipient domain without requiring them to poll a large number of devices, it is suitable for roaming where the client moves between different domains. Accounting data is typically not transferred as a single file and this results in low processing delay but high bandwidth consumption and high overhead.

### Conclusion

The requirements we had when choosing an accounting protocol were:

- Easy to implement support for roaming
- Scalable
- Easy to use and well tested
- Freely available open source implementation

RADIUS is the dominant accounting protocol used by ISPs today. It's fairly easy to understand and use, and there are both commercial and open source implementations available. RADIUS also has support for authentication and authorization, and this makes it an overall good choice. Additional advantages are integration possibilities with Mobile IP and that its successor DIAMETER is backward compatible with RADIUS.

## **5.4.1 RADIUS Accounting**

When the NAS has successfully authenticated a user, either by Kerberos, RADIUS, or in some other way, the NAS will start the Accounting associated with this user. The NAS starts the accounting by sending an Accounting-Request with the Acct-Status-Type set to START.

In the same way that an Access-Request is used in RADIUS for authentication, the Accounting-Request will be proxied to the correct home RADIUS server for that user, although the accounting information will also be stored in the visited network RADIUS server.

An extra feature in the Freeradius implementation is the possibility to create a unique session ID, although this is not defined in the RADIUS protocol [25,26,27]. The unique session ID is a MD5 [106] digest value of the following concatenated attribute values, NAS-IP-Address, Session-ID, and User-Name.

The attributes used in the Accounting-Request packet when starting the accounting are:

- |  |  |
|--|--|
| <input type="checkbox"/> User-Name:        | <b>The username.</b>   |
| <input type="checkbox"/> Acct-Status-Type  | The type of the Accounting-Request packet, 1=START   |
| <input type="checkbox"/> Framed-IP-Address | The IP-Address used by the user.   |
| <input type="checkbox"/> NAS-IP-Address    | The IP-Address of the NAS the user is connected to.  |
| <input type="checkbox"/> Acct-Session-ID   | This attribute is used to send information on when the session started, the value is the number of seconds since 1 Jan. 1970.                          |
| <input type="checkbox"/> Callback-Number   | This attribute is used if the user use credit card, the credit card number is then stored here. If credit card is not used the value is then set to 0. |

When the Accounting-Request reach the RADIUS server, the server will then send an SQL query (described in appendix E). This will creates an entry in the accounting database. Before sending an acknowledgement back, another SQL query is sent to get the unique session ID created by the RADIUS server. The RADIUS server acknowledges the Accounting-Request by sending an Accounting-Reply. The only attribute present is Acct-Session-ID.

Accounting is stopped by sending an Accounting-Request with the Acct-Status-Type set to STOP. All the attributes that will be sent when stopping the accounting are listed below:

- |   |  |
|---|--|
| <input type="checkbox"/> User-Name            | <b>The username</b>  |
| <input type="checkbox"/> Acct-Status-Type     | The type of the Accounting-Request packet, 2=STOP.   |
| <input type="checkbox"/> Acct-Session-ID      | The unique session id created by the RADIUS server when the accounting started. This makes sure that the correct session in ended.   |
| <input type="checkbox"/> Acct-Session-Time    | The session time length  |
| <input type="checkbox"/> Acct-Input-Octets    | Number of bytes the user has received during the session   |
| <input type="checkbox"/> Acct-Output-Octets   | Number of bytes the user has sent during the session   |
| <input type="checkbox"/> Session-Timeout      | If the account is time limited, the remaining time after the session is sent to the server, to be used next time the user sign in  |
| <input type="checkbox"/> Acct-Terminate-Cause | What cause the session to end, three types are used all defined by RADIUS:<br>Acct-Terminate-Cause = 1 (User Request)<br>Acct-Terminate-Cause = 2 (Lost Carrier)<br>Acct-Terminate-Cause = 5 (Session Timeout) |

The number of bytes sent and received by the user is taken from iptables. When the session started and the Session-Timeout (if present) is stored in the shared table described in 5.2.1, with that information it's trivial to calculate the Acct-Session-Time and the new Session-Timeout.

When the RADIUS server receives the Accounting-Request, it launches an SQL query that saves the information associated with the session. To acknowledge the Accounting-Request an Accounting-Response is sent back to the client.

### 5.4.2 SQL database

The SQL database is a standard installation of mySQL version 11.13 that ships as standard with RedHat 7.1. No modification has been done to the database software. The tables that are used are based on those that come with Freeradius [31]. Some modification has been done to allow us to store what we need. The tables used are:

- radcheck: This is the table where all the usernames and their respective hashed passwords are stored.
- radreply: In this table possible replies to Access-Request messages are stored, these are used to send Session-Timeout messages to the prepaid customers.
- radacct: This is where all the accounting data is stored with one entry for each session a user initiates.

This is the content of the radacct table, some new attributes have been added and some of the attributes are not used.

Field	Type	Key	Default	Extra
Id	bigint(21)	PRI	NULL	auto_increment
AcctSessionId	varchar(32)	MUL		Used
Value	varchar(32)	MUL	NULL	Added
UserName	varchar(32)	MUL		Used
Realm	varchar(30)			Used
NASIPAddress	varchar(15)	MUL		Used
NASPortId	int(12)		NULL	Not used
NASPortType	varchar(32)		NULL	Not used
AcctStartTime	datetime	MUL	0000-00-00 00:00:00	Used
AcctStopTime	datetime	MUL	0000-00-00 00:00:00	Used
AcctSessionTime	int(12)		NULL	Used
AcctAuthentic	varchar(32)		NULL	Not Used
ConnectInfo_start	varchar(32)		NULL	Not used
ConnectInfo_stop	varchar(32)		NULL	Not used
AcctInputOctets	int(12)		NULL	Used
AcctOutputOctets	int(12)		NULL	Used
CalledStationId	varchar(10)			Not used
CallingStationId	varchar(10)			Not used
AcctTerminateCause	varchar(32)			Used
ServiceType	varchar(32)		NULL	Not used
FramedProtocol	varchar(32)		NULL	Not used
FramedIPAddress	varchar(15)	MUL		Used
AcctStartDelay	int(12)		NULL	Not used
AcctStopDelay	int(12)		NULL	Not used
ConnectInfo	varchar(20)		NULL	Not used
Attribute	varchar(32)		NULL	Added
cardnumber	varchar(32)		NULL	Added

The following table shows the contents of the radreply table; no modifications have been made.

Field	Type	Key	Default	Extra
id	int(10)	PRI	NULL	auto_increment
UserName	varchar(30)	MUL		Used
Attribute	varchar(30)		NULL	Used
Value	varchar(40)		NULL	Used



The following table presents the contents of the radcheck table. The only attribute added is the mac address.

Field	Type	Key	Default	Extra
id	int(10)	PRI	NULL	auto_increment
UserName	varchar(30)	MUL		Used
Attribute	varchar(30)		NULL	Used
Value	varchar(40)		NULL	Used
macaddress	varchar(30)		NULL	Added

## 5.5 Security

A good security policy should be established in several layers in order to make it more difficult for an attacker to launch an attack or break into the system. The ISO model describes 7 layers in the networking model and security provisions can be made in most of them to increase the protection of the system.

Layer 7	Application	PGP
Layer 6	Presentation	
Layer 5	Session	SSL/TLS
Layer 4	Transport	
Layer 3	Network	IPSec
Layer 2	Link	802.1x
Layer 1	Physical	WEP

**Figure 5.5 Protocol Stack**

Security in access networks should be transparent to applications. Security can therefore be placed in all layers up to the Session Layer (ISO layer 5) since the Presentation Layer (ISO layer 6) is rarely used. No modifications to applications are needed when the security is provided in the link or network layers (ISO layers 2 & 3).

Security in the link layer (ISO layer 2) is provided in a hop-by-hop basis while security in upper layers is provided in an end-to-end basis. An example of how to provide several layers of security is provided in Figure 5.5.

Different WLAN usage scenarios have quite different security requirements. In this thesis security was analysed for public WLAN scenarios, such as a WISP providing public access or a community access network, while private WLAN scenarios, including corporations and domestic use, have not been studied. The security discussion has centred around the measures needed in the design of the system. For a more general public WLAN security description a reader may refer to [7].

Public access networks usually have different security requirements based on the services they are providing. In the system described in this thesis where two different access services in public hotspots are provided, the security requirements are tighter for the corporate access case than for the Internet access case. If local services were also provided, their security requirements should be analysed.

Different security considerations also apply to different client terminals. Memory, CPU power, and battery life limitations allow more limited security possibilities for handheld devices than for stationary desktops or laptops. The suitability of existing techniques for secure communication in handheld devices was studied in [56].

Because of limited bandwidth on wireless links and the fact that payment might be made based on data sent and received, the proposed solutions should cause little additional overhead.

The balance between system security and user friendliness is always difficult to achieve when designing a system. In the system presented, access to a corporate network gives security preference over easiness of use, but the simpler case of Internet access is designed to more carefully balance both requirements.

### 5.5.1 Assumptions about the system

For the design of the security of the system the following assumptions were made:

- ❑ User's credentials are located in a centralized database (if only one administrative domain is present; no inter domain roaming allowed)
- ❑ Some scalable distributed system is needed to store user's credentials (if several administrative domains are present; inter domain roaming is allowed)
- ❑ Both authorized and non-authorized users have physical access to the network
- ❑ Only authorized users are granted access to network resources
- ❑ Physical protection of the Access Points is needed

Other assumptions that influenced the design of the security were:

- ❑ The system should be user-friendly
- ❑ Minimum requirements on the client side
- ❑ Easy to administer by a System Administrator
- ❑ Scalability: security should pose no limit on the scalability of the whole system
- ❑ Flexibility: different security levels for different users, and different authentication methods

### 5.5.2 Risk Analysis

The severity of a threat depends on the amount of resources than an attacker needs in order to be successful. Time, computational power, and money are the most common resources.

The threat analysis is based on the one provided by [7] for wireless public networks and focused on the system described in this thesis.

- ❑ Basic user error: high risk  
Basic users are not expected to have any security background and thus no confidence should be placed in their behaviour.
- ❑ Advanced user error: moderate risk  
An advanced user, who requires Mobile IP, IPSec, or Kerberos, might have some notions of security and therefore is less inclined than a basic user to make a security mistake.
- ❑ Personal privacy: high risk  
User's profiles, credentials, and location information should be kept confidential.
- ❑ Industrial espionage: low risk  
Only advanced users are expected to transmit data over the network interesting enough for an attacker to try to steal it.
- ❑ Sabotage: low risk  
It is unlikely than a competitor, an unhappy customer, or former employee will try to sabotage the system.

- ❑ Abuse of resources: high risk  
In an IEEE 802.11 segment users compete for resources and therefore some users might be tempted to keep more resources than allowed.
- ❑ Hackers: high risk  
Getting free access to the Internet is a juicy target for a hacker.

Threat / Risk	High	Moderate	Low
Basic user error	X		
Advanced user error		X	
Privacy	X		
Industrial espionage			X
Sabotage			X
Abuse of resources	X		
Hackers	X		

Table 5.1 Summary of the Risk Analysis

### 5.5.3 Security services

The following security services should be provided in the system:

- ❑ Denial of Service (DoS) prevention
- ❑ Auditing and logging
- ❑ Non-repudiation (of source and destination)
- ❑ Integrity (of data)
- ❑ Confidentiality (of data)
- ❑ Access Control
- ❑ Accounting
- ❑ Authorization (of the user)
- ❑ Authentication (both user and network)

The importance of the security requirements is shown in the following figure:

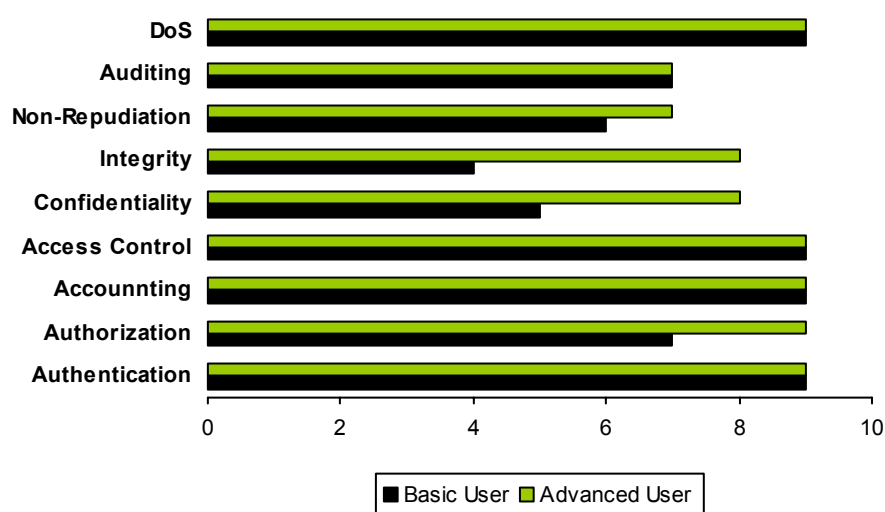


Figure 5.6 Security Services Importance

Traffic flow analysis prevention is considered out of the scope of this thesis.

### 5.5.4 WLAN security

WLAN security refers to the security provisions included in the diverse WLAN access standards, mainly the various IEEE 802.11 standards and the Hiperlan2 standards.

The different IEEE 802.11 standards only specify the physical layer and the MAC layer, a sublayer of the link layer. Though IEEE 802.2 (LLC) is also used in the Link layer, no security provisions are made there.

Thus WLAN security mainly deals with the security provisions in the link and physical layers of the ISO model. These provisions are implemented both in the APs and in the Client Adapters and therefore they only provide security over the wireless segment and cannot provide any end-to-end security.

The IEEE 802.11 standard provides some security provisions but as will be shown, they provide very little security at all. The standard defines two authentication methods (Open System and Shared Key) and a privacy method (WEP). It mandates the use of authentication for infrastructure BSS, but allows optional use for IBSS (Ad-hoc networks). WEP is optional both for infrastructure BSS and IBSS.

Another security service implemented in the standard is Access Control because the authentication process is always done before allowing the mobile station to perform an association with the AP (Figure 2.6). An unauthenticated station is not allowed to send or receive any traffic. Mutual authentication is not provided. The AP authenticates the mobile station, but the mobile station is not able to authenticate the AP.

Open System authentication is really a null authentication method since any wireless node can associate with the AP without providing valid credentials.

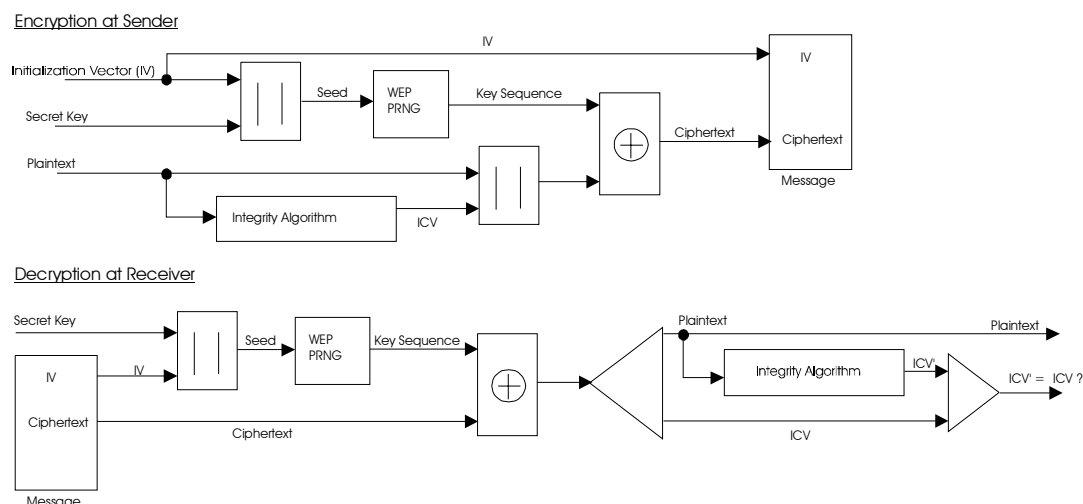
Shared Key authentication is based on a secret key shared between the clients and the AP. All clients allowed connecting to the network share the same key with the AP, thus this key is considered a global key. The secret key is never transmitted in the clear; however using Shared Key authentication requires the use of the WEP privacy mechanism.

The Shared Key authentication works like the following: an AP sends a challenge packet to the client, which has to encrypt it with the correct key and return it to the AP. If the response is right, the user is associated with the AP.

The standard also provides a privacy method called Wired Equivalent Privacy (WEP), which allows for confidentiality and integrity of both the data sent and received by the user and also the management frames of subtype Authentication. The algorithm was not designed for high security, but rather to be at least as secure as a wired LAN. It uses the RC4 Pseudo Random Number Generation (PRNG) algorithm from RSA inc, which is not publicly available but has been reverse engineered.

There exists a shared key between all users and the AP which is actually the same key used for Shared Key authentication. The key has a length of 64 bits where 24 bits are an Initialization Vector (IV) and the other 40 bits are a shared secret. Key management, for example, how to get the key in place prior to any communication, is not addressed by the standard.

Integrity is provided by WEP with an Integrity Check Vector (ICV). A frame contains the cipher text data block (encrypted data) together with the ICV and the clear text IV. The IV has to be different in every frame sent, but it can be reused after a limited time. This is one of the major weaknesses of WEP.



**Figure 5.7 WEP Mechanism**

### 5.5.4.1 WEP Security

Standard WEP does not provide per packet authentication. It is a well known fact that encryption without data authentication does not provide good security. Very simple attacks, such as modifications of DNS replies to redirect connections to rogue servers or replay attacks, exist against WEP due to this fact. A keyed Message Authentication Code should be added to WEP to make it stronger.

The problems related to WEP come from the way WEP uses cryptography, not from the key size. Besides, the 802.11 standard left key management and authentication mechanism as open problems, making the whole design even worse.

The first published vulnerability of WEP [114] was the implementation of the Initialisation Vector. WEP appends the 24-bit IV to the shared key to form  $2^{24}$  keys. In practice keys are not replaced as often as needed and an 11 Mbps AP with a typical load can exhaust the derived key space in about an hour hence creating IV collisions. Once two encrypted packets using the same IV are discovered, various attacks can be launched to recover the plaintext, one of these was described in the referenced paper.

In [18] new security flaws in the design of WEP were pointed out and several more attacks were described. An attack on the Shared Key authentication mechanism is provided in [115]. Combining these three papers all security mechanisms used in 802.11 had been compromised. The final blow to the WEP protocol was given by [19] where a major flaw was found in the design of the WEP protocol. From that moment several cryptanalysts teams have implemented attacks using that flaw [20]. Some of the attackers have even published their code on the Internet [71,72]. There even exists a tool to try to brute force into a WLAN system [73].

This renders WEP ineffective for security unless good key management is designed. If the WEP keys are updated often enough, then even if an attacker can compromise the key, this one is only useful for a limited time. Besides less data is transmitted using the same key and this makes the collection of information (e.g. collisions) more difficult for the attacker.

In summary, the link layer security provisions in the IEEE 802.11 are all vulnerable to attack. Therefore the system should put no trust at all on the link layer until new measures have been implemented.

Some hardware companies have included extra security provisions in their hardware, which are not defined in the standard. Here we analyse the security provisions in Agere Systems (formerly Lucent) Orinoco systems.

#### 5.5.4.2 Agere Systems Extra Security provisions

Agere Systems is a spin off company created from the former Microelectronics Group of Lucent Technologies. It provides wireless networking through its ORINOCO™ product line.

They provide some added security features to the IEEE 802.11 standard.

##### 1. Close Wireless System option

The IEEE 802.11 allows clients to connect to any IEEE 802.11 network present at a location if no SSID is provided by the client (empty string). With this option, you can instruct your AP to reject any client trying to connect using an empty string as SSID.

This option implies that a client can only connect to the network if it knows the network name. That is hardly difficult since you can sniff the SSID, which is broadcasted in the beacon frames and then use it to associate with the AP. Besides in a public access WLAN system, any user might be allowed to connect to the system (e.g. providing credit card credentials) and no security should be protecting the SSID. Therefore this feature does not provide any useful security to our system.

##### 2. 128 bit WEP key

The IEEE 802.11 standard defines the WEP encryption key to have a length of 64 bits, 24 from an IV and the other 40 as a user-selected secret key. ORINOCO has the option of using a longer encryption key of 128 bits of which 104 are a user-defined secret key. This increases the time to obtain the key using brute force attacks.

The complexity of the attack described in [19] against WEP grows linearly with the key length, and longer keys are also affected although the attack needs more time. It has been already proved with successful attacks [20] that vulnerabilities in the WEP implementation in IEEE 802.11 make 128 bit keys almost as vulnerable as 64 bit keys.

##### 3. MAC address based Authentication

It allows for a Radius Server to have a database of the MAC addresses of all the clients allowed in the system. When a client tries to associate with the AP, a check of the MAC address is conducted in the RADIUS database. If a client's address is not in the database, its traffic is filtered out by the AP. For this operation, a Radius client is provided in the AP. However, the RADIUS client included in the AP-500 and AP-1000 from Lucent do not support RADIUS accounting.

Authentication based solely on MAC addresses cannot be used in the system since the operator will probably allow a user to provide their own WLAN card. Therefore the operator cannot know beforehand the MAC address is allowed in the system. Once the user has provided a MAC address it can be used for increasing the security level, see Section 5.5.6.3.

In addition most wireless cards allow a change in the MAC address through software and since these can be easily sniffed on the air (no protection at all) hence registering the MAC addresses of the users provides a very limited level of security. User-based authentication is preferable to system-based authentication.

4. ORINOCO systems do not store the keys in Flash ROM inside the PCMCIA card. In that way stealing the card does not compromise the encryption key. The WEP key is encrypted and stored in the registry of the Operating System. So instead of stealing the card, the attacker

only needs to steal the computer to get access to the key. This feature provides very little security over the storage of keys in the Flash ROM.

5. Key distribution can be done by software allowing changing the WEP key in client stations and access points remotely, and having it activated the next time the adapter's driver is loaded (probably at the next reboot). The user is not aware of this change and continues with normal usage. This solution is quite neat because it helps to deal with WEP inefficiencies, but updating the key would need to be done very frequently and the process would need to be automated.

6. Agere's Access Server 2000 (AS-2000) introduces a new concept. The AP negotiates a session key using the Diffie-Hellman algorithm for every connection with the client. Every client has a different session key, thus even if you can attack WEP and figure out the key, it would only be valid for that user and for that session. It also provides for PAP and CHAP authentication using a RADIUS server in preparation for support of 802.1x.

Other vendors have similar proprietary solutions in their hardware, but they all pose the same problem to systems such as the one described in this thesis. In order for these solutions to work all access points and adapters have to come from the same vendor and that is nearly impossible in public WLAN usage. Besides most of these solutions imply the installation of proprietary software in the client which is not allowed in the specified requirements for a basic user.

#### **5.5.4.3 Physical security of the base station**

Base Stations should be placed not only to provide a suitable coverage area, but they should also be difficult to access. For example in visible places, where lots of people can see if someone is manipulating it, and out of normal reach (e.g. high enough a ladder is needed to reach it). Unauthorised physical access might allow an attacker to bypass other security measures (e.g. disconnecting the AP from the DS or modifying the configuration by direct connection).

Risk of theft is important to take into account in your system. One preventive measure used at the IT University wireless network is to use the SNMP agent available in the base station. If the base station is switched off or the PCMCIA card extracted, an SNMP Trap message is sent to the server. However this way if you unplug the Ethernet and then take the base station, the AP is not able to send the message. So you have to check if the base station responds to SNMP messages periodically and alert the system administrator if it does not.

#### **5.5.5 Network Access Authentication**

This section presents several trends in network authentication.

Authentication, Authorization, and Access Control are all very closely tied together in a public Wireless LAN system. In fact all systems analysed during this thesis have both Authentication and Authorization completely integrated together. The SPINACH system at Stanford University [35] provides a separation of both systems in theory, but was not implemented that way.

One conclusion of our work is that Authentication and Authorization will need to be separated in public WLAN access whenever more specific access services need to be provided. In the system described in this thesis Authentication and Authorization are tied together because a user is either granted access to the Internet or denied that same access. If the operator wishes to offer different access services, like access to specific local services, different bit rates for

different users, tunnels to other networks, or Virtual LANs (VLANs) there is a need to separate the two services in order to differentiate between users with different access rights.

The RADIUS protocol does not provide for separation of Authentication and Authorization because the server can only send an Access-Accept or an Access-Reject, therefore it does not allow distinguishing between different access rights. On the other hand, Kerberos provides just Authentication and leaves the Authorization to the kerberized applications. A user authenticated by the KDC gets a Kerberos ticket with a limited validity. When the user presents this ticket to the proper application server it performs Authentication. Access rights could be added in the tickets so that an application can differentiate if a properly authorized user is allowed to use the service or not.

Another problem identified in public WLAN systems is whether authentication should be performed on the client or the user. Since accounting and auditing are done on a per user basis, it makes more sense to have user based network authentication. Besides several problems arise when the system authenticates the client:

- ❑ In a multi-user host if you authenticate the client then several users are allowed access.
- ❑ Should a user be logged onto two different clients at the same time?
- ❑ What happens when the client is stolen?

Centralized authentication is needed in public WLAN access networks. In early deployments, authentication could be done at each AP individually, but this meant lots of administrative work when a new user needed to be added or removed. The need for a centralized database that stores the user's credentials is clear and in return it allows the separation between the provider of the service and the provider of the architecture. For example in an airport all the wireless infrastructure including the APs and the different NASs could be set up and paid for by the airport, but the authentication server, accounting database, and the user support could be provided by one or more Service Providers. Between these entities a revenue sharing contract would be negotiated.

Network access authentication can be implemented in several layers:

- ❑ Physical layer: 802.11
- ❑ Link Layer: 802.1x
- ❑ Network layer: IPSec client authentication
- ❑ Session layer: SSL client authentication
- ❑ Application layer: Web login

There are two different ways to provide authentication: authentication of the user and per packet authentication (making sure each packet comes from where it is supposed). Per-packet authentication, based on a key derived during the authentication process, thus linking each packet to the identity claimed in the authentication is not supported in Point-to-Point (PPP) and WEP.

### 5.5.6 Credentials

Credentials are pieces of information passed from one entity to another, in order to establish the sending entity's access rights. The system involves the use of two kinds of credentials: user's credentials and client's credentials.

In order to identify itself to the system, the user has to present some kind of credentials, mainly username and password, but sometimes the user cannot previously be identified



because they have no credentials (test user for example). In this case client credentials, namely the MAC address, are used in order to check the access rights.

### 5.5.6.1 NAI

One system requirement is that any user must be assigned a unique identifier. Network Access Identifiers (NAIs) are a standardized method for identifying users [57]. The main use for NAIs is roaming where the same username can be assigned (independently) in different administrative domains and when it is necessary to identify the user's home domain. Nowadays most AAA servers identify their clients by using the NAI.

The syntax of the NAI is: `nai = username@realm`

Mobile IP describes a NAI extension to the Mobile IP registration request [58] in order for the foreign AAA server (AAAF) to know where to forward the authentication request (i.e. to which realm) and uniquely identify the mobile node.

### 5.5.6.2 Password

User's credentials are stored in a centralized database. If the database is compromised in any way, an attacker gains access to the credentials. A common way to protect the credentials, used both by the Unix and Windows NT operating systems, is to keep them only in hashed form.

A hash is the result of a one-way function. One-way functions are mathematical functions which provide a fixed size output given an input, but with the special property that from the output it is not practically possible to obtain the original input.

The hashing algorithm used by our system is SHA-1 [83]. The SHA-1 algorithm is a second version of the Secure Hash Algorithm (SHA) [84], which provides a 160-bit length hash, usually 32-bit aligned. It is more resistant to brute force attack than MD4 or MD5 (128-bit hashes) though it has worse computational performance. The SHA-1 performance is around 6.8 Megabytes per Second on a 90 MHz Pentium. SHA is 62% as fast as MD5 and 80% as fast as DES hashing [85].

The user presents their credentials in the login webpage and the browser sends them, encrypted with SSL/TLS [69], to the NAS. The NAS hashes the password and runs the authentication process (either Radius or Kerberos authentication) using the hashed password as the user's credentials along with the username/NAI, which does not need to be hashed.

Only hashed passwords are stored in the database so the authentication process performs a comparison of both hashed passwords in order to authorize the user. If an attacker gains access to the database, they will only have access to the hashed passwords, which are not valid for login since the NAS would hash the hashed password again, providing a wrong value to the authentication process. A problem referring to storage of passwords and CHAP authentication is presented in Section 5.5.14.

An attacker can use two different types of attacks to try to recover the original password from the hash: brute force attack or dictionary attack.

Brute force attack means trying all the possible input combinations and comparing the hashes to the stored hashed passwords. When a match is found a valid password is known. This process is computationally very expensive.

Dictionary attacks are more intelligent and require less computational power. The attacker hashes only selected inputs (typical passwords and variations) instead of all the possible

combinations. Salting of passwords can slow down dictionary attacks, but this has not been yet implemented in the system.

User's passwords should be carefully chosen to be robust to brute force and dictionary attacks. They should not be single words in any language or any type of acronyms. Ideally they will be long and easy to remember to the user, mixing upper and lower case character, digits and other characters. The system administrator could check for the strength of the password against a basic dictionary attack before allowing a user to use it. For more information on passwords a reader can refer to Section 4.1 in [86]

### **5.5.6.3 MAC address**

The system does not require the user to have a specific WLAN adapter, but requires the user to notify the system administrator if he changes adapter. The reason for this is that the system keeps track of the MAC address of the adapter in order to make more difficult for an attacker to steal a session. See Section 5.5.6.3 for details. Therefore a new user has no MAC address associated in the database. At the time of the first login, if the rest of credentials are valid, the MAC address of the user is stored in the database.

This shows the balance between ease of use and security. If no MAC address control is performed at all, the user does not need to notify the administrator of a change of WLAN adapter, but the security is lowered because it becomes easier to steal a session.

When the NAS receives the user's credentials from the web interface, a Radius Authentication Request is sent to the Radius Server. This request includes the MAC address of the client in an Accounting Multi-Session-ID attribute since no better suited attribute was found. The Radius Server compares this value to the MAC address stored in the database for that user. If the comparison fails, then the user will be denied access. If the comparison holds and the rest of the credentials are valid, then an Access Accept is sent back to the NAS.

## **5.5.7 Access Control**

Access Control determines how the system limits the use of resources to only authorized clients.

It is usual in many access systems for access control to be implemented at the same place that authentication is. For example, if a gateway is to perform the access control, then it will probably also take care of the authentication process which will provide the user's credentials to an authentication server, which can be located locally at the gateway or in a remote host. In other words, the usual approach is to place an authentication client in the same host where the access control module is located.

### **5.5.7.1 Access Point versus Gateway Access Control**

There exist two possible locations for performing Access Control in a system architecture like the one described in this thesis, namely the APs or the gateway.

Systems similar to the one described in this thesis [11], have usually performed access control at a gateway using a packet filtering firewall. This was also the approach used in the implementation part of this thesis, but during the analysis part of the thesis another possibility has been studied.

Using IEEE 802.11 in infrastructure mode, in order for a mobile node to be able to send or receive traffic through a certain AP it first needs to associate to that AP. This association is needed in order for the distribution system (DS) to know where the client is located and to which AP must the traffic to this client be directed. This makes the AP a very good place to

perform access control since the user does not gain any access to the network if the client has not first performed an association.

As has already been explained, the IEEE 802.11 standard defines several ways to provide Access Control at the AP, but these are very basic and several flaws have been found especially in the Shared Key authentication method [115]. Different vendors have implemented proprietary solutions in order to allow Access Control at the APs to be centralized from an authentication server.

One approach used by Agere Systems has already been introduced in Section 5.5.4.2. A Radius client in the AP allows authentication based on MAC addresses. A Radius server keeps a centralized database of MAC addresses and the AP consults the server providing the client's MAC address prior to making an association. This type of MAC address authentication was used in [11].

Access point access control has the advantage that an unauthorized user cannot send or receive any packets, not even to other users in the same cell, since the AP does not bridge these packets. This solves the problem of how to control traffic between clients in the same Basic Service Set (BSS). It also establishes a first line of defense against other type of attacks like a denial of service attack on the DHCP or DNS server.

Clients could directly communicate with each other in Ad-hoc mode but this does not consume any resources from the AP, although it would create interference if they were in close channels.

An unauthorized user with a suitable driver for their adapter will still be able to listen to all the traffic in the cell being sent or received by other users. In order to prevent the attacker gaining any valuable information from this traffic some kind of confidentiality mechanism, such as encryption, is needed.

At the time of designing the system to be implemented in this thesis only proprietary solutions like the one mentioned in 5.5.4.2 were available for access point access control. This means that all the APs installed by the operator had to be from the same hardware vendor in order for the access control to work. Most operators might find this unacceptable. A standardized way to provide access control at the AP is provided in IEEE 802.1x and will be discussed in the next section.

Gateway Authentication does not require any vendor specific AP. The problem lays in how to control traffic exchange between different users in the same BSS when the access control is placed at the gateway. As long as they do not send traffic through the firewall (do not try to access outside networks) it is difficult to control this. The only way proposed in this thesis work is using some authentication with the DHCP server, such as proposed in [8]. This is thoroughly explained in Section 5.5.7.5.

In this way, two users who might be playing an online game between them, sending and receiving traffic through the AP, but not through the gateway could be stopped, since they need IP addresses in order to do that. But of course they could simply pick two IP addresses and this would be hard to control if the AP does not provide some type of filtering on IP address.

It is recommended that both AP access control and gateway access control be implemented in a public hotspot because they can provide complimentary functionality. For example, the Access Control performed at the AP cannot differentiate between different services while this type of Access Control can easily be performed at a gateway using a packet filtering firewall.

### 5.5.7.2 IEEE 802.1x

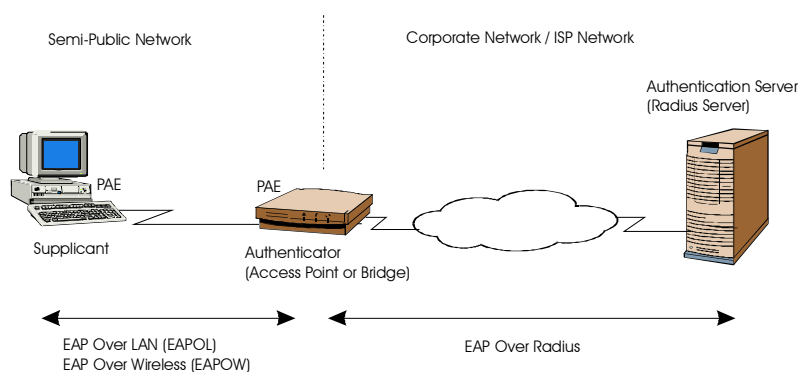
The IEEE standard for Port based Network Access Control [147] was ratified in June 2001. It is based upon the Extensible Authentication Protocol (EAP) defined in [75] and which might be made obsolete by [77].

It provides a framework for authentication and key management in all IEEE 802 networks. It helps to provide authentication, authorization, and access control to devices attached to a LAN port that has point-to-point connection characteristics. Examples of such LAN ports include the ports used to attach a server or router to the LAN infrastructure and associations between stations and AP as defined in IEEE 802.11.

IEEE 802.1X can be used to derive keys, which can be used to provide per-packet authentication, integrity, and confidentiality. However, it does not provide these per se, it only helps to derive keys for the session; a cipher like WEP, 3DES, or AES [70] is needed to provide encryption. It is commonly used with key derivation algorithms like TLS [69], Kerberos [42] or SRP [68].

It supports multiple authentication mechanisms without needing changes to the AP or the NIC. This allows for great flexibility since different security solutions can be used and the reaction time to security issues is reduced greatly because authentication and key exchange mechanisms can be upgraded without hardware modifications, avoiding major problems like the security holes found in WEP.

Only EAP authentication methods are supported by the standard, PAP and CHAP are not supported although there exists an MD5 challenge type that is equivalent to the CHAP Challenge. This is hardly a drawback since the security provided with PAP and CHAP is limited and most EAP methods provide a higher level of security. EAP methods supporting mutual authentication (e.g. TLS or SRP) are recommended in order to make sure that the derived keys arrive to the right entity, avoiding for example attacks based on rogue APs.



**Figure 5.8 802.1x Topology**

The standard defines the following entities, which are shown in the Figure 5.8.

#### Authenticator

An Authenticator is an entity at one end of a point-to-point LAN segment that facilitates authentication of the entity attached to the other end of that link. The authenticator does not need to understand the authentication mechanism used, it acts simply as a pass through agent for the Authentication Server.

### Authentication Server

An Authentication Server is an entity that provides an Authentication Service to an Authenticator. It determines, from the credentials provided by the Supplicant, whether the Supplicant is authorized to access the services provided by the Authenticator.

### Port Access Entity (PAE)

The protocol entity associated with a Port. A given PAE can support the protocol functionality associated with the Authenticator, the Supplicant, or both.

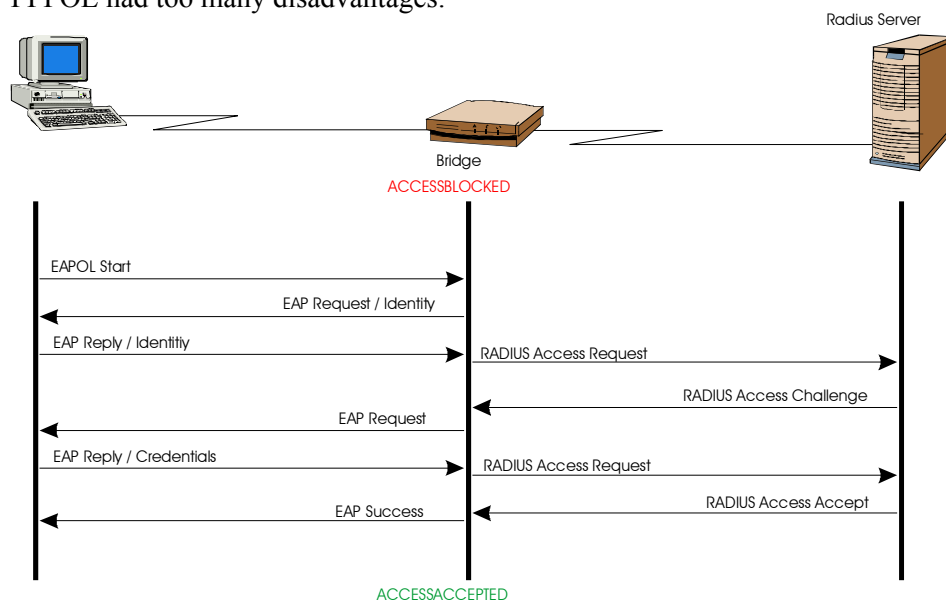
### Supplicant

A Supplicant is an entity at one end of a point-to-point LAN segment that is being authenticated by an Authenticator attached to the other end of that link.

The standard does not mandate use of an Authentication Server, and thus can be deployed with stand-alone APs or switches, as well as in centrally managed scenarios. Deployment of an AAA server to manage Access Control in a centralized manner is desired in several different scenarios. In such situations, it is expected that the Authenticator will behave as an AAA client.

EAP defines the security communication between Supplicant and Authentication Server using the Authenticator to forward the frames.

One main idea behind IEEE 802.1x was to control the access to a public network in an inexpensive way (on deployed switches) and using existing network infrastructure such as RADIUS or LDAP. Other previous approaches such as DHCP authentication, VLAN, or PPPOE had too many disadvantages.



**Figure 5.9 802.1x Authentication Exchange**

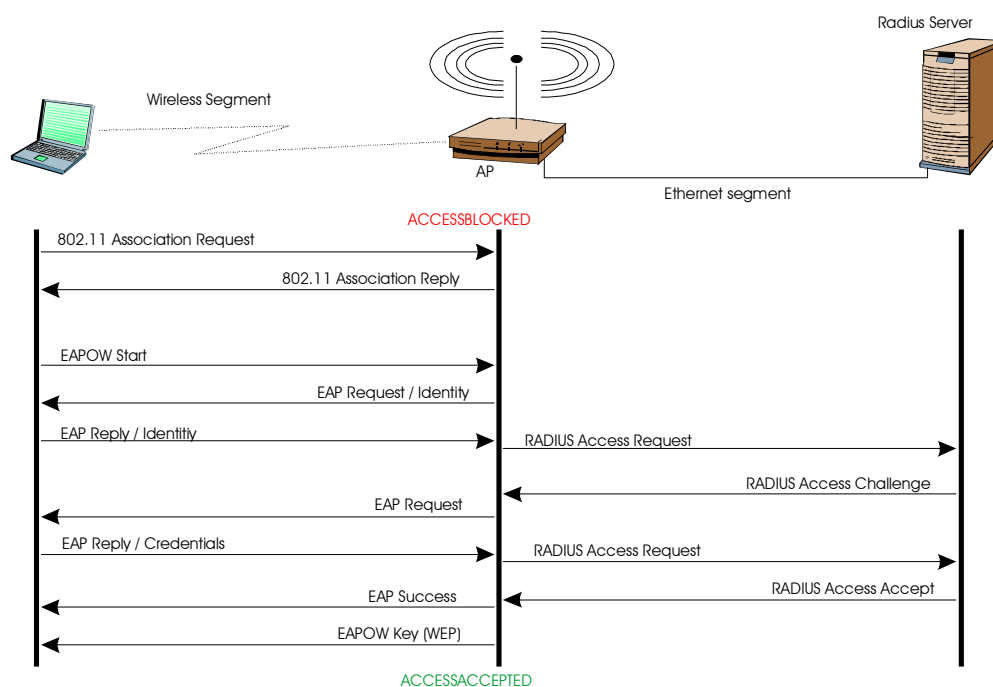
Another fundamental requirement for 802.1x was to support user authentication instead of device authentication, something that was lacking in other network authentication solutions. Finally, providing an extensible framework was desired after facing problems in upgrading the RC4 algorithm in APs because it is embedded in the MAC layer and thus became difficult to upgrade.

The security conversation implemented by 802.1x is shown in Figure 5.9. A backend RADIUS server is used as Authentication Server to keep centralized user credentials.

### 5.5.7.3 802.1x with 802.11

The 802.1x standard can be used with any IEEE 802 access network. Combined with 802.11 it provides a controlled wireless access network with user identification, centralized authentication, and key management.

The association process is carried in the way defined in 802.11, but after association the 802.1x authentication process is started. Prior to authentication the AP filters all non-802.1x traffic from and to the client. Once the client has been successfully authenticated using the Authentication Server, the traffic from and to that client is allowed (see Figure 5.10).



**Figure 5.10 802.1x Authentication**

The keys are derived between client and the RADIUS server and then transmitted by the RADIUS server to the AP. Since the RADIUS server and the RADIUS client (AP) share a secret, this is used to encrypt the attribute containing the keys to avoid eavesdropping.

One session key can be derived for each user per session, but if global keys are used then the session key is only used to encrypt the transmission of this global key to the supplicant which is done using an EAPOL-Key packet for the WEP key, as shown in Figure 5.10.

Per-user keys were supported in IEEE 802.11, but most implementations only support global keys. The problem with global keys is a common one in security: when secrets are shared between many people they become easily compromised. Static keys pose a problem for managing and securing them in the clients. Using IEEE 802.1x per session user keys are easy to employ.

One main problem with 802.11 was there was no mutual authentication. APs could authenticate clients, but clients could not authenticate APs, this opened the door for attacks based on rogue APs. Mutual authentication is not explicitly provided by 802.1x and Appendix C in the standard mentions a possible attack by using a rogue bridge. In such attack, the attacker could send an EAP-Request with limited authentication (such as EAP-MD5 with a

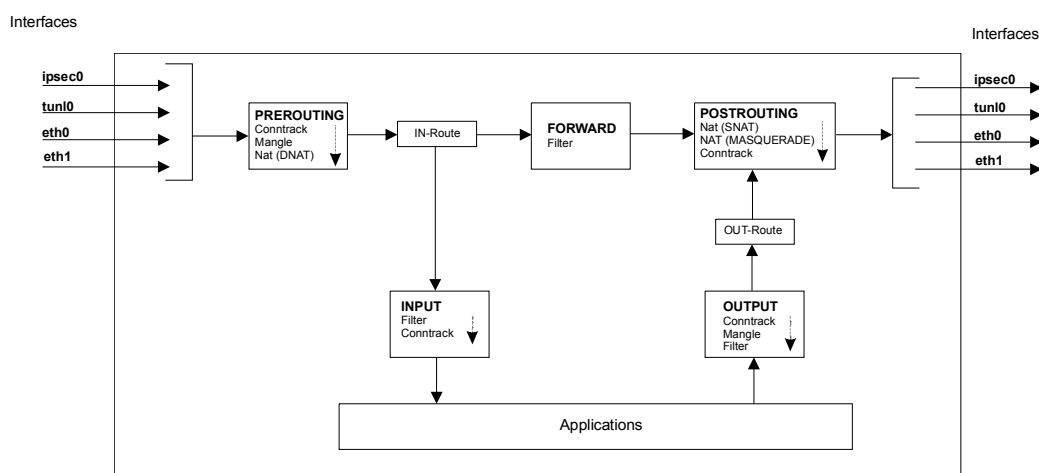
static challenge) in order to perpetrate a dictionary attack on the user's password. The standard proposes configuring the client to require an EAP type which supports mutual authentication as the solution to this attack.

Support for any AAA protocol is optional in IEEE 802.1x, although suggestions of RADIUS usage with IEEE 802.1x Authenticators are provided in an appendix of the specification. The standard is currently being updated by the IEEE and also discussed in informational terms with the IETF [105].

#### 5.5.7.4 Firewall

The access control mechanism in the system presented in this thesis is based on a dynamic packet filtering firewall provided by the iptables software which has been lately included as part of the Linux 2.4.x kernel distributions.

Iptables is the successor to ipchains, which was the packet filtering firewall in Linux kernels 2.2.x distributions. It is recommended in this thesis to use iptables over ipchains if Linux is used. All Linux firewall development is nowadays done using iptables, hence a lot of new functionalities are available (e.g. filtering on MAC addresses) and the architecture has also been slightly improved. A very useful description of how packets traverse the different firewall chains in iptables is provided in Figure 5.11.



**Figure 5.11 Iptables Firewall**

Iptables firewalling is controlled by rules, which perform actions over the different chains. These rules have to be changed dynamically in order to allow traffic only coming or going to authorized users. In our implementation the authentication daemon (`auth-d`) takes care of inserting firewall rules when the user logs in and the control daemon (`ctrl-d`) takes care of removing those rules when the user needs to be logged out.

The previous figure is especially useful when analysing how tunnelled packets (e.g. IPSec or Mobile IP) traverse the NAS. Most of the implementations use kernel sockets and virtual interfaces (e.g. `tunl0` or `ipsec0`) to manage these encapsulations and packets do not traverse all the chains that a normal IP packet would.

#### 5.5.7.5 DHCP Security

The Dynamic Host Configuration Protocol [111] (DHCP) provides a framework for passing configuration information to hosts. It is one of the most extended protocols in the TCP/IP suite. Configuration parameters that can be sent to hosts are: IP address, default gateway, network mask, DNS server address, DNS domain, and many others.

The limited availability of IP addresses has helped to spread the idea that they are a valuable resource and tighter control over them might be desirable. A network administrator might wish to constrain the allocation of IP address to authorized hosts, especially in a medium that is not physically secured such as a wireless network. In this way access control to the network can be implemented in the DHCP server because a client needs an IP address before it can communicate with other hosts.

The problem is that the protocol was not designed for this function and other solutions might be better suited for providing access control to the network. For example if 802.1x is used, then when the client requests an IP address from the DHCP server it has already been authorized although authentication of DHCP packets is still needed to stop insider attacks. In general if user authentication and access control is moved to the link layer, then the threats to DHCP are limited to an insider threat.

### Attacks against DHCP

Most of the threats related to DHCP security arise because the DHCP packets are not authenticated. Most typical attacks trying to gain advantage through this are DoS attacks.

For example, placing a fake DHCP server in the wireless segment. The fake server could provide IP addresses in any range different from 192.168.x.x to the clients and this would result in the firewall rejecting the authentication requests. A client that receives fake configuration settings might also send the traffic to a sniffer instead of to the NAS.

Another threat is exhaustion of the IP addresses by an attacker who changes MAC addresses between requests. This implies flashing a new MAC address to the adapter quite fast and might only be possible when the leases are long. The best general solution against DoS attacks is redundancy of the resources (e.g. two NAS or two RADIUS servers).

The necessary security requirements identified in the original DHCP protocol are:

- Mutual authentication between client and server
- Access control in the server
- Integrity of the messages
- Replay protection
- Backwards compatibility with non-secure DHCP clients

A new solution for authenticating the source of DHCP packets has been standardized [8]. The Delayed authentication method defined in that standard provides integrity, protection against replay attacks, and source spoofing to the DHCP packets. Both the client and the server have to share a secret known only to both of them and the packets are signed using a keyed hash (HMAC-MD5). On reception the signature is checked using the shared secret. Sharing of keys between different clients is strongly discouraged. This of course leads to key distribution problems (see end of this subsection).

Some attacks are still possible against this method like flooding the DHCP server with DHCPDISCOVER messages because these packets are not authenticated. Flooding the server with authenticated messages might also be possible.

Other solutions for applying security to DHCP were discussed in [7], but have not been standardized. The reader should refer to that paper for further information.



### A possible improvement

One way to add some DHCP security to the system could be the following. When a new user arrives, the DHCP server provides this client with a temporary IP address with a short lease time. That IP address is used by the client to perform the authentication process and it comes from a specific range of addresses used only for this purpose. Packets with source IP address coming from the range of temporary addresses are never allowed to traverse the firewall, but could be used for local services. Once the user has been properly authorized by the system through the AAA infrastructure, another IP address is provided to the user. This time the lease time is longer and packets coming from that IP address (and from the specific MAC address) are allowed to traverse the firewall.

The exact way to implement this has not been designed in this thesis, but a very similar scenario is described in a very recent Request For Comments (RFC) [55], which defines a way for DHCP servers to force a DHCP client to renew its configuration information. No known DHCP servers or clients support this yet.

This solution would help (but not solve) against denial of service attacks (see Section 5.5.8.4) trying to exhaust the range of valid IP addresses. Since the lease time is very short, an attacker would need to change MAC address very fast in order to be able to exhaust the pool of IP addresses. Let's say that the lease time is 2 minutes and the pool of addresses contains 1 000 addresses, this would mean that an attacker would have to change MAC address every 120 ms.

A reader should realize that if 802.1x is used in the link layer, the user has already been authenticated and there is no need to provide it with a temporary address. The problem comes when the client needs an IP address in order to perform the authentication.

### How to apply this to the system

In our system, no security has been added to the DHCP server, which in turn means that no modifications have been made to the DHCP client or server software.

The main problem using [8] is roaming between different administrative domains, so called inter-domain roaming. Then the secret shared with the DHCP server is not valid when the client moves to the new domain and the whole security of the model is lowered. The proposed solution is to combine the key generation and access control capabilities of 802.1x with support for [8] and [55] in the DHCP server and client. In this way the shared secret needed between to provide the authentication is dynamically generated by 802.1x when the client authenticates through the AAA infrastructure.

This solution would solve most of the scalability problems though some implementations issues still would have to be addressed (e.g. getting the key dynamically to the DHCP server). This is left to future work.

## **5.5.8 Some security threats to a WLAN**

### **5.5.8.1 Spoofing**

An attacker could spoof both the MAC address and IP address of a user in order to fake its identity. Still the attacker cannot open the firewall if it is closed unless they know the password, therefore they need to come along during the time that the firewall is still open but after the user has already left. In order to know the MAC address and the IP address that the user was using (since it was dynamically allocated by DHCP) the attacker had to be listening to the user's traffic.

The only way to prevent this form of attack is to allow encryption of IP headers (with IPSec tunnel mode for example) or to have some sort of anti-sniffer hardware. In addition, if the user closes the session when he leaves, then the attacker will need to steal the personal password (which travels encrypted with SSL) to succeed. These problems are more thoroughly explained in Section 5.5.13.

Also an attacker can perform a DoS attack by spoofing the IP and MAC address of a specific user to create collisions and prevent communication from the proper user.

#### **5.5.8.2 Rogue Access Point**

When a mobile station is turned on, it will usually try to associate with the AP that generates the strongest signal of the several signals it is receiving. If an attacker places a rogue AP in the same area as a valid AP, sending the same SSID, but with a stronger signal than the valid one, it will fool the mobile stations into logging in to the rogue AP.

The attacker can reject all the authentication requests and record all the traffic. When enough data has been stored, it can try to find the shared key from that information. This is difficult to detect because mobile stations do not usually report unsuccessful authentications to the upper layers. The attacker in this way is also performing a DoS attack.

Mutual authentication is required to solve this. A possible solution is using 802.1x with clients that require an EAP type supporting mutual authentication. Detecting a rogue AP is not too difficult since the attacker has to be located in the area of coverage of the valid AP.

#### **5.5.8.3 Fake NAS Servers**

An attacker could pose his computer as a fake NAS giving out DHCP leases and accepting authorization requests. It could not gain access to the password since it is encrypted with SSL and it lacks the appropriate certificate. The user can detect this attack by checking that they are accessing the right host: <https://nas.r2m.net> through the certificate that the NAS server (either valid or fake) has to present.

#### **5.5.8.4 Denial of Service**

One possible attack is to place an AP, which generates a very strong signal, causing so much interference with other AP as to render the system useless. This kind of attack is usually called jamming. As with rogue APs, this attack is quite easy to detect and find, since the attacking device has to be located close to the valid AP.

#### **5.5.8.5 Monitoring of traffic**

One of the main problems when deploying WLAN corporate networks is that the signal leaks through the company boundaries, but this is hardly a problem with public networks where the larger area reached, the better (being careful with the interference of course).

One basic solution to avoid monitoring of traffic is to allow WEP but as was already explained an attacker using existing tools could easily bypass this protection. Besides key distribution becomes a real problem. A better solution implemented in this thesis is to allow encryption with IPSec over the wireless link. The only problem is that not all the users might own an IPSec client but the operator could provide it to them.

In the near future, employing 802.1x monitoring of traffic will become an insider threat and intruders will be easier to detect (see Section 5.5.7.2 for details).

#### **5.5.8.6 Abuse of resources**

Controlling internal traffic poses a problem as mentioned in Section 5.5.7.1.

### 5.5.8.7 AP Configuration

The different AP available present different configuration interfaces such as Simple Network Management Protocol (SNMP), serial (console), web, or telnet access. Some protection is needed to avoid an attacker to write new configuration to the AP using these interfaces. Examples of interfaces used by hardware vendors are:

- ❑ Cisco & 3Com: SNMP, serial, web and telnet
- ❑ Lucent: SNMP, and serial

### 5.5.9 Intrusion Tolerance

The damage that an attacker can do if he gets access to the NAS, database or Authentication Server, is described below:

#### 1. Breaking into the NAS

The NAS acts as an Authorization Server and an Access Control Server. An attacker who gains access to the NAS compromises the keys shared with other machines, which should be reissued. It also gains access to the network and could perform Denial of Service attacks against any user since it controls the Access Control module. It can also modify log files to hide its presence.

On the other hand the attacker does not get access to any user authentication info such as usernames and passwords, which is stored in the database in order to provide a centralized authentication scheme as explained before.

#### 2. Breaking into the Authentication Server.

The attacker compromises the keys shared with the database and the NAS. These keys (or shared secrets) are used for both RADIUS and IPSec communication. No user information is stored in the Authentication Server.

#### 3. Breaking into the Database

This is probably the worst case since the attacker gets access to all the accounting information and can modify and delete the records. The attacker gains also access to the hashed passwords from the users, but still needs to break the hash or session encryption to get useful passwords.

### 5.5.10 IP Security (IPSec)

This section talks about how IPSec support has been added to the system. For an introduction to IPSec the reader should refer to Section 2.1.3.

IPSec has been used in the system, to provide extra security over the air interface and to provide secure links between the NAS and the authentication server and between the authentication server and the database. For this, an IPSec gateway was added to the NAS. The IPSec usage in the system is described in Figure 5.12.

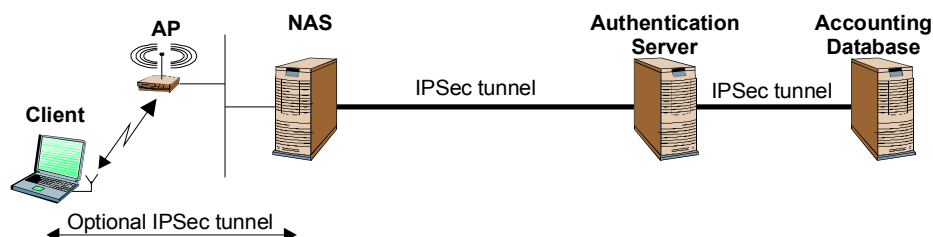


Figure 5.12 IPSec usage in the System

The IPSec distribution used is FreeSwan 1.91 [112] with support for x509 certificates added through a patch [101]. Newer versions of FreeSwan already have the patch applied in the source code.

The IPSec links between the NAS and the authentication server and between the authentication server and the database are configured to use ESP in tunnel mode. They are managed by the System Administrator and need the exchange of keys or certificates between the different hosts. In the case that all the three entities (NAS, authentication server and database) are placed in the same host, no IPSec tunnels are used.

The IPSec tunnel between the NAS and the authentication server is important because the encryption of the user password provided by the PAP method in RADIUS is not very strong. Using an IPSec tunnel, there exists a triple protection over the password. First it has been hashed in the NAS, then encrypted by the RADIUS protocol and finally encrypted again by the algorithm defined in the IPSec link (e.g. Triple DES).

Roaming agreements with other domains could include the negotiation of an IPSec link between the authentication servers in order to increase the security. Deciding the algorithms to be used and exchanging keys or certificates are the needed steps.

A user who desires extra security over the air interface, including confidentiality, integrity and packet source authentication, should install IPSec software in their client. Two different IPSec clients have been tested: a FreeSwan client using the same distribution as in the NAS and a Windows 2000 Professional client, using the embedded IPSec configuration in the operating system.

It was found that the FreeSwan client presented several advantages over the Windows client including that once the IPSec software is installed it could be configured to start the IKE negotiation automatically at boot time making it more transparent to the user. In both cases the user needs to configure the software, though some installation software could be provided to the user in a possible future deployment.

A reader interested in using a Windows 2000 or Windows XP client to talk to a FreeSwan gateway (road-warrior configuration) might have a look at [146] where a tool is provided to ease the configuration. The problem detected when using Windows 2000, as road-warrior is that Microsoft has not allowed configuring the subnet behind the NAS as “any”. This means that you have to specify which is the subnet behind the NAS and only traffic to that subnet is encrypted. With FreeSwan, the subnet behind the NAS can be defined as “any” and all traffic from the Mobile Node to the NAS or to an outside network is encrypted between the MN and the NAS though it is unencrypted from the NAS to the final destination (e.g. [www.google.com](http://www.google.com)). We hope that this behaviour can be modified in a following Service Pack. However, other IPSec clients (e.g. PGPNet) could be used to work around this problem.

Different independent groups have tested the FreeSwan distribution used in the NAS to interoperate with several other IPSec clients, but we cannot guarantee that it would work with any specific one. If the user were allowed to use its own IPSec client this should be considered. If the user were provided the IPSec software by the Administrator this would pose no problem.

Public Key cryptography is preferred over shared secrets (very difficult to manage with large numbers of users) to identify a user and x.509 certificates are used for distributing and authenticating the keys. A Certification Authority (CA) was installed in the NAS to generate and store client’s certificates. See the following section for key distribution to the user.

If the user has some Virtual Private Network (VPN) software to connect to their corporate network it should not use our optional IPSec link to the NAS since that would mean much overhead (and the user might be charged by data sent and received). It is recommended (as mentioned in Section 5.5.11) that a user who wishes an end-to-end security solution to access its corporate network (or at least to the corporate gateway) use some type of VPN software.

The implementation of the system allows an advanced user with an IPSec client to establish an IPSec connection with the NAS and then requires authentication through the webpage. This means that the user is really authenticating twice to the system. One unnecessary authentication is performed. FreeSwan can be configured to execute a script when a connection goes up or down, which could be used to open the firewall and start the accounting (or to close the firewall and stop the accounting). This is left as future work.

An IPSec link between the Mobile Node and the NAS would break if the Mobile Node moves to another hotspot (different NAS) or even to another segment pending from the same NAS. Handover between IP subnets when using IPSec would require a new IKE negotiation that would slow the handover. This has not been studied in this thesis and is left as future work.

### 5.5.10.1 Key distribution

This section presents the procedure needed to provide an advanced user with the keys they need to use IPSec on the wireless link. This process needs to be performed the first time that the user gets their keys and every time it needs to change them (e.g. when the private key has been compromised).

When a user needs a new pair of keys, it needs to contact the system administrator (or an authorized entity). The specific procedure of how a user contacts their system administrator might vary depending on the wireless operator. Usually an IPSec client needs its private key, host certificate, and the CA certificate and the most convenient way to provide all this information to the user is usually a PKCS#12 file.

The system administrator does the following:

1. Generates a private key and a certificate request for the user
2. Signs the certificate request using the CA
3. Generates a random string to use as password to protect the keys
4. Prints the password on a piece of paper
5. Creates a PKCS#12 file with the private key, the certificate request signed by the CA, and the CA certificate, encrypted with the password generated in the previous step
6. Copies the PKCS#12 file to a floppy disk
7. Installs the host certificate and the user identifier in the central database
8. Gives the piece of paper with the password and the floppy disk to the user

All the previous steps can be performed, for example, using the OpenSSL library and reader might refer to the x509 patch Installation and Configuration Guide [101] for more specific details.

The user should install the PKCS#12 file in their client using the password provided to him to extract the information. In a real commercial deployment the user would be provided an executable file that reads the file from the floppy, prompts for the password and installs the private key and certificates in the proper places.

### 5.5.10.2 Ericsson's WLAN security solution

Ericsson has a WLAN security solution called “WirelessGuard” [87] to enhance security over the wireless segment, which has been analyzed during this thesis. It is an IPSec/IKE gateway, which uses ESP in tunnel mode to provide confidentiality, integrity, and packet authentication. The solution is not designed for public access networks, but rather for corporations. User authentication can be done using shared key or X.509 public key certificates.

The solution does not provide any extra functionality compared to the NAS acting as an IPSec gateway using the FreeSwan Linux IPSec distribution as implemented in the system presented in this thesis.

### 5.5.11 VPN

A user may want to access their employer's corporate resources from a remote access network, such as another corporation, a public hotspot, or their home network. This kind of remote access is called a Virtual Private Network (VPN) and the security requirements in this case closely resemble those of access in a private network even if access is provided over a public access network and data is transferred over insecure links.

Advanced users will probably have VPN solutions and the system has to allow the use of these. This is currently allowed in our system although a user first needs to authenticate through the web interface before being able to establish the VPN to the corporate network.

The potential VPN solution for users vary from IPSec to tunnelling protocols (e.g. PPTP, L2F, or L2TP) although tunnelling protocols generate more overhead and if the user is paying based on the volume of data sent and received this kind of solution is not appropriate.

### 5.5.12 Cookie security

For a basic user who does not have a Mobile IP client installed, the system provides for a very basic mobility solution using cookies (see Section 5.7.2). Cookies are pieces of data, which web servers can use to store and retrieve information on the client. A cookie is usually as simple as some text, which is stored by the client's browser usually in a special directory as an independent file (e.g. Internet Explorer) or as some lines of data in a special file (e.g. Netscape Navigator or Opera).

During this thesis several problems were identified and solved for the case when using cookies to ease the login for a user:

#### Stealing the client's cookie

The cookie stored in the client, allows the user to relogin automatically to the system without having to provide the username and password manually every time he comes back into the system. This is true for as long as the cookie is valid as determined by the expiration field. If an attacker steals the client's cookie, he will be able to login with the user's id as long as the cookie is valid.

A feasible scenario is that the user leaves the computer unattended while going to have a coffee for example. At this moment, an attacker comes in and copies the cookie to a floppy disk. The attacker then copies the cookie into the proper location on their own client and they are now granted access to the system with the original user's id. From that moment on, any resources used by the attacker are charged to the original user.

### Expiration control

Cookies have an expiration date that is set by the web server that places the cookie in the client. The problem is that a cookie's expiration time is checked against the clock in the client not against the clock in the server. That means that if the attacker that has already stolen a valid cookie and copied it into its own client now changes its system time to a past date, he will be granted access to the system forever.

### Hashed password

One of the fields in the cookie set in the client by the NAS is the user's hashed password. This is not completely necessary for implementing the system, but it simplifies significantly the implementation. An attacker who steals the cookie as previously stated, can launch a dictionary or brute force attack on the hashed password until it is able to recover the original password of the user since the hashing algorithm is SHA-1.

#### **5.5.12.1 Solution**

All the above security holes have been solved in our system by using an encrypted cookie with some extra data fields. The cookie includes a timestamp stating the expiration time (currently 8 hours) that the NAS will check against its clock when it retrieves the cookie from the client and before granting access to the system. The MAC address of the client requesting authentication is compared against the MAC address stored in the database for that user and they if they are not the same, access is denied.

Finally, the whole cookie is encrypted by the NAS before sending it to the client using the Blowfish algorithm [136]. This way the hashed password is not accessible for the attacker who will now need first to break the Blowfish encryption before they can attack the underlying data. Being able to retrieve the user's password from the cookie it is considered highly unlikely since it requires to break first the Blowfish encryption and then launching a dictionary or brute force attack to retrieve the original password.

Blowfish is a symmetric 64-bit block cipher designed in 1993 by Bruce Schneier as a fast, free alternative to existing encryption algorithms. It uses a variable length key, from 32 bits to 448 bits and is slowly gaining acceptance as a strong encryption algorithm. The key length used in our system was 120 bits.

When a client tries to login to the system, if it has a cookie from the proper NAS, it is retrieved, decrypted, the timestamp is checked against the NAS clock and finally the authentication data (username, password and MAC address) is verified with the data stored in the database.

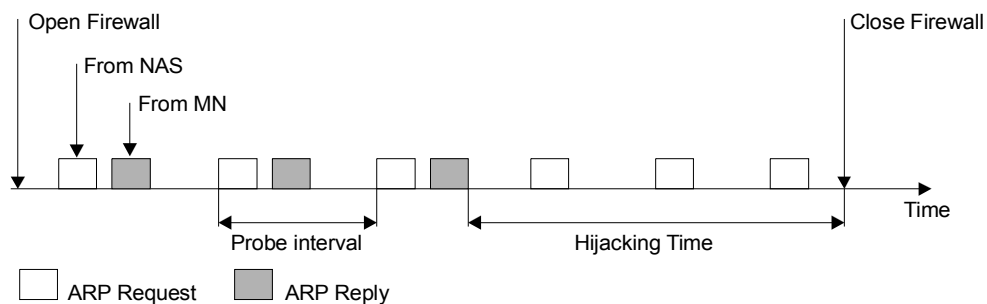
It is still possible (although harder) for an attacker to steal the cookie and spoof the original owner's MAC address in order to get into the system, but he will still only be granted access to the system for as long as the cookie is valid. The best solution is that the user prevents anyone from stealing the cookie, for example, by setting a password for the computer and locking their screen when the computer is left unattended.

### **5.5.13 Keep-Alive time window security problem**

Any access system, which keeps control of the users logged by probing them, is affected by the same security problem. This problem was stated in [11].

From the moment the user is authenticated and the firewall is open for traffic coming from its client, the system starts periodically probing the client to be sure that it did not leave the

hotspot. In our particular system, whenever the client does not answer three consecutive probes the firewall is closed for its traffic.



**Figure 5.13** Time window for keep-alive attack

There exists a window of time in which an attacker can try to steal the session. From the moment the client leaves the hotspot until the firewall is closed for that client (after 3 consecutive failed probes), any attacker, which fakes the identity of the departed user, can start answering the probes on behalf of the departed user in order to gain access to the system.

Since most access systems (including the one described in this thesis) use a packet filtering firewall, there are limited ways to filter packets coming from a client. Usually the IP address and the MAC address of the client might be used to differentiate between similar packets coming from different clients. However, as noted earlier this is not secure.

The described window can theoretically be reduced close to zero, by reducing the time difference between consecutive probes and the number of unanswered probes needed before closing the firewall, but there are some problems with this. The time difference between consecutive probes can be reduced, but this will increase the bandwidth wasted in controlling traffic.

Reducing the number of unanswered probes needed to log out a client presents some limitations in a wireless link where due to environmental conditions the error probability for a certain packet can vary and be quite high compared to the error probability for a wired link. If the system uses a single probe to check the availability of the user then it might happen that he is in an area with bad coverage and it would be log out when failing to reply to that probe. This presents inconvenience to the user who will probably be forced to relogin too frequently. It is not recommended to reduce this value below three.

In the system described in this thesis, the firewall identifies the client's packet by both IP address and MAC address. Thus, any attacker trying to use the time window to steal a session has to perform both an IP address spoofing and a MAC address spoofing. This is feasible but not trivial. IP address spoofing is quite simple but MAC address spoofing is more complicated.

A solution for this is described next, based on using a shared secret instead of probing the client, although it has not been implemented in our system due to a lack of time. It is considered to be a future improvement to the system.

### Solution

A proposed solution for the above problem uses the feedback module provided by the applet to do the session keep-alive function. When the applet is downloaded to the client from the NAS using SSL, a shared secret is passed to the applet, this is only valid for as long as the session stays open.



Every fixed amount of time, the applet opens a socket and transmits a UDP packet to a certain port in the NAS where a daemon is listening for keep-alive packets. This UDP datagram contains the shared secret for that session between the client and the server encrypted in the application data field. Authentication and integrity protection needs to be added to the UDP exchange.

Some retransmission mechanism for lost packets has to be implemented at the application layer since UDP packets are used, but still UDP datagrams are recommended over TCP connections because the opening and closing of a TCP connection poses too heavy a burden for this kind of operation.

The main disadvantage of this approach is that a new server, which would be located at the NAS, has to be written to substitute for the control daemon.

The main advantage is that security is tightened. The time window is still present for an attacker to fake the user's identity, but with this solution the attacker not only needs to fake its IP and MAC addresses, but it also needs access to the shared secret which is always transmitted encrypted over the network.

#### **5.5.14 Mobile IP security risks**

Two security issues were raised when the integration RADIUS-Mobile IP was implemented.

The main problem is that RADIUS authentication for Mobile Nodes uses the CHAP method included in the RADIUS standard. When using CHAP authentication, passwords have to be stored in clear text in the database or authentication server. If they are hashed they cannot be used as the key for the MD5 hash of the challenge response. This is specified in the CHAP standard [102] and therefore is a general CHAP problem; it is not a specific problem derived from the integration of AAA with Mobile IP.

One solution considered for many systems using CHAP authentication is to encrypt the passwords stored in the database using a symmetric cipher (e.g. 3DES or Blowfish). The problem is that the RADIUS server needs to access the password dynamically when it gets a Challenge Request and therefore needs to know the key to decrypt the passwords. This key would be embedded in the code, not in a configuration file (it would be the same problem as before). It changes a clear-text password file with a key embedded in software, but it is not a neat solution at all and does not provide much extra security (e.g. a memory dump of the program would disclose the key).

Another possible solution is to modify the Mobile IP client so it hashes the password shared with the AAA infrastructure before calculating the Authenticator. Because the RADIUS server would use the hashed password to calculate the MD5 hash, this solution might work. But it has not been implemented in the system and thus cannot be evaluated. However, the password still is in clear-text form in the configuration file of the client although at least now only a single password could be compromised at a time, instead of all the passwords at the same time. Still if the hashed password of a user is compromised in the database, the attacker gains access to the system.

Both solutions are very limited and the recommendation would be not to use CHAP authentication; however, at the moment is the only authentication defined with RADIUS servers so the only possible action is to strongly protect the database.

Another problem is that the FA has to accept all traffic from the Mobile Nodes in order to provide the IP over IP encapsulation. This means that any user with an IP address in the range

used by the Mobile Nodes can send any traffic to the NAS. This would open the door for numerous attacks and should not be allowed by a security administrator.

Fortunately this problem could be solved because of the special way that the *ipip* module, used in HUT's Dynamics IP-IP encapsulation, behaves. A packet sent by the Mobile Node to the FA passes through the Prerouting chain (See Figure 5.11) and then goes to the Forward chain, instead of the usual path through the Input chain. A couple of proper rules placed in the Forward chain allow the traffic that needs to be forwarded to the HA to continue its way without giving access to the NAS to the client.

```
iptables -A FORWARD -d <ipaddress> --in-interface <tunl0>
        --out-interface <interface> -j ACCEPT

iptables -A FORWARD -s <ipaddress> --in-interface <interface>
        --out-interface <tunl0> -j ACCEPT
```

### 5.5.15 Applet security

The applet is downloaded to the user using SSL, thus it is not possible for an attacker to modify the code of the applet. All the communication from the applet to the scripts is also done using SSL and every time the applet calls the logout script or the update script it passes a shared secret with the NAS in order to authenticate the remote call and avoid, for example, an attacker logging out a valid user.

The secret shared between the NAS and the applet is generated dynamically every time the user logs in. The RADIUS server generates it when the user logs in and transmits it to the NAS and the applet. See Section 5.2.2 for details.

A signed applet could be used by the system to implement a solution like the one proposed in Section 5.2.10, but this has not been done and is left as future work. The applet could also provide its own encryption using an appropriate Java library instead of using the TLS/SSL provided by the browser.

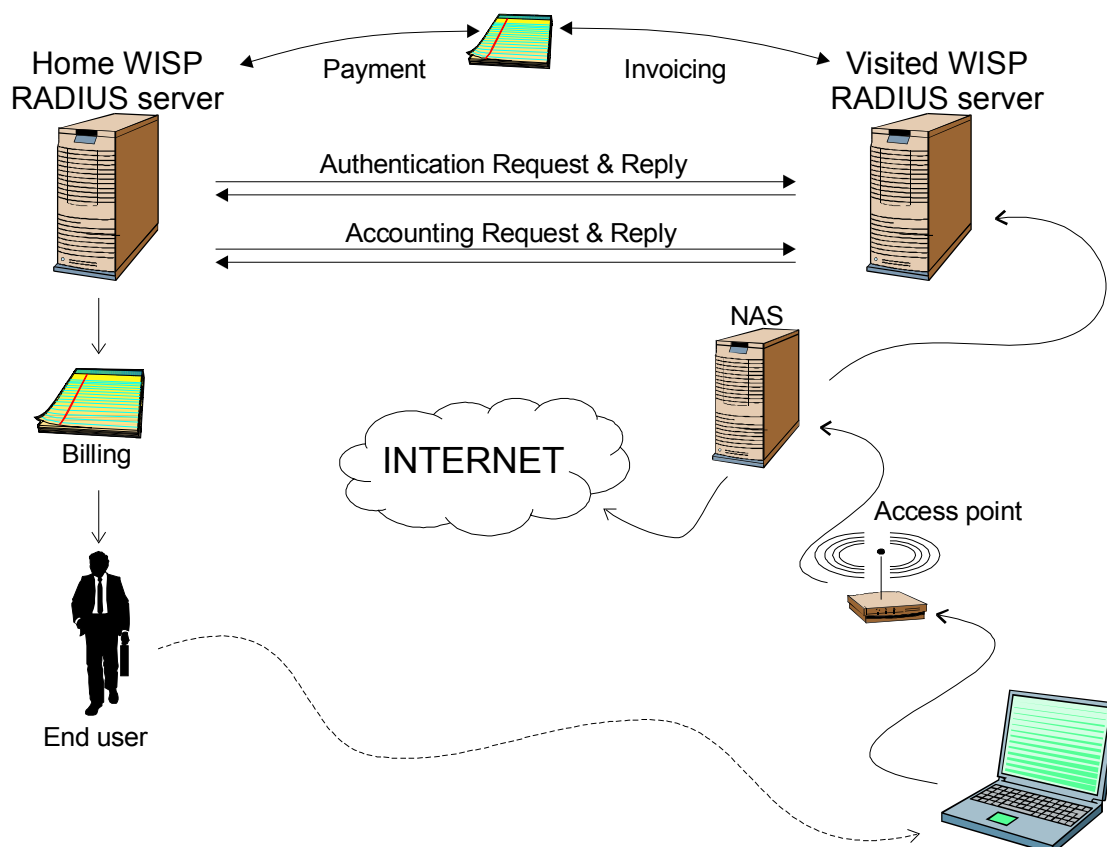
## 5.6 Roaming

Roaming may be loosely defined as the ability to use multiple ISPs, while maintaining a formal, customer-vendor relationship with only one [126]. Figure 5.14 shows an example how the roaming can work.

Roaming services can be implemented by using proxies. In such a system authentication, authorization, and accounting packets generated by the NAS are routed between the NAS and the home server through a series of proxies. Some of the advantages of proxy chaining are:

- ❑ Improved scalability
- ❑ Easy roaming policy implementation

In a large scale roaming system, it necessary to provide a scalable method of managing keys, used for authentication and integrity protection between the different roaming partners. This is essential for an AAA protocol, which has no automated key management, e.g. RADIUS. With roaming through proxy chaining, it's possible to implement a hierarchical topology.



**Figure 5.14** Roaming example

This way you only have to have a security association between the roaming partners above and under you in the hierarchical model. Or a number of roaming partners can together form a roaming consortium and construct a logical star model, as shown in Figure 5.15.

This reduces the number of security associations drastically, e.g. in a roaming consortium with 100 roaming partners, it would require 4950 shared secrets if every partner should have direct contact with every other partner. If they were to form a logical star with a central proxy they would only have to use 100 shared secrets. This would also reduce the number of bilateral agreements, which would considerably decrease the administration required. This model can also be referred to as a broker model. Figure 5.15 shows both cases, each line represents a shared secret.

Between different administrative domains it is often desirable to implement different policies, e.g. two partners might only have roaming agreement during a certain period of the day. These policies can be implemented in the proxies, where they act as a filter between the different administrative domains.

There are also some drawbacks when using proxies, some of them are:

- ❑ Decreased reliability
- ❑ Increased security vulnerabilities

Setting up a single central proxy server puts high demands on its reliability; as a proxy server crash would affect all roaming functionality. Thus, having a back up system is essential.

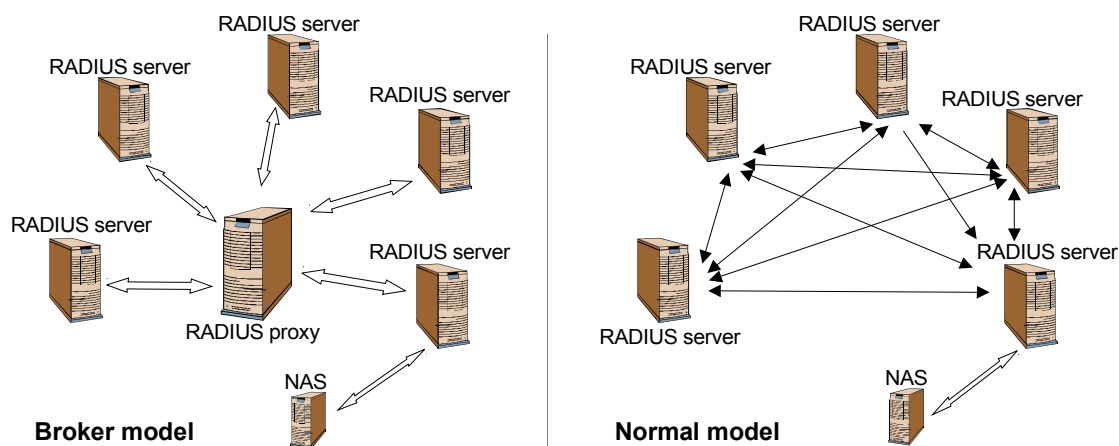


Figure 5.15 Example of a Normal and a Broker model

To make it secure against hacking attacks is also vital. Since all authentication and accounting traffic goes through the proxy, a successful attack upon it would be disastrous. If someone would take control of the proxy server he could change whatever packet is proxied, for example a protocol such as RADIUS where no end-to-end security exists. Such an attack could change authentication, authorization requests, and edit accounting requests thus allowing fraud. However, there is a draft that explains how end-to-end authentication can be achieved by using Kerberos V5 [127]. When putting up a central proxy server security should be the main concern.

### 5.6.1 Roaming with RADIUS

Roaming is achieved by the use of a RADIUS proxy, this means that a RADIUS server acts as a RADIUS client towards other RADIUS servers and can therefore forward authentication and accounting requests, as described earlier. The decision of and how a request should be forwarded is based on the realm part in the username (`user@realm`). All of the realms that one has roaming agreements with are defined in the `proxy.conf` configuration file in the FreeRADIUS distribution.

Since RADIUS doesn't support end-to-end security, a lot of man-in-the-middle attacks can be perpetrated, include modifying messages and theft of password (if authentication is done using PAP). The implication of this is that trust between the roaming partners is important to avoid fraud.

## 5.7 Mobility

### 5.7.1 Mobile IP integration with AAA infrastructure

The existing AAA infrastructure is mainly composed of RADIUS servers and proxies mixed with some other less extended protocols like TACACS+. The IETF through the AAA working group [104] is currently defining a new AAA standard named DIAMETER.

The IETF through the Mobile IP working group (MIP WG) [118] defined the AAA requirements needed for Mobile IP [59]. In order to satisfy these requirements the IETF decided to accept the protocol currently being designed by the AAA working group to provide AAA support for Mobile IP.

Experience has shown that after a protocol is standardized several years are needed before it becomes widely used. Even providing backwards compatibility with RADIUS, as it is

included in the protocol, it might take several years before DIAMETER clearly replaces RADIUS as the leading AAA protocol. Before that happens any implementation of Mobile IP which might wish to use the existing available infrastructure will have to deal with RADIUS.

RADIUS was not design with support for Mobile IP integrated in the protocol, as DIAMETER has been. This means that many of the Mobile IP-AAA requirements defined by the IETF [59] are difficult to meet using RADIUS. This remains mainly an implementation problem and research has moved to Mobile IP integration with DIAMETER, but still the authors believe that if Mobile IP is to be used in the short term (i.e. the next one and a half years or so) in large deployments (such as current mobile operator networks) RADIUS support will have to be provided.

In the system presented in this thesis, the AAA infrastructure used is RADIUS for two main reasons. First, because there was not available open source DIAMETER implementation and second because in this way Mobile IP can be deployed **today** as it is the intention of the sponsor of this work.

### 5.7.1.1 Mobile IP-AAA architecture

In [59] the requirements and the architecture for interaction of AAA and Mobile IP were defined. We have tried to follow these as much as possible; the architecture is the same but some requirements cannot be met using RADIUS. Figure 5.16 shows the security associations as defined in that document. It is important that the Mobile Node only shares one security association, with its Home Domain (either with the AAAH or the HA).

A user could obtain an IP address from the DHCP server located at the NAS, perform a web login authentication with the browser, and finally start its mobile IP client in co-located care-of-address mode. The MN would authenticate with the HA and be provided a binding in the Home Agent's binding table, but in this case all the benefits of Mobile IP are lost since the user needs to detect a subnet change and manually relogin every time it changes from a NAS to another. This is an example of why Mobile IP-AAA integration is needed in the system. In addition two authentication processes were performed (one with the RADIUS architecture and another with the HA) which seems like a waste of time and resources.

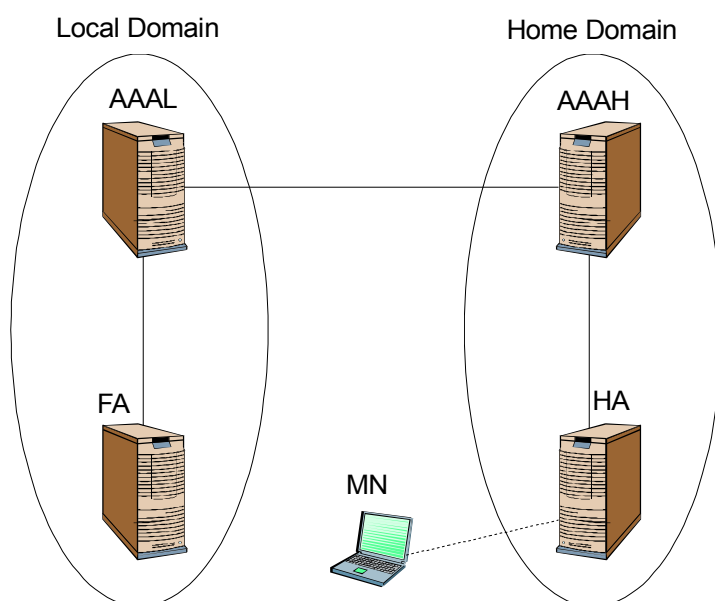


Figure 5.16 MIP-AAA Security Associations

### 5.7.1.2 Challenge-Response

The original Mobile IP specification [100] defines an optional authentication extension in the registration packets between the MN and the FA. The extension does not provide strong replay protection for the FA and poses a heavy administrative burden on large networks where numerous shared secrets have to be distributed.

In addition, a Mobile Node roaming between different administrative domains does not usually share a secret with the foreign domain and the FA cannot authenticate the user unless some extra mechanism is provided. This authentication is needed in order for an unknown client to be allowed access to the network.

Extensions for the Agent Advertisements and the Registration Request were defined in [60] to allow a FA to use a challenge/response mechanism to authenticate the MN. A new extension to provide authentication between a Mobile Node and an AAA server was also defined in the document.

The FA includes a different challenge (random value) of at least 32 bits in every Agent Advertisement using the newly defined Mobile IP Agent Advertisement Challenge extension. If the Mobile Node does not have a security association with the FA, it has to include this challenge value in its Registration Request using the Challenge/Response extension and must also include the newly defined MN-AAA Authentication Extension in order to provide authentication and integrity to the challenge.

In addition, the Mobile Node should include the Network Access Identifier extension to enable the use of the AAA infrastructure (e.g. to identify the user's administrative domain).

The FA may check the Registration Request for the Challenge extension and compare the value to previous values used by the mobile node to make sure that the MN is not trying to replay an old request. The FA only accepts the challenge if it was offered in the last successful Registration Reply issued to the MN or it has been advertised in the last agent advertisement. The number of previously issued challenges checked by the FA is defined by a parameter called the Challenge Window.

The standard states that a FA receiving a Registration Request that contains the MN-AAA Authentication extension with a correct challenge may start an authentication process using the AAA infrastructure, but does not specify the method to do it. Section 8 in that document specifies a reserved SPI for RADIUS authentication, which paves the way to use CHAP authentication [102] between the FA and the RADIUS server, allowing the calculation of the Authenticator over a maximum of 253 bytes because that is the maximum length of a RADIUS attribute.

### 5.7.1.3 AAA keys

When the MN shares one security association with its home AAA server, it is possible to use that security association to generate new security associations between the mobile and its home agent (MN-HA) and between the mobile node and the foreign agent it is currently attached to (MN-FA) [98]. In this way, only one security association needs to be shared between the mobile node and the home domain.

It could also happen that the mobile node does not have an assigned HA before hand and has to be provided one. Since it is mandatory that the MN and its HA share a security association, this one has to be provided dynamically by the AAA infrastructure.

In the case that the MN does not share a security association with its FA, it includes a MN-FA Key Request extension [99] in the Registration Request. If it does not share a security

association with its HA either, it also includes a MN-HA Key Request extension. The MN-AAA Authentication extension is added also in order to authorize and provide integrity to the previous extensions.

When the AAA server verifies the MN-AAA Authentication extension, it also generates key material for the keys requested by the mobile node. Then it fills the proper fields in the MN-FA Key Reply extension of subtype 1 (Unsolicited MN-HA Key Material from AAA) included in the Registration Reply using the key material, the algorithm information, the key lifetime, SPI assigned, and the replay protection method. The AAA keys are also distributed to the HA and the FA using the AAA protocol.

The mobile node extracts the key material and the rest of the security association's information from the Registration Reply and generates the appropriate keys that will be used to compute the MN-FA authentication extension and the MN-HA Authentication extension respectively in following registrations.

The document only describes one method to generate the key material and derive the proper keys from it. It uses MD5 in prefix+suffix mode and uses a random value of at least 64 bits as the key material. The mobile node derives the key by calculating:

$$\text{Key} = \text{MD5}(\text{AAA-key} \parallel \text{key material} \parallel \text{node-address} \parallel \text{AAA-key})$$

and indexes the key, replay method, and algorithm, using the specified SPI. Other methods might be described in later standards.

#### 5.7.1.4 Security Associations

An advanced user with Mobile IP shares two security associations with its home domain: one with the HA and the other one with the AAA infrastructure. A basic user only has one security association with the AAA infrastructure.

When a user decides that it wants to use Mobile IP, it has to be provided with a shared secret to be placed in its Mobile IP client, so it can authenticate via the AAA infrastructure.

#### 5.7.1.5 Implementation

The Mobile IP implementation used comes from the Dynamics group at the University of Helsinki (HUT) [128]. The version used in the system is 0.8.1 and it already provided some support for interaction with AAA infrastructure mainly located in the MN. Support for Challenge/Response [60] is already included in that distribution and partial support for AAA Registration Keys is also already implemented [98].

Not present yet in the Dynamics implementation are the functions needed to call the AAA infrastructure from the FA or the HA and these need to be implemented.

A MN can obtain a care-of-address through a FA or a dynamic address assignment protocol such as DHCP. The usual behaviour for a MN is to use a FA if one is available, but this is not mandated by the standard.

The FA in the NAS is configured to set the R bit in its Agent Advertisements. This bit lets the MN know that registration is required with this FA **even** if the MN is using co-located care-of-address. This is done in order to make sure that all MNs are authenticated through the RADIUS infrastructure. If an attacker tries to register directly with its HA without following the behaviour mandated by the R bit its Registration Requests are filtered at the NAS.

A decision has to be taken by the FA to start an authentication process every time it receives a Registration Request from a MN. The following conditions have to be met in the implementation in order for the AAA authentication to take place:

- ❑ No valid MN-FA Authentication Extension is present in the Registration Request
- ❑ The FA is the highest FA if a hierarchical structure of FA is being used
- ❑ The Network Access Identifier extension is included in the Registration Request
- ❑ The Challenge/Response extension is included in the Registration Request
- ❑ The MN is not already registered with that FA

If a MN already shares a secret with that FA, then no AAA authentication is needed, but that will not be the case in our system since advanced users will not be configured to share a secret with the FA belonging to the home domain. Sharing secrets between different clients is discouraged and placing a secret in each FA for each client poses too much of an administrative burden on the system compared to the benefit of reducing the amount of authentication processing performed.

If a hierarchical FA structure is in place, the lowest FA has to take care of generating and checking the values of the challenges, but the highest FA should take care of the AAA authentication. The reason for this is explained in Section 5.7.1.5. One problem detected in the Dynamics implementation of [60] is that the Highest FA does not forward the Challenge-Response extension included in the Registration Reply to the lower FA which means that the user is denied access since the challenge expected by the lowest FA is missing. Therefore, our system has been implemented with only one FA placed at the NAS; no hierarchy has been built.

Roaming agreements might be in place with other WISPs and mobile nodes coming from those administrative domains have to be allowed access to the system after they have been properly authenticated. In order to perform the authentication the AAA infrastructure has to know the home domain of the MN trying to access the resources. This information has to be provided by the MN by including the NAI extension in the Registration Request.

### Access Request

The FA has to extract information from the Registration Request in order to build the RADIUS Access Request. The following attributes are necessary in the Access Request:

- ❑ User-Name = NAI
- ❑ CHAP Password = First byte from Challenge || Authenticator
- ❑ CHAP Challenge = MD5 (Preceding Mobile IP data || Type || Subtype || Length || SPI)|| Least order 237 bytes from challenge

The NAI is extracted from the Network Access Identifier extension. The Challenge is extracted from the Challenge/Response extension. The Authenticator is extracted from the MN-AAA Authentication extension. The Preceding Mobile IP data includes all the Mobile IP fixed header plus all the extensions preceding the MN-AAA Authentication extension. The Type, Subtype, Length and SPI are all fields from the MN-AAA Authentication extension.

The Radius Server receives the request, checks the domain in the NAI and decides if the request is addressed to it or has to be forwarded to another server. In the case when the user belongs to the same domain as the RADIUS Server, it computes the MD5 hash of the first byte from the challenge, the user password stored in the database, and the CHAP Challenge and compares them with the CHAP Password attribute. If the comparison holds then an Access Accept is generated, otherwise an Access Reject is sent back to the FA.



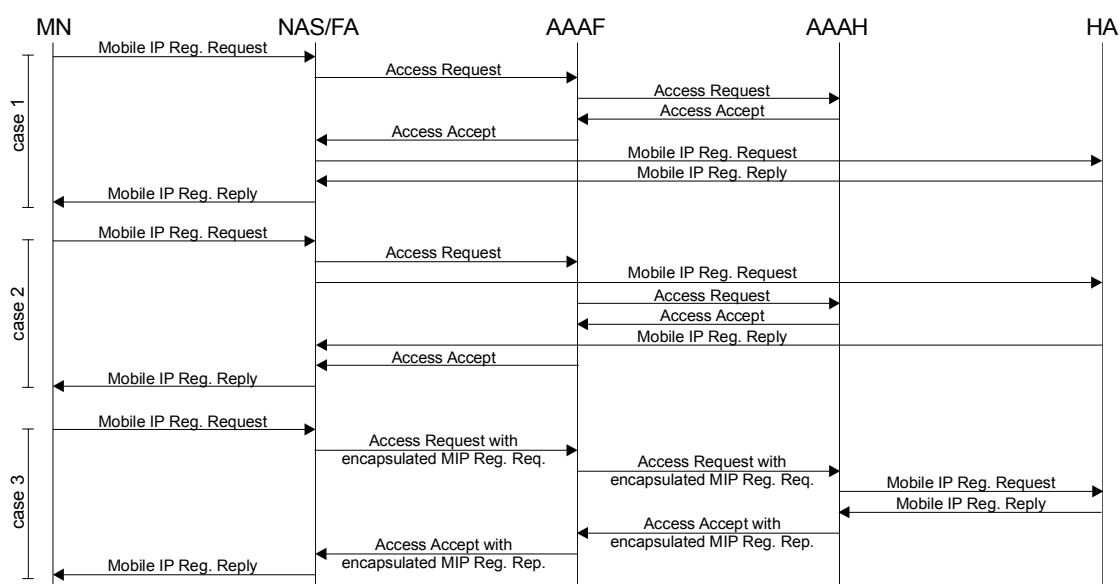
```
if (MD5 (First byte from Challenge || user password in database || CHAP challenge) ==
    CHAP Password) then Access Accept sent to NAS
```

A HA receiving a Registration Request with a MN-AAA Authentication extension or a Challenge/Response extension has to process the request even if it does not understand those extensions since the extension type is within the range 128-255 for both of them.

### Analysis of Possible Solutions

A Registration Request arriving at the FA triggers an authentication process through the AAA infrastructure. The authentication process can be integrated in three different ways:

1. The FA blocks the Registration Request, performs the authentication with the RADIUS infrastructure, and depending on the received authentication reply forwards the request to the HA or stops processing and sends a denied Registration Reply to the MN.
2. The FA starts the RADIUS authentication process and forwards the Registration Request to the HA. Both processes are executed in parallel. The FA blocks any Registration Reply for that MN until the authentication reply has been received. Depending on the reply it forwards the Registration Reply to the MN with an accept code or with a reject code.
3. The FA encapsulates the Registration Request in the authentication request (e.g. as an attribute of the Authentication Request) and waits for the authentication reply. If the Home RADIUS server accepts the authentication it has to communicate that to the HA and also send the authentication reply back to the Authentication, Authorization, Accounting Foreign (AAAF). Then the AAAF forwards this reply back to the FA and this one sends a Registration Reply back to the MN with the appropriate code.



**Figure 5.17 RADIUS-Mobile IP integration alternatives**

One of the requirements established in [59] is that all the necessary Mobile IP and AAA functions should be processed during a single Internet traversal. This requirement means tight integration of the AAA functions with Mobile IP functionality (as provided in DIAMETER) and can only be accomplished using the last proposal above.

The other two proposals need two Internet traversals, one for the authentication process and another for the Mobile IP registration. Thus, using the third solution would usually mean the

lowest delay and that means a faster registration process and that the user could access the system faster.

But that solution was discarded because it meant numerous changes to the RADIUS server in order to pass it the Registration Request and encapsulate it in an attribute. Due to the different extensions that might be present in a Registration Request, the Registration Request could become larger than 253 bytes, which is the maximum size of a RADIUS attribute, making it even harder to implement the encapsulation of the Registration Request in the Authentication Request. This could happen, for example, when the challenge value sent by the FA is large.

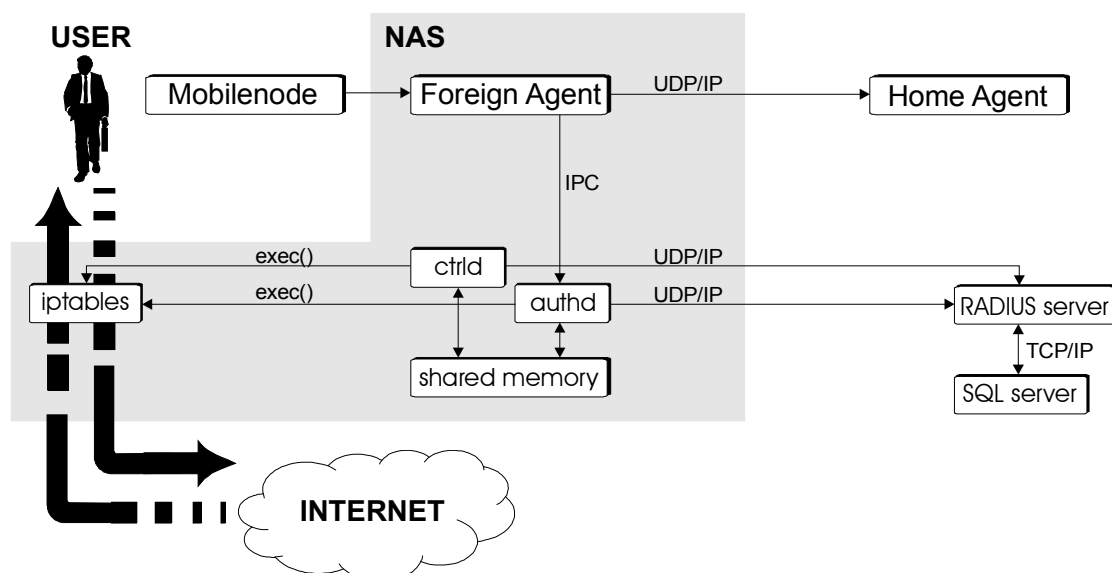
Between the other two proposals the second one is preferred because processing both requests in parallel can reduce the delay to perform the MN's authentication, but due to lack of time and because more modifications were needed in the Dynamics Mobile IP implementation, the first proposal was chosen and implemented.

There are two conclusions to all this. First, DIAMETER is needed to provide a neat AAA-Mobile IP integration since RADIUS was not designed to support this integration. Second sending the Authentication Request and Registration Request in parallel from the FA can reduce the authentication delay. This is left to future work.

#### Communication with the system

A RADIUS client is needed to perform the authentication process with the RADIUS server. This client could be integrated in the Mobile IP distribution or could be an independent process. In the latter case, Inter Process Communication (IPC) is needed between the FA and the RADIUS client since the FA has to provide all the information needed by the client to generate an Access Request.

The Dynamics Mobile IP distribution used in the system does not have an integrated RADIUS client, but the IP-Login that is the heart of the system already has one for authenticating basic users. The FA has been given IPC functionality to pass all the required information to the Authentication Daemon.



**Figure 5.18 Mobile IP communication**

When the FA decides that it needs to perform an AAA authentication of the user, it collects the necessary information and passes it to the authentication daemon using IPC. It is really the auth-d that takes care of the RADIUS communication. When the reply from the RADIUS

server is received by the auth-d, it passes that information back to the FA so it can continue with the registration process. The auth-d then places the user information on the shared memory and opens the firewall for that client.

### Key Distribution

The draft [98] is intended for use with DIAMETER and poses some difficulties to implement with RADIUS. The following is a proposal of how this could be done in the system, but it has not been implemented during this thesis.

In our system no communication has still been implemented between the RADIUS server and the HA. Thus, this solution only addresses the generation of a key between the MN and the FA that currently supports it.

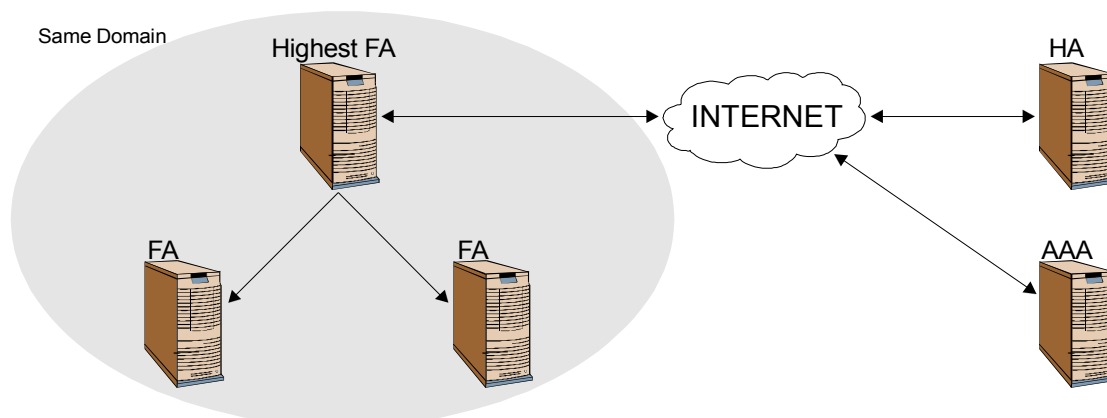
The RADIUS Access Request could include a Service Type attribute with type larger than 11 (e.g. 15) to indicate that the user is requesting Mobile IP support. Then if the authentication is granted using the data provided in the other attributes, the RADIUS server adds an attribute of a previously defined type, including a random number of at least 64 bit length, in the Access Accept sent to the FA.

### Single Authentication process

The Mobile IP AAA requirements state that the authentication process using the AAA infrastructure must be carried out whenever the user arrives at the foreign domain, but only performed once. If a MN roams between different FA belonging to the same administrative domain no further authentication should be needed. The system as it is implemented requires a new AAA authentication process whenever the MN moves to a new FA even if it belongs to the same administrative domain.

A proposed solution to meet the requirement of a single authentication process is to build a hierarchy of FAs. All the FAs belonging to the same administrative domain are structured in a tree hierarchy where only the highest FA shares a security association with the authentication server. Every other FA needs to share a security association with its upper and lower FA.

A MN moving from the area controlled by a FA to the area controlled by another FA of the same domain sends a Registration Request to the new FA as soon as it detects the movement to a new subnet. The Registration Request goes up the hierarchy to the highest FA, which already has an entry for that Mobile Node and does not need to perform another AAA process since the MN had already been authenticated.



**Figure 5.19 Hierarchical Mobile IP**

### 5.7.1.6 Further issues

The Mobile IP basic protocol [100] became a standard in 1996. It has been investigated for several years, but has not been widely deployed yet. The reasons for this might be diverse:

- ❑ No clear support for AAA infrastructure was available.
- ❑ It only provides useful functionality with long-lived applications. For short-lived applications such as HTTP it might make more sense to establish a new connection.
- ❑ Its most attractive capability: roaming between heterogeneous networks is lacking
- ❑ Deployment problems: NAT and firewall traversal.
- ❑ Mobile IP is useful for roaming between IP subnets.
- ❑ Incoming traffic initiated by a Correspondent Host (CH) also benefits.

### 5.7.2 Using cookies for mobility

This is actually not really true mobility, it's more of a way to facilitate changing of subnets connected to the same NAS or changing to a new subnet in a different NAS. When the user changes subnet, normally the user has to login again. To avoid this, a cookie will be placed in the users browser.

#### What is a cookie?

A cookie is a small file, not larger than 4 KB, which is set by adding to the HTTP header:

```
Set-Cookie: NAME=VALUE; expires=DATE; path=PATH;
domain=DOMAIN_NAME; secure
```

The cookie specification was originally written for Netscape Navigator 1.0 [82]. Cookies are a mechanism that allows the server side to store and retrieve information on the client side. The possibility to store state information at the client end significantly extends the capabilities of web-based client/server application.

When a server returns a HTTP object to the client, the server may also send a piece of state information (cookie) to be stored in the client. In the state object, information about the range of URLs that the state is valid for is included. Any future HTTP request sent by the client to the server (in that range) will include the state object (cookie).

#### How it's used

When a user first comes to the network he is requested to login, if the login was successful a cookie will be created. The information in the cookie will be the following:

- ❑ Username, e.g. steve
- ❑ Hashed password, e.g. 76329b77ee40d6fdc045f8abeb1f788d9822895a
- ❑ Expiration time, seconds since 1 of January 1970, e.g. 1293796791

Before the cookie will be placed in the user browser, it will be encrypted using Blowfish [136]. The security aspects of this are discussed in 5.5.12.

Next time the user has to login, it will be as usual redirected to the NAS web server. Before the login page is shown, there will be a check to see if the user has any cookie in the browser. If there is a cookie, an attempt to authenticate the user with the use of the information kept in the cookie will be performed. If it's successful the user will be redirected to its original requested web page. This way, the user will hardly notice the login procedure.

## 5.8 User Feedback

### 5.8.1 Requirements

The system needs a way to give feedback to the user. The user would like to have updated information about its session in order to control how much he is spending.

There are several different ways to provide feedback to the user. All of them have advantages and disadvantages. In order to be able to choose the best one, first we need to define the type of information that we need to pass back to the user.

In our system, at least the time connected and the bytes sent and received are parameters that we should pass back to the user. With this information, the user has knowledge about how expensive this session is and can take decisions about it (unfortunately after incurring the expense).

Some users might use prepaid cards. These cards could be based on time or amount of data. The decision is up to the operator, since the system can handle both. The latter might make better sense in a packet-based system like this one, but still some operator might want to provide prepaid cards based on time, since users are already used to this kind of payment.

For prepaid users, the useful information is not the bytes sent and received or even the time connected, but the time left for surfing or the bytes left to send and receive.

A subscriber to the system would probably like to know how much he is spending in that session. He might not care about the time connected or the bytes sent and received but the final amount that will be charged to him. If the feedback module has the ability to get up-to-date fares, it could display the actual cost instead of parameters such as time or bytes. It is an operator's decision what might be best suited to their business model.

Therefore, the first conclusion was that User Feedback should depend mainly on the payment method chosen by that user. That means that feedback to the user must be flexible. If you add new payment methods later on, you should be able to pass the relevant information back to the user.

### 5.8.2 Implementation alternatives

One big limitation in providing feedback to the user is that no extra software should be needed to install in the user's machine. One of the prerequisites of the whole system is that the basic user should need no special extra software.

Two different ways of implementing the feedback module were considered during this thesis: using HTML pages or using a Java Applet. Finally the Java Applet option was chosen and implemented.

Using HTML pages has the advantage of simplicity, but it provides a less flexible method. Every fixed amount of time, let's say 30 seconds, the HTML page has to be refreshed in order to provide updated information.

This method was used in Mathias Rönnblom's [9] Master's Thesis and the following problems were identified:

- ❑ Problem with caching
- ❑ Coordination problem

In that thesis it was also considered to implement a Java applet, but some problems were stated about this also:

- ❑ Java is not always supported by the client's browser
- ❑ Java is not always turned on in the client's browser
- ❑ Java can slow down the system

From our point of view, most browser include Java, most users have it activated as default, and laptops are usually powerful enough not to be slowed significantly by using Java. Java has become standard in the Internet and should be treated as such. But the main reason that HTML pages were used in Mathias' thesis was simplicity and in his conclusions Matthias states that a Java applet might have proved a better way to provide the feedback to the user.

Actually, in that thesis, the author was only worried about one way to provide the time left back to the user. As we have stated, we need to pass several other parameters back to the user such as bytes sent and received. The problem is that this kind of information is only available at the NAS. Thus some kind of communication between the client and the NAS has to be established if we want the user to get this kind of feedback.

An applet can solve this problem since it can communicate back to the server where it comes from. Because of all these reasons the applet option was chosen to provide a flexible feedback module into our system. Summarizing, we believe that the applet solution is far better than the HTML solution, providing more flexibility to the system.

One of the main security issues is that the applet can only communicate back to the server it came from. This means that in order to communicate with the NAS, which is the only part of the system that controls the traffic from the users (mainly the bytes sent and received), the applet has to be downloaded from the NAS. Since we want to download the applet to the user at the time that he logs in, it makes more sense to have the login page at the NAS instead of having it in a centralized authentication server like other system do (No-Cat [16], LANRoamer [15]).

Once we decided to use a Java applet, there were several different possibilities to make the applet communicate back to the server it comes from. For example, the applet could open a socket and try to communicate with the server it originated from. But that solution means that a server has to be listening on a fixed port on the NAS in order to take care of communication with the applet. This solution was not used mainly because of time limitations, but it provides very nice possibilities since the communication with the applet can be bi-directional. This allows for another way of performing the keep-alive function as was described in Section 5.2.10.

Instead we tried a different approach, which has already been used in several successful systems [137]. You use HTTP traffic in order to execute actions at the NAS. You just call a CGI-BIN script located at the NAS. The advantage of this approach is that all communication is done using HTTP, which means that all the system sees is just normal web traffic (to port 80). It also means that we did not need to program a server to be listening to a fixed port, because the Web Server used at the NAS will take care of communication.

In other words, when the user performs an action (e.g. pressing the Update or Logout button), the applet sends a request to the Web server through port 80.

Now, some security issues arise. Since all the users communicate with the same scripts in the NAS. How does the server know which is the correct user information that it has to pass back?

The authentication server provides a Session ID every time that a successful login is performed. This session ID is unique and generated from different parameters like the NAS address, the username, and the IP address of the user. This Session ID is passed back to the applet. So whenever the applet wants to communicate with the NAS, it has to pass this Session ID back in order to be authenticated.

The Session ID works pretty much the same as a shared secret between the NAS and the User. This way even if the Session ID is compromised, the stolen information is only valid for the duration of that session. As the rest of the applet, the Session ID is passed encrypted with SSL so cannot be easily compromised.

RMI or CORBA might be better solutions for this system, but we did not consider them as valid options because they complicated the system a lot, but only achieved a little improvement. But a further investigation into this might have different results.

The applet provides a lot of flexibility to the feedback module. Any new information that we might need to pass to the user or any new functionality that we wish to provide can be added in the applet code and then the next time the user logs in, it would download the upgraded version automatically, without thinking about downloading the new version.

### 5.8.3 Applet

A capture of the applet used to provide feedback to the user is shown in Figure 5.20. The username and IP address are passed back to the user to provide additional information. Others parameters such as MAC address, Session ID or Java version could be printed by the applet if the operator thinks it is useful.

Mainly the applet gives two buttons and some information about the current session to the user. The figure below shows the applet for a prepaid user based on time. The information about the session passed to the user is the bytes sent (870 bytes), the bytes received (34.7 Kbytes) and the time left in its prepaid card (35 minutes). For other types of user the information changes slightly.



**Figure 5.20** Capture of the feedback Applet

The Logout button allows the user to close its session. It calls the logout script located at the NAS to basically stop the accounting and close the firewall. The Update button calls the update script at the NAS and obtains information about the bytes sent and received until that

moment during the current session. This functionality could be done real-time (no Update button needed). The modifications take only some minutes but the problem is that the traffic generated by each user in order to check the NAS about its personal information periodically. Depending on the number of users and the goals of the operator this can be easily modified.

The applet also presents a link to the User Manual, as mentioned in Section 5.8.5, and a link to the password change webpage, though these are not shown in the above figure.

### 5.8.4 Password change

One extra feature is the ability to change password. This is done through an HTTPS form that calls a CGI-script, the web page is accessible from a link in the applet. The information that is requested from the user is the user-name, old-password and the new-password. The first thing the CGI-script does is hash the passwords, since no password is stored in plain text. After that the CGI script sends a SQL query direct to the SQL database, the query looks like this:

```
UPDATE radcheck SET Value = "Hashed-New-Password" WHERE UserName = "User-Name" AND Value = "Hashed-Old-Password";
```

As you see the password will not be updated if the old password is not correct. The CGI-script then checks if the query was successful, and gives appropriate user feedback.

The communication between the user and the web server will be encrypted with SSL, and between the web server and the SQL database there needs to be some sort of secure connection.

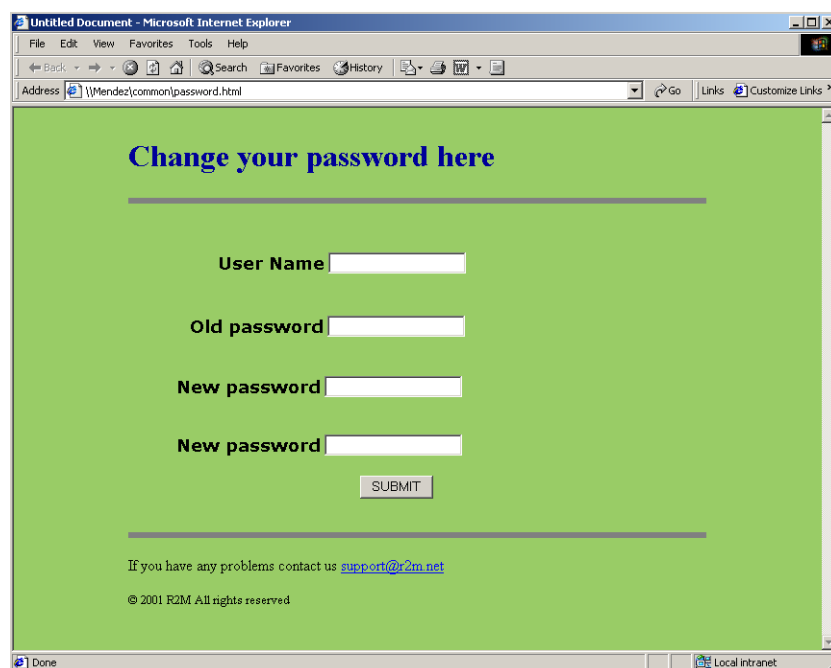


Figure 5.21 Password change webpage

To avoid having too many security associations in the system, the best place to put this webpage is probably not in the NAS. Instead the web page could be placed in a central place or perhaps in the same machine as the RADIUS server or the SQL server, this way no extra security association is needed.



A possible improvement to the password change would be to run a test over the new password provided by the user to check the strength against a possible dictionary attack. At least requiring a minimum length of 8 characters and mixing of upper and lower case characters with digits and other special characters should be considered.

### **5.8.5 Manual**

If the user should run into any problems when he is trying to use the system, there is a Frequently Asked Questions (FAQ) and a manual page that can be reached from the login page and from the applet. Using both the manual and the FAQ the user hopefully could easily get the help it needs.

## 6 Analysis

### 6.1 Analysis of the solution

In chapter 3 the system requirements were presented. In this section we will see how they have been met and discuss general reflections about our solution.

#### 6.1.1 Security

A lot of work has been done to increase the security of the system. The security can be split in different sections, the security for the user, the security in the NAS, and the security when transferring data to the central server.

##### User security

When a user uses a wireless medium to access a network, anyone within range can listen to the traffic. To ensure the confidentiality of the traffic, some sort of protection has to be in place. When using IEEE 802.11, which was considered to be the main access method in this thesis, a link layer encryption solution is available, called WEP. Although it has been shown that the WEP encryption easily can be broken [20, 72]. The solution that is provided in this thesis is an IPSec tunnel from the user to the gateway, this way all the traffic that is sent over the wireless medium will be safe. For this to work the user has to have an IPSec client installed.

##### NAS security

It's essential that no unauthorized user can login to the NAS. To ensure this we rely on the software that is running in the NAS. The important elements are Apache HTTP server, iptables, and an SSH daemon; a security hole in any of these could jeopardize the security of the NAS. Careful monitoring of security updates for all of these programs and the OS should be done.

##### Transport security

When transmitting authentication and accounting information over the Internet to the RADIUS server, the integrity and confidentiality of the data has to be ensured. In this thesis an IPSec tunnel between NAS and RADIUS server ensures this security.

#### 6.1.2 AAA

RADIUS has been used as AAA protocol, it have proven to be suitable protocol in this thesis. The wide use of it can be seen as a proof of its reliability. The implementation used, Freeradius, is still under development, but has shown to be stable and there are no reasons not to continuing using it. No modification has been made to the actual RADIUS server. Smaller modifications have been done in a module that handles the SQL queries; thus an upgrade of the server can still easily be done without requiring any major new modifications.

In RFC 3169 [113] requirements for new Network Access Servers Protocols are listed. The protocol that is used in this system (RADIUS) is a few years old and unfortunately doesn't meet all the requirements. Some of the more important problems are:

- ❑ Experience has shown that attribute space tends to run out quickly. In order to provide room for expansion in the attribute space, the AAA protocol must support a minimum of 64K attributes. RADIUS only supports 256 attributes.
- ❑ Often IP addresses are carried within the AAA protocol, therefore the AAA protocol needs to be able to carry both IPv4 **and** IPv6 addresses. RADIUS is only capable of carrying IPv4 addresses.
- ❑ Some attributes are more sensitive than others, and in a multi-domain scenario attributes may be inserted from different administrative domains. Therefore the AAA protocol must support selective encryption of attributes on an attribute-by-attribute basis. RADIUS only supports encryption of the password attribute.
- ❑ In a multi-domain scenario the AAA protocol must support end-to-end hiding of user credentials. RADIUS password encryption works on a hop-by-hop basis.

The problems listed above are all solved in DIAMETER and that is the main reason why a migration to DIAMETER is desirable.

With the use of RADIUS proxy servers, roaming functionality between different domains can be achieved. This improves the scalability of the solutions.

In our thesis we have tried to be as flexible as possible when it comes to different payment methods. In Section 3.4.4 a number of payment methods the system should support are listed. We have succeeded in supporting all of them, for the credit card payment the communication with the banking network needs still to be implemented. An operator doesn't have to support all payment methods and could choose the ones that fit their needs.

In addition to the username and password the client's MAC address is also used in the authentication. This makes it a bit harder for an attacker, as described in detail in Section 5.5.6.

### 6.1.3 Database

All the accounting data is put into a SQL database, this makes it easy to access and manipulate. In this thesis no evaluation of different billing systems has been done, but the flexibility offered by using a backend database should probably suit most billing systems.

The storage of the data is done in a mySQL [116] database, but almost any other SQL database can be used e.g. Oracle, DB2, etc. Storing data in a database is very common and experience from other fields can be reused here to ensure reliability and security.

### 6.1.4 NAS

IP-login is the main program in the NAS and it is based on the program written by Enrico Pelletta that is still in use at the KTH's IT-University in Stockholm. Although substantial changes have been made to this program, its structure has proven to work well.

The NAS doesn't hold any state information other than information about the users currently logged. In the case of an unintentional reboot, accounting information about the ongoing login sessions will be lost. In the case where the login session is very long, interim accounting request should be send to the RADIUS server, or be stored in a non-volatile memory in the NAS for later transmission to the RADIUS server. This is something that should be looked in to in the future.

### 6.1.5 User experience

Since no end user tests have been performed, it's difficult to draw any conclusions about user experience. When compared to other existing solutions that we have had the opportunity to look at, this solution provides some extra features that would probably make the user experience more pleasant, as described below.

In this thesis we have looked at two kinds of users, a basic user and an advanced user (see Section 3.4.1). The extra features that the basic user probably will appreciate, features that other solutions lack are:

- The automated redirections, first to the login page, then to the originated requested page. This will make it very easy for the user to get logged in.
- The feedback applet, where the user can get real-time information about its usage. This is especially useful when the price model is other than flat rate.

For the advanced user that makes use of mobile IP, the integration with the AAA system will probably be appreciated. If the user has configured his/her mobile IP client correctly, no additional web based login is necessary. This makes it very convenient for the user, especially if he/she changes subnets often. A drawback is that the advanced user won't be shown the applet, so he/she will not get any real-time information about usage. A way around this would be to enable the user to manually start the applet by manually entering its URL in a web browser.

### 6.1.6 Network Management

One thing that this solution lacks is network management. When setting up a system like this in a production environment network management is essential. It would be enormously inconvenient to have to go to the physical location of each NAS, every time maintenance has to be done. With good management software all the control and monitoring can be done in one place. The most commonly used protocol for network management is SNMP [95] and it should be suitable for this job.

Management was not part of the scope of this thesis, but would be an obvious element of any further work.

## 6.2 Comparison with other solutions

To give some perspective of the work done in this thesis, a short comparison with other similar systems has been done. The systems described below are just a selection of the system that are out on the market or are being developed.

### 6.2.1 Homerun

Homerun is a commercial network that has been running for approximately a year. Since Telia, the owner, doesn't want to reveal how the system works, it's very hard to make a comparison, but from the outside look and our user experience of the system, one can identify several features:

The login credential that the user supplies in a web login page is transmitted in clear text over the wireless medium. This is unacceptable and can easily be fixed by using HTTPS instead of regular HTTP. The rest of the security is hard to say anything about. Our solution in this thesis offers a higher degree of security for the user, by using SSL and the possibility to set up an IPSec tunnel from the client to the gateway.

Homerun only provides a few payment methods to the customer, if this is because of the system or if it's by choice is hard to tell. At the time of writing this report all payment methods supported were flatrate.

There is no mention about mobility in Homerun's presentation of their system. By mobility we mean the ability to keep a session alive when changing between subnets, e.g. with the use of Mobile IP.

On their strong side they have a long experience as an operator. Their access points are spread throughout Sweden and they are obviously able to administer them, they probably have a good network management system.

### **6.2.2 IT-University 2001-2002**

As explained earlier the IP-login program (Section 5.2) shares some common code with the IP-login program that is used in the IT-University in Kista [109] though some of the code is common, the rest of the infrastructure is different. For example, their firewall is on a different machine and not included in the NAS as in this thesis. The IT-University uses a Kerberos KDC instead of a RADIUS server when authenticating the user. The login part is also a bit different, as their user has to explicitly type in the address of the login web page.

No feedback is given to the user during the session, unlike the applet in this thesis. As far as the authors know their system doesn't support any accounting. That is not odd since the system is designed to be only used by teachers and students within the campus.

The system has been up and running during the autumn of 2001, and has proven to work very well with few errors. The number of users the system handles is approximately 400.

### **6.2.3 LAN Roamer [15]**

Still in an early stage, seems to need a lot of work to get the functionality that this thesis offers. During the time of this thesis the work seems to have stopped, at least their web page hasn't been updated.

Some of the many things their webpage states that they will implement, are already provided in our system:

- Use RADIUS
- SQL based accounting system
- Auto login via browser-cookie
- Add support for taking user to original web page after successful login
- Add MAC based access control when filtering the traffic with iptables

The LAN Roamer project seems to be more of a hobby, although they already have 11 access points running, most of them in private homes.

### **6.2.4 NOCAT [16]**

Still in early stage, development seems to be going slowly it doesn't seem do support any kind of accounting only authentication.

The system consists of a wireless gateway (NAS) and a central host running an authentication server and a web server. For a user to get access to the network he type in the URL of the web server, which is located in the authentication server. The user will authenticate himself by

filling in username and password via a web page. When the user is authenticated the authentication server will send a PGP signed cleartext message to the originating gateway, telling it to open the firewall for that user. The message contains username, MAC address and authentication status.

The user will be logged off after a predetermined time period, and then the user has to reauthenticate themselves.

Their solution seems to lack a lot of functionality including scalability support, strong security, and roaming. It adds some new functionality (not provided in this thesis) to allow three different classes of service (priority, co-op member and public) but the way to implement it (using the TOS bits in the IP header) is not very good. Several community networks (especially in California) are being set up using this code.

### **6.2.5 IP Unplugged [145]**

IP Unplugged, founded in Stockholm in December 1999, offers an interesting commercial package since the second quarter 2001. They provide software to the end user (Mobile Client Software for Windows 2000) and both hardware and software to the operator (Mobile Service Router and Mobile Service Manager) in order to allow mobility across heterogeneous networks for operators and VPN solutions to corporations.

The main component is the Mobile Service Router (MSR), which is a router with support for Mobile IP. It can act as both HA and FA. Routing protocols (e.g. OSPF) are still not implemented.

They have implemented a proprietary solution to solve the NAT traversal problem that affects Mobile IP [132]. This is the only available solution of the problem that we are aware of, but poses the big inconvenience of a high price.

The end user software gives the user the opportunity to change between different interfaces and keep the session running, e.g. change to a GPRS network when connection to the WLAN is broken. Mobile IP is used for the mobility part; a secure connection to the HA can be set up with IPSec. All this can be controlled with a nice user interface.

Instead of using RADIUS they are developing their own DIAMETER version, which they plan to use in conjunction with Mobile IP. This might be the right approach as has already been mentioned.

Management through a web interface is provided through their Mobile Service Manager (MSM 200). It is software, which runs on a Solaris host and allows configuring interfaces in the system with addresses, NAT, filtering rules, policies, security associations, and others.

## **6.3 Evaluation of the capacity of our system**

To give an idea on what the system can handle in terms of users and load, an analysis of the capacity has been done. One should keep in mind that the system hasn't been tested in a live environment with real users. The estimation is more on a theoretical level, and in a live environment the results could be different.

### 6.3.1 Client

No specific software has been done for the client. It depends only on the software and the hardware each client already has. The client has to share the bandwidth resource with other clients.

### 6.3.2 NAS

The resources needed by the NAS are related to what kinds of services the user requires. The most capacity consuming individual service is by far the IPsec encryption and decryption. The requirements of the NAS will probably be dominated by the resources required for a number of users to be using IPsec. Beside IPsec there are other programs running in the NAS, evaluations of the important ones are listed below.

#### IPsec

The throughput performance of the NAS could in worst case be drastically reduced, if the demand for IPsec encryption is high from the clients. This of course depends on what machine is used as a NAS.

In this thesis the FreeSwan [112] implementation of IPsec has been used. In the documentation that comes with FreeSwan distribution a theoretical estimation of what CPU speed is necessary for different throughputs is provided. The estimation is based on how many clock cycles are required for each processed byte (Pentium II clock cycles). Their estimate is that it requires 140 cycles / byte, with the triple DES encryption the dominant part with 116 cycles/byte. With this estimate, to saturate a link with the capacity  $C$  Mbit/s would require a processor with the speed:  $C*140/8 = C*17.5$  MHz. This estimation is not precise and ignores several factors. A more realistic formula would be something like  $C*25$  MHz. Table 6.1 shows some example of what kind of machine would be required to saturate different network connections.

Interface		Machine speed in MHz		
Type	Mbit/s	Estimated speed	Minimum IPsec gateway	Minimum with other load (e.g. firewall)
DSL	1	25	whatever you have	133, or better if you have it
Cable	3	75		
Ethernet	10	250	surplus 266 or 300	500+
Full duplex Ethernet	20	500	500	800+
T3 or E3	45	1125	1200	1500+
Bi-directional T3/E3	90	2250	(Not feasible in software on current machines)	
Fast Ethernet	100	2500		

**Table 6.1 IPsec capacity**

The estimation said that it matches real tests reasonably well, and could be seen as a guide when planning a system. More data about the performance of Free S/Wan can be read in the documentation of Free S/Wan [112].

#### Mobile IP Foreign Agent

Requirements for the Mobile IP FA are very low and it can be run in almost any host. It has been run successfully in 486/100 hosts with 8 MB of RAM. Detailed analysis of the capacity required by this agent are not available but we believe they are not a limiting factor.

### Iptables

Iptables is a well-used program, used in many environments. From information taken mainly from the iptables mailing list [96], normal usage doesn't require much CPU power or memory. Some questions have been raised about the speed of adding new rules when a lot of rules already are present. This occurs when the number of rules approaches 50 000. This scenario is not what iptables was designed for, and in these cases other solutions would be preferable.

In our case a rough estimate of the number of simultaneous users in one NAS are most probably less than 1000. Each user generates three rules, which means that the total number of rules won't normally exceed 3000. Being on the safe side, we can say that the system should not require more than 5000 rules at any time. And on this level there are no indications that the time to insert new rules would be a problem but this might need further checks.

### IP-login

The two daemons don't require significant processing power or memory usage, so a very modest computer could do the job. A more important question is the stability of the program. In this thesis we haven't had the chance to test the system in a real deployment, but since the structure of the code is shared with the IP-login system, used at the IT-University [109], some parallels can be drawn. At the IT-University the system has been used throughout the whole autumn of 2001, with approximately 400 users listed. Everything has seemed to work fine with almost no errors. In this thesis we have of course changed the original code a lot, but the structure of the two daemon and the different scripts are the same.

Due to lack of time and lack of hardware thorough tests have not been done. Based on the authors experience of using the program and the similarities with the system used at the IT-University in Kista, the program is considered to be reasonable stable and efficient.

### ARP bandwidth usage

Each time a user is probed to see if it is still in the network, two ARP packets are generated: one request and one reply. An important question is how much bandwidth this keep-alive method consumes. Of course this depends on the number of users that are currently logged in and how often ARP is used.

The size of both the request and the reply packet is 60 bytes (including the Ethernet frame header), so each keep-alive check uses 120 bytes. If there for example are 100 users logged in and the keep-alive check is done every 60 second, then the keep-alive check would transfer 12 Kb of data per minute or 0.2 Kbyte/sec. This is of course a theoretical estimation; in reality it would probably be more since some ARP messages need to be sent more than once.

## **6.3.3 RADIUS server**

In a large system the RADIUS server could obviously be a weak link, every NAS in the system will send authentication and accounting request to the same machine. The scalability of the system is significantly limited by the capacity of the RADIUS server. The RADIUS protocol is quite simple and for that reason each request can be handled quickly.

As said earlier the RADIUS server used is Freeradius, and is available under GNU General Public License (GPL). No figures about the capacity of Freeradius have been found, primarily because the results heavily depend on what hardware, OS, and the many configuration parameters used. To get an idea of the capacity, a few simple tests have been run.



First 10 000 users were created, and were put in the database and to a file that could be used as input to a standalone RADIUS client. The RADIUS client and the SQL database were located on the same host; the RADIUS server was located on a different machine. Both computers were connected to the same 10 Mbit/s Ethernet LAN. In reality the RADIUS client would be a NAS located on a different LAN. The machine used for the RADIUS server was an Intel Pentium III 500 MHz with 128 MB memory. The other machine used for the SQL server and for the RADIUS client, is the same machine described in Appendix B.

The RADIUS client's job is to send as many RADIUS requests as possible. The client takes information about the first username and password from a file, after the client gets a reply from the RADIUS server it proceeds to the next user until all 10 000 user have been authenticated. This way the RADIUS server will only receive one request at a time. In reality it's more likely that multiple requests will be received at the same time another request is handled. To test this, four different RADIUS clients were started the same time, all four authenticate 2500 user each. The time for the RADIUS client/clients to finish all requests were measured. The results can be seen below:

	<b>Total time 10 000 users</b>	<b>Average time per authentication</b>
<b>1 RADIUS client running</b>	64.5 s	6.45 ms
<b>4 RADIUS clients running</b>	35.0 s	3.50 ms

**Table 6.2 RADIUS capacity**

A similar test was run, but this time an accounting request was sent. Accounting requests are a little bit different than authentication requests. In authentication requests the password is encrypted and has to be decrypted, and when an accounting START request is received a new entry in the database has to be created. The differences were not large and the time that was measured was more or less the same.

This test should be taken with some reservations. Several parameters have not been taken into consideration, e.g. memory usage, CPU load, network throughput, etc. The Freeradius server gives the opportunity to adjust different parameters to optimise the performance, e.g. number of threads running the same time, number of database handles, etc.

### 6.3.4 SQL server

The SQL database is used to store accounting information. Each session generates a new entry in the database and the size of each entry is approximately 530 bytes. The size of each session record can significantly be reduced in size by removing information you don't want to store (see Section 5.4.2). In this thesis we use the open source database MySQL [116] on a Linux x86 platform, kernel 2.4.8 and the file system used is ext2. The maximum size of the database in MySQL in this case is 2 GB, depending on OS and what file system used the maximum size could reach 4-8 GB [74]. So the maximum number of stored sessions for us would approximately be 4 million. If the demands are larger, then there are commercial database alternatives that can be used, this out of the scope of this thesis.

The database is also used for storing authentication data and reply attributes. The size of this data is probably considerable smaller than for the accounting data. Both in the authentication case and for the accounting case the time it takes for each query is probably most important along with how many simultaneous requests can be handled. Unfortunately this is hard to estimate, much of this is depend on what hardware is used and what software was installed. With the right configuration this should not be the weakest link in the system.

## 7 Conclusions

### 7.1 Meeting our goals

The main goal of this thesis was to build and evaluate a prototype of a public WLAN access network and especially the NAS needed at hotspots. This goal has been achieved and the functionality needed to provide public WLAN access to a basic user as defined in the requirements is working well. Some work remains to be done to provide advanced users with Mobile IP, Kerberos and IPSec support though a large step has been done towards that goal.

The following topics composed the scope of this thesis: accounting, security, user-friendliness, mobility, and multi-access. All of them have been studied, but the results vary from one to another.

#### Accounting

Support for flexible accounting and specifically for those payment methods defined in the requirements has been implemented and this part of the thesis is considered fully completed.

#### Security

Major improvements compared to other presented systems have been done in the area of security. All communications with the NAS have been secured, passwords are kept only in hashed form, cookies are protected, and Access control has been implemented using two types of authentication: RADIUS and Kerberos. Security of the wireless link has been studied and an IPSec solution, similar to other commercial solutions, has been set up and studied. Also a new protocol IEEE 802.1x has been analysed to see how it could raise the overall security.

#### User-friendliness

A significant part of the thesis was spent on providing an easy way for the user to login, and providing feedback to allow the user to get information about their session. Avoiding unnecessary logins has also been implemented with a cookie mechanism and new solutions for the keep-alive module have also been presented. This part of the work is considered also fully completed.

#### Mobility

Integration of the RADIUS architecture with the Mobile IP distribution has been accomplished. Handover between WLAN and GPRS has not been implemented, but the road to implementing it has been clearly drawn.

#### Multi-access

Some problems have been detected in implementing multiaccess with the Dynamics Mobile IP distribution, but no solution has been implemented. This area remains to be carefully studied.

Several different solutions for public WLAN access networks have been analysed and integration of the different functionalities found into a single solution has been tried and mostly accomplished.

### 7.1.1 Fulfilling the system requirements

The system requirements were presented in Section 3.4. Most of the requirements marked as necessary (e.g. with a **must** or a **should**) have been achieved. Analysis of the capacity of the

system has only partially been accomplished and how to provide redundancy in the system remains to be studied.

Among the optional functionality (e.g. marked with a **may**) some remains to be done. Below is the degree of fulfillment for each requirement in more detail presented.

The grading can be one of these:

- ✓ Completely Fulfilled
- 1/2 Partly Fulfilled
- ✗ Not Fulfilled at all

#### 7.1.1.1 General

- ✓ It **must** be possible to run different combinations of the NAS modules.
- ✓ Connectivity to outside networks (e.g. Internet) **must** only be provided through the NAS.
- ✓ The system **must** at least support two kinds of clients:
  - The basic user is only required to
    - A laptop or handheld device with an IEEE 802.11b compliant interface,
    - An OS with TCP/IP support and a DHCP client, and
    - A browser with HTTPS support (SSL)
  - The advanced user, needs to have or to be provided with:
    - IPsec client software,
    - Mobile IP client, and
    - Kerberos V client
- ✓ The user **should** be able to choose its WLAN adapter. The user **may** need to notify the system a change in adapter.
- ✗ Redundancy requirements **should** be studied and **may** be implemented in order to allow a fault-protected system.
- 1/2 An analysis of the capacity of the system **should** be performed after the implementation is finished. That means being able to measure how many users can the system handle.
- ✓ Roaming with another operators **should** be included in the design of the system.
- ✗ The advanced user **may** optionally have a GPRS subscription in order to use wide area mobility though the wireless operator could provide this subscription.
- ✗ Each user **may** have a User Profile, which contains all the information about him and can be retrieved by some external means.
- ✓ The system **may** provide a method for users to securely update their User Profiles, at least the Password and the MAC address.
- ✗ The system **may** provide support for local services.

#### 7.1.1.2 Login

- ✓ The user **must** be identified by a unique user-name.

- ✓ The user **must** not be able to start several sessions simultaneously.
- ✓ After authenticating itself, the user **must** be redirected to the webpage first requested.
- ✓ The system **should** not require the user to log in more than once for a session, for its convenience.
- ✓ The system **should** provide feedback to the user about the status of its connection.
- ✓ Unsuccessful logins **should** be recorded for security purposes.
- ✓ A User Manual **may** be implemented and be accessible from the applet.
- ✗ After a fixed number (e.g. 5) of unsuccessful logins the system **MAY** block that account for several minutes (e.g. 10).

#### 7.1.1.3 Security

- ✓ The user **must** be able to authenticate the system before revealing his username and password. In other words, mutual authentication is needed.
- ✓ Filtering at the firewall **must** be done on both IP and MAC to avoid IP address spoofing.
- ✓ It **must not** be possible for any non-root user to change firewall rules.
- ✓ The system **must** be able to close the session when the user stops using it. A manual logout **MUST** exist but **should** not be the only procedure. A keep-alive procedure **should** be implemented.
- ✓ Protection from IP and MAC spoofing **should** be provided.
- ✗ WEP **should** only be used if keys are dynamically changed quite often. Even in that case it **MAY** not be the primary protection in the wireless segment.
- ✓ The system **should** have the possibility to control MAC addresses (possibly at access points) in authorization but this is disabled by default.
- ✓ The status of the NAS **should** be logged.
- 1/2 Passwords **should** never be stored in clear text. They **should** be hashed when provided and stored in hashed form in the database. The incoming requests **should** compare hashed passwords.
- ✗ Internal traffic between clients on the wireless network **may** need to be controlled.
- ✓ All unnecessary accounts **may** be removed and all unused ports closed at the NAS.

#### 7.1.1.4 Accounting

- ✓ The system **must** support different payments methods.
- ✓ The system **should** allow late additions of new payment methods.
- ✓ At least the following payment methods **must** be supported by the system.

**Prepaid** - The user pays in advanced for a limited amount of time or data to be transferred. When the time or data paid for is up, the user is blocked.

- Information the user has to provide: username, and password
- Information stored in the database: username, time connected, time left, bytes left to send, and bytes left to receive

- Authentication based on: username, password, and MAC address

**Credit card** - The user pays with its credit card.

- Information the user has to provide: name, credit card number, expiration date, and type of card (e.g. VISA or MasterCard)
- Information stored in the database: credit card number, time connected, bytes sent, and bytes received
- Authentication based on: Credit Card information, and MAC address

**Test** - The user gets a chance to test the service for five minutes, but only once for each MAC address.

- Information the user has to provide: none
- Information stored in the database: MAC address, and time connected
- Authentication based on: MAC address

**Subscription** - The user gets billed every fixed amount of time (e.g. at the end of each month) for the time connected or amount data transferred during that period.

- Information the user has to provide: username, and password
- Information stored in the database: username, time connected, bytes sent, bytes received, bank account, bank name, name, and person number
- Authentication based on: username, password, and MAC address

**Invoice** - Like a temporary subscription, e.g. the user stays at a hotel and just wants to use the service during that visit. The user can get the amount to be paid in the hotel's bill.

- Information the user has to provide: username, and password
- Information stored in the database: username, time connected, bytes sent, bytes received, address, name, and person number
- Authentication based on: username, password, and MAC address

## 7.2 Suggestions & Lessons Learned

We would like to suggest the following to a researcher who continues with this work:

- ❑ Limit the scope of the problem.
- ❑ Do not underestimate the importance of a good literature study.
- ❑ Specific requirements help to speed up the implementation phase.
- ❑ Do not make too optimistic a time schedule.
- ❑ If a testbed needs to be set up, start doing it as early as possible as many unforeseen problems usually appear.
- ❑ Most of the problems encountered are not new and some old solutions might still be valid.
- ❑ Do not forget that your code has to be understood by the next researcher.

## 8 Future work

This section presents the future work to make the solution presented in this thesis into a commercial system and also further areas of research for public WLAN area networks.

### 8.1 System upgrades

The following improvements could be added to the system:

- ❑ Fix the parsing of the IP-login configuration file to work with all the required parameters. No configuration information should be embedded in the code.
- ❑ To speed up dynamic insertion and removal of rules to the iptables firewall, the libiptc library should be used instead of a system call.
- ❑ A soft configuration reload should be added so there is no need to restart the daemons when the configuration information needs to be changed.
- ❑ An installation procedure needs to be developed.
- ❑ Further tests in a more realistic environment (e.g. using Access Points) are needed. Some extra effort should be made in realistic capacity tests of the system.
- ❑ Add Network Management to the system. Remote configuration of the NAS, Authentication Servers, Database and Access Points is needed for commercial deployment.
- ❑ Client software could increase the overall security and usage of the system. It could replace the applet as feedback module and handle both the login and the keep-alive functionality.
- ❑ Coding of the UDP-IP encapsulation and integrating it with the Mobile IP distribution to allow handovers between GPRS and WLAN.
- ❑ Substitute the RADIUS infrastructure for DIAMETER infrastructure whenever an implementation of this one is available. Another option is to code a DIAMETER implementation.
- ❑ All types of duplicated logins (e.g. using Kerberos) should be prevented. The NAS could be checked using SNMP when a duplicate login is suspected.
- ❑ Implement interim accounting records for long lived sessions.
- ❑ The applet could be signed to allow give it extra privileges.
- ❑ A new keep-alive solution with no probing and based in an applet as mentioned in Section 5.5.13
- ❑ RMI or CORBA could be added as mentioned in Section 5.8
- ❑ Other RADIUS-Mobile IP integration solutions as mentioned in Section 5.7.1.5
- ❑ Allowing a FA hierarchy as stated at the end of Section 5.7.1.5

## 8.2 Other areas of investigation

Further investigation is needed in the following areas:

- ❑ Multiaccess
- ❑ Integration of 802.1x in public WLAN access networks
- ❑ Mobility support in applications
  - Adaptation of the service to the underlying access network
- ❑ Ad-Hoc networking
- ❑ Interference in the 2.4 and 5 GHz bands
- ❑ Shared access networks
- ❑ Fast Handover with MobileIPv4 (including 802.11)
  - Minimizing handover times when changing subnets connected to the same NAS and specially the handover to GPRS networks is important in this system. Handover between different Network Access Servers is probably is less important.

## 8.3 Related research groups

### IETF

- ❑ AAA Working Group (aaa) [104]
- ❑ Network Access Server Requirements (NASREQ) Working group [117]
- ❑ Mobile IP Working Group (mip) [118]
- ❑ IP Security Protocol Working Group (ipsec) [119]
- ❑ Point-to-Point Protocol Extensions Working Group (pppext) [120]
- ❑ IP Security Remote Access Working Group (ipsra) [121]
- ❑ Mobile Ad-hoc Networks Working Group (manet) [122]

### IEEE

- ❑ 802.1X – Network Port Authentication [147]
- ❑ 802.1w – Spanning tree rapid convergence [124]
- ❑ 802.11e – Quality of Service [123]
- ❑ 802.11f – Inter-Access Point Protocol [123]
- ❑ 802.11i – Extended security [123]

### ETSI

- ❑ Broadband Radio Access Networks Group (BRAN) [125]

## 9 References

- [1] Johan Ervenius and Filip Tysk. "Dual-mode Capability in a WLAN-equipped PC for Roaming and Mobility between WLANs and GPRS Networks". Master's thesis, Department of Microelectronics and Information Technology (IMIT), Royal Institute of Technology (KTH). February 2001.
- [2] Juan Caballero, Kenneth Sundström, Jurgen Klein, Claes Rosenberg, Shelley Wu, Kamil Mohummad, and Yong Jiang. "TurnCoat Project: An Investigation into Intersystem Handover". Technical Report, Department of Microelectronics and Information Technology (IMIT), Royal Institute of Technology (KTH). May 2001.
- [3] O. H. Levkowitz, J. Forslow, and H. Sjostrand. "NAT Traversal for Mobile IP using UDP Tunnelling". Internet-draft (Work in progress) <draft-levkowitz-mobileip-nat-tunnel-00.txt>. July 2001.
- [4] Ralf Schmitz. "Seamless handoff in Mobile IP using Simultaneous Bindings". Diploma thesis, Department of Communications Engineering, University of Applied Sciences, Cologne. August 2000.
- [5] MosquitoNet project, <http://mosquitonet.stanford.edu>, accessed September 2001.
- [6] Xinhua Zhao, Claude Castelluccia, and Mary Baker. "Flexible Network Support for Mobile Hosts". Mobile Networks and Applications (MONET) Special Issue on Management of Mobility in Distributed Systems, volume 6, number 2. March/April 2001.
- [7] Fabio Moiola. "Security in Public Access Wireless Networks". Master's Thesis, Department of Microelectronics and Information Technology (IMIT), Royal Institute of Technology (KTH). May 2000.
- [8] R. Droms and W. Arbaugh, editors. "Authentication for DHCP Messages". RFC 3118. June 2001.
- [9] Mathias Rönnblom. "A Network Access Server for controlling air time in a public wireless LAN environment". Master's Thesis, Department of Microelectronics and Information Technology (IMIT), Royal Institute of Technology (KTH). June 2001.
- [10] Guido Appenzeller, Mema Roussopoulos, and Mary Baker. "User-Friendly Access Control for Public Network Ports". Proceedings of IEEE INFOCOM'99. March 1999.
- [11] A. Escudero, B. Pehrson, E. Pelletta, J.O. Vatn, and P. Wiatr. "Wireless access in Kista - IT University: MobileIPv4 integration in a IEEE 802.11b". 11th IEEE Work-shop on Local and Metropolitan Area Networks. LAN-MAN2001. March 2001.
- [12] StockholmOpen, <http://www.stockholmopen.net>, accessed September 2001.
- [13] Telia Homerun, <http://www.homerun.telia.com>, accessed September 2001.
- [14] "IP Mobility vs. Session Mobility", White Paper, Columbitech, 2001.
- [15] SoHo wireless, <http://www.lanroamer.com>, accessed September 2001.
- [16] No Cat project, <http://nocat.net>, accessed September 2001.



- [17] WECA, <http://www.wirelessethernet.org>, accessed September 2001.
- [18] Nikita Borisov, Ian Goldberg, and David Wagner. "Intercepting Mobile Communications: The Insecurity of 802.11" Proceedings of the 7<sup>th</sup> Annual International Conference on Mobile Computing And Networking. July 2001.
- [19] Scott Fluhrer, Itsik Mantin, and Adi Shamir. "Weaknesses in the Key Scheduling Algorithm of RC4". Proceeding at the Selected Areas in Cryptography, SAC'2001. August 2001.
- [20] Adam Stubblefield, John Ioannidis, and Aviel D. Rubin. "Using the Fluhrer, Mantin, and Shamir Attack to Break WEP". AT&T Labs Technical Report TD-4ZCPZZ. August 2001.
- [21] Martin Johnsson. "Hiperlan/2- The Broadband Radio Transmission Technology Operating in the 5 GHz Frequency Band". White Paper, HiperLAN/2 Global Forum, 1999.
- [22] RadioLan, <http://www.radiolan.com>, accessed September 2001.
- [23] R. Droms. "Dynamic Host Configuration Protocol". RFC 2131. March 1997.
- [24] C. Finseth. "An Access Control Protocol, Sometimes Called TACACS". RFC 1492. July 1993.
- [25] C. Rigney, S. Willens, A. Rubens, and W. Simpson. "Remote Authentication Dial In User Service (RADIUS)". RFC 2865. June 2000.
- [26] C. Rigney. "Radius Accounting". RFC 2866. June 2000.
- [27] C. Rigney, W. Willats, and P. Calhoun. "RADIUS Extensions". RFC 2869. June 2000.
- [28] P. R. Calhoun, G. Zorn, P. Pan, and H. Akhtar. "Diameter Framework Document". Internet-draft (Work in progress) <draft-ietf-aaa-diameter-framework-01.txt>. March 2001.
- [29] P. R. Calhoun, G. Zorn, H. Akhtar, J. Arkko, E. Guttman, and A. C. Rubens. "Diameter Base Protocol". Internet-draft (Work in progress) <draft-ietf-aaa-diameter-07.txt>. July 2001.
- [30] P. R. Calhoun, J. Arkko, and G. Zorn. "Diameter Accounting Extensions". Internet-draft (Work in progress) <draft-ietf-aaa-diameter-accounting-01.txt>. March 2001.
- [31] FreeRadius, <http://www.freeradius.org>, accessed September 2001.
- [32] Cisco. "Single-User Network Access Security TACACS+", <http://www.cisco.com/warp/public/614/7.html>, accessed September 2001.
- [33] GNU General Public License, <http://www.gnu.org/copyleft/gpl.html>, accessed September 2001.
- [34] Mobile IP working group, <http://www.ietf.org/html.charters/mobileip-charter.html>, accessed September 2001.
- [35] SPINACH, [http://mosquitonet.stanford.edu/spinach/spinach\\_info/index.html](http://mosquitonet.stanford.edu/spinach/spinach_info/index.html), accessed September 2001.

- [36] Department of Microelectronics and Information Technology, KTH, <http://www.imit.kth.se/>, accessed September 2001.
- [37] ANSI/IEEE Std 802.11-1999. "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications". ISBN 0-7381-1658-0. Institute of Electrical and Electronics Engineering, Inc. March 1999.
- [38] Hiperlan2 standard, <http://www.hiperlan2.com>, accessed September 2001.
- [39] Bluetooth standard, <http://www.bluetooth.com>, accessed September 2001.
- [40] Nokia C110/111 wireless LAN card, [http://www.nokia.com/networks/wireless\\_lan/hl\\_c110.html](http://www.nokia.com/networks/wireless_lan/hl_c110.html), accessed September 2001.
- [41] Ulf Gustafson and Jan Forsl w. "Network design with mobile IP". Proceedings at INET2001. June 2001.
- [42] B. Clifford Neuman and Theodore Ts'o. "Kerberos: An Authentication Service for Computer Networks". IEEE Communications Magazines. September 1994.
- [43] Senthil Sengodan and Raj Bansal. "Use of RSIP within GPRS Networks". Paper at INET2001. June, 2001.
- [44] Bernard Aboba. "IPSec-NAT Compatibility Requirements". Internet-draft (Work in progress) <draft-ietf-ipsec-nat-reqts-00.txt>. June 2001.
- [45] W. Dixon, B. Swander, A. Huttunen, J. Sierwald, V. Volpe, L. DiBurro, M. Stenberg, and T. Kivinen. "IPSec over NAT Justification for UDP Encapsulation". Internet-draft (Work in progress) <draft-ietf-ipsec-udp-encaps-justification-00.txt>. June 2001.
- [46] W. Dixon, A. Huttunen, B. Swander, T. Kivinen, M. Stenberg, V. Volpe, and L. DiBurro. "UDP Encapsulation of IPSec Packets". Internet-draft (Work in progress) <draft-ietf-ipsec-udp-encaps-00.txt>. June 2001.
- [47] T. Kivinen, M. Stenberg, A. Huttunen, W. Dixon, B. Swander, V. Volpe, and L. DiBurro. "Negotiation of NAT-Traversal in the IKE". Internet-draft (Work in progress) <draft-ietf-ipsec-nat-t-ike-00.txt>. June 2001.
- [48] S. M. Shahrier. "Incorporating NAT boxes in Mobile IPv4". Internet-draft (Work in progress) <draft-shahrier-mobileip-nat-00.txt>. May 2001.
- [49] C. Perkins, editor. "IP Mobility Support for IPv4, revised". Internet-draft (Work in progress) <draft-ietf-mobileip-rfc2002-bis-06.txt>. June 2001.
- [50] Pawel Wiatr. "Authentication, Authorization and Accounting services, security and monitoring issues in the IT-University wireless network". Pre-graduation report submitted to National Institute of Applied Sciences at Lyon, France. February 2001.
- [51] G tz Brasche and Bernhard Walke. "Concepts, Services, and Protocols of the New GSM Phase 2+ General Packet Radio Service". IEEE Communications Magazine. August 1997.
- [52] Brian P. Crow, Indra Widjaja, Jeong Geun Kim, and Prescott T. Sakai. "IEEE 802.11 Wireless Local Area Networks". IEEE Communications Magazine. September 1997.

- [53] Mobile Computing Group at Stanford University. "MosquitoNet Mobile IPv4: User's manual".
- [54] R. Droms. "Dynamic Host Configuration Protocol". RFC 2131. March 1997.
- [55] Y. T'Joens, C. Hublet, and P. De Schrijver. "DHCP reconfigure extension". RFC 3203. December 2001.
- [56] Marika Mattila. "Secure Communication in Mobile Internet". Master's Thesis, Department of Microelectronics and Information Technology (IMIT), Royal Institute of Technology (KTH). February 2001.
- [57] B. Aboba and M. Beadles. "The Network Access Identifier". RFC 2486. January 1999.
- [58] P. Calhoun and C. Perkins. "Mobile IP Network Access Identifier Extension for IPv4". RFC 2290. March 2000.
- [59] S. Glass, T. Hiller, S. Jacobs, and C. Perkins. "Mobile IP Authentication, Authorization, and Accounting Requirements". RFC 2977. October 2000.
- [60] C. Perkins and P. Calhoun. "Mobile IPv4 Challenge/Response Extensions". RFC 3012. November 2000.
- [61] L. Blunk and J. Vollbrecht. "PPP Extensible Authentication Protocol (EAP)". RFC 2284. March 1998.
- [62] Jamie Jaworski and Paul J. Perrone. "Java Security Handbook". Sams Publishing. September 2000. ISBN 0-672-31602-1.
- [63] P. Congdon, B. Aboba, T. Moore, A. Palekar, A. Smith, G. Zorn, D. Halasz, A. Li, A. P. Young and J. Roes. "IEEE 802.1x RADIUS Usage Guidelines". Internet-draft (Work in progress) <draft-congdon-radius-8021x-17.txt>. November 2001.
- [64] Jon-Olov Vatn and Gerald Q. Maguire Jr., "The effect of using co-located care-of addresses on macro handover latency". Fourteenth Nordic Tele-traffic Seminar (NTS 14). August 1998.
- [65] B. Aboba, J. Arkko, and D. Harrington, "Introduction to Accounting Management". RFC 2975. October 2000.
- [66] B. Aboba and J. Vollbrecht, "Proxy Chaining and Policy Implementation in Roaming". RFC 2607. June 1999.
- [67] K. Narayan, "Radius Security Extensions using Kerberos V5", Internet-draft (Work in progress) <draft-kaushik-radius-sec-ext-06.txt>, July 2001.
- [68] T. Wu. "The SRP Authentication and Key Exchange System". RFC 2945. September 2000.
- [69] T. Dierks and C. Allen. "The TLS Protocol". RFC 2246. January 1999.
- [70] U.S. Department of Commerce. "Advanced Encryption Standard (AES)". Federal Information Processing Standard (FIPS) Publication 197. November 2001.
- [71] AirSnort project, <http://sourceforge.net/projects/airsnort/>, accessed November 2001.

- [72] WepCrack project, <http://sourceforge.net/projects/wepcrack/>, accessed November 2001.
- [73] THC-RUT, <http://www.thehackerschoice.com/releases.php?s=8&q=>, accessed November 2001.
- [74] MySQL documentation “1.2.4 How Big Can MySQL Tables Be?”, [http://www.mysql.com/doc/T/a/Table\\_size.html](http://www.mysql.com/doc/T/a/Table_size.html), accessed December 2001.
- [75] B. Aboba and D. Simon. “PPP EAP TLS Authentication Protocol”. RFC 2716. October 1999.
- [76] Jonathan Trostle, Michael Swift, Bernard Aboba, and Glen Zorn. “Initial and Pass Through Authentication Using Kerberos V5 and the GSS-API (IAKERB)”. Internet-draft (Work in progress) <draft-ietf-cat-iakerb-08.txt>. September 2001.
- [77] L. Blunk, J. Vollbrecht and B. Aboba. “Extensible Authentication Protocol (EAP)”. Internet-draft (Work in progress) <draft-ietf-pppext-rfc2284bis-01.txt>. November 2001.
- [78] T. Raivisto. “Applying Cryptography to GSM Short Message Services”. Master’s thesis, Helsinki University of Technology (HUT). October 1997.
- [79] IEEE Std 802.11b-1999 (Supplement to ANSI/IEEE Std 802.11, 1999 Edition). “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. Higher-Speed Physical Layer Extension in the 2.4 GHz Band”. ISBN 0-7381-1811-7. Institute of Electrical and Electronics Engineering, Inc. September 1999.
- [80] IEEE Std 802.11a-1999 (Supplement to ANSI/IEEE Std 802.11, 1999 Edition). “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. High-speed Physical Layer in the 5 GHz Band”. Institute of Electrical and Electronics Engineering, Inc. September 1999.
- [81] International Telecommunications Union (ITU), <http://www.itu.int>, accessed November 2001.
- [82] Persistent Client State HTTP Cookies, [http://www.netscape.com/newsref/std/cookie\\_spec.html](http://www.netscape.com/newsref/std/cookie_spec.html), accessed December 2001.
- [83] U.S. Department of Commerce. “Secure Hash Standard”. Federal Information Processing Standard (FIPS) Publication 180-1. April 1995.
- [84] U.S. Department of Commerce. “Secure Hash Standard”. Federal Information Processing Standard (FIPS) Publication 180. May 1993.
- [85] P. Metzger and W. Simpson. “IP Authentication using Keyed SHA1 with Interleaved Padding (IP-MAC)”. RFC 2841. November 2000.
- [86] B. Fraser, Editor. “Site Security Handbook”. RFC 2196. September 1997.
- [87] Ericsson Wireless Guard, <http://www.ericsson.com/wlan/pdf/guard11.pdf>, accessed December 2001.
- [88] S. Kent and R. Atkinson. “Security Architecture for the Internet Protocol”. RFC 2401. November 1998.

- [89] S. Kent and R. Atkinson. "IP Authentication Header". RFC 2402. November 1998.
- [90] S. Kent and R. Atkinson. "IP Encapsulating Security Payload (ESP)". RFC 2406. November 1998.
- [91] D. Maughan, M. Schertler, M. Schneider, and J. Turner. "Internet Security Association and Key Management Protocol (ISAKMP)". RFC 2408. November 1998.
- [92] D. Harkins and D. Carrel. "The Internet Key Exchange (IKE)". RFC 2409. November 1998.
- [93] R. Thayer, N. Doraswamy, and R. Glenn. "IP Security Document Roadmap". RFC 2411. November 1998.
- [94] H. Orman. "The OAKLEY Key Determination Protocol". RFC 2412. November 1998.
- [95] J. Case, M. Fedor, M. Schoffstall, and J. Davis. "A Simple Network Management Protocol (SNMP)". RFC 1098, May 1990.
- [96] netfilter/iptables mailing list, <http://lists.samba.org/mailman/listinfo/netfilter>, accessed December 2001.
- [97] IPsec White paper, Cisco, 1998.
- [98] Charles E. Perkins and Pat R. Calhoun. "AAA Registration Keys for Mobile IP". Internet-draft (Work in progress) <draft-ietf-mobileip-aaa-key-08.txt>. July 2001.
- [99] Charles E. Perkins and Pat R. Calhoun. "Generalized Key Distribution Extensions for Mobile IP". Internet-draft (Work in progress) <draft-ietf-mobileip-gen-key-01.txt>. August 2001.
- [100] C. Perkins, Editor. "IP Mobility Support". RFC 2002. October 1996.
- [101] x509 Patch for FreeSwan Installation & Configuration Guide, <http://www.strongsec.com/fresswan/install.htm>, accessed October 2001.
- [102] W. Simpson. "PPP Challenge Handshake Authentication Protocol (CHAP)". RFC 1334. August 1996.
- [103] IETF IPsec Working Group, <http://www.ietf.org/html.charters/ipsec-charter.html>, accessed December 2001.
- [104] IETF Authentication Authorization Accounting (AAA) Working Group, <http://www.ietf.org/html.charters/aaa-charter.html>, accessed December 2001.
- [105] Internet Engineers Task Force (IETF), <http://www.ietf.org>, accessed December 2001.
- [106] R. Rivest and S. Dusse. "The MD5 Message-Digest Algorithm", RFC 1321, MIT Laboratory for Computer Science, RSA Data Security Inc., April 1992.
- [107] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear, "Address Allocation for Private Internets", RFC 1918, February 1996.
- [108] Default City, WISP, <http://www.defaultcity.net>, accessed December 2001.

- [109] IT-University, Kista, <http://www.it.kth.se/itusysadm>, accessed December 2001.
- [110] Linköping University Netlogon, <http://www.unit.liu.se/netlogon/>, accessed December 2001.
- [111] R. Droms, "Dynamic Host Configuration Protocol", RFC 2131, March 1997.
- [112] FreeSwan (IPSec implementation), <http://www.freeswan.org/>, accessed December 2001.
- [113] M. Beadles and D. Mitton, "Criteria for Evaluating Network Access Server Protocols", RFC 3169, September 2001.
- [114] Jesse R. Walker. "Unsafe at any key size; An analysis of the WEP encapsulation". IEEE doc 802.11-00/362. October 2000.
- [115] William A. Arbaugh, Narendar Shankar, and Y.C. Justin Wan. "Your 802.11 Network has No Clothes". Department of Computer Science, University of Maryland. March 2001.
- [116] MySQL, <http://www.mysql.com/>, accessed December 2001.
- [117] Network Access Server Requirements (nasreq) Working Group, <http://www.ietf.org/html.charters/nasreq-charter.html>, accessed December 2001.
- [118] IP Routing for Wireless/Mobile Hosts (mobileip) Working Group, <http://www.ietf.org/html.charters/mobileip-charter.html>, accessed December 2001.
- [119] IP Security (IPSec) Working Group, <http://www.ietf.org/html.charters/ipsec-charter.html>, accessed December 2001.
- [120] Point-to-Point Protocol Extension (pppext) Working Group, <http://www.ietf.org/html.charters/pppext-charter.html>, accessed December 2001.
- [121] IP Security Remote Access (ipsra), <http://www.ietf.org/html.charters/ipsra-charter.html>, accessed December 2001.
- [122] Mobile Ad-Hoc Networks (manet), <http://www.ietf.org/html.charters/manet-charter.html>, accessed December 2001.
- [123] IEEE 802.11 Working Groups, <http://grouper.ieee.org/groups/802/11/>, accessed December 2001.
- [124] IEEE 802.1w Rapid Reconfiguration of Spanning Tree, <http://www.ieee802.org/1/pages/802.1w.html>, accessed December 2001.
- [125] ETSI Broadband Radio Access Network (BRAN), [http://portal.etsi.org/portal\\_common/home.asp?tbkey1=BRAN](http://portal.etsi.org/portal_common/home.asp?tbkey1=BRAN), accessed December 2001.
- [126] B. Aboba, J. Lu, J. Alsop, J. Ding, and W. Wang, "Review of Roaming Implementations", RFC 2194, September 1997.
- [127] J. Kohl and C. Neuman, "The Kerberos Network Authentication Service (V5)", RFC 1510, September 1993.

- [128] Dynamics - HUT Mobile IP, <http://www.cs.hut.fi/Research/Dynamics/>, accessed December 2001.
- [129] Home RF, <http://www.homerf.org>, accessed December 2001.
- [130] P. Srisuresh and K. Egevang. "Traditional IP Network Address Translator (Traditional NAT)". RFC 3022. January 2001.
- [131] Linksys WAP 11 AP, <http://www.linksys.com/products/>, accessed December 2001.
- [132] H. Levkowitz and S. Vaarala. "Mobile IP NAT/NAPT Traversal using UDP Tunnelling". Internet-draft (Work in progress) <draft-levkowitz-vaarala-mobileip-nat-traversal-00.txt>. November 2001.
- [133] S. Vaarala. "Mobile IP NAT/NAPT/Firewall Traversal". Internet-draft (Work in progress) <draft-vaarala-mobileip-nat-traversal-00>. July 2001.
- [134] Amazing Ports, <http://www.amazingports.com/index-en.html>, accessed December 2001.
- [135] Columbitech, <http://www.columbitech.com/>, accessed July 2001.
- [136] Blowfish cipher, <http://www.counterpane.com/blowfish.html>, accessed December 2001.
- [137] Combining Java and CGI scripts, <http://www.webtechniques.com/archives/1997/09/pierce/>, accessed August 2001.
- [138] ETSI EN 301 344 v7.4.1. "Digital cellular telecommunications system (Phase 2+); General Packet Radio Service (GPRS); Service description; Stage 2". September 2000.
- [139] ETSI TS 123 121 v3.4.0. "Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); Architectural Requirements". October 2000.
- [140] ETSI ETS 300 522 ed.1. "European digital cellular telecommunications system (Phase 2); Network architecture (GSM 03.02)". September 1994.
- [141] ETSI TR 150 059 V4.0.1. "Digital cellular telecommunications system (Phase 2+); Enhanced Data rates for GSM Evolution (EDGE); Project scheduling and open issues (3GPP TR 50.059 version 4.0.1 Release 4)". August 2001.
- [142] 3GPP2 C.S0001-0 Version 1.0. "Introduction to cdma2000 Standards for Spread Spectrum Systems". July 1999.
- [143] D. Mitton, M. St.Johns, S. Barkley, D. Nelson, B. Patil, M. Stevens and B. Wolff, "Authentication, Authorization, and Accounting: Protocol Evaluation", RFC 3127, June 2001.
- [144] T. Mancill, "Linux Routers: A Primer For Network Administrators", Prentice Hall, ISBN 0130861138.
- [145] IP-Unplugged, <http://www.ipunplugged.com>, accessed January 2002.
- [146] Windows 2000/XP VPN tool. <http://vpn.ebootis.de>, accessed January 2002.
- [147] IEEE Draft P802.1X/D11. "Standard for Port based Network Access Control". Institute of Electrical and Electronics Engineering, Inc. March 2001.

## Appendix A: Glossary

AAA	Authentication, Authorization and Accounting
AAAF	Foreign AAA server
AAAH	Home AAA server
ACK	Acknowledgement
AES	Advanced Encryption Standard
AH	Authentication Header
AP	Access Point
ARP	Address Resolution Protocol
AVP	Attribute-Value-Pairs
BER	Bit Error Rate
BSS	Basic Service Set
CA	Certification Authority
CCK	Complementary Code Keying
CDMA	Code Division Multiple Access
CGI	Common Gateway Interface
CHAP	Challenge Handshake Authentication Protocol
CRC	Cyclic Redundancy Check
CSMA/CA	Carrier-Sense Multiple Access, Collision Avoidance
CSMA/CD	Carrier-Sense Multiple Access, Collision detection
CTS	Clear To Send
DES	Data Encryption Standard
DFS	Dynamic Frequency Selection
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DoS	Denial of Service
DS	Distribution System
DSSS	Direct Sequence Spread Spectrum
EAP	Extensible Authentication Protocol
ESP	Encapsulating Security Payload
ESS	Extended Service Set
ETSI	European Telecommunications Standards Institute
FA	Foreign Agent
FHSS	Frequency Hopping Spread Spectrum
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communication
HA	Home Agent
HIPERLAN	High Performance Radio Local Area Network
HR/DSSS	High Rate Direct Sequence Spread Spectrum
HTTP	HyperText Transfer Protocol
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
ICV	Integrity Check Value
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
IP	Internet Protocol
IPSec	IP Security
IR	Infrared
ISAKMP	Internet Security Association and Key Management Protocol
ISM	Industrial, Scientific, and Medical
ISO	International Organization for Standardization
ISP	Internet Service Provider



---

ITU	International Telecommunication Union
KDC	Key Distribution Centre
L2TP	Layer Two Tunnelling Protocol
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LLC	Logical Link Control
MAC	Message Authentication Code
MIB	Management Information Base
MIP	Mobile IP
MN	Mobile Node
NAI	Network Access Identifier
NAS	Network Access Server
NAT	Network Address Translator
NIC	Network Interface Card
NTP	Network Time Protocol
OFDM	Orthogonal Frequency Digital Multiplexing
PAE	Port Access Entity
PAN	Personal Area Network
PAP	Password Authentication Protocol
PGP	Pretty Good Privacy
PHY	Physical Layer
PKCS	Public Key Cryptography Standard
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPTP	Point-to-Point Tunnelling Protocol
PRN	Pseudo Random Number
QoS	Quality of Service
RADIUS	Remote Authentication Dial-In User Service
RF	Radio Frequency
RFC	Request for Comments
RSA	Rivest, Shamir, and Adelman
RTS	Ready To Send
SA	Security Association
SHA	Secure Hash Algorithm
SKIP	Simple Key-management for Internet Protocols
SNMP	Simple Network Management Protocol
SOHO	Small Office and Homes
SPI	Security Parameters Index
SRP	Secure Remote Password
SSID	Service Set Identifier
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TDMA	Time-Division Multiple Access
TLS	Transport Layer Security
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications Service
UNII	Unlicensed National Information Infrastructure
WCDMA	Wideband Code Division Multiple Access
WECA	Wireless Ethernet Compatibility Alliance
WEP	Wire Equivalent Privacy
WLAN	Wireless Local Area Network
VoIP	Voice over IP
VPN	Virtual Private Network

## Appendix B: NAS Architecture

### *Hardware specifications*

Pentium II MMX at 350 MHz  
64 MB of RAM memory in a single DIMM slot  
10 GB hard drive: Western Digital Caviar AC310100  
3.5" Floppy disk drive  
48x CD-ROM drive: Aopen  
3 PCI Ethernet cards: 3 COM 3C905B-TXNM

### *Software specifications*

Installed Red Hat Linux 7.1 in an empty hard drive. Kernel 2.4.2

The partitions in the hard drive are as follows:

Hda1	Root partition	/	500MB
Hda2	Swap partition		128MB
Hda5	Usr partition	/usr	4000MB
Hda6	Var partition	/var	1000MB
Hda7	Home partition	/home	4200MB

Updated the kernel to 2.4.8 from [www.kernel.org](http://www.kernel.org)

Compiled a new image file based on kernel 2.4.8 with IPsec in it (Linux FreeSwan 1.91)

The following are the modules included in the NAS. If a module was installed and has been replaced by another one, only the newer is mentioned

### **NAS main modules**

- ❑ DHCP server  
The Internet Software Consortium (ISC) DHCP server, "dhcpd" is used.  
Version unknown but is the one shipped with Red Hat 7.1
- ❑ DNS server  
The Internet Software Consortium (ISC) DNS server, "named" is used.  
We have installed BIND 9.1.3 version
- ❑ Firewall  
Iptables v1.2.1a is installed by default with Red Hat 7.1
- ❑ NAT  
NAT functionality is provided by Iptables v1.2.1a
- ❑ Database  
Installed MySQL version 11.13 Distribution 3.23.36 for redhat-linux-gnu (i386)
- ❑ Web Server  
Installed Apache Server 1.3.19
- ❑ NTP Server  
Installed NTP 4.1.0 release

- ❑ Radius server  
Installed Freeradius version 0.2 for host i686-pc-linux-gnu  
Upgraded to Freeradius version 0.3 for host i686-pc-linux-gnu
- ❑ Kerberos server  
Installed KTH-Kerberos IV distribution krb4-1.0.8  
Red Hat 7.1 already has Kerberos V
- ❑ Mobile IP  
HA and FA functionality provided by the Dynamics 0.8.1 release

We are also experimenting with Stanford's Mosquitonet 2.0.2beta but in other machines.

## Other modules

- ❑ CIPE  
Installed in order to try the UDP encapsulation. Version 1.5.2
- ❑ IP-login  
Code from Enrico Pelleta that gives the skeleton of the whole system  
Version 1.0
- ❑ mhash library  
Version 0.8.9
- ❑ OpenSSL 0.9.6b  
Crypto library for Blowfish encryption for example
- ❑ x509 patch for FreeSwan  
Allows using x509 certificates with IPSec. Useful for road-warriors  
Version 0.9.2
- ❑ mySQL GUI  
Graphical interface to MySQL  
Version 1.7.5
- ❑ gcc compiler  
Version 2.96-81

## Appendix C: 802.1x support

### *802.1X OS support*

#### **Microsoft**

Windows XP has support for 802.1X built-in using EAP-TLS [75] (certificate based authentication) and MD-5 challenge. It also provides automatic detection of wireless networks through the BSSID and allows easy configuration

#### **Cisco**

Windows 9x, NT4, 2000, Mac OS, Linux

### *RADIUS servers supporting 802.1x*

Funk Software: Steel-Belted Radius Server v3.0

Interlink Networks: RAD-E Enterprise Access Server v5.1

Microsoft: Windows 2000 Server

Cisco: Access Control Server 2000 Version 2.6

### *Access Points vendors supporting 802.1x*

#### **Agere Systems**

In the Orinoco series the AP-2000 has support for 802.1x from November 2001. A software upgrade is needed which can be downloaded from the web site.

#### **Cisco**

Cisco offers 802.1x support in its Aironet© 350 line of products.

[http://www.security-informer.com/english/crd\\_security\\_495312.html](http://www.security-informer.com/english/crd_security_495312.html)

[http://www.cisco.com/warp/public/cc/pd/witc/ao350ap/prodlit/carto\\_in.htm](http://www.cisco.com/warp/public/cc/pd/witc/ao350ap/prodlit/carto_in.htm)

#### **3Com**

3 Com has vowed support to 802.1x but at the time of writing this thesis no firmware updates had been released to support it.

<http://emea.3com.com/news/news01/mar26.html>

#### **Enterasys**

Enterasys has pledged support to 802.1x but it will start implementing first in LAN switches. No software for its Access Points yet.

#### **Compaq**

Compaq has pledged support to the standard but its Access points (WL510 & WL410) still do not support it

**HP**

HP has pledged support to the standard but its Access points (hp wireless gateway hn200w) still do not support it.

**Symbol**

Symbol Technologies has pledged support to the standard but its Access points (Spectrum24<sup>®</sup> High Rate AP 41X1) still do not support it.

**Intel**

Intel has pledged support to the standard but its Access points (PRO/Wireless 5000 LAN Access Point) still do not support it.

**Dell**

Dell has pledged support to the standard but its Access points (TrueMobile 1150 Wireless Access Point & TrueMobile 1170 Wireless Base Station) still do not support it.

## Appendix D: RADIUS Server modifications

As stated in the body, Freeradius v0.3 has been used in this master thesis. No changes have been made in the server, except for some minor changes made in the `rlm_sql` module. This was done to let the server make some special SQL queries and to make multiple queries where earlier only one query could be done. Some SQL queries that were not used were removed. Changes are not extensive, and an upgrade of the RADIUS server should be fairly easy.

The modified files are:

```
freeradius-0.3/src/modules/rlm_sql/rlm_sql.c
freeradius-0.3/src/modules/rlm_sql/conf.h
```

Some SQL queries that were not used have been removed:

```
authorize_group_check_query
authorize_group_reply_query
accounting_onoff_query_alt
accounting_stop_query_alt
```

The SQL queries that were added are:

- ❑ **authenticate\_test\_user**: If a user wants to try the test mode, the system has to check that the user hasn't tried it before. Saving the MAC address of all the users that uses the test mode does this. All MAC addresses are stored in the `test_users` table. This query checks if the user's address is already stored in the table. If the address is in the table, then the user has already used the free test mode before and therefore is not allowed to do it again; if the address is not in the table the user is granted access.
- ❑ **add\_test\_user**: When a user tries the test mode and has been accepted, the users MAC address has to be stored in the `test_users` table, the reason is explained above.
- ❑ **accounting\_reply\_query**: When accounting is started, a unique pseudo random session ID is created. This session ID should be sent back to the NAS in the accounting reply packet, the query extracts the session id from the `radacct` table.
- ❑ **change\_login**: To prevent that account is being used more than once at the same time, a value in the `radcheck` table is changed. This is to prevent duplicate logins.
- ❑ **change\_logout**: Change the value in `radcheck` that was change by the `change_login` query back to the original value. The reason for this is clear if one reads the explanation above.

## Appendix E: SQL queries

The RADIUS server sends a number of queries to the SQL database to fetch information. The following queries are used:

- ❑ **authenticate\_query**: Extract the password (Value) from the database. Attribute indicates if the password is encrypted or not. Check at the same time that the user is using the same MAC address that has been registered before. If it is the user's first login to the system the MAC-address value in the database should be NULL.

```
SELECT Value, Attribute FROM radcheck WHERE UserName = "User-Name"
AND (Attribute = "Password" OR Attribute = "Crypt-Password") AND
(macaddress = "Acct-Multi-Session-Id" OR macaddress is NULL) ORDER BY
Attribute DESC
```

- ❑ **authenticate\_test\_user**: Check if a test user has tested the service before. If the users MAC address is stored in the test\_user table then the user has tested the service before.

```
SELECT Id, UserName, Attribute, Value FROM test_users WHERE Username =
"User-Name" AND macaddress = "Acct-Multi-Session-Id"
```

- ❑ **authorize\_reply\_query**: Extracts the attributes and values that will be sent back to the NAS, the attribute describes what type of authorization the user has. In this thesis the only limitation in access is the time that a prepaid user has left (Session-Timeout). This information should be stored here.

```
SELECT id, UserName, Attribute, Value FROM radreply WHERE Username =
"User-Name" ORDER BY id
```

- ❑ **add\_test\_user**: After a user that test the service first has been checked that the user has not test the service before, the users MAC address will be added to the test\_user table. This is to prevent duplicate logins.

```
INSERT into test_users (UserName, Attribute, Value, macaddress)
VALUES("User-Name", "used", "TEST", "Acct-Multi-Session-Id")
```

- ❑ **add\_mac\_address**: Binds the user's MAC address to the username the first time the account is used. This is done to make it harder to steal an account.

```
UPDATE radcheck SET macaddress = "Acct-Multi-Session-Id" WHERE
username = "User-Name" AND macaddress is NULL
```

- ❑ **change\_login**: When a user logs in, the attribute value will be changed from "Password" to "busy". This way no more than one will be able to use the account at the same time.

```
UPDATE radcheck SET Attribute = "busy" WHERE Username = "User-Name"
```

- ❑ **change\_logout**: When a user logs out, the attribute value will be changed from "busy" to "Password" so the user will be able to log in again.

```
UPDATE radcheck SET Attribute = "Password" WHERE Username = "User-Name"
```

- **accounting\_start\_query**: Initiate the accounting entry for the session, puts in the initial accounting data (e.g. Starttime), other values are set to zero.

```
INSERT into radacct (Id, AcctSessionId, Value, UserName, Realm,
NASIPAddress, AcctStartTime, AcctStopTime, AcctSessionTime,
ConnectInfo_stop, AcctInputOctets, AcctOutputOctets, CallingStationId,
AcctTerminateCause, FramedIPAddress, AcctStopDelay, Attribute, cardnumber)
values("", "Acct-Session-Id", "Acct-Unique-Session-Id", "User-Name", "Realm",
"NAS-IP-Address", "StartTime", "0", "0", "", "0", "0", "Calling-Station-Id", "",
"Framed-IP-Address", "0", "Acct-Session-Id", "Callback-Number")
```

- **accounting\_stop\_query**: When the session ends the accounting must stop. This query inserts session parameters (e.g. sessiontime, bytes sent, etc) in the database.

```
UPDATE radacct SET AcctStopTime = "StopTime", AcctSessionTime = "Acct-
Session-Time", AcctInputOctets = "Acct-Input-Octets", AcctOutputOctets =
"Acct-Output-Octets", AcctTerminateCause = "Acct-Terminate-Cause" WHERE
Value = "Acct-Session-Id" AND UserName = "User-Name"
```

- **accounting\_reply\_query**: This query will extract a unique session ID that is stored in the "Value" field. The unique session ID is created by the RADIUS server and is a pseudo random string. The unique session ID will be sent back in the accounting reply packet.

```
SELECT Id,UserName,Attribute,Value FROM radacct WHERE UserName =
"User-Name" AND Value = "Acct-Unique-Session-Id"
```

- **accounting\_update\_query**: When a prepaid customer has not used all his time, the remaining time should be stored so that is could be used the next time the customer loges in.

```
UPDATE radreply SET Value = "Session-Timeout" WHERE UserName = "User-
Name" AND Attribute = "Session-Timeout"
```



## Appendix F: Testbed setup

### General testbed

When building the system a test bed was set up. A schematic illustration of the test bed can be seen in figure I. The computers `daniel.r2m.net` and `antares.r2m.net` were mainly used by the authors to work on and had standard installation of RedHat 7.2 respective 6.2. The two computers also had Windows 2000 Professional installed. The computer `nas.r2m.net` works as gateway for the computers attached to the two networks `192.168.10.0/24` and `192.168.20.0/24`. A description of the computer `nas.r2m.net` can be found in Appendix B. The mobile node (MN) used was a laptop with RedHat 6.2 installed. The gateway in the upper right corner in figure I is R2M own gateway that connects all the computers on network `192.168.1.0/24` to the Internet.

### General testbed

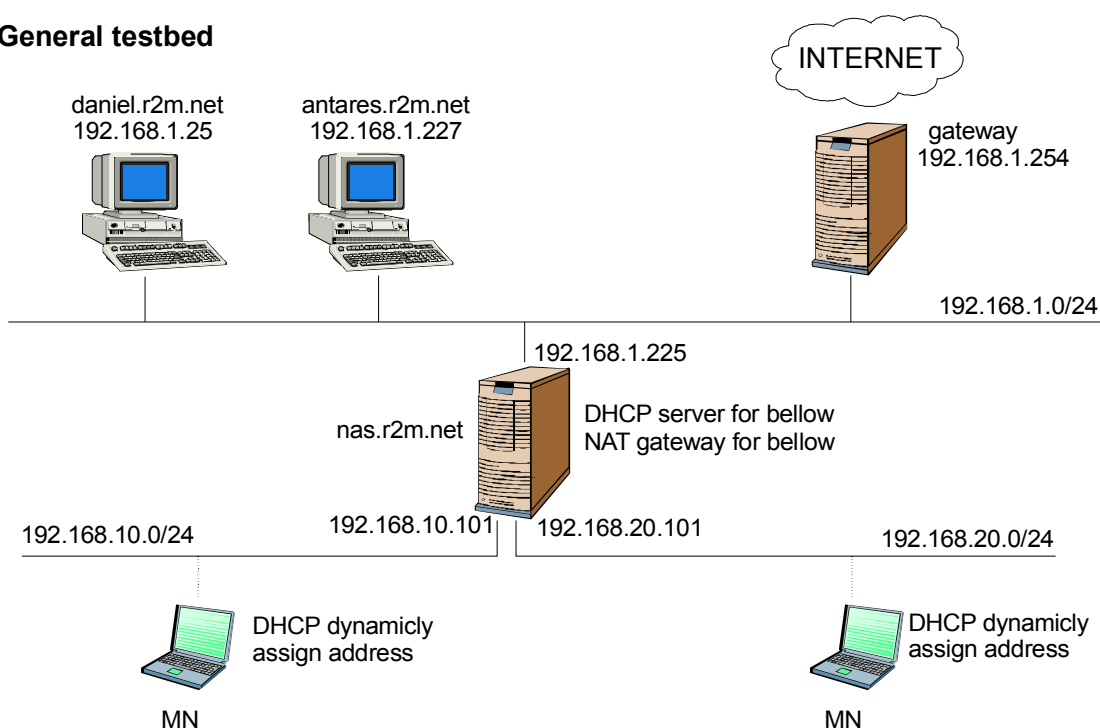
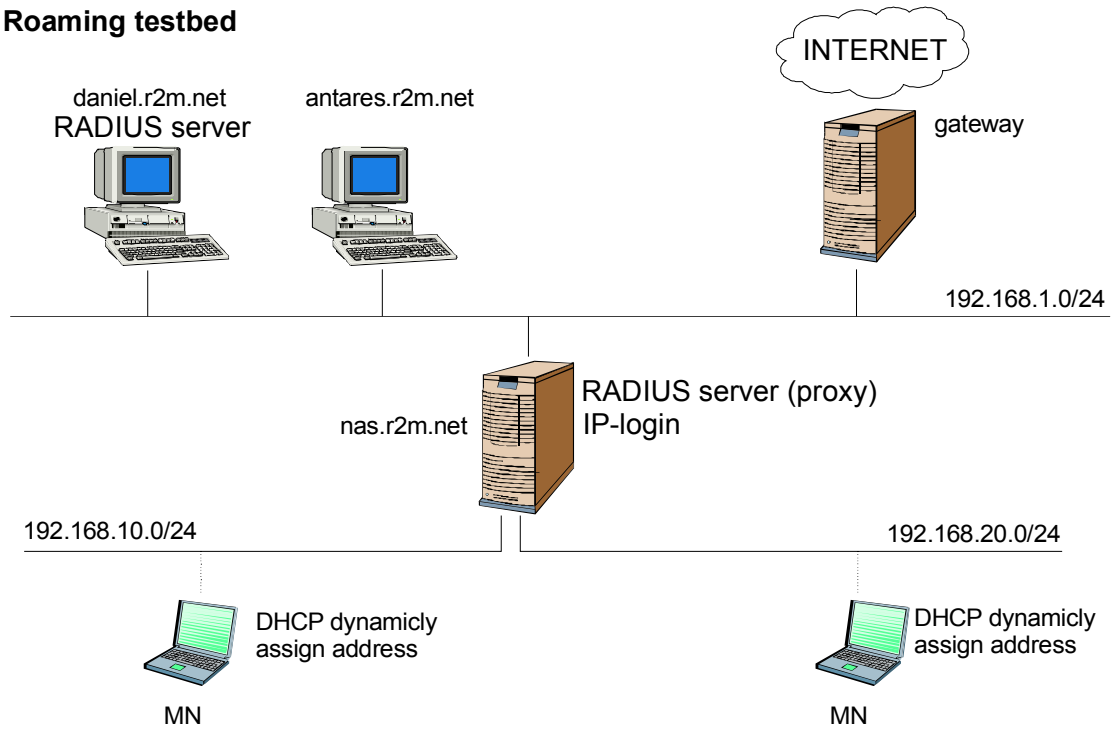


Figure I General testbed setup

### RADIUS roaming testbed

When testing the RADIUS roaming functionality another RADIUS server was needed. The new RADIUS server was placed in the computer `daniel.r2m.net`. When roaming was done, the RADIUS server in `nas.r2m.net` worked as a RADIUS proxy.

**Roaming testbed**

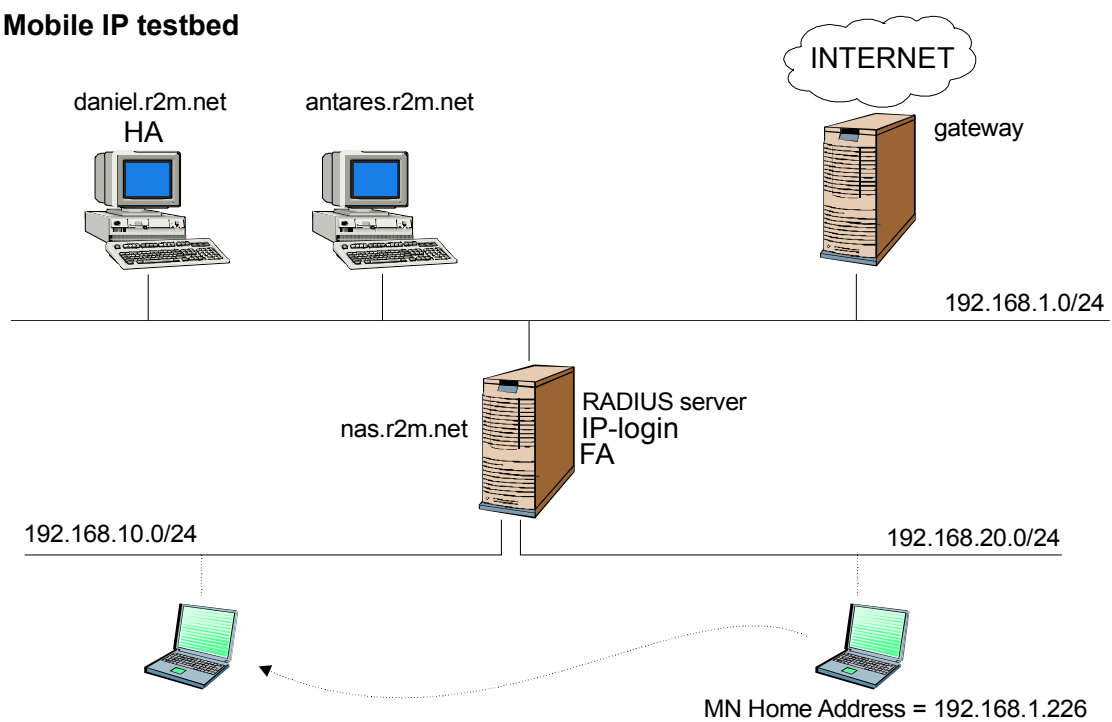


**Figure II Roaming testbed setup**

**Mobile IP testbed**

When Mobile IP was used the testbed looked like the illustration in figure III. The FA was placed in the NAS (nas.r2m.net) and the HA was placed in daniel.r2m.net. A RADIUS server that took care of the AAA authentication was also placed in the NAS.

**Mobile IP testbed**



**Figure III Mobile IP testbed setup**