Umeå University
Department of Computing Science
Ulf Jonsson
c99ujn@cs.umu.se

4th February 2004

Security and cooperation considerations for Skekraft.net's wireless network

External supervisor: Jan Wennerström
Internal supervisor: Jerry Eriksson
Examiner: Per Lindström

Abstract

Security has always been the Achilles' heel of wireless local area networks (WLANs). In this paper the authentication aspect of security is investigated. Since the media used to connect to the network is air, it is more difficult to control the network access than in a regular wired network. Many authentication methods already exists and more are making their way. This paper therefore examines the most interesting ones and the most suitable is chosen for implementation in Skellefteå Kraft's WLAN.

As more companies, organizations and persons are setting up access points (APs) to form WLANs, the available radio frequencies are becoming a scarce resource. Each AP uses a radio frequency to send and receive data. Therefore would it be desirable to share APs at least between service providers to save frequencies, money and shorten time to market. Vendors of APs have discovered this demand. However the development of APs that support this kind of cooperation is at an early stage and different vendors have solved the problem in their own way. This paper examines this feature and shows how it can be used by Skellefteå Kraft to share APs with the network SkellefteOpen. An alternative solution to the cooperation problem is also presented in order to show the benefits with the first solution.

Acknowledgments

- To the company and employees at Skellefteå Kraft for making this Master's Thesis possible and for a great time.
- Special thanks to my supervisor Jan Wennerström at Skellefteå Kraft for all the help and support despite your heavy workload.
- Thank you very much, Torgny Holmlund at Skellefteå Kraft, for the invaluable help with the tests and all equipment.
- Also great thanks to MobileCity for letting me use SkellefteOpen and take part of valuable information.
- Special thanks to Jim Engman and Magnus Andersson at MobileCity for all the help and support.
- Also a great thank to Jan-Åke Rönnblom at Skeria Utveckling for all the help with the test runs and expertise.
- Thank you, Jerry Eriksson at Umeå University, for commenting on the report.
- Last but certainly not least thank you, Jenny Vikström, for the support and understanding.

Contents

List of Figures

1 Introduction

Wireless Local Area Networks (WLANs) have gained increasing popularity among Internet Service Providers (ISPs). Especially the IEEE 802.11 standard of WLANs has been accepted. Setting up WLANs is an easy way to provide Internet access since there is no requirement of connecting the users with a cable. This feature can be employed by both broadband providers and Hot Spot providers. Broadband providers can utilize this technique by setting up Access Points (APs) which users establishes a wireless connection with. WLAN gives a fast time to market and is more economical than digging down cables which are two important factors for broadband providers. With just a few APs and external aerials it is possible to cover an area of over a kilometer in radius. The use of powerful aerials extends the distance between the AP and the user greatly if they are in line of sight. Hot Spot providers on the other hand often covers a smaller area but inside this area a high bandwidth is provided together with mobility support. Both the broadband and Hot Spot providers use the same kind of equipment but in different ways. A cooperation with two or more providers is therefore desirable, especially between a broadband and a Hot Spot provider which offers services to two different market segments. By performing a cooperation it is possible to share the infrastructure and in this way save money, shorten time to market and last but not least house keeping with frequencies. Since more and more devices uses the air as media, available frequencies for sending and receiving data are becoming a scarce resource.

Already from the beginning of WLAN deployments, security has been its weak link. Since the media that transfers the data is shared among others, the security considerations are higher than in a wired network. In a wired network you need at least a physical connection to access the network. This is not the case in a wireless network where the only requirement to access the network is to be located sufficient close to the AP. However there are techniques to authenticate users in a wireless network and it is almost indispensable to use one of them in order to keep unauthorized users out of the network.

In this Master's Thesis the two mentioned aspects of a IEEE 802.11 network; cooperation and authentication, will be investigated. Authentication is almost a requirement in order to share APs with other providers. With a very limited authentication, the network will be even more vulnerable if a cooperation with another provider is performed. We will therefore examine and evaluate different authentication methods and chose one of them which will

be implemented and modified to be suitable for an existing WLAN. Then it will be tested in the WLAN owned by Skellefteå Kraft. Other already known theoretical frameworks will also be examined and described in order to provide a solution of how to share APs with different providers. Two solutions to this problem will be implemented and tested in two existing WLANs. The test results are then presented in the paper. Finally a concluding discussion of the solutions and test results is provided at the end.

2 Background

This Master's Thesis was carried out at Skellefteå Kraft which is a power company that has become a large provider of broadband and services such as IP telephony. Through both wired and wireless connections, customers can connect to the Internet Service Provider (ISP) AC-net[1]. Skellefteå Kraft offers a WLAN product for customers that want a broadband connection as soon as possible. The customers are resided at locations where the extension of broadband via optofibre cable has not yet been performed. Today Skellefteå Kraft has approximately 600 customers who use their WLAN product.

2.1 Skekraft.net

In the following of this paper both the wired and the wireless network will be called Skekraft.net. The network that Skellefteå Kraft has built and owns has the topology shown in Figure 1. APs are located at approximately 50 places. Most of them are located in the villages around the City of Skellefteå but some are placed in central Skellefteå. They are also divided in different districts, e.g. Byske is one and Burträsk is another.

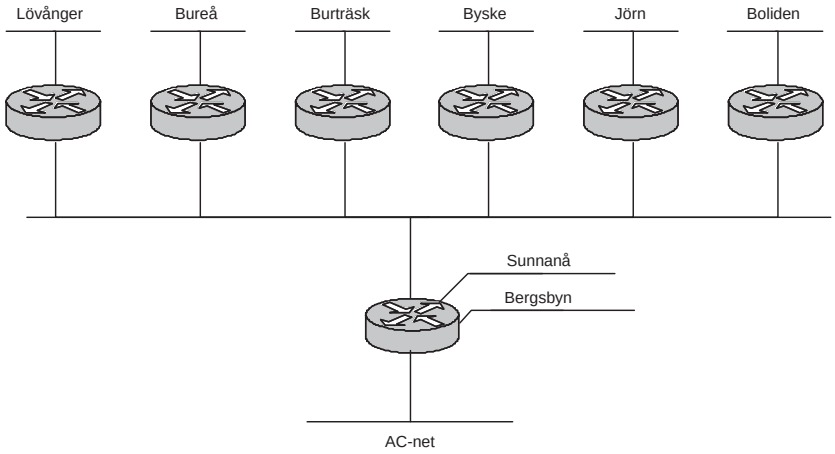

Figure 1: The topology of Skekraft.net

The WLAN technique 802.11b (described in section 3.1) is used to carry data between customers and a fixed AP. Therefore Skellefteå Kraft's WLAN product is not intended to support mobility. Every WLAN customer has an antenna mounted on their house or apartment. This antenna is directed to

[1] www.ac-net.net

the closest AP, which has to be in line of sight and located within a specified distance. The antenna is connected to an ethernet converter, which has a built in wireless network card. The ethernet converter is further attached with a cable to the network card in the customer's computer. This makes the ethernet converter function as a bridge which converts traffic from IEEE 802.3, ethernet, to IEEE 802.11 and vice versa. The wireless interface captures and saves the hardware address, called the Media Access Control (MAC) address, of the customer's network card when the ethernet converter is booted. The two network interfaces have therefore the same MAC address. Every customer is given a public IP address and the ethernet converter is assigned a management IP address used for the staff at Skellefteå Kraft to configure and manage it. The ethernet converters are already configured when they are bought and cannot be reconfigured by the customers since they are protected with a password.

2.2 SkellefteOpen.net

Originally a research project called StockholmOpen.net [2] was started at the Royal Institute of Technology (KTH). This project has since been developed among others by Martin Hedenfalk at KTH. The design and implementation of a Network Access Server (NAS) and the design of the access network is described in his Master's Thesis, (Hedenfalk, 2001). SkellefteOpen.net is a spin-off of StockholmOpen.net and uses the same concept of an operator neutral access network, called Open.net. The design of the two networks is therefore pretty much the same. The idea behind the Open.net concept is that anyone can build an access network which end users can access. Operators can then provide their services through the access network to the end users which are free to choose whatever operator they want to use.

The implementation of SkellefteOpen.net was performed 2002 during the project course "Communication System Design" given at KTH. This project mostly focused on the two services that were developed, a location based messenger and a basic location tool. However the core of SkellefteOpen.net was also set up and tested (Ljungberg et al., 2002). The following year the course was given again and this time the focus was on expanding the Open.net community by connecting StockholmOpen.net with SkellefteOpen.net. The solution to the problem is described in (Andersson et al., 2003). However the full scale expanding has not yet been carried out but the solution is tested and ready to be used.

[2] www.stockholmopen.net

2.3 Project specification

There are two tasks within this project but they are both related and should be considered together during the whole project. However, the first one is the most important one since it has to be solved before the second one can be carried out in an appropriate way. Below is a description of each task.

2.3.1 Authentication of customers

There is a very limited authentication of customers in Skekraft.net. Two problems arises from this. First is the possibility that a non-customer with some knowledge about WLANs can connect to an AP and use the bandwidth free and also cause trouble for some customers. The second problem arises if one customer accidently configures his/her network settings with an IP address, which another customer who is connected to the same AP is using. This will cause IP collisions and trouble for both users with the same IP address. One task of this Master's Thesis is therefore to investigate and evaluate different authentication methods that can be used in Skekraft.net. One of them should then be chosen and implemented. The requirement of the chosen authentication method is that there should not be any changes, if possible, for the customers and in the network, such as new software for the customers.

2.3.2 Cooperation with SkellefteOpen

MobileCity is an EC financed project in Skellefteå that runs an open wireless network called SkellefteOpen[3]. Everyone that is able to associate to one of the network's APs can choose the ISP the user wants to use. Currently there are two ISPs, one that students at Skeria can use free and one that sells tickets for Internet access. Unlike Skekraft.net, SkellefteOpen.net uses the Dynamic Host Configuration Protocol (DHCP) to administrate IP addresses. This means that a server will decide what IP address a user will have and a user maybe gets a different IP address when connecting a second time. MobileCity has installed several APs in Skellefteå but uses Skellefteå Kraft's wired network to connect them. Therefore it would be desirable to share APs with MobileCity. That is, users of SkellefteOpen should be able to use Skellefteå Kraft's APs to get access to their ISP. Skellefteå Kraft's customers should also be able to use SkellefteOpen's APs to be able to use their accounts at Skellefteå Kraft to connect to the Internet.

[3] www.skellefteopen.net

2.4 Method

At first a literature study of existing authentication methods will be performed. The investigated methods will be evaluated and one of them will be chosen to implement. After the implementation, the authentication system will be tested. Finally the second task concerning the cooperation problem will be investigated by examining existing and finding new solutions to the problem. A theoretical study of related protocols and frameworks will also be performed in order to solve the problem.

3 Theoretical framework

The following sections describes the relevant protocols and techniques used in this Master's Thesis. At first an overview of the standard IEEE 802.11 will be given. Then the principles of authentication will be explained and how cryptography can be used to carry out the authenticaion. All of the examined authentication methods are also explained. After that a short description of the Simple Network Management Protocol (SNMP) will be given. This protocol will be used by the authentication system. The following section explains the concept behind virtual APs, which will be used to solve the second task. Then will Virtual Local Area Networks (VLANs) be described, which is a vital technique in order to use the concept of virtual APs. Finally a deeper description of SkellefteOpen will be given.

3.1 Overview of IEEE 802.11

The standard IEEE 802.11, (IEEE, 1999), specifies the Media Access Control (MAC) and the PHYsical layer (PHY) in the Internet protocol stack, Figure 2, for wireless connectivity.

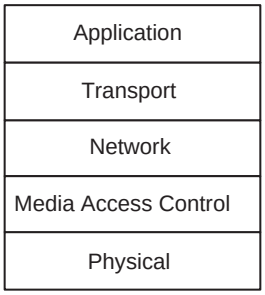

Figure 2: The Internet protocol stack.

Both an ad hoc and an infrastructure mode are specified in the standard. However, only the infrastructure mode will be covered. In the infrastructure mode an AP is used to send and receive packets to and from STAtions (STAs). It will therefore not be any traffic directly between STAs, only between the AP and the STAs. Before a STA can send any data messages via an AP, the STA has to associate with one AP. An example is shown in Figure 3 where STA 1 communicates with STA 2 through the AP. The two STAs and the AP in the figure form a Basic Service Set (BSS). Several BSSs can be connected with a Distribution System (DS) which together form an Extended Service Set (ESS). This makes it possible for a STA in one BSS to communicate with

another STA in a different BSS. The DS could be a traditional wired LAN which maybe is connected to the Internet by a portal, as shown in Figure 4.

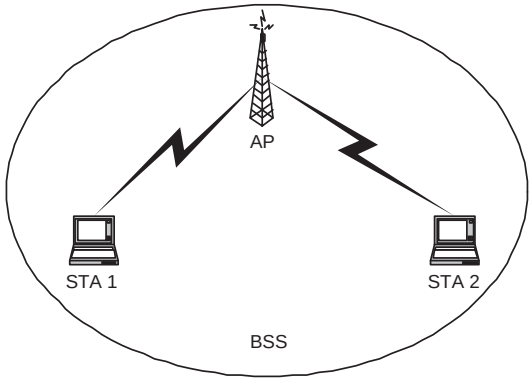

Figure 3: The Basic Service Set (BSS), adopted from IEEE (1999).

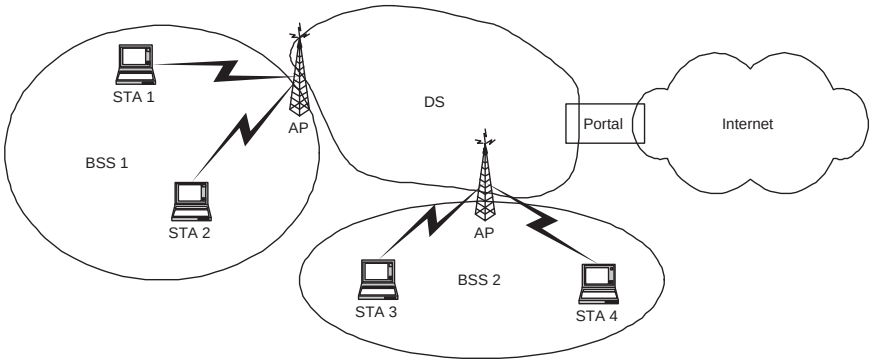

Figure 4: The Extended Service Set (ESS), adopted from IEEE (1999).

There are several flavors of IEEE 802.11 and more are making their way to become a standard. The currently most interesting versions of the standard are the `a`, `b` and `g`. The differences between them are the maximum bandwidth and at what frequencies they operate on. The IEEE 802.11b was the first to reach the market and consequently is the one that has the most implementations. Then the IEEE 802.11a became adopted by the market. However this standard operates on a different frequency band, 5 GHz, than its successor which is at 2,4 GHz. These two versions are therefore not compatible with each other. The major benefit with `a` is the support of higher bandwidth. Instead of 11 Mega bit per second (Mbps), `a` has the theoretical bandwidth of 54 Mbps. The latest standard of the IEEE 802.11 is `g`, which operates on the same frequency band as `b` and is backward compatible with `b`. Moreover, the maximum bandwidth is increased to 54 Mbps.

The frequency bands that IEEE 802.11 operates on are divided in several channels. The number of channels is determined by the radio frequency regulations in the country where the equipment is used. Most of the countries in Europe have 13 available channels. The more available channels there are, the easier is the frequency planning. It is important to configure adjacent APs to use different channels which are sufficient separated from each other. In countries where 13 channels are available it is possible to use three channels on the same geographical area without loss of bandwidth caused by interference of other APs (IEEE, 1999).

Each AP is configured with a Service Set IDentifier (SSID), which functions as a network name and indicates the identity of the ESS. A single BSS is identified by its Basic Service Set IDentifier (BSSID), which is the MAC address of the AP's wireless network card. The SSID is a string of at most 32 characters. All STAs that want to connect to a particular ESS should be configured with the same SSID as the APs in the ESS. A STA can set the SSID with the special value "ANY" or the empty string, which indicates that the STA will connect to the first ESS as it discovers. However, the APs can be configured to not allow associations with STAs which have the SSID set to "ANY". There are two ways for a STA to discover SSIDs and which channel they exists on, namely passive and active scanning. In passive scanning, the STA listens for beacons and probe responses on each of the available channels for a specified time per channel. When the STA has received either a beacon or probe response, it knows which channel the AP operates on. Active scanning is much faster but requires that the STA sends probe requests containing the SSID on each of the available channels and listens for responses. Alternatively, the STA can send a probe request containing the "ANY" SSID.

3.2 Principles of authentication

Authentication is often related to the AAA concept, which stands for Authentication, Authorization and Accounting. AAA is a framework for access control. In this paradigm a user first connects to a NAS, which asks a AAA server if the user is allowed to access the network. While this Master's Thesis only concerns authentication, the other two components are also worth mentioning.

- Authentication is the process of proving who someone is for someone else (Kurose and Ross, 2001). The one that wants to prove who he or

she is will be called the supplicant and the other part will be called the authenticator in the following of this paper. Mutual authentication is the process when both the supplicant and the authenticator authenticates to the other. In that case both parties know who they are communicating with.

- Authorization is the process of granting or denying a user access rights to a resource. The authorization is often performed after the authentication but it could be integrated to the authentication process.
- Accounting is the process of gathering information of resource usage with the intention of trend analysis, auditing, billing or cost allocation.

There are three general ways of authentication; providing something the supplicant has, knows or is to the authenticator (Schneier, 2000). Below is a description and an example of each type.

- **Something the supplicant has.** The supplicant has something, e.g. a key, that the authenticator knows belong to the supplicant. If the supplicant uses the key to unlock a door then the supplicant is authenticated by the door and may pass through.
- **Something the supplicant knows.** The supplicant knows something, often a password or code, that the authenticator knows belong to the supplicant. Suppose the lock to the door in the example above is replaced with a code lock. If the supplicant enters his or her code and the doors gets unlocked then the supplicant is authenticated and may pass through.
- **Something the supplicant is.** The supplicant is something that will make the authenticator sure of the identity of the supplicant. Suppose that the lock to the door in the example above is replaced with a fingerprint scanner. If the supplicant puts his or her thumb to the fingerprint scanner and the door opens then is the supplicant authenticated and may pass through.

3.3 Cryptography

Originally cryptography is a field from the mathematics, (Singh, 1999). Although mathematics is an old science, the most prominent advances in cryptography was made in the past 30 years (Kurose and Ross, 2001). The basics

of cryptography are the components; plaintext, encryption/decryption algorithm, ciphertext and keys. The plaintext is the readable string or message which will be encrypted. The encryption algorithm specifies how the encryption will be carried out and uses the key to produce the ciphertext. The ciphertext is unintelligible to any intruder. To decrypt the ciphertext, the key and the decryption algorithm is used. In most cases the same encryption and decryption algorithm is used. Often the encryption algorithm is public so everybody may examine it in detail, even a potential intruder. When designing a secure system, one should choose a published and standardized encryption technique according to Saltzer and Schroeder (1975). Then is the security resided in the key and not in the protocol itself.

Cryptography is often used by the authentication process in a computer system since it makes it harder for an intruder to attack the system. Generally there are two kinds of cryptographic systems; symmetric key and public key. They will be described in the following two sections.

3.3.1 Symmetric key

In a symmetric key system, two parties share a common key which is only known by them. This key can then be used by the two parties to authenticate each other. The most commonly used authentication method is to have a username together with a password, called user credentials (Schneier, 2000). This method can be placed under the second type of authentication described in section 3.2. Both the supplicant and the authenticator knows the user's username and password. A very simple and not secure authentication method, that e.g. the File Transfer Protocol (FTP) (Postel and Reynolds, 1985) uses, sends the username and password in cleartext. However, a malicious person might eavesdrop on the communication between the two parties and use the same username and password later to log in to the FTP server. Another more secure authentication method that uses the symmetric key system is the challenge response concept. In this concept the authenticator sends a challenge text as plaintext to the supplicant. The supplicant then encrypts the challenge text and sends it back. If the authenticator successfully decrypts the encrypted challenge text then the supplicant is authenticated. However the challenge text should not be used more than once since an attack could be performed with an old challenge text. Suppose that a malicious person has eavesdropped on the communication between the two parties a number of times. Then the malicious user might have gathered several challenge texts and their corresponding ciphertexts. The malicious user can then send the username of the eavesdropped party and hope that the authentica-

tor answers with a challenge text that is known by the malicious user. The malicious user then responds with the corresponding ciphertext.

3.3.2 Public key

Symmetric keys have to be distributed via a secure channel. This could be difficult to achieve if the supplicant and authenticator are far away from each other. Therefore Diffie and Hellman (1976) developed the public key cryptosystem which does not need this cumbersome key distribution system. In a public key cryptosystem, everybody has two keys, one private and one public. The private key is only known by the owner but the public key is published so that everybody can get it. If a plaintext is encrypted with the private key, then it is possible to decrypt the ciphertext with the public key. Furthermore, it is only possible to decrypt a ciphertext, encrypted with the public key, with the private key. Another important characteristic of the public key system is that the two keys are very difficult to derive from the other.

Authentication can be carried out by using the public key cryptosystem. The same challenge respond protocol described above can be used with public and private keys. The supplicant sends a message claiming its identity. Then the authenticator responds by sending back a challenge text. The supplicant now encrypts the challenge text with its private key and sends the encrypted challenge text to the authenticator. If the authenticator has the supplicant's public key and successfully decrypts the ciphertext, then the supplicant is authenticated.

However Schneier (1996) clarifies that a man-in-the-middle attack can generally be used against any protocol that does not involve a shared secret. This is a consequence of the fact that the authenticator has to get the supplicant's public key from someone and be sure that the public key really belongs to the supplicant. There are several ways of publishing a public key, e.g. posting it on a Web page, storing it in a public key server or by sending the key by e-mail to the one requesting the key (Kurose and Ross, 2001). To assure that a public key belongs to a certain person, a Certification Authority (CA) is needed to sign public keys. The two main tasks for a CA are to verify that an entity, e.g. person, is who it says it is and to digitally sign a certificate that binds the public key to the owner. However, the authenticator now has to trust the CA, creating a chain of trust. The CA is an important component of a Public Key Infrastructure (PKI), which is needed in a public key system.

3.4 Authentication protocols

In the following sections different authentication protocols and techniques are presented. They are all described to be used in a wireless network.

3.4.1 MAC table

Most of the APs on the market supports a MAC address based authentication system. If the feature is used, a network administrator enters the MAC addresses of all allowed users in a table resided in the AP. When a STA tries to associate to the AP, the MAC address of the STA is look upped in the MAC table. If the MAC address is found the STA is authenticated and allowed to associate to the AP. Otherwise the STA cannot associate to the AP.

3.4.2 RADIUS

Remote Authentication Dial In User Service (RADIUS), defined in (Rigney et al., 2000), specifies a protocol for a client/server model where there is one central server which contains the list of allowed users. The server has a shared secret with all clients. This shared secret is never sent over the network but it is used to authenticate clients to the server. Figure 5 shows an example of a network where there are four users (STAs), two clients (APs) and one server.

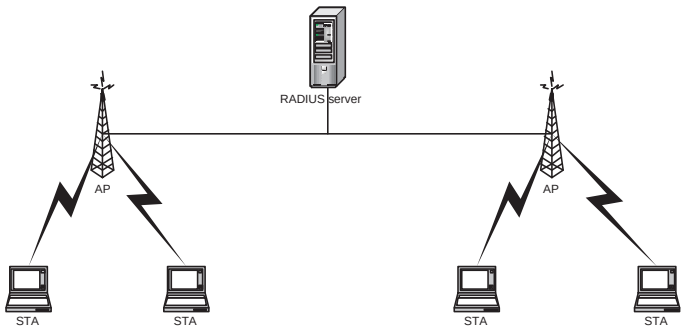

Figure 5: An example of a network where RADIUS is used.

When a user wants to connect to the network it first contacts the client, in this case the AP. The AP then sends an `Access-request` message to the RADIUS server containing attributes such as the user's name and password. If a password is sent then it is encrypted by a method based on the RSA Message Digest Algorithm MD5, defined by Rivest (1992). The client decides

what attributes that will be sent, which determines what kind of authentication method that will be used by the server. This is one nice property of RADIUS, namely that it is possible to choose between several different authentication methods. After the server has received the `Access-request` message the server runs the authentication process and determines whether the user should get access to the network. During this process the server and the client may exchange other messages if the authentication method requires that. If the authentication process was successful, an `Access-accept` message is sent back to the client, i.e. the AP. Otherwise an `Access-reject` message is sent.

However, Hassell (2003) describes that there is a security problem with RADIUS. Since when several proxy RADIUS servers are used, all data packets must be viewed and performed logic on at every hop.

Another disadvantage with RADIUS, as mentioned in Rigney et al. (2000) is the use of the User Datagram Protocol (UDP) as transport protocol. When RADIUS is used in a large scale system, the network could easily be congested since the lack of congestion control in UDP.

3.4.3 Diameter

Because of the two problems with RADIUS mentioned in section 3.4.2 and other design flaws, a new authentication protocol is needed. Probably will the Diameter protocol (Calhoun et al., 2003) be the next generation authentication protocol. This protocol will correct the design flaws in RADIUS and be backward compatible with it. Diameter uses the Transmission Control Protocol (TCP) or Stream Control Transmission Protocol (SCTP), (Stewart et al., 2000) instead of UDP as transport protocol. The protocol standard also specifies that either IP security (IPsec) or Transport Layer Security (TLS) must be used to provide security of messages. IPsec, defined in Kent and Atkinson (1998), provides security services at the Network layer in the Internet protocol stack. It uses symmetric keys to perform the cryptographic algorithms. However the shared keys may be distributed by using public key protocols. TLS, defined in Dierks and Allen (1999), on the other hand operates between the Transport layer and the Application layer, providing privacy and data integrity between two communicating entities. Both symmetric and public key cryptographic algorithms are supported and may be used by TLS. As with RADIUS, the Diameter protocol is a base protocol. The applications should therefore extend the protocol with the desired authentication method.

3.4.4 TACACS(+)

Terminal Access Controller Access Control System (TACACS), described in Finseth (1993), is a simple protocol for authentication. One of its disadvantages is that usernames and passwords are sent in plaintext. This makes it easy for an attacker to sniff the wireless network for collecting usernames and passwords. Therefore a newer version of this protocol was developed which was named TACACS+, defined in Carrel and Grant (1997). TACACS+ does not only encrypt the passwords as RADIUS do, it encrypts the whole packet. Another advantage with TACACS+ is that it separates the authentication, authorization and accounting which allows the usage of for example Kerberos authentication. Furthermore is TCP used as transport protocol. Since the developers of TACACS+ are people from Cisco, this protocol is mostly used in Cisco equipments.

3.4.5 Kerberos

When Kerberos (Kohl and Neuman, 1993; Steiner et al., 1998) was developed at MIT it was intended to provide authentication and security in the campus computing network at MIT and to other intranets. Today it is used by many companies and universities (Coulouris et al., 2001). Kerberos is partly based on the Needham and Schroeder authentication protocol (Needham and Schroeder, 1978). In their protocol they specify an authentication server which contains a list of users and their passwords. Everybody on the network must therefore trust the authentication server. In Kerberos there are two services on the authentication server, the authentication service and the Ticket Granting Service (TGS). The authentication service authenticates the client and replies to the client with a ticket to the TGS. The TGS receives the ticket from the client and checks its validity and replies to the client with a new ticket to the server the client wishes to make a request to. The hosts on the network are required to be loosely synchronized to handle replay attacks. If the synchronization is performed over the network the synchronization protocol must itself be secure.

3.4.6 802.1x

IEEE has developed a standard called 802.1x for authenticating and authorizing devices in an IEEE 802 LAN. The standard, defined in IEEE (2001), is a port-based network access control. A port in this context is either a physical connection to the LAN or a logical (e.g. an association between an AP and a STA in IEEE 802.11). The standard specifies three entities, the

supplicant and the authenticator (described in section 3.2) and the authentication server. In a WLAN the STA is the supplicant and the AP is the authenticator. At first when the supplicant wants to connect to the network it contacts the authenticator. Alternatively, the authenticator may initialize the authentication process. The authenticator has two logical ports, one called the uncontrolled port, that is only used to establish authentication and another called the controlled port, which only accepts packets from authorized devices. The supplicant therefore communicates via the uncontrolled port with the authenticator which asks for the supplicant's identity. The supplicant answers with its identity and the authenticator forwards it to the authentication server. It is up to the authentication server to decide which authentication protocol that should be used. The authentication server then performs the authentication process and optionally requests information from the supplicant, e.g. an encrypted challenge text. Depending on the result of the authentication process, the server sends either an accept or a reject message to the authenticator. If the authenticator receives an accept message it will then open the controlled port and let the supplicant use that port until the authenticator closes the port of some reason (e.g. the supplicant explicitly sends a logoff request or does not reauthorize within an expiry time). All authentication messages are sent using the Extensible Authentication Protocol (EAP), described below. Figure 6 shows the information that is exchanged during the authentication initiated by the supplicant.

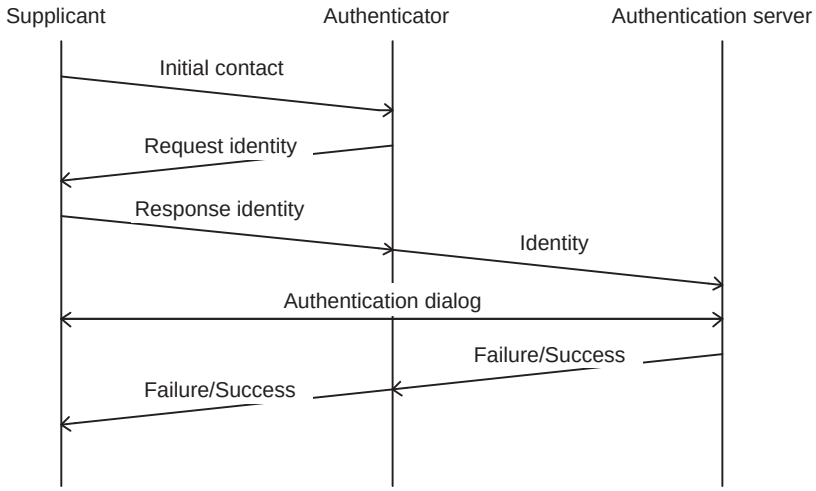

Figure 6: The flow of messages in 802.1x authentication.

The authentication is performed at the MAC layer, see Figure 2. The MAC address is therefore used to authenticate a device. After the supplicant has got access to the network, i.e. the authenticator has added the supplicant's

MAC address to a list of authorized devices, the authenticator only looks at the MAC address of the received packets to determine whether they should pass or be dropped.

EAP, defined in Blunk and Vollbrecht (1998), is a general protocol for Point-to-Point Protocol (PPP) authentication and supports several authentication methods. The PPP, (Simpson, 1994), defines how multi-protocol datagrams could be transported over a point-to-point link, e.g. between a STA and an AP. The EAP itself is very simple, the authenticator simply requests what information it wants from the supplicant and the supplicant responds with the requested data. Some of the supported authentication methods are identity check (e.g. by MAC address), Challenge-Handshake Authentication Protocol (CHAP) (Simpson, 1996) with MD-5 as encryption algorithm and One-Time Password (OTP) (Haller et al., 1998). However, the commonly implemented authentication method for EAP is the EAP-Transport Level Security (EAP-TLS), defined in Aboba and Simon (1999). This authentication method uses digital certificates to assure mutual authentication. However, this is only true if both the supplicant and the authentication server can validate the other's certificate. This could be accomplished by having both certificates issued by the same CA and providing the certificate of the CA to both participants.

Using EAP-TLS provides mutual authentication but this is not mandatory and even if it was, it could be bypassed as explained in Mishra and Arbaugh (2002). Furthermore, Mishra and Arbaugh (2002) clarifies that both a man-in-the-middle attack and a session hijacking could be carried out against a 802.11 network if 802.1x is used since in combination with each other they have some design flaws.

3.4.7 Wi-Fi Protected Access

The security mechanism in IEEE 802.11, Wired Equivalent Privacy (WEP), has proven to have cryptographic weaknesses (Walker, 2000). Therefore the Wi-Fi Alliance together with IEEE began to develop a specification for enhancing the security in WLANs. The specification is called Wi-Fi Protected Access (WPA) and is a subset of the IEEE 802.11i draft, which currently is under development but should have been standardized in September 2003. WPA is designed to run on existing hardware and to be forward compatible with IEEE 802.11i. The user authentication in WPA implements IEEE 802.1x and EAP, as described in section 3.4.6.

3.4.8 IEEE 802.11 MAC layer authentication

The IEEE 802.11 standard itself defines two types of authentication, namely open system and shared key. The simplest one is open system which is the default one. If the AP is configured to use the open system authentication then all STAs that want to associate to an AP will be authenticated by the AP. However it is possible for APs to decline STAs.

The other type of authentication, shared key, requires that the STA and the AP have a shared key and that the Wired Equivalent Privacy (WEP) encryption algorithm is implemented (specified in the IEEE 802.11 standard). The IEEE 802.11 standard specify that the key should be distributed via a secure channel independent of 802.11.

Figure 7 illustrates the operation of the shared key authentication method. When the STA wants to connect to the network it sends an authentication request to the AP. The AP will then responds with a random generated challenge text. The receiving STA will then encrypt the challenge text with the shared key and send the result back to the AP. If the AP successfully decrypts the encrypted challenge text then the STA is authenticated.

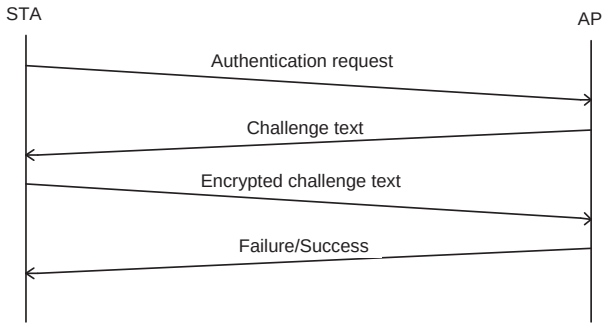

Figure 7: The flow of messages in the shared key authentication.

3.5 Simple Network Management Protocol

Simple Network Management Protocol (SNMP), (Case et al., 1990), was designed to help network administrators to manage, monitor and control the network. Since the devices in a network are spread over a wide area, a tool was needed by the administrators to remotely access the devices. The development of SNMP has evolved from version 1 to the more secure versions 2 and 3 (Case et al., 2002). However the enhancements in the later versions are out of scope in this paper and will therefore not be covered. The SNMP

specifies the operations that can be performed by an network administrator. The `GetRequest` and `GetNextRequest` are two examples of operations that exist in SNMP. These operations are used to get information from a network device.

The information in network devices that can be queried with SNMP operations, such as `GetRequest`, is defined by the Management Information Base (MIB). Different network devices have its own MIB and different vendors have also its own MIB. Therefore a standard is needed to define what a MIB may contain. The International Organization for Standardization (ISO) has specified a hierarchical structure, like a tree, for all kind of standardized objects. The tree is incredible large but SNMP typical only operates on a small fraction of it. Every node in the tree has both by a number and a name associated to it. A MIB object is identified by either the numbers or names of the nodes on the path from the root to the particular node in the tree.

3.6 Virtual Local Area Network

The IEEE has specified a standard called 802.1Q (IEEE, 2003), which defines Virtual Local Area Networks (VLANs). By using VLANs, nodes on different physical LANs can appear to be on the same LAN. Furthermore, it is possible to separate nodes on the same physical LAN to be grouped in different virtual LANs. In 802.1Q a special tag of four bytes is added to the ethernet frame. This tag identifies which VLAN the frame came from or should be sent to. However there exists frames that does not have this tag. These frames are called untagged and they are often originated from end stations which are not VLAN-aware devices. All switches in the network should be VLAN-aware devices which means that they can interpret and understand the special VLAN tag inserted in the ethernet frames. A VLAN-unaware device will drop all frames that contains the VLAN tag. There are two different ways of classifying a VLAN according to IEEE (2003), either by port or by port and protocol. In the port-based VLAN classification, an untagged frame is considered to belong to the VLAN associated to the port where the frame arrived. The other classification, port-and-protocol, determines the VLAN which the frame should be sent to by the port of arrival together with the protocol identifier of the frame.

3.7 Virtual APs

The vendors of IEEE 802.11 equipment have noticed the benefits of letting operators share APs with each other. Therefore, some of them have started to implement a feature in their APs that enables the AP to act as several virtual APs. One physical AP can simulate several virtual APs by using multiple SSIDs. Each SSID represents a virtual AP and is mapped to a specific VLAN. In this way it is possible to segment the network into different VLANs all the way out to the end users. This is not possible with an "ordinary" AP since all associated users must belong to the same VLAN. However, this technology is rather new which have lead to different implementations of virtual APs. Aboba (2003) explains that different vendors solve the multiple SSIDs problem in different ways. He describes four possible approaches to implement virtual APs. They are shortly described in the following sections.

The first approach is to include multiple SSIDs in each beacon and probe response and use one BSSID for all virtual APs. In this approach the beacon interval is unchanged, i.e. the same as if no virtual APs were present. The IEEE 802.11 standard does not explicitly forbid several SSIDs in beacons and probe responses. However most implementations of STAs does not expect more than one SSID and therefore might not behave well upon receiving these kinds of frames. This approach supports both passive and active scanning. Since the same BSSID is used for all virtual APs, a STA receives broadcast and multicast traffic from all virtual APs resided in the same physical AP. However the STA drops broadcast and multicast traffic from virtual APs which the STA is not associated with.

In the second approach a primary SSID is set, which is the only SSID present in beacons and probe responses to requests with SSID set to "ANY". However the virtual AP responds to probe requests where the SSID is set to any of the secondary SSIDs as well. This implies that passive scanning is not supported but active scanning is. Therefore it is not possible to discover all networks without knowing all SSIDs in advance. All virtual APs use the same BSSID which results in the same problem with broadcast and multicast traffic as occurred in the previous design.

The next approach is similar to the previous except that the virtual AP sends a beacon for each SSID. This approach supports probe requests with the SSID set to "ANY" and also passive scanning. If the original beacon interval was ΔT and there are N virtual APs then a passive scan would take N times longer to complete since the beacon interval for each virtual AP is $N\Delta T$. However all STAs times out after a predefined time and if this time-

out is too short all SSIDs will not be discovered. As the previous approaches this one also uses only one BSSID which will cause the same problem with broadcast and multicast traffic as described earlier.

The final approach solves all problems as the previous ones have suffered from. As the second and third designs, this one also include only one SSID in each beacon and probe response. Another similarity with the third approach is the use of a beacon for each SSID. However the beacon interval for a virtual AP is kept unchanged to ΔT. This implies N times more beacon traffic if the AP supports N SSIDs. The major difference with this approach compared with the others is however the use of multiple BSSIDs, one for each SSID. The hardware in the STAs now filters the broadcast and multicast traffic from the other virtual APs instead of letting the frames being processed by the software which discovers that the frames were not destined for the STA. This approach is clearly the superior one compared with the others presented.

3.8 SkellefteOpen.net in depth

An overview of SkellefteOpen.net was given in section 2.2. This section examines the technical aspect of SkellefteOpen.net in more detail. The logical view of SkellefteOpen.net is shown in Figure 8. To keep the figure perspicuous, all details are not present in the figure since they do not have any impact of a cooperation with another network.

The components located in the public service frame are all maintained by the administrators of SkellefteOpen.net. However the components in the ISPs frames are controlled by the ISPs themselfs. The DHCP relay agent receives all DHCP requests sent by the users. Since the relay agent is located on the same VLAN as all APs in the access network, the relay agent receives the MAC addresses of all users. These MAC addresses are stored in the MAC database together with the identity of the ISP the user chooses. A MAC address of a user that never has logged in is therefore not stored in the database. In that case the user gets an IP address from the DHCP relay with a very short lease time. The access relay has a web page where the user chooses the desired provider. If the user already has chosen a provider then the access relay redirects the user to the login page of the desired ISP and the user is given a new IP address from the DHCP server at the ISP.

Each ISP has to set up an own DHCP server. It is important that the server is not located on the same network as SkellefteOpen.net, since it would then interfere with the relay agent by answering DHCP requests directly to the

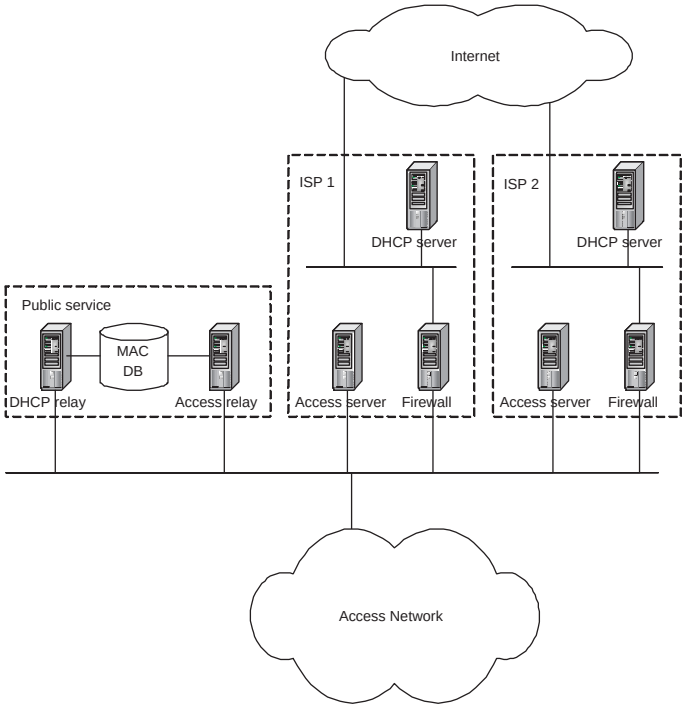

Figure 8: The logical view of SkellefteOpen.net, adopted from Andersson et al. (2003).

users. The access server provides the user with a login page where the user enters his/her credentials. Upon successful authentication the firewall will allow traffic from that user to pass through. The provider is free to use any kind of software for authentication of users. However, there is an authentication software called Oasis, developed and described in (Hedenfalk, 2001). This software is released as Open Source and could be downloaded from Hedenfalk (2002). Oasis communicates with the login page which is located on a Secure Sockets Layer (SSL) (Freier et al., 1996) secured web server. From the login page the user submits his/her credentials which are forwarded to Oasis via a UNIX socket. The actual authentication process is performed at the ISP. Oasis is using Pluggable Authentication Modules (PAM) to communicate with the authentication system at the ISP to do the authentication. PAM abstracts a number of authentication related operations and provides an interface for dynamical loaded modules which implements the authentication operations. It is therefore possible to chose whatever authentication scheme without modifying the software by simply load the module for the desired authentication type. See Morgan (2003) for more information about PAM.

A typical login procedure to SkellefteOpen.net by a new user is described in

the following algorithm.

1. The user associates to one of the APs.
2. The user must have DHCP enabled on his/her laptop to be able to acquire an IP address. The user sends a DHCP request which the relay agent receives. If the relay agent finds the MAC address of the user in the MAC database, then the relay agent forwards the DHCP request to the ISP's DHCP server. Otherwise the relay agent assigns an IP address to the user.
3. The user chooses a provider. The MAC address of the user is stored in the MAC database together with the identity of the ISP. The user is given a new IP address, this time from the selected ISP.
4. The user submits his/her credentials at the login web page of the ISP.
5. The credentials are sent from the web server to the Oasis, which tries to authenticate the user through PAM.
6. If the authentication was successful then Oasis opens the firewall for the user's IP and MAC addresses.
7. Oasis closes the firewall for the user's IP and MAC address when the user fails to respond to a specified number of ARP-ping probes.

4 Solution to the authentication problem

Since one of the requirement of the authentication is that it should be transperant to the customer, it is not acceptable to use any kind of user credentials that the user has to submit. Therefore an authentication method where the user submits his/her username and password is not a possible solution. Furthermore is IEEE 802.1x not supported by all operating systems which eliminates that authentication method. Although Diameter is designed specifically for WLANs it needs some time before vendors of APs adopt the protocol and implement it. The APs in Skekraft.net do not have support for this and will probably not get it since the vendor has not released any new software for the APs for a long time. This applies to WPA as well since the APs in Skekraft.net do not have support for WPA. TACACS+ is not supported by the APs either which rules out that protocol. The usage of Kerberos requires new software for the customers, making it a bad alternative. When several users connect to a network it is often done through a NAS. An AP could for example be a NAS. This is a consequence of the design of IEEE 802.11, which specifies that all traffic in the infrastructure mode must pass through the AP, as mentioned in section 3.1. Often there are several NASs which could be the case in IEEE 802.11. It would therefore be desirable for a network administrator to only administrate one "database" of allowed users. This make the MAC table alternative unsuitable.

4.1 Design of the authentication system

Consequently is RADIUS the only one that meets the requirements specified in section 2.3.1. Since user credentials cannot be used, the authentication has to be performed on MAC addresses. The big challenge with this solution is to determine which MAC addresses that should get access to the network. It is not feasible to demand that the customers shall inform Skellefeå Kraft what MAC address they are using. This would require to much work of the support staff and also be cumbersome for the customers. Therefore some kind of intelligence should gather the allowed MAC addresses and store them in a dynamical central database. Dynamical means that MAC addresses should be added and removed depending on whether they are allowed or not to get access to the network. An intelligent RADIUS server is therefore the preferably solution.

4.1.1 The intelligence

As mentioned above, the big challenge with this solution is to find out which MAC addresses that are allowed to get access to the network. As mentioned in section 2.1, both the customer and the corresponding ethernet converter will get an own IP address. This together with the fact that the customer's ethernet converter has the same MAC address as the customers network card can be used to set up a rule for network access. When a customer with a new MAC address tries to connect to the network, the two corresponding public and management IP addresses should be looked up. If the two IP addresses are found and they belong together then the MAC address is a valid one and the user gets network access. However if the two IP addresses does not match or at least one is not found, then should the user be denied network access. The user should however get a new chance to get network access in some time. When the user tries to connect after a while the user should be granted access if the requirements are fulfilled. Therefore a time-out function is needed.

4.2 Implementation of the authentication system

It would take an incredible large amount of time to write a RADIUS server from scratch. Therefore an existing one would be preferably. Since the software of the RADIUS server has to be modified, an open source RADIUS server is needed. The currently most maintained one is FreeRADIUS, which is one of the most modular and featureful RADIUS server today (The FreeRADIUS project, 2003). As platform, Trustix 2.0 was chosen. However during this Master's Thesis was carried out, Trustix went bankruptcy but the software project changed name to Tawie and is still maintained by the same developers (The Tawie Team, 2003). Tawie 2.0 has the Linux 2.4.22 kernel and only limited packages included. After the installation, no network services are running which forces the administrator to enable the ones that are needed. This makes the system safe at the startup and the administrator has a good control over the system. The dynamic database is a MySQL database running on the same machine as FreeRADIUS. Another database which stores information about the customers, e.g. relationships between public and management IP addresses, is running on another machine. That database is accessed through a TCP/IP connection. The open source project NET-SNMP, (The NET-SNMP Project, 2003), is used to send and receive SNMP queries to the router closest to the AP for retrieving MAC and IP addresses. Since that software supports several SNMP operations, it is stripped down and modified to fit to this project. Figure 9 shows the logical view of

the system.

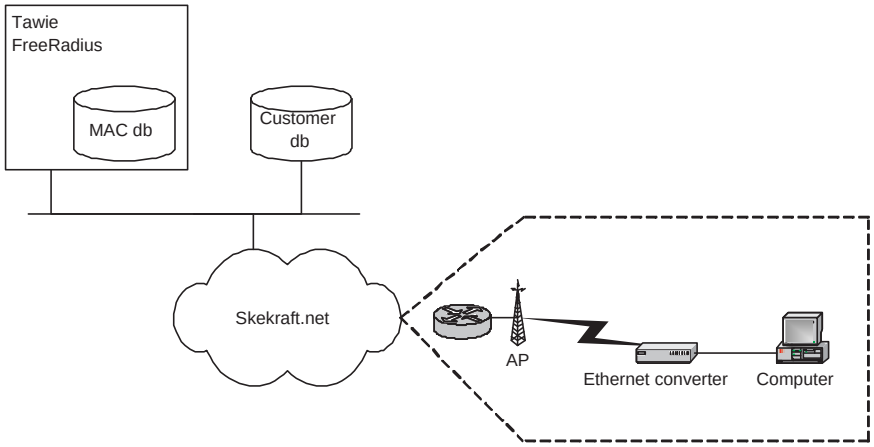

Figure 9: The logical view of the authentication system.

The APs are located in the Skekraft.net and configured to ask the RADIUS server if a MAC address of a user is allowed to get access. The MAC and IP addresses of a customer and his/her ethernet converter and the used AP are all stored in the same router as shown in Figure 9. The SNMP queries are therefore sent to that router.

4.2.1 The intelligence

At the beginning is the dynamic database with allowed MAC addresses, called allowed_MAC_list, empty. There is also an initially empty database with disallowed MAC addresses, called disallowed_MAC_list. A separate database, called IP_list, stores information about which ethernet converter a particular customer has. The following algorithm describes how a STA, with a MAC address called MAC_X, is authenticated.

1. If MAC_X is in the allowed_MAC_list, then the STA is authenticated and gets access to the network. The algorithm terminates.
2. Else if MAC_X is in the disallowed_MAC_list and has not yet been there for 30 minutes, then deny the STA access to the network. The algorithm terminates.
3. Else if MAC_X is in the disallowed_MAC_list and has been there for at least 30 minutes, then remove the MAC address from the disallowed_MAC_list.

4. Allow the STA access to the network.
5. Find the IP address of the ethernet converter that has MAC_X by asking the closest router with a SNMP `GetRequest`.
6. If the router does not know which IP address the ethernet converter with MAC_X has.
   (a) Send a ping request to all ethernet converters on the same subnet as the AP.
   (b) Ask the router again with a SNMP `GetRequest` for the IP address of the ethernet converter with MAC_X.
   (c) If the router still does not know the IP address of the ethernet converter with MAC_X, then add MAC_X to disallowed_MAC_list. The algorithm terminates.
7. Get the IP address of the customer that corresponds to the IP address of the ethernet converter by looking in IP_list, call that IP address IP_user.
8. If there is no such IP_user, then add MAC_X to disallowed_MAC_list. The algorithm terminates.
9. Send a ping request to the IP_user.
10. Find the MAC address of the customer with IP address IP_user, called MAC_user, by asking the closest router with a SNMP `GetRequest`.
11. If the router does not know MAC_user, then add MAC_X to disallowed_MAC_list. The algorithm terminates.
12. If MAC_X is equal to MAC_user, then add MAC_X to allowed_MAC_list, else add MAC_X to disallowed_MAC_list.

According to the algorithm, a MAC address that is not present in any of the lists or has been present in disallowed_MAC_list for more than 30 minutes must get network access to be authenticated. However if the MAC address turns out to be a disallowed one then it will be denied access after a maximum of 15 minutes when the authentication process repeats itself by initiative of the AP.

By asking a router for a special address with a SNMP `GetRequest`, results in receiving all known addresses by the router. Therefore if the address is not

in the returned set then the router must be forced to ask the host with that address. This is done by the Address Resolution Protocol (ARP) (Plummer, 1982). However ARP cannot be used between different ethernet networks. Therefore a higher layer protocol must be used to force the router to use ARP to get the requested address. This is done by sending an Internet Control Message Protocol (ICMP), (Postel, 1981), echo request to the IP address. The ping program sends such a echo request. However it is not possible to ping a MAC address from a remote sub network. Therefore all hosts on that sub network has to be pinged to be sure that the router gets the requested address. Then the router stores all IP addresses and their corresponding MAC addresses of a particular sub network in its ARP table. If a following SNMP `GetRequest` does not return the right address then it is not present on the sub network.

The time a MAC address will be disallowed does not necessarily have to be 30 minutes. It is possible to set it to any other value. However it may be necessary to change the ARP timeout value on the affected routers to the same value since the router may keep the ARP entry of an IP address until it times out. This has been tested and yielded different result from time to time.

5 Solution of sharing APs

This section describes the designs and implementations of how to share APs with SkellefteOpen.net. Three different solutions are first presented. Two implementations of them are later described. Unfortunately all solutions requires that new hardware has to be bought since the existing equipment does not have the required functionality.

5.1 Design of how to share APs

As described in section 2.1, Skekraft.net is a routed network. This means that several LANs are connected to each other by routers. Therefore MAC layer broadcast and multicast traffic is not transmitted to other nodes than the ones on the same broadcast domain, i.e. LAN, as the sender. SkellefteOpen.net on the other hand is a single layer two network where all nodes are on the same LAN. However the nodes in SkellefteOpen.net are located on physical different broadcast domains but by using a VLAN all nodes appears to be located on the same LAN. This difference between the two networks is essential and puts constraints on the solution of the cooperation task. Three proposals of how to share the access points are presented in sections 5.1.1, 5.1.2 and 5.1.3.

5.1.1 Two radio cards in one AP

With two radio cards located in the same AP, it is possible for some APs to define two SSIDs and VLANs. Each radio card has its own SSID and VLAN configured. Currently there are APs on the market that support this functionality but unfortunately not the ones used in Skekraft.net. However, when two radio cards are used on the same AP they are configured to use different channels. This solution is therefore almost as bad as setting up another AP just next to the existing one. The only benefits are probably that this will be a little cheaper and it saves one connection to the wired backbone.

5.1.2 Two SSIDs and VLANs

A clean and elegant solution would be to use the concept of virtual APs as described in section 3.7. However the existing APs does not support this feature. Therefore new equipment has to be invested. Since the APs vendors have adopted different techniques of how to solve the problem it is important

to investigate which ones that support the desired functionality. Most of the ethernet converters are configured with the SSID "ANY". Therefore it would be desired that the APs will respond to probe request with SSID set to "ANY". However the SSID used by Skellefteå Kraft does not need to be advertised since passive scanning is not used by the ethernet converters. On the other hand must the SSID SkellefteOpen be advertised in order for the SkellefteOpen users to easily find and select the correct SSID. Since the lack of documentation from several vendors it is difficult to know in advance what the APs is capable to do. If the reseller do not know how the APs work an evaluation of different APs would be desired. However this is outside the scope of this paper.

5.1.3 One SSID and tunneling

Another approach to share APs with SkellefteOpen.net is to use the same SSID and tunnel data to the backbone of SkellefteOpen.net. This solution focuses on how to share an AP owned by Skellefteå Kraft with SkellefteOpen but not vice versa. Since the ethernet converters of Skellefteå Kraft's customers are configured to associate to any SSID, it would be possible to change the SSID of Skellefteå Kraft's APs to SkellefteOpen. However, this could imply some problems if some ethernet converter has to be configured to associate to a specific AP. A possible scenario would then be that the ethernet converter associates to an AP that only supports SkellefteOpen.net.

All traffic from Skellefteå Kraft's APs should not be tunneled. Therefore, a filter is needed which selects the data from the APs that should be tunneled to SkellefteOpen.net and leaves the other traffic untouched. A schematic picture of this filtering is shown in Figure 10.

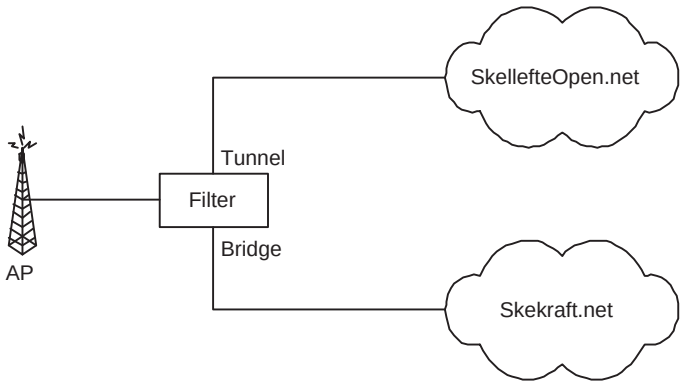

Figure 10: A schematic view of the filter.

The major problem with this solution is to decide how to filter the data. Since the MAC addresses of neither the Skellefteå Kraft or SkellefteOpen users are known in advance, it is not possible to determine the filtering rules by the source MAC addresses. The IP addresses of Skellefteå Kraft's customers are however known, but all data packets does not contain source IP addresses, e.g. ARP requests. Therefore the filtering must be based on the destination address. Since the destination IP addresses of data packets are arbitrary, the destination MAC address is the only remaining alternative. All packets from SkellefteOpen users that is not destined to other SkellefteOpen users must pass one of the public service machines or some ISP gateway. The MAC addresses of these four machines are known in advance and could be used to filter the data traffic form the APs. Let us call the four MAC addresses A, B, C and D. A possible filtering rule would then be to tunnel all broadcast and multicast traffic and packets with destination MAC address A, B, C or D.

5.2 Implementation of multiple SSIDs and VLANs

The best and easiest solution to the cooperation problem is to use two different SSIDs for one AP and in that way create a virtual AP. However, the APs in Skekraft.net does not support this. Neither does the APs in SkellefteOpen support multiple SSIDs but they could be upgraded to 802.11g and in that way get the desired support. Unfortunately, if the APs in SkellefteOpen are upgraded they still are not configurable to use different RADIUS servers for different SSIDs. This is a desirable feature since the authentication in Skekraft.net is performed with the RADIUS protocol. The ultimate feature of a virtual AP would be to assign a specific IP address to it and let the administrator control all parameters of it independently of the other virtual APs. However, this is neither possible.

Since the authentication of Skekraft.net users is performed with RADIUS, the APs should be configured to use this. However, one problem arises from this, SkellefteOpen.net users should not authenticated by the RADIUS server at Skekraft.net. Therefore, the RADIUS server must know to which network the users belong. The best solution would be if the AP could include in the RADIUS request packet which SSID the user has used. The RADIUS server would then respond with an `Access-accept` message if the SSID is SkellefteOpen. A minor modification of the RADIUS server would then be necessary.

5.3 Implementation of tunneling

The fully implementation of the solution using a tunnel and a bridge could unfortunately not be performed since a limitation of the used software. However the tunneling part of the solution was implemented in order to compare the overhead of using a tunnel instead of a VLAN. This section describes how the full implementation should be carried out. Furthermore is the limited implementation described.

The idea behind this solution is to connect a machine, called the client, near the AP that will be shared and another machine, called the server, at SkellefteOpen.net. These two machines will function as tunnel endpoints and as bridges, see Figure 11. The connection between Skekraft.net and SkellefteOpen.net is not the only one as the figure shows. Therefore all traffic between the two networks does not need to and cannot pass the server machine between them.

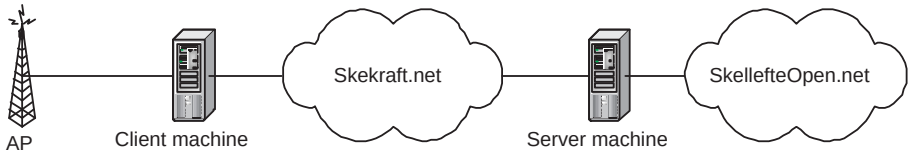

Figure 11: Logic view of the tunneling and bridging.

5.3.1 Full implementation

The client machine should bridge traffic between Skellefteå Kraft's customers and Skekraft.net. However the traffic from SkellefteOpen users should be tunneled to and from the server machine which bridges the traffic to and from SkellefteOpen.net. The functionality of the server machine is simple, bridge all incoming traffic from the tunnel to the SkellefteOpen.net and tunnel all traffic from SkellefteOpen.net to the remote tunnel endpoint. The functionality of the client machine is however slightly more advanced. It has to decide which traffic to tunnel and which to bridge.

Both machines are running FreeBSD, (The FreeBSD Project, 2003), as operating system. It has some useful features that can be used to set up tunnels and bridges. To set up a tunnel, the Virtual Tunnel (`VTun`), (Krasnyansky, 2003), software is used. This software can be configured to tunnel ethernet traffic over an IP network. Other properties that `VTun` have are encryption, compression and traffic shaping. However none of them are used but at

least encryption should be used in order to keep the tunnel secure against eavesdropping for example.

In order to decide which traffic to bridge and which to tunnel, the program `tcpdump` can be used. This packet sniffer prints the headers of all captured packets. If it is given the option `-ddd`, it dumps a packet matching code for a given pattern that can be given to a Berkeley Packet Filter (bpf) to filter packets. To control where matched and unmatched packets should be forwarded by the bpf, the `Netgraph` software is used. In this graph based kernel networking subsystem there exists nodes, e.g. network interfaces and bpfs, and hooks which connects the nodes with each other. If two nodes have for example one hook each which are connected to each other, then there is an edge between the two nodes. Figure 12 shows how the netgraph looks like in the client machine.

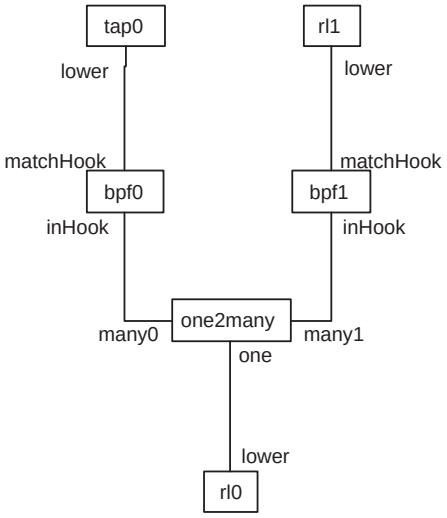

Figure 12: The netgraph at the client machine.

In the figure, rl0 and rl1 symbols the two network cards in the machine where rl0 is connected to the AP and rl1 to Skekraft.net. The tap0 node is a ethernet tunnel software network interface which is used by `VTun` to send and receive packets over the tunnel. The one2many node transmits all raw ethernet frames that are received by the hook named *one* to all other connected hooks, named *many0* and *many1*. However ethernet frames received by any of the *many* hooks are only forwarded to the *one* hook. The two bpfX nodes filters ethernet frames and only forwards those that matches the filtering rules. Each hook on the bpfs are configured with a bpf code which determines which packets to forward.

The netgraph in the server machine is shown in Figure 13. There is only one network interface, xl0, in this one except for the tap0 interface. The xl0 network interface is connected to SkellefteOpen.net and another is connected to Skekraft.net. However the other one is not used by netgraph. The two interfaces, xl0 and tap0, are connected to each other by their *lower* hooks.

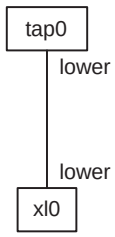

Figure 13: The netgraph at the server machine.

5.3.2 Limited implementation

Unfortunately when the above implementation was tested it turned out that it was not as simple as it looked. The problem arises from the fact that all incoming ethernet frames from Skekraft.net to the client machine passes the rl1 interface. The design of the solution is however based on that the tunneling packets are delivered to the virtual tap0 interface. A way to forward all tunneling packets to the tap0 interface would therefore solve the problem. This can be done by connecting bpf1 to tap0. However by doing this, all outgoing packets from tap0 are delivered to the bpf1 node instead to the VTun software.

To be able to test the tunneling mechanism, a limited implementation was carried out instead, where the rl1 interface was not included in the netgraph. This way the client machine functions the same as the server machine with the exception of the filtering of packets. Figure 14 shows the netgraph for this implementation.

This limited implementation disables therefore the Skekraft.net functionality of the AP. The limited implementation is therefore not useful in a large scale deployment. However the test of it can be used to compare the performance with the virtual AP solution.

UDP was chosen as transport protocol in the tunnel and no encryption, compression or traffic shaping was enabled. The reason to use UDP instead of TCP as transport protocol is simple. If the user data is sent with TCP

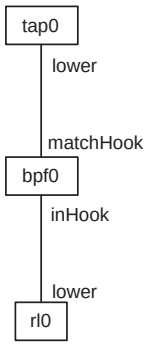

Figure 14: The netgraph at the client machine in a limited implementation.

then would every acknowledgment be acknowledged by TCP in the tunnel which is clearly unnecessary. However if the user data is sent with UDP, then may the data be lost any way.

6 Test results

The authentication system and the solution that uses two SSIDs and VLANs to share APs have been tested. To compare the solution that utilizes two SSIDs and VLANs with the one that uses a tunnel, the limited implementation of the solution based on tunneling was tested.

6.1 Authentication system

The authentication system has been tested on one AP in Skellefteå Kraft's network. An AP with only two users were selected since if a failure of the system would occur it would not cause trouble for many users. A third user was simulating an intruder with a laptop.

When the AP was reconfigured to use RADIUS authentication it had to be restarted, causing all users to reconnect to the AP. However, this is not seen by the end users since the ethernet converter handles this. One of the customers that was using the AP had his computer switched off but his ethernet converter was on. The RADIUS server did therefore not approve the MAC address of the customer since no public IP address were found in the router acting as the customer's gateway. The consequence of this is that the customer's MAC address was banned for a time. When the customer powers on his computer he has to restart the ethernet converter, causing it to reconnect to the AP. This time the RADIUS server will find the correct IP address and add the customer's MAC address to the list of approved users.

The other customer had both the computer and ethernet converter up and running. Therefore, the RADIUS server authenticated the customer at once.

The simulated intruder was given network access at first, as the authentication system is supposed to do. However when the AP reauthenticated the intruder, he was denied access to the network.

Since the router that the customers were using as the default gateway was shared by many other users, a large ARP table was retrieved from the router. Most of the time it took for the RADIUS server to authenticate the users was spent on transferring the ARP table from the router to the RADIUS server. This is clearly a performance bottleneck of the authentication system.

6.2 Sharing APs

The throughput of two users, one belonging to Skekraft.net and the other to SkellefteOpen.net, were measured with the performance measurement tool Distributed Benchmark System (DBS) (Murayama, 1998). DBS is also a traffic generator which sends data in a predefined way. Information about the data sent and received is logged and written to a file which can be used to produce graphs for visualizing throughput over time. DBS is easy to set up by using a single command file for all TCP or UDP connections. The command file specifies which TCP and UDP connections that will be set up and measured. In the command file, it is possible to define how much data or the duration of the data transmission. At each host where a TCP or UDP connection will exist, a daemon is started. The controller of all connections is a process running on one of the hosts or on any other host. The controller starts the course of events by sending information about the connections to the daemons. At a specified time the daemons start the data transmissions. This implies that the clock at the hosts have to be synchronized to achieve correct results. When the data transmissions have ended the daemons send their log files to the controller which stores them on disk.

To synchronize the internal clocks at the hosts, `ntpdate` was used. This program is an implementation of the Network Time Protocol (NTP) Mills (1995).

The transmit rate of the network cards should have been set to the same fixed rate. However the ethernet converter had only the three undefined values *low*, *medium* and *high* apart from the the *auto* mode. Therefore, both the network card of the SkellefteOpen user and the ethernet converter were configured to use *auto*. By using the *auto* mode, the network senses at which rate it is possible to send data.

6.2.1 Virtual APs

A testbed was set up to test this solution. The AP, an Orinoco AP-600, was configured with two SSIDs, skekraft and TestNet4. These two SSIDs were then mapped to two different VLANs. However, the AP was not configured to use RADIUS authentication since then would all users be authenticated by the RADIUS server. If the AP had included what SSID or VLAN the user belongs to in the RADIUS packets then it would have been possible to enable RADIUS authentication. The server would then simply respond with an `Access-accept` message for all SkellefteOpen users but perform the

normal authentication process for all other users.

The following scenarios were tested when the AP was configured with two SSIDs and VLANs.

1. A Skekraft.net user sending data to a computer on a different network.
2. A SkellefteOpen.net user sending data to a computer on a different network.
3. A Skekraft.net and a SkellefteOpen.net user sending data to the same computer on a different network.
4. A Skekraft.net and a SkellefteOpen.net user sending data to each other.

All users were associated to the tested AP. However the computer resided on a different network was located at Umeå University, 130 kilometers away. The four scenarios were tested twice, one time using TCP as transport protocol and the other time using UDP as transport protocol.

The first scenario yielded some strange results when TCP was used, as shown in Figure 15. The throughput is fairly constant except for the spikes after the throughput has dropped to zero. The mean throughput was 0,2579 Mbps. TCP normally increases the transmit rate as more data is received by the end host. However the figure shows the opposite. A possible explanation to this might be that the data is buffered at some point between the user and the foreign network. If so, the buffered data might be sent very quickly when it eventually is sent.

When UDP was used instead of TCP in the first scenario, the mean throughput dropped slightly to 0,2315 Mbps. However the mean throughput at the sending node was significantly increased to 9,4122 Mbps. This is a consequence of the design of UDP since it just sends data without checking if it reaches the destination. Figure 16 shows the throughput at the sending node.

The second scenario did not yield the strange results as the first scenario did. This probably depends on that the data is taking another way than in the first scenario and does not get bufferd in the same way. The data has to pass the backbone of SkellefteOpen.net, which has another Internet connection than Skekraft.net has. The mean throughput when TCP was used was 0,4845 Mbps, i.e. higher than the first scenario. Figure 17 shows the throughput over time at the receiving node.

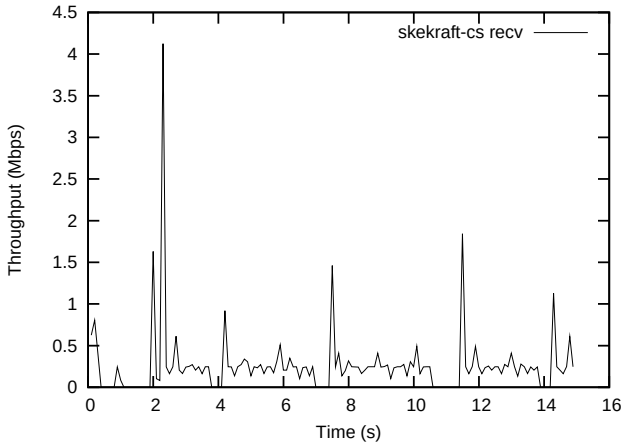

Figure 15: Throughput at the receiving host when a Skekraft.net user is sending.

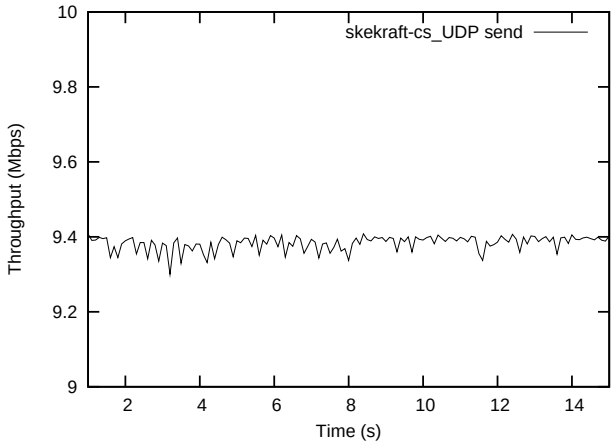

Figure 16: Throughput at the sending host when a Skekraft.net user is sending data using UDP.

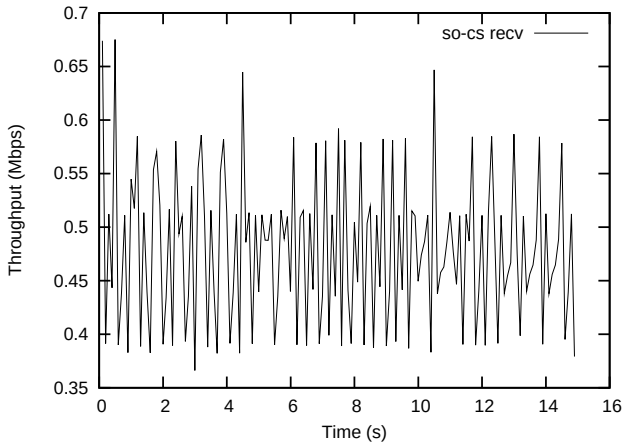

Figure 17: Throughput at the receiving host when a SkellefteOpen.net user is sending.

The same behavior as in the first scenario occurred when TCP was replaced by UDP in the second scenario. The sending node had a constant high throughput, 5,0686 Mbps. However it was not as high as in the previous scenario. This indicates that the ethernet converter and the network card in the laptop used by the SkellefteOpen user did not operate with the same transmit rate.

In the third scenario, which is like scenario one and two together, the same behavior occurred. Namely, when TCP was used throughput spikes were registered for the Skekraft.net user. Figure 18 clarifies this. The mean throughput for the two users was about the same as in scenario one and two. This implies that the performance did not degrade by the fact that there were two sending nodes at the same time.

The use of UDP instead of TCP in the third scenario showed that the ethernet converter operated with a higher transmit rate than network card in the SkellefteOpen's laptop, at least when UDP was used. Almost the same mean throughput was achieved in this scenario as in the two previous.

During the fourth scenario when both users were sending to each other using TCP, the throughput of the two users were almost the same, see Figure 19. This is a consequence that TCP is fair, i.e. all hosts that are TCP will eventually get the same throughput. The mean throughput at the receiving end were 0,3423 Mbps and 0,3494 Mbps respectively.

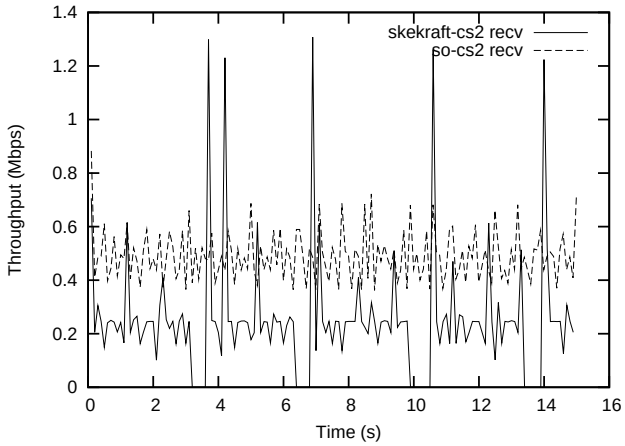

Figure 18: Throughput at the receiving host when both a Skekraft.net and a SkellefteOpen.net user are sending.

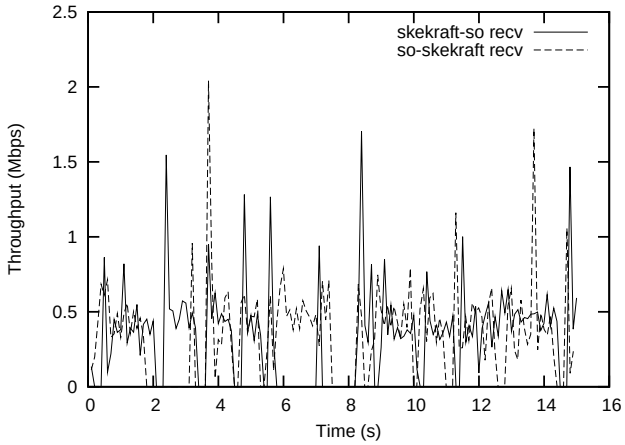

Figure 19: Throughput at the receivers when both users are sending to each other using TCP.

When UDP was used in the fourth scenario, a poor performance was achieved, as shown in Figure 20. The mean throughput at the receivers were 0,0667 and 0,0264 Mbps respectively. As UDP is not fair, this is not a suprising result.

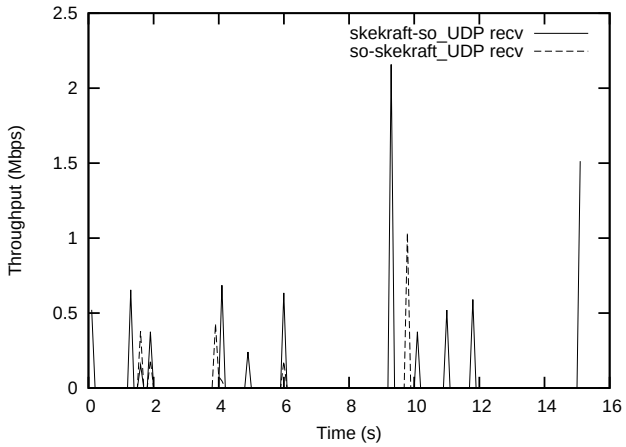

Figure 20: Throughput at the receivers when both users are sending to each other using UDP.

The AP that was used did not advertise any of the two SSIDs it was configured with. Therefore the clients had to be configured manually with the correct SSID.

6.2.2 Tunneling and bridging

Since only a limited implementation of the solution based on tunneling and bridging was performed, the second scenario described in 6.2.1 could only be carried out. However, that scenario, where a SkellefteOpen user is sending data to a different network is suitable for comparing the two solutions.

Figure 21 shows the throughput at the receiving host located at Umeå University when TCP was used. The mean throughput is only 0,1194 Mbps compared with 0,4845 Mbps which was achieved in the solution based on virtual APs. In this solution all data has to be encapsulated and decapsulated in order to get the data to the destination. This degrade the performance essential.

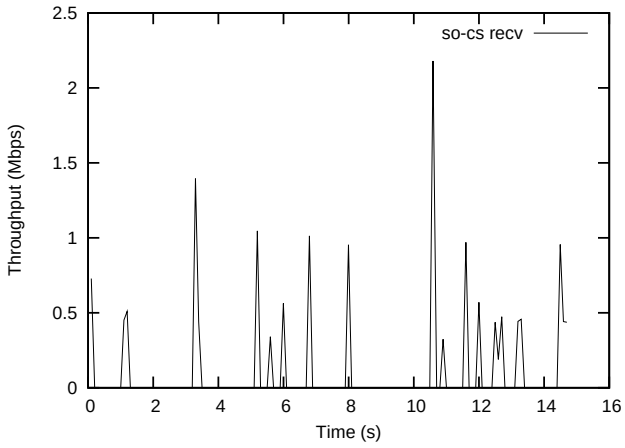

Figure 21: Throughput at the receiver when a SkellefteOpen user is sending using TCP, tunnel solution.

The use of UDP instead of TCP did actually increase the performance as Figure 22 shows where the mean throughput is 0,2545 Mbps. However the achieved throughput is only about half as high as in the virtual AP solution. When UDP is used no acknowledgements are sent over the tunnel, no retransmission is either performed.

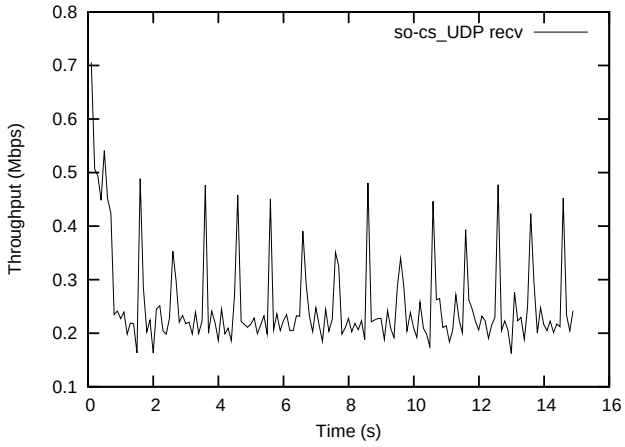

Figure 22: Throughput at the receiver when a SkellefteOpen user is sending using UDP, tunnel solution.

The same transmit rate was used in this solution as the in the previous one. Furthermore was the same hardware and software used to get as comparable results as possible.

7 Discussion and future work

The authentication system passed the small scale test, where only two users were involved. However before a large scale deployment can be performed, a test with more users should be carried out. As mentioned in the test result section, it took a long time to retrieve the ARP table. Therefore a clever algorithm should be designed and implemented. This new algorithm should then use the retrieved ARP table more than once before it is dropped. This will improve the performance significantly when or if the authentication is deployed because many customers have the same default gateway. A possible solution would be to wait to retrieve the ARP table until for example ten new users are unauthenticated or they have been waiting to be authenticated for ten minutes. That is however a suggestion to future work.

If the authentication system will be deployed it should be done at one AP at a time to not jeopardize a system crash on the server. Furthermore should the concerned customers be notified that they maybe have to restart their ethernet converters in order to get network access.

In order to make the authentication system redundant, a secondary RADIUS server should be configured. This server will authenticate users when the primary sever does not have enough resources or is unable to answer authentication requests. Almost all APs have support to be configured with a secondary RADIUS server. However, a design decision has to be taken concerning where to place the dynamical database. Should it be located on the primary server or secondary server or maybe on a different host? If the database is resided on one of the servers and that host goes down then will the database be unreachable. Probably is the best alternative to place the database on a different host or to let each server have its own database. However in the latter case some kind of synchronization of the two databases has to be performed to avoid inconsistency.

When modifying a large and complex software there are a lot of things to consider. The slightest change in one place could cause a devastating consequence in another. Furthermore, it is a bit cumbersome to update FreeRADIUS to a newer version since the software has been updated in a couple of places. It would therefore be desirable to write some kind of module which can be loaded in the FreeRADIUS server after an update. However this would probably still require some hacking the server in order to get it to work. This is also a suggestion to future work.

If the customers turn off their ethernet converter, an intruder can pretty easy

connect to Skekraft.net. However the customers were informed when they bought the equipment that they should keep their ethernet converters on in order for the administrators to access it remotely. The overall advantage with the authentication system is that possible intruders have significally harder to connect to Skekraft.net. A completely secure network is difficult to achieve without the use of user credentials. Another advantage is that the customers does not notice the authentication and therefore are not bothered with usernames and passwords.

Since Skellefteå Kraft is extending their broadband via optofibre cable, the demand for their WLAN product will maybe drop. This should be beard in mind when the cooperation with SkellefteOpen is considered. The superior solution to the cooperation with SkellefteOpen is namely the utilization of virtual APs which require new APs. The test results clearly showed that this solution has better performance than the solution based on tunneling. Furthermore the two networks are separated all the way to the end users. This is actually the most important advantage since no traffic is sent to both networks. If for example both networks would provide IP addresses by DHCP, then will the end users only get one response to a DHCP request. If the solution based on tunneling had been used, it would be impossible to separate the DHCP traffic correctly.

The difference of the throughput between a Skekraft.net user and a SkellefteOpen.net user sending data to a host at Umeå University could be due to that the data are transferred different ways. However this is something in the periphery of this Master's Thesis.

The vendors of APs are still at an early stage in developing APs with support for multiple SSIDs and VLANs. The tested AP, an Orinoco AP-600, is clearly not fully developed since the lack of many useful features. For example should the RADIUS packets at least contain information about what SSID the user is using. If this information was provided, the authentication system could be used. The AP did neither advertise any of its configured SSIDs which makes it a bad choice for SkellefteOpen. Hopefully will the vendors agree on a standard of this concept of virtual APs and implement them. The concept is very useful and could be used in many situations. For example could someone build the infrastructure needed for a WLAN at an airport. ISPs or other providers could then lease this infrastructure in order to provide its services to their customers.

References

Aboba, B. (2003). Virtual Access Points. Technical report, The Institute of Electrical and Electronics Engineers. http://www.ieee2.org/11/-Documents/DocumentHolder/3-154.zip; accessed November 11, 2003.

Aboba, B. and Simon, D. (1999). PPP EAP TLS Authentication Protocol. RFC 2716, Internet Engineering Task Force. http://www.ietf.org/-rfc/rfc2716.txt; accessed August 12, 2003.

Andersson, M., Engman, J., Karlsson, T., Lilja, N., and Stridfeldt, M. (2003). OpenLINC, a solution for expanding the Open.Net community. Webpage. http://www.beam.to/openlinc; accessed September 2, 2003.

Blunk, L. and Vollbrecht, J. (1998). PPP Extensible Authentication Protocol (EAP). RFC 2284, Internet Engineering Task Force. http://-www.ietf.org/rfc/rfc2284.txt; accessed August 12, 2003.

Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and Arkko, J. (2003). Diameter Base Protocol. Internet-draft, Internet Engineering Task Force. http://www.ietf.org/rfc/rfc3588.txt; accessed October 27, 2003.

Carrel, D. and Grant, L. (1997). The TACACS+ Protocol Version 1.78. Webpage. ftp://ftpeng.cisco.com/pub/tacacs/tac-rfc.1.78.txt; accessed September 4, 2003.

Case, J., Fedor, M., Schoffstall, M., and Davin, J. (1990). Simple Network Management Protocol (SNMP). RFC 1157, Internet Engineering Task Force. http://www.ietf.org/rfc/rfc1157.txt; accessed October 27, 2003.

Case, J., Mundy, R., Partain, D., and Stewart, B. (2002). Introduction and Applicability Statements for Internet Standard Management Framework. RFC 3410, Internet Engineering Task Force. http://www.ietf.org/-rfc/rfc3410.txt; accessed October 27, 2003.

Coulouris, G., Dollimore, J., and Kindberg, T. (2001). *Distrubuted Systems, Concepts and Design*. Addison Wesley, Harlow England, third edition.

Dierks, T. and Allen, C. (1999). The TLS Protocol Version 1.0. RFC 2246, Internet Engineering Task Force. http://www.ietf.org/rfc/rfc2246.txt; accessed November 3, 2003.

Diffie, W. and Hellman, M. E. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654.

Finseth, C. (1993). An Access Control Protocol, Sometimes Called TACACS. RFC 1492, Internet Engineering Task Force. `http://www.ietf.org/rfc/rfc1492.txt`; accessed September 4, 2003.

Freier, A. O., Karlton, P., and Kocher, P. C. (1996). The SSL Protocol Version 3.0. Internet draft, Internet Engineering Task Force. `http://wp.netscape.com/eng/ssl3/draft302.txt`; accessed November 18, 2003.

Haller, N., Metz, C., Nesser, P., and Straw, M. (1998). A One-Time Password System. RFC 2289, Internet Engineering Task Force. `http://www.ietf.org/rfc/rfc2289.txt`; accessed August 20, 2003.

Hassell, J. (2003). *RADIUS: Securing Public Access to Private Resources*. O'Reilly & Associates, Inc., Sebastopol, CA, USA.

Hedenfalk, M. (2001). Access Control in an Operator Neutral Public Access Network. Master's thesis, The Royal Institute of Technology. `http://www.e.kth.se/~e97_mhe/thesis/`; accessed August 27, 2003.

Hedenfalk, M. (2002). Open.Net :: main. `http://software.stockholmopen.net/`; accessed November 18, 2003.

IEEE (1999). Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Technical report, The Institute of Electrical and Electronics Engineers. `http://standards.ieee.org/getieee802/download/802.11-1999.pdf`; accessed August 13, 2003.

IEEE (2001). Standards for local and metropolitan area networks: Port-Based Network Access Control. Technical report, The Institute of Electrical and Electronics Engineers. `http://standards.ieee.org/getieee802/download/802.1X-2001.pdf`; accessed August 12, 2003.

IEEE (2003). IEEE Standards for Local and metropolitan area networks: Virtual Bridged Local Area Networks. Technical report, The Institute of Electrical and Electronics Engineers. `http://standards.ieee.org/getieee802/download/802.1Q-2003.pdf`; accessed November 17, 2003.

Kent, S. and Atkinson, R. (1998). Security Architecture for the Internet Protocol. RFC 2401, Internet Engineering Task Force. `http://www.ietf.org/rfc/rfc2401.txt`; accessed November 3, 2003.

Kohl, J. and Neuman, C. (1993). The Kerberos Network Authentication Service (V5). RFC 1510, Internet Engineering Task Force. `http://www.ietf.org/rfc/rfc1510.txt`; accessed August 18, 2003.

Krasnyansky, M. (2003). VTun - Virtual Tunnels over TCP/IP networks. `http://vtun.sourceforge.net/`; accessed November 25, 2003.

Kurose, J. F. and Ross, K. W. (2001). *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison Wesley, Boston MA.

Ljungberg, E., Fransson, J., Westin, A., and Wiklund, J. (2002). Final Report SkellefteOpen. Webpage. `http://2g1319.ssvl.kth.se/ csd2002-skellefteopen/`; accessed August 27, 2003.

Mills, D. (1995). Improved Algorithms for Synchronizing Computer Network Clocks. In *IEEE Transactions Networks*, pages 245–254.

Mishra, A. and Arbaugh, W. A. (2002). An initial security analysis of the IEEE 802.1X standard. Technical report, University of Maryland, Department of Computer Science. `http://www.cs.umd.edu/Library/TRs/-CS-TR-4328/CS-TR-4328.ps.zip`; accessed August 12, 2003.

Morgan, A. G. (2003). A Linux-PAM page. Webpage. `http://-www.kernel.org/pub/linux/libs/pam/`; accessed November 26, 2003.

Murayama, Y. (1998). DBS : A TCP Benchmark Tool. `http://www.kusa.ac.jp/ yukio-m/dbs/`; accessed December 1, 2003.

Needham, R. M. and Schroeder, M. D. (1978). Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12):993–999.

Plummer, D. C. (1982). An Ethernet Address Resolution Protocol – or – Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware. RFC 826, Internet Engineering Task Force. `http://www.ietf.org/rfc/rfc826.txt`; accessed October 22, 2003.

Postel, J. (1981). INTERNET CONTROL MESSAGE PROTOCOL. RFC 792, Internet Engineering Task Force. `http://www.ietf.org/rfc/-rfc792.txt`; accessed October 23, 2003.

Postel, J. and Reynolds, J. (1985). File Transfer Protocol (FTP). RFC 959, Internet Engineering Task Force. `http://www.ietf.org/rfc/-rfc959.txt`; accessed October 28, 2003.

Rigney, C., Willens, S., Rubens, A., and Simpson, W. (2000). Remote Authentication Dial In User Service (RADIUS). RFC 2865, Internet Engineering Task Force. `http://www.ietf.org/rfc/rfc2865.txt`; accessed August 12, 2003.

Rivest, R. (1992). The MD5 Message-Digest Algorithm. RFC 1321, Internet Engineering Task Force. `http://www.ietf.org/rfc/rfc1321.txt`; accessed August 15, 2003.

Saltzer, J. H. and Schroeder, M. D. (1975). The protection of information in computer systems. *Proceedings of the IEEE*, 63(9):1278–1308.

Schneier, B. (1996). *Applied Crtptography: Protocols, Algorithms and Source Code in C*. John Wiley & Sons, Inc, second edition.

Schneier, B. (2000). *Secrets & Lies: Digital Security in a Networked World*. John Wiley & Sons, Inc, New York, USA.

Simpson, W. (1994). The Point-to-Point Protocol (PPP). RFC 1661, Internet Engineering Task Force. `http://www.ietf.org/rfc/rfc1661.txt`; accessed November 3, 2003.

Simpson, W. (1996). PPP Challenge Handshake Authentication Protocol (CHAP). RFC 1994, Internet Engineering Task Force. `http://www.ietf.org/rfc/rfc1994.txt`; accessed August 20, 2003.

Singh, S. (1999). *The Code Book*. Norstedts Förlag, Stockholm Sweden.

Steiner, J. G., Neuman, C., and Schiller, J. I. (1998). Kerberos: An Authentication Service for Open Network Systems. In *Winter 1998 USENIX Conference*, pages 191–2001, Dallas, Texas.

Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L., and Paxson, V. (2000). Stream Control Transmission Protocol. RFC 2960, Internet Engineering Task Force. `http://www.ietf.org/rfc/rfc2960.txt`; accessed November 3, 2003.

The FreeBSD Project (2003). The FreeBSD Project. `http://www.freebsd.org/`; accessed November 26, 2003.

The FreeRADIUS project (2003). FreeRADIUS – building the perfect RADIUS server. `http://www.freeradius.org/`; accessed October 22, 2003.

The NET-SNMP Project (2003). The NET-SNMP Project Home Page. `http://www.net-snmp.org/`; accessed November 7, 2003.

The Tawie Team (2003). TSL: Home. `http://www.tawie.net/`; accessed October 22, 2003.

Walker, J. R. (2000). Unsafe at any key size; An analysis of the WEP encapsulation. Technical report, The Institute of Electrical and Electronics Engineers. `http://grouper.ieee.org/groups/802/11/Documents/-DocumentHolder/0-362.zip`; accessed November 7, 2003.

A Abbreviations and acronyms

AP	Access Point
ARP	Address Resolution Protocol
BSS	Basic Service Set
BSSID	Basic Service Set IDentifier
CA	Certification Authority
CHAP	Challenge-Handshake Authentication Protocol
DHCP	Dynamic Host Configuration Protocol
DS	Distribution System
EAP	Extensible Authentication Protocol
ESS	Extended Service Set
ISO	International Organization for Standardization
ISP	Internet Service Provider
LAN	Local Area Network
MAC	Media Access Control
Mbps	Mega bit per second
MIB	Management Information Base
NAS	Network Access Server
NTP	Network Time Protocol
OTP	One-Time Password
PHY	PHYsical layer
PKI	Public Key Infrastructure
RADIUS	Remote Authentication Dial In User Service
SNMP	Simple Network Management Protocol
SSID	Service Set IDentifier
STA	STAtion
TCP	Transmission Control Protocol
TLS	Transport Level Security
UDP	User Datagram Protocol
VLAN	Virtual Local Area Network
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access