HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Computer Science and Engineering
Degree Programme in Computer Science and Engineering

A Gateway for Wireless Ad-Hoc Networks

Master's Thesis

Erno Alhoniemi

Telecommunications Software and Multimedia Laboratory
Espoo 2004

HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Computer Science and Engineering
Degree Programme in Computer Science and Engineering

ABSTRACT OF MASTER'S THESIS

Author: Erno Alhoniemi

Title of thesis:
A Gateway for Wireless Ad-Hoc Networks

Date: December 14 2004 **Pages:** 12 + 80
Professorship: Telecommunications software **Code:** T-110
Supervisor: Professor Antti Ylä-Jääski
Instructor: Lic.Sc. (Tech.) Sanna Liimatainen

A gateway enables secure access from a wireless ad-hoc network to an infrastructured network such as the Internet or an intranet. The wireless ad-hoc network is based on Wireless Local Area Network (WLAN) that is used in ad-hoc mode, and the hosts communicate in the same link. They negotiate unique Internet Protocol (IP) addresses, and use IP to communicate. They can also access the infrastructured network through the gateway. The gateway is a generic solution that works with most frequently used applications. It does not require any changes to the external network, and it can be used with any network technologies that enable IP based communication. It provides multiple security levels for different network environments. The gateway uses Network Address Translation (NAT) to enable access from the wireless ad-hoc network to the infrastructured network. IP security (IPsec) provides mutual authentication, mutual authorization, communication integrity, and communication confidentiality between the gateway and the gateway client. The architecture includes Domain Name Service (DNS) that can resolve host names and addresses in the wireless ad-hoc network and in the infrastructured network. The gateway clients use IP based Service Discovery Service (SDS) to discover the gateway. This thesis includes the gateway specification, and reports the details of a gateway implemented on the Linux platform.

Keywords: ad-hoc networks, gateway, WLAN

Language: English

TEKNILLINEN KORKEAKOULU DIPLOMITYÖN TIIVISTELMÄ
Tietotekniikan osasto
Tietotekniikan koulutusohjelma

Tekijä:	Erno Alhoniemi	
Työn nimi:	A Gateway for Wireless Ad-Hoc Networks	
Päiväys:	14. joulukuuta 2004	**Sivumäärä:** 12 + 80
Professuuri:	Tietoliikenneohjelmistot	**Koodi:** T-110
Työn valvoja:	prof. Antti Ylä-Jääski	
Työn ohjaaja:	TkL Sanna Liimatainen	

Yhdyskäytävä tarjoaa turvallisen pääsyn langattomasta ad-hoc verkosta ulkoiseen verkkoon, joka voi olla Internet tai intranet. Verkko perustuu langattomaan lähiverkkoon (WLAN) jota käytetään ad-hoc tilassa. Kaikki koneet voivat viestiä verkossa suoraan toistensa kanssa. Neuvoteltuaan osoitteet koneet viestivät Internet-protokollan (IP) avulla. Ne voivat myös viestiä ulkoisen verkon kanssa yhdyskäytävän kautta. Yhdyskäytävä on yleinen ratkaisu, joka toimii eniten käytettyjen sovellusten kanssa. Yhdyskäytävä ei vaadi mitään muutoksia ulkoiseen verkkoon. Se soveltuu käytettäväksi kaikkien IP-pohjaisen liikenteen sallivien verkkoteknologioiden kanssa. Se tarjoaa monta eri turvatasoa erilaisia verkkoympäristöjä varten. Yhdyskäytävä tarjoaa yhteyden ulkoiseen verkkoon osoitemuunnosta (NAT) käyttäen. Turvattu IP (IPsec) tarjoaa yhdyskäytävän ja sen asiakkaan välillä molemminpuolisen todennuksen, käyttöoikeuksien tarkistuksen, sekä tietoliikenteen eheyden ja luottamuksellisuuden. Arkkitehtuuriin sisältyy nimipalvelu (DNS) joka muuntaa nimiä osoitteiksi ja osoitteita nimiksi sekä ad-hoc verkossa että ulkoisessa verkossa. Yhdyskäytävän asiakkaat käyttävät palvelujen hakupalvelua (SDS) löytääkseen yhdyskäytävän. Diplomityö sisältää yhdyskäytävän määritykset ja raportoi yksityiskohdat toteutetusta yhdyskäytävästä Linux-alustalla.

Avainsanat:	ad-hoc verkot, yhdyskäytävä, WLAN
Kieli:	Englanti

Acknowledgements

This thesis is a result from the SESSI project, and I thank the funders and participants for the opportunity to work in this project. I thank my instructor Sanna Liimatainen and my supervisor Antti Ylä-Jääski. I thank Linda Källström for helping all the way and Sasu Tarkoma for feedback.

Espoo December 14th 2004

Erno Alhoniemi

Acronyms and Definitions

Acronyms

3DES: Triple DES

3G: 3rd Generation

3GPP: 3rd Generation Partnership Project

AA: Authentication and Authorization

AAA: Authentication, Authorization, and Accounting

AES: Advanced Encryption Standard

AH: Authentication Header

AODV: Ad-hoc On-Demand Distance Vector Routing

AP: Access Point

API: Application Program Interface

CPU: Central Processing Unit

DES: Data Encryption Standard

DNAT: Destination NAT

DNS: Domain Name Server/Service/System

DoS: Denial of Service

EAP: Extensible Authentication Protocol

EAPOL: EAP Over LANs

ESP: Encapsulating Security Payload

FTP: File Transfer Protocol

FSB: Frontside Bus

GRE: Generic Routing Encapsulation

HTTP: HyperText Transfer Protocol

IBSS: Independent Basic Service Set

IEEE: Institute of Electrical and Electronics Engineers, Inc.

ICMP: Internet Control Message Protocol

IKE: Internet Key Exchange

IP: Internet Protocol

IPsec: IP security

MAC: Media Access Control

MB: 1048576 Bytes

Mbps: Million bits per second

mDNS: multicast DNS

MHz: Million Hertz

MitM: Man in the Middle

NAPT: Network Address and Port Translation

NAT: Network Address Translation

ND: Neighbor Discovery

PC: Personal Computer

PDG: Packet Data Gateway

PLMN: Public Land Mobile Network

PS: Packet Switched

RADIUS: Remote Authentication Dial In User Service

SDS: Service Discovery Service

SEND: SEcure Neighbor Discovery

SIM: Subscriber Identity Module

SLP: Service Location Protocol

SNAT: Source NAT

SMTP: Simple Mail Transfer Protocol

SSID: Service Set Identifier

TTL: Time To Live

URL: Uniform Resource Locator

USIM: Universal Subscriber Identity Module

WLAN: Wireless Local Area Network

WAN: Wide Area Network

WWW: World Wide Web

Definitions

Authentication ensures that an entity is what it claims to be.

Authorization ensures that an entity is authorized to do an action.

Integrity protects data against unauthorized modification.

Confidentiality protects data against unauthorized disclosure.

Privacy protects the information about an an entity against unauthorized disclosure.

Availability ensures that authorized access to a resource is available.

Contents

List of Figures

Chapter 1

Introduction

Most computers communicate with each other by using wired networks. This approach is well suited for stationary computers, but it is not appropriate for mobile devices. Mobile devices can use wireless networks almost anywhere and anytime by using one or more wireless network technologies. These technologies enable the use of infrastructured networks and ad-hoc networks. Infrastructured networks enable communication in an entire network infrastructure. Ad-hoc networks allow the devices to set up quickly a network that enables only local communication without a network infrastructure.

In a wireless ad-hoc network, devices can use a gateway to access an infrastructured network. One device in the ad-hoc network acts as gateway that enables communication from the wireless ad-hoc network to the infrastructured network. The infrastructured network can be the Internet or a private network. In this thesis, I present a secure gateway implementation for wireless ad-hoc networks. In particular, this thesis considers using the Wireless Local Area Network (WLAN) based on the IEEE 802.11 [26] standard as a wireless ad-hoc network. On top of that, the communication is based on the Internet Protocol (IP) [51]. It is also assumed that the devices can communicate in the same link. Thus, ad-hoc routing protocols are not needed.

The gateway implementation provides a secure generic solution for accessing the infrastructured network from the ad-hoc network, and it provides multiple security levels for different network environments. It works with most frequently used applications. It does not require any changes to the external network, and it can be used with any network technologies that enable IP based communication. For example, WLAN can be based on IEEE 802.11 standard in the wireless ad-hoc network, and the gateway can use 3G

functionality to access the infrastructured network.

Several important choices were made for the gateway implementation. NAT provides a generic solution for accessing external services from the private network, and it works with most frequently used applications. The gateway implementation can provide security for all applications with IPsec. However, using IPsec significantly decreases the performance, and demanding applications may not deliver adequate performance with IPsec. The gateway implementation also provides a DNS service that can resolve host names and addresses in the ad-hoc network and in the infrastructured network without any changes to existing applications. In addition, using existing services in a wireless ad-hoc network is discussed, and guidelines for implementing services for such networks are provided.

The rest of the thesis is organized as follows. Chapter 2 describes the environment in which the gateway is used, introduces potential gateway solutions, and chooses the most appropriate gateway solution. Next, Chapter 3 provides the gateway specification. Chapter 4 describes the gateway implementation and testing the gateway, and Chapter 5 discusses them. Finally, the conclusions are presented in chapter 6.

Chapter 2

Gateway for Wireless Ad-Hoc Networks

In this chapter, I describe the characteristics of wireless ad-hoc networks and the environment for the gateway implementation. I also describe different possible gateway solutions, compare them, and select the most appropriate gateway solution for the described environment.

2.1 Characteristics of Wireless Ad-Hoc Networks

The devices form a wireless ad-hoc network without a network infrastructure by using WLAN. This type of network can change constantly. First, the devices can freely move in the network. Second, the devices can leave and join the network at any time. Finally, the network disappears when the last devices leave the network.

Isolated wireless ad-hoc networks are not suitable for today's applications that require accessing services in the Internet. To overcome this limitation, one or more devices in the wireless ad-hoc network can provide a gateway to an external network. This external network can be the Internet or a local area network (LAN).

Wireless networks are more vulnerable to misuse than wired networks. In a wireless network, all devices share the same radio band. If two or more devices transmit simultaneously, the communication fails. In addition, a malicious device may be present in the network. It can analyze the communication in the network and do several attacks by sending invalid data. It can masquerade as another device, or it may do various Man-in-the-Middle

(MitM) or Denial-of-Service (DoS) attacks. In particular, it can even block all communication by constantly interfering the transmission.

There are several security mechanisms that partially protect communication in WLAN. WLAN may provide security on the lower layers that correspond the physical and link layers of the Open Systems Interconnection (OSI) reference model [31]. These mechanisms protect communication authenticity, integrity and confidentiality by using cryptographic methods. Moreover, these mechanisms depend on the WLAN technology. On the other hand, security can be provided independently on upper layers. However, none of these mechanisms protect against DoS attacks because it is impossible to prevent a malicious device from interfering the transmission in a wireless ad-hoc network.

It is also possible that all devices cannot communicate directly in the wireless ad-hoc network. Such a scenario is shown in Figure 2.1 in which device B can communicate with devices A and C directly, but devices A and C cannot communicate directly. This has an impact on the network-layer and application-layer protocols. In the network-layer, all devices cannot communicate directly with each other by using IP addresses. Moreover, some applications do not work unless the communication is link-local. For example, Dynamic Configuration of IPv4 Link-Local Addresses [12] requires link-local communication to successfully configure and maintain IPv4 addresses.

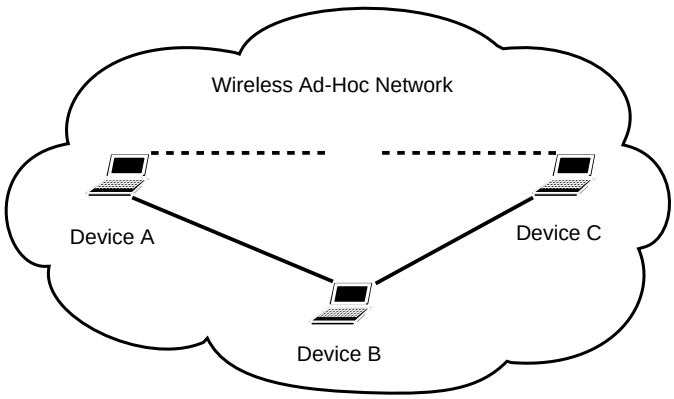

Figure 2.1: Direct communication unavailable between two devices.

Routing enables communication between devices that cannot communicate directly. In the ad-hoc network, this is done by using an ad-hoc routing protocol. There are two types of routing protocols for ad-hoc networks:

proactive and reactive. In proactive routing, routes are actively maintained, and they are available when needed. In reactive routing, routes are discovered on demand.

An ad-hoc network can be isolated, or it can have a gateway that provides a connection to another network. Consequently, the devices must be able to communicate when the gateway is available and when it unavailable. Moreover, a gateway becoming available or unavailable should not should not interfere local connections in the wireless ad-hoc network. Further, many gateways should be able to coexist in the network.

The gateway must also provide services. First, it must provide fundamental network services such as DNS. Second, it must provide the gateway discovery service that allows gateway clients to discover the gateway. Finally, it must be possible to configure the gateway and the gateway clients.

2.2 The Ad-Hoc Network Environment

This section introduces the gateway and its environment, and it describes the used ad-hoc network environment first from the lowest layer upwards and then from the logical point of view.

2.2.1 Introduction

The users use wireless devices to communicate with other users in proximity. The devices use Wireless Local Area Network (WLAN) to form a wireless ad-hoc network. The devices can be desktops, laptops, and mobile phones.

The communication takes place within a group of two or more people. The communication group may be formed for one communication session only or for many communication sessions. The communication group may remain unchanged during the communication, or it may change constantly. The communication can be personal, professional, or between unknown people. Friends and relatives can use the devices for personal communication. Company employees can use them to organize a project meeting. The users can also use them to communicate temporarily with unknown people.

The users may also need to use services that are not available in the wireless ad-hoc network. They may need to browse web pages, send and receive email, and communicate with other users that are not present in the wireless ad-hoc network. These services are available in infrastructured networks, such

as in the Internet or in intranets. A user usually communicates with the infrastructured network through an access network, e.g. through a cellular network. However, this option is available only to the customers of the operator that provides the access network.

2.2.2 WLAN Technology

WLAN is based on the IEEE 802.11 standard [26] that defines a family of WLAN standards. More specifically, it is based on the 802.11b standard [27] that enables the data transfer rate of 11 Mbps making the performance comparable to that of a wired LAN. In addition, the ad-hoc network is implemented by using the Independent Basic Service Set (IBSS) type of network. This allows devices within range communicate only directly. However, devices must use the same physical channel to communicate, and they must use choose to use the same ad-hoc network because the standard allows coexistence of many networks in the same physical channel.

Although WLAN implements the functionality of the physical layer and the link layer, to enable the use of network-layer addresses in the current link, the network-layer addresses must be mapped into Ethernet addresses by using An Ethernet Address Resolution Protocol (ARP) [48] or Neighbor Discovery (ND) [39].

2.2.3 Network Layer

In the network layer, the communication is based on IP. More specifically, either IPv4 [51] or IPv6 [17] is used. It is also possible to use a dual stack in which both IPv4 and IPv6 are used. Also Internet Control Message Protocol (ICMP) [50] is used along with IPv4 for various purposes, e.g. testing the reachability of a host. Similarly, ICMPv6 [14] is used along with IPv6.

2.2.4 Transport Layer and Session Layer

The Internet Protocol provide unreliable packet delivery of upper level protocols. The User Datagram Protocol (UDP) [49] enables connectionless delivery of application data. The Transmission Control Protocol (TCP) [52] enables connection-oriented communication.

IP can also be encapsulated inside other protocols. When IPsec is used, IP can be encapsulated by using Authentication Header (AH) [33] or IP

Encapsulating Security Payload (ESP) [34]. Using IPsec also includes using Internet Key Exchange (IKE) [22]. Moreover, IP can be encapsulated by using IP Encapsulation within IP [45] and Generic Routing Encapsulation (GRE) [19].

2.2.5 Presentation Layer and Application Layer

Using the network requires several services that are provided in the application layer. These services are service discovery service, address configuration, and domain name service (DNS). The service discovery service allows gateway clients to discover the gateway. The address configuration service allows clients to negotiate unique IP addresses. DNS is used to resolve host names and IP addresses in the external network. The next sections describe first services and then logical network structures.

2.2.6 Devices

This work considers devices that are laptops with a WLAN interface, and the operating system used is Linux. In addition, using mobile devices based on the Symbian operating system is an optional solution that is considered here.

A gateway can also provide access to an infrastructured network. A gateway can be either wired or wireless. A wired gateway is usually based on Ethernet (described in [29]) but other technologies can also be used. A wireless gateway can offer network access, for example, by using 3G or WLAN.

2.2.7 Network Structure

In the wireless ad-hoc network, the communication is restricted to the link-local communication, but the gateway is used enable communication between the wireless ad-hoc network and the infrastructured network. The communication works properly only if all the devices are within each other's communication range. The gateway may also be able to provide globally routable addresses to the gateway clients.

2.3 Gateway Solutions

In this section, I describe the potential gateway solutions, compare them, and select the most suitable solution for the environment described above.

2.3.1 3GPP System and WLAN Interworking

The 3GPP System and WLAN Interworking specification [2] allows 3G devices to use WLAN as a radio access technology. The specification describes Authentication, Authorization, and Accounting (AAA) through the 3GPP system, the use of an infrastructured network through a WLAN, and Packet Switched (PS) services through a Public Land Mobile Network (PLMN).

Figure 2.2 presents the trust model entities in the 3GPP System and WLAN interworking: user, WLAN access provider, and cellular operator. The user uses the 3G functionality of the cellular operator through the WLAN access provider. The WLAN access provider offers WLAN connectivity to the user and an access network to the cellular operator. The WLAN access provider can also be a part of the cellular operator. The cellular operator provides 3G services to the user through the WLAN access provider. The user-operator trust relation (U-O) is based on a legal agreement between a the user and the cellular operator. The operator-WLAN trust relation (O-W) is based on roaming agreements or other agreements, or it is internal to the cellular operator if the WLAN access provider is part of the cellular operator. Finally, The user-WLAN trust relation (U-W) is derived from U-O and O-W. Next, I describe the 3GPP system and WLAN interworking architecture in more detail and explain how it is modified to provide a gateway implementation for the environment described above.

Architecture

Figure 2.3 presents the simplified architecture. A user can access an infrastructured network directly through WLAN or through a Packet Data Gateway (PDG). This requires that the user is successfully authenticated and authorized for access by using the 3GPP AAA server. The 3GPP system and WLAN interworking provides two reference models: the non-roaming and the roaming reference model. In the non-roaming model, a user uses WLAN connected to the 3GPP home network. In contrast, in the roaming model, a user uses WLAN connected to the 3GPP visited network. Here, a user can access the PDG either in the 3GPP home network or in the 3GPP

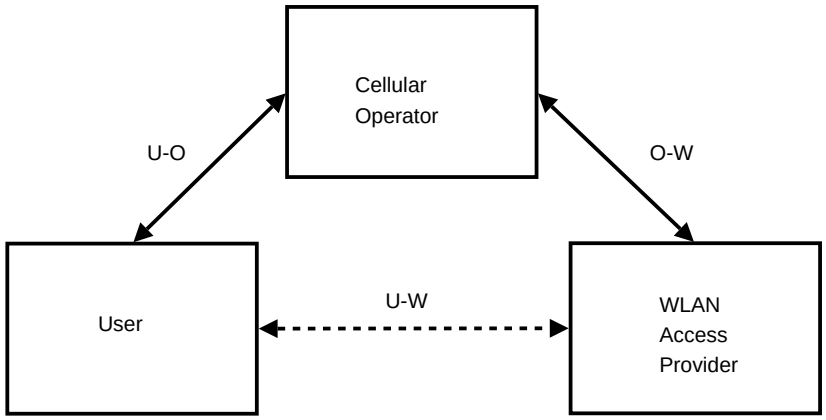

Figure 2.2: The trust model in 3GPP system and WLAN interworking.

visited network.

Authentication, Authorization, and Accounting

Using 3GPP system and WLAN interworking requires that the cellular operator trusts the WLAN access provider. The cellular operator must allow the WLAN access provider to use its AAA server to authenticate and authorize the users. This also allows the WLAN access provider to generate accounting information that can be used for billing and other purposes. A WLAN access point (AP) uses the AAA information in the AAA server of the 3GPP home network by using an AAA protocol. Both RADIUS [54] or DIAMETER [11] can be used to transport the AAA information over IP. Authentication is based on the Extensible Authentication Protocol (EAP) [3] that supports multiple authentication methods. Authentication is done by using a Universal Subscriber Identity Module (USIM) (described in [6]) or by using a Subscriber Identity Module (SIM) (described in [23]). Using EAP between a 3G device and an AAA server is shown in Figure 2.4. The authentication information is transported between a device and an AAA server. Between a device and AP, EAP is transported over a WLAN protocol. In a 802.11 WLAN, the EAP Over LANs (EAPOL) (described in [28]) is used. Further, between AP and an AAA server, EAP transported over an AAA protocol. In addition, there can be proxies between AP and the AAA server. The AAA proxy is responsible for obtaining the AAA information from the appropriate AAA proxy or server. Proxies can participate in authorization

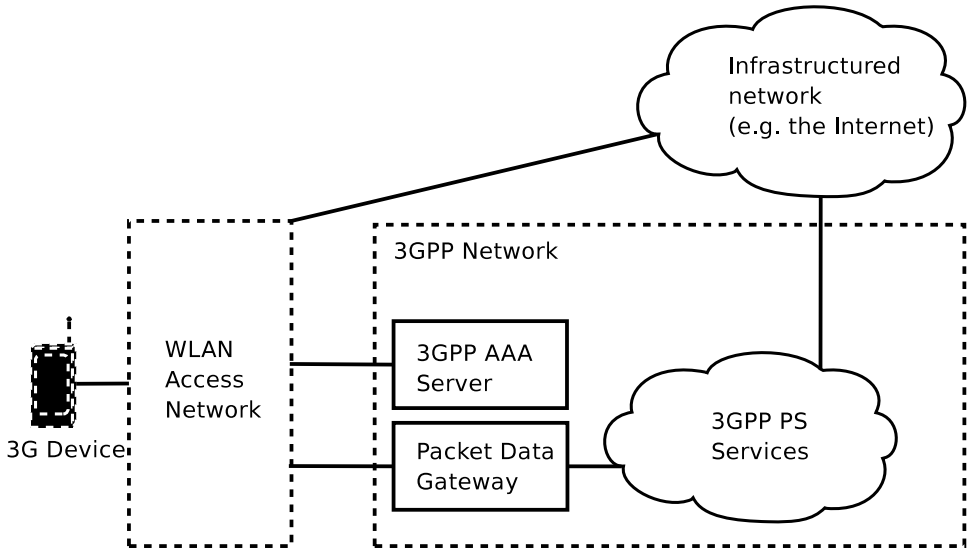

Figure 2.3: Simplified 3GPP system and WLAN interworking architecture.

by further restricting access, and they can store accounting information.

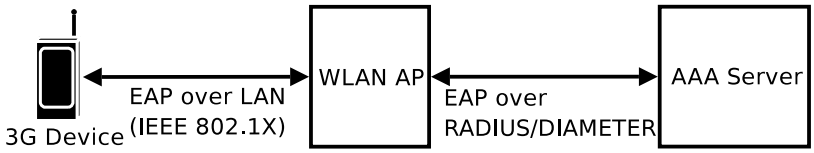

Figure 2.4: EAP between a 3G device and an AAA server.

Communication Integrity and Confidentiality

In 802.11 WLAN, link-layer communication integrity and confidentiality are based on the IEEE 802.11i specification [30]. This specification enhances WLAN security by introducing the use of Advanced Encryption Standard (AES) [43]. However, according to the Wi-Fi Alliance's white paper [63], old IEEE 802.11 hardware may not be able to support it.

In the network layer, communication integrity and confidentiality between WLAN and PDG can be provided by using IPsec [60]. Using IPsec is described in [1], but the mechanism to set up a secure tunnel between WLAN

and PDG is not yet finalized. However, it does propose a solution. First, a security association is made by using the Internet Key Exchange (IKEv2) Protocol [32], and PDG is authenticated by using public key cryptography with certificates. Second, a device can be authenticated by using EAP with USIM or SIM. Alternatively, IKEv2 or the older version, IKE is used, and both the device and PDG mutually authenticate each other with certificates.

2.3.2 Using 3GPP System and WLAN Interworking as a Gateway

The modified 3GPP system and WLAN interworking allows a 3G device to provide a gateway service in a wireless ad-hoc network as shown in Figure 2.5. The 3G device uses its standard 3G functionality for IP connectivity and DNS. It allows the gateway clients to access an infrastructured network by using Network Address Translation (NAT) [57]. It also provides a caching DNS service and advertises the gateway service by using ICMP Router Discovery Messages [16] or Neighbor Discovery. Finally, IPsec provides communication integrity and confidentiality between the gateway and the gateway client.

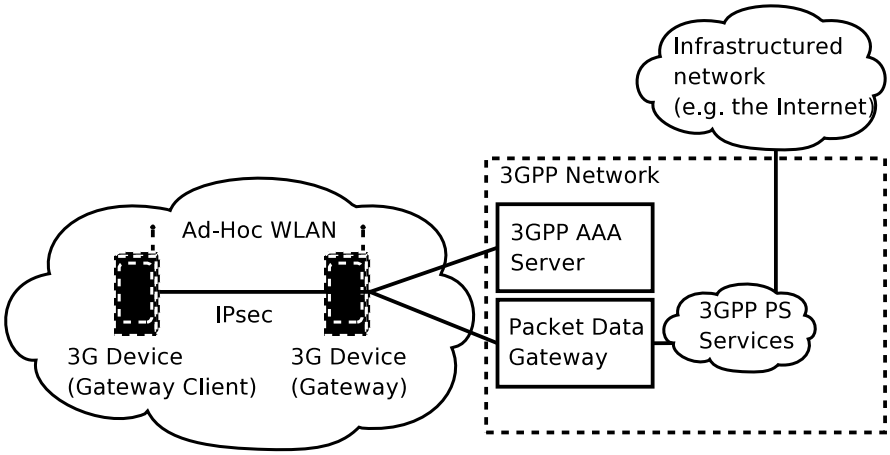

Figure 2.5: 3GPP System and WLAN Interworking as a gateway.

Authentication, Authorization, and Accounting

As the gateway acts as a client to the cellular operator, the gateway and the cellular operator mutually authenticate and authorize each other by using the standard 3G functionality. The cellular operator also checks that the gateway is authorized to use the AAA server. This allows the gateway and the gateway client to mutually authenticate and authorize each other by using EAP with USIM or SIM as the gateway and the gateway client establish an IPsec tunnel by using IKEv2. The gateway can also further restrict its use by denying access. In addition, the cellular operator stores the total accounting information of the gateway by using standard 3G functionality. This information represents the total amount of network traffic through the gateway, that is, the traffic caused by the gateway clients the traffic caused by the gateway itself. In addition, the gateway can be allowed to generate accounting information on behalf of the gateway clients. The amount of traffic caused by the gateway is obtained when the amount of client traffic is subtracted from the total amount. However, this requires that the cellular operator trusts that the gateway generates the accounting information correctly. Alternatively, the gateway does not generate accounting information, and only the total amount of network traffic through the gateway is available.

2.3.3 Local 3G Radio Link

A 3G device can use its own 3G radio link to access an infrastructured network, and it can use a WLAN interface to access the wireless ad-hoc network. Consequently, the device is a multi-homed host that has a valid IP address in both networks. The device must have routes to both networks, but it does not need to provide routing between the networks. Authentication, authorization, and accounting are based on existing 3G functionality. Each device that needs to access the infrastructured network must mutually authenticate with the cellular operator by using USIM or SIM.

2.3.4 Generic Gateway

A multi-homed device can act as a gateway that enables communication between the wireless ad-hoc network and the infrastructured network as presented in Figure 2.6. As the gateway provides access to an infrastructured network by using NAT, it does not need to configure IP addresses for the gateway clients. IPsec provides Communication integrity and confidentiality

between the gateway and the gateway client. The gateway can advertise its availability by using ICMP Router Discovery Messages or Neighbor Discovery. In short, the gateway is just a router. This approach is so generic that it is applicable with any network technology that enables IP based communication.

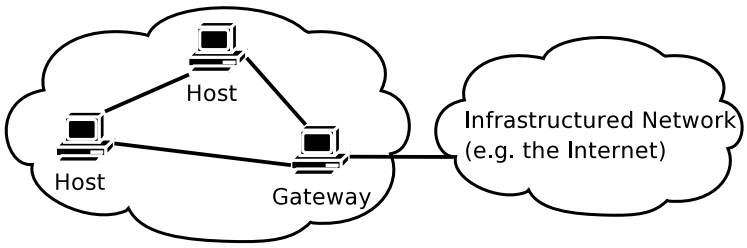

Figure 2.6: The generic gateway.

Authentication, Authorization, and Accounting

The gateway and the gateway client mutually authenticate and authorize each other as they establish an IPsec tunnel by using IKE with certificates. The AAA information may reside in the gateway or in a remote AAA server. Optionally, the gateway generates accounting information.

2.3.5 Application-Level Gateway

In a wireless ad-hoc network, the gateway can also provides access to an infrastructured network on the application level. The gateway can be a generic proxy for all applications, or it can provide an application-specific proxy for each application.

The gateway can provide a generic SOCKS proxy (described in [37]) that can provide IP connectivity for all applications that support the SOCKS protocol. The SOCKS proxy is shown in Figure 2.7. To access an infrastructured network, a gateway client uses the SOCKS protocol to communicate with a SOCKS proxy which in turn communicates in an infrastructured network on behalf of the client.

Alternatively, a gateway can provide an application-specific proxy for each application as shown in Figure 2.8. This requires that the client can use

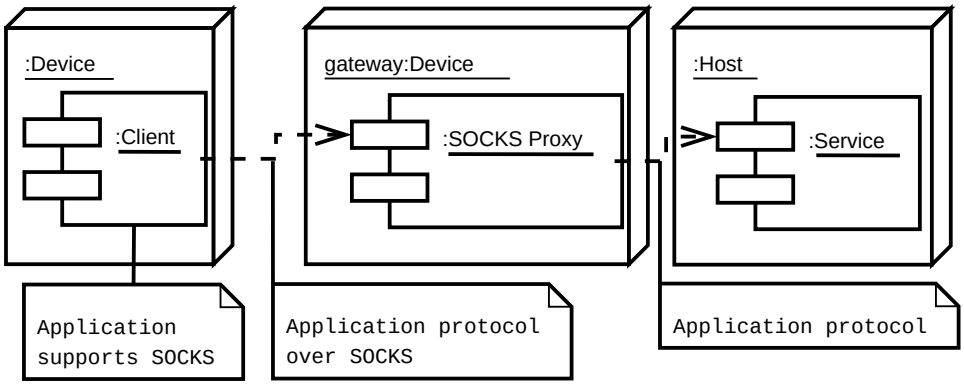

Figure 2.7: The SOCKS proxy.

the protocol through a single proxy only. Here, a gateway client uses an application-specific protocol to communicate with a proxy which in turn communicates in an infrastructured network on behalf of the client. However, using application-specific proxies is possible only with known applications and compatible software versions.

Optionally, an application-specific proxy is transparent. A transparent proxy is shown in Figure 2.9. Using a transparent proxy does not require any modification to a gateway client. Here, a gateway client assumes that it can access an infrastructured network by using the IP address of an infrastructured network as a destination address. The gateway intercepts the application-specific communication sent by a gateway client, and it acts as a client on behalf of the gateway client in an infrastructured network. It relays the application-specific communication from an infrastructured network to a gateway client as if it came from the infrastructured network.

Authentication, Authorization, and Accounting

Although some applications provide proxy authentication, IPsec can provide authentication and authorization for all applications, and it can also provide communication integrity and confidentiality between the gateway and the gateway client. Optionally, the gateway can also use a remote AAA server. The application-specific proxies do not store traditional accounting information, but they can do application-specific logging that provides application-level information.

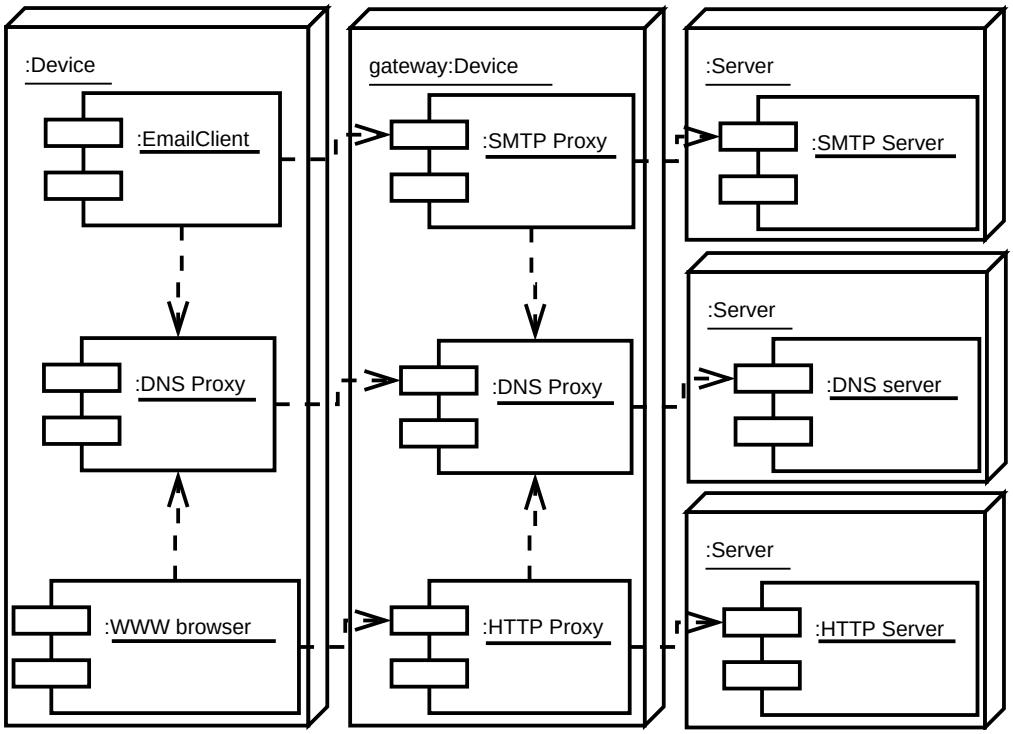

Figure 2.8: Application-specific proxies.

2.4 Other Solutions

Striegel, Ramanujan, and Bonney [59] describe a protocol-independent gateway for wireless ad-hoc networks. This gateway can support various ad-hoc routing protocols in the wireless ad-hoc network. The gateway routes traffic between the wireless ad-hoc network and the Internet. It enables Internet access by using either NAT or Mobile IP [46]. However, this solution does not provide any security.

Nilsson et. al [41] discuss how IPv6 and Ad-hoc On-Demand Distance Vector Routing (AODV) [47] can be used for Internet access. In this solution, the gateway allocates a globally routable prefix for the ad-hoc network. However, this disables coexistence of multiple gateways because there can be only one global prefix.

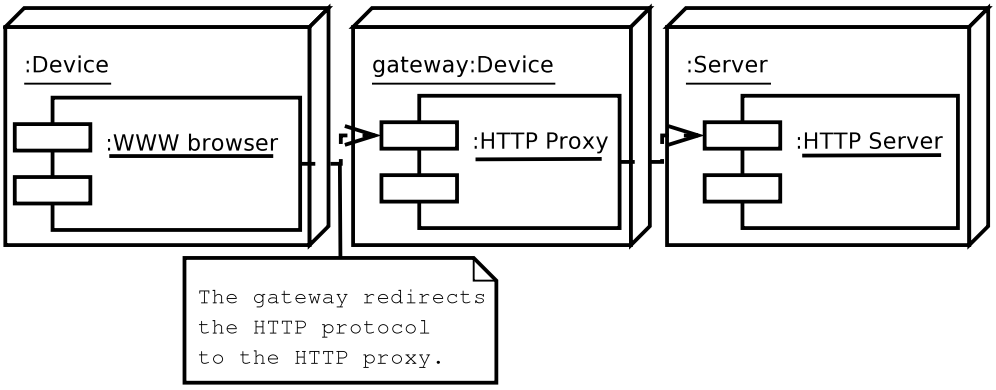

Figure 2.9: Transparent proxy.

2.5 Choosing the Gateway Solution

Next, the most appropriate gateway solution is chosen from the previously presented gateway solutions according to a brief comparison. A 3G device can use its local 3G radio link to provide the gateway service to itself. When configured correctly, it can directly access both the ad-hoc network and the infrastructured network. However, this option is available to 3G devices only. Moreover, to use the gateway based on 3GPP System and WLAN Interworking, the device must be a 3G device. From the technical point of view, this solution is obsolete because the device does not need to use this solution because it can use its local 3G radio link to provide the gateway service to itself. However, this solution can be meaningful from the business perspective. For example, if the devices are used in a foreign country, one of the devices might have a local USIM or SIM that enables Internet access at a moderate charge. Still, this solution is not very useful under normal circumstances. Although the application-level gateway is independent of the network technology, it depends on the applications. The gateway must support all applications that are used. In addition, when a new application is added, a new proxy must be added. Consequently, as the application-level gateway supports a fixed set of applications only, it lacks generality, and it is not practical for evolving applications. In contrast, the generic gateway is independent of the device type, the network technology, and the applications. Therefore, it is the preferred gateway solution.

Chapter 3

Gateway Specification

In this chapter I describe the development of the conventional gateway.

3.1 Introduction

This specification describes a gateway for wireless ad-hoc networks. The business model of the gateway is shown in Figure 3.1. In the wireless ad-hoc network, a gateway can allow other devices to communicate with an infrastructured network. The gateway provides the DNS service and IP based access to the infrastructured network, and it uses the service discovery service that allows devices to discover available gateways. The gateway is available only to authenticated and authorized users, and the users can choose which devices are authenticated and authorized to provide the gateway. The integrity and confidentiality of the gateway communication can be protected in the wireless ad-hoc network.

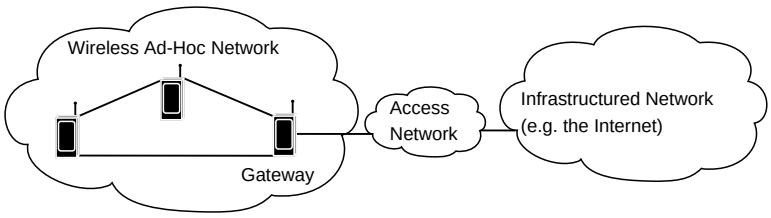

Figure 3.1: The business model of the gateway.

In the gateway business model, there are four roles: the user, the gateway

provider, the operator, and the infrastructured network provider. These roles are shown in Figure 3.2.

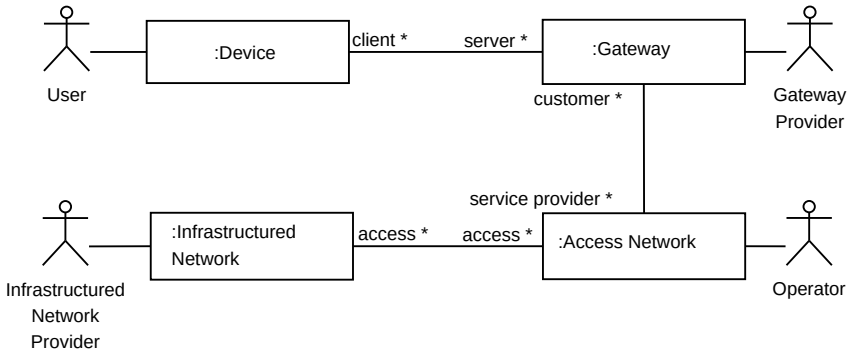

Figure 3.2: The roles.

The user accesses the infrastructured network through the gateway. Although the user does not pay directly for using the gateway, the gateway provider may require some compensation for providing the gateway. For example, the gateway provider may expect that other users occasionally provide the gateway service or pay for using the gateway.

The gateway provider allows other users in the wireless ad-hoc network to use the gateway. The gateway provider is a customer to the operator, and he or she is directly responsible for all communication costs. These costs result both from gateway communication and from the gateway provider's personal communication.

The operator provides IP based access to the infrastructured network. This access is based on existing technology and the gateway does not require any modifications to it. The infrastructured network can be the Internet or an intranet. The operator also charges for using the network. The operator may charge a monthly fee or according to the transferred data. In addition, if the gateway provider is visiting a foreign network, the foreign network may charge the roaming costs.

The infrastructured network provider is either the Internet community or an organization with an intranet. In case of an intranet, the services necessary for network access must be available in the intranet. The intranet may also provide access to the Internet. It is also possible that the infrastructured network provider and the operator are the same entity.

3.2 Use Cases

The use cases illustrate the basic functionality of the gateway. The use cases are shown in Figure 3.3.

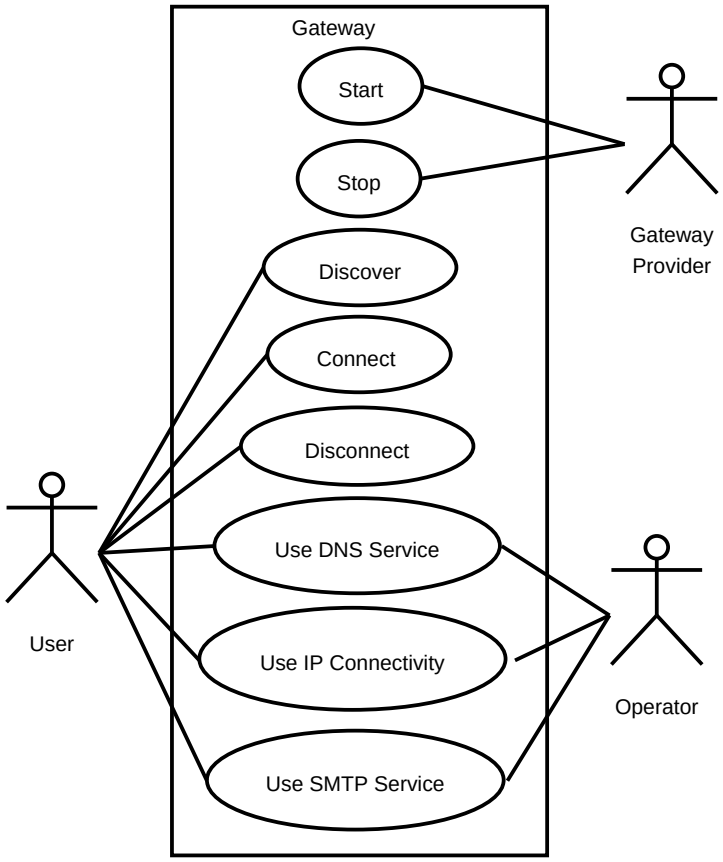

Figure 3.3: Use cases.

3.2.1 Actors

The following actors are present in the use cases:

- **User**: The user uses the gateway to access the infrastructured network.
- **Gateway Provider**: The gateway provider provides the gateway to the users.

- **Operator**: The operator provides access to the infrastructured network.

3.2.2 User Functions

The user may use the following functions:

- **Discover**: The user discovers the available gateways.
- **Connect**: The user connects to the gateway. This enables the DNS service and IP based access to the infrastructured network.
- **Disconnect**: The user disconnects from the gateway. The user does this explicitly, or this may occur implicitly when the gateway is unavailable. Disconnecting from the gateway disables the DNS service and IP based access to the infrastructured network.
- **Use DNS Service**: The DNS service allows applications to resolve host names into IP addresses. Other types of DNS queries are also possible.
- **Use IP Connectivity**: IP connectivity allows applications to communicate with the infrastructured network through the gateway.
- **Use SMTP Service**: The Simple Mail Transfer Protocol (SMTP) [36] service allows applications to send e-mail to the infrastructured network.

3.2.3 Gateway Provider Functions

The gateway provider can use the following functions:

- **Start**: The gateway provider starts the gateway. This allows other devices to discover the gateway and connect to it.
- **Stop**: The gateway provider stops the gateway. This disconnects all connected devices and prevents devices from discovering the gateway.

3.3 Requirements

There are two types of requirements: functional requirements and non-functional requirements. Functional requirements are related to required functionalities, and non-functional requirements are related to the properties of the functionalities, e.g. performance. The minimum functional requirements are given in Table 3.1.

ID	Name	Description
A1	Gateway IP Connectivity	The gateway must have IP connectivity to the infrastructured network.
A2	Gateway DNS Service	The gateway must be able to use the DNS service that can resolve host names and addresses in the infrastructured network.
A3	Gateway Discovery	The gateway must be able to allow gateway clients to discover the gateway.
A4	Mutual Authentication and Authorization	The gateway and the gateway client must mutually authenticate and authorize each other before the gateway client can use the gateway.
A5	Gateway Configuration	The gateway must provide all necessary configuration for network access to the gateway client.
A6	External IP Connectivity	The gateway must provide IP connectivity to the infrastructured network for the gateway clients.
A7	External DNS Service	The gateway must provide the DNS service that the gateway client can use to resolve host names and addresses in the infrastructured network.
A8	Communication Integrity	The integrity of the communication between the gateway and the gateway client must be protected.

Table 3.1: Functional requirements.

There are also additional requirements that extend the usability of the gateway. They do not belong to the minimum requirements; nevertheless, they are mandatory. The additional functional requirements are given in Table 3.2. The gateway must also conform to the non-functional requirements given in Table 3.3.

Finally, optional functional requirements are not implemented, but they describe features that may be implemented in the future. These requirements are given in Table 3.4.

ID	Name	Description
B1	SMTP Service	The gateway must provide a SMTP service in the wireless ad-hoc network.
B2	Multiple Gateways	More than one gateways must be able to coexist in the same wireless ad-hoc network.

Table 3.2: Additional functional requirements.

ID	Name	Description
N1	Link-Layer Independence	The gateway must be independent of the link-layer technology.

Table 3.3: Non-functional requirements.

ID	Name	Description
X1	Ad-Hoc Routing	The gateway must use an ad-hoc routing protocol and provide access to the infrastructured network through zero or more intermediate nodes.
X2	Routable IP Address	The gateway must provide a routable IP address to the gateway clients. The address must also be routable from the infrastructured network.
X3	Authoritative DNS Service	The gateway must provide the DNS service to the infrastructured network that resolves host names and addressed in the wireless ad-hoc network.
X4	Inter-Ad-Hoc Gateway	The gateway must enable IP based access between two or more wireless ad-hoc networks.
X5	Inter-Ad-Hoc DNS Service	In the wireless ad-hoc network, the gateway must provide the DNS service that resolves host names and addresses in those ad-hoc networks that the gateway provides IP based access to.
X6	IPv4 and IPv6	The gateway must support both IPv4 and IPv6.

Table 3.4: Optional functional requirements.

3.4 Architecture

The architecture of the gateway is shown in Figure 3.4. The gateway allows the gateway clients in the wireless ad-hoc network to communicate with the infrastructured network through the access network. Because the gateway

and the gateway client share the same components, they share the same implementation that can act as a gateway or as a gateway client. The access network enables communication with the infrastructured network, and it provides services such as DNS and SMTP. The infrastructured network is the Internet or an intranet. It is also possible that the infrastructured network and the access network are the same entity.

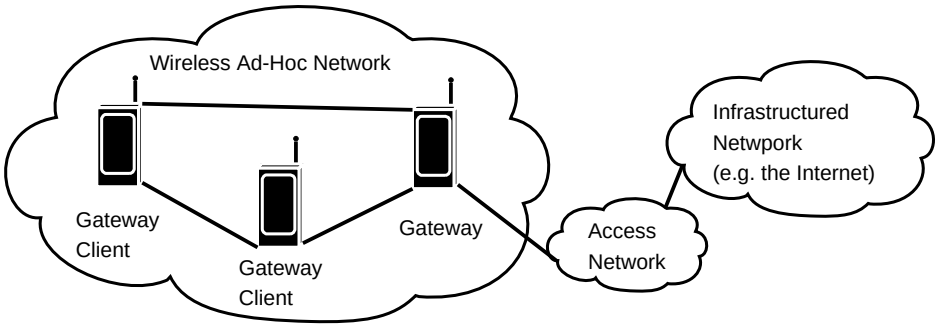

Figure 3.4: The gateway architecture.

3.4.1 Wireless Ad-Hoc Network

In the wireless ad-hoc network, WLAN is based on the IEEE 802.11b standard [27] which enables data transfer rate of 11 Mbps. WLAN is operated in IBSS mode which enables ad-hoc communication without using network infrastructure. To communicate, the devices must use the same network. They must use the same physical channel, and they must use the same Service Set Identifier (SSID).

The local communication in the wireless ad-hoc network takes place within the link-local scope, that is, all communicating devices are able to communicate directly with each other. The wireless ad-hoc network contains at most 100 devices. In addition, the devices can communicate with the infrastructured network through the gateway.

3.4.2 Access Network

The gateway can communicate with the infrastructured network through the access network. The access network provides IP based access to the

infrastructured network and the DNS service. Optionally, it provides the SMTP service.

3.4.3 Security

The architecture from the security point of view is shown in Figure 3.5. The wireless ad-hoc network, the access network, and the infrastructured network may or may not provide security. The communication between the gateway and the gateway client can be protected. Alternatively, the wireless ad-hoc network can be protected by using link-layer security. Nevertheless, the ad-hoc network is vulnerable to DoS attacks, and the availability of the communication cannot be guaranteed. The gateway can provide the gateway clients the same level of security that the access network and the infrastructured network can provide to the gateway. To enable optimal settings for different network configurations, the gateway supports the following security levels defined in the SESSI project [61]:

- *None*: No security is provided.
- *Authentication*: Authentication, authorization, and integrity are provided.
- *Confidentiality*: Authentication, authorization, integrity, and confidentiality are provided.

The security levels supported by the gateway components are shown in Table 3.5. The gateway discovery (described in [61]) can support all security levels. The communication between the gateway and the gateway client can be protected with IPsec, which provides multiple algorithms and protocols that can support all security levels. *None* can be implemented without using IPsec. Also, IPsec provides security to the DNS service that is used to resolve host names and addresses in the infrastructured network. However, IPsec does not provide security to mDNS that enables local DNS in the wireless ad-hoc network. Currently, there are no suitable solutions for securing multicast and broadcast traffic in an ad-hoc network.

To protect against unauthorized use of services, the devices should have a personal firewall that protects against unauthorized use of resources. The firewall should enable only the services chosen by the user. Finally, the firewall should be as simple as possible to avoid unnecessary configuration. However, the firewall is not a part of the gateway implementation.

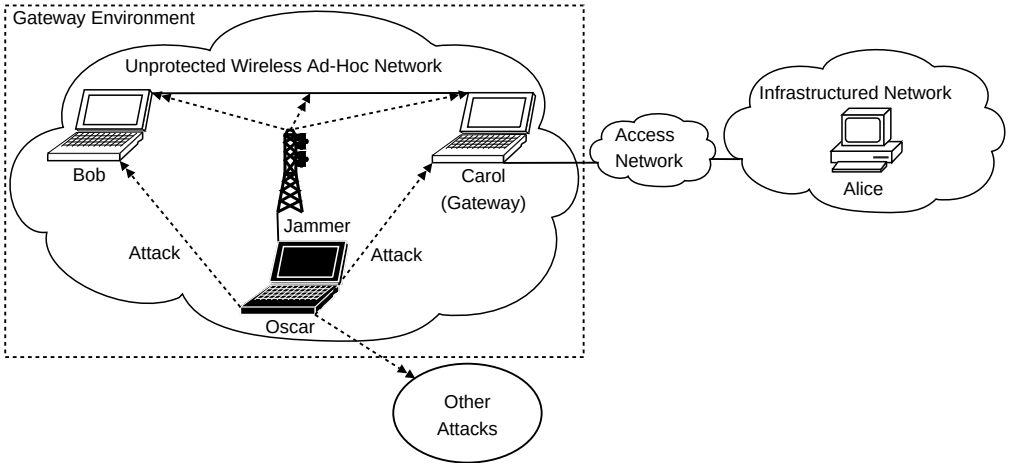

Figure 3.5: Security.

	None	Authentication	Confidentiality
Gateway Discovery	**yes**	**yes**	**yes**
IP Connectivity	**yes**	**yes**	**yes**
DNS Service (Infrastructured Network)	**yes**	**yes**	**yes**
DNS Service (Wireless Ad-Hoc Network)	**yes**	**no**	**no**

Table 3.5: Security levels supported by the gateway.

3.5 Internal Architecture

This section describes the architectural components of the gateway.

3.5.1 Gateway Components

The gateway and the gateway client are similar enough to share the same implementation. This implementation can be started as a gateway or a gateway client. The internal architecture of the gateway implementation is shown in

Figure 3.6.

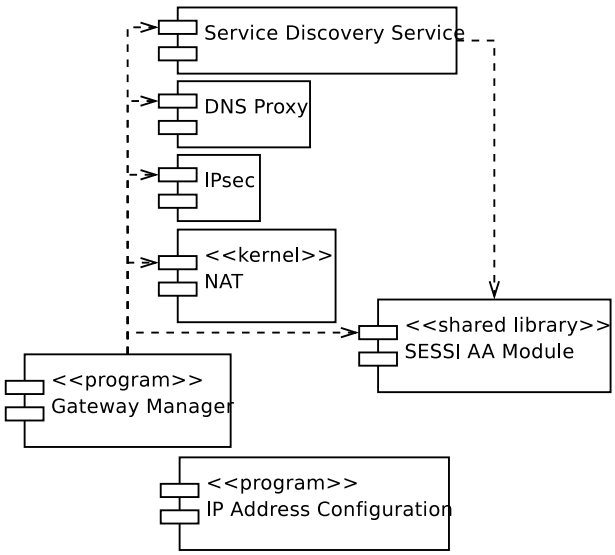

Figure 3.6: The internal architecture of the gateway.

The gateway implementation is composed of several components. The gateway manager controls the other components. The service discovery service advertises the gateway to the gateway clients. IPsec can protect communication integrity and confidentiality between the gateway and the gateway client. The gateway shares its IP address with the gateway clients by using NAT. The DNS proxy resolves host names and addresses in the wireless ad-hoc network and in the infrastructured network. The IP address configuration is executed independently of the gateway implementation. However, the SMTP server is not a part of the internal architecture; instead, the gateway only provides a connection to the SMTP server. Finally, the components that need authentication and authorization information use the SESSI authentication and authorization module.

3.5.2 Network Interfaces

The network interfaces used by the gateway are shown in Figure 3.7.

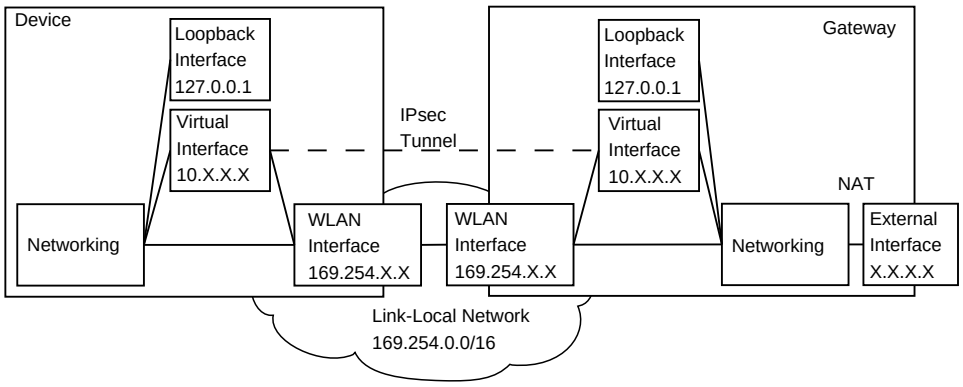

Figure 3.7: Network Interfaces

Loopback Interface

The DNS proxy uses the loopback interface to provide DNS locally in the current device.

Virtual Interface

Using IPsec in tunnel mode between the gateway and the gateway client creates virtual interfaces that enable communication through the IPsec tunnel. This requires that the tunneled data is communicated by using concrete interfaces such as WLAN interfaces.

WLAN Interface

The WLAN interface enables communication in the wireless ad-hoc network. Therefore, WLAN must support the ad-hoc mode. WLAN is used for link-local communication only.

External Interface

The external interface allows the gateway to communicate with the infrastructured network.

3.5.3 Gateway Manager

The gateway manager controls the other gateway components. It is a program that provides a command line interface. It can start gateway component as a gateway or as a gateway client. As a gateway, it provides the following functions:

- Start the gateway.
- Stop the gateway.

On the other hand, as a gateway client, it provides the following functions:

- Connect to the gateway.
- Disconnect from the gateway.

The operation system is responsible for starting and stopping the networking components. This also disables the gateway component.

The gateway manager executes single commands and acts as a daemon. It continuously monitors other gateway components and the status of the connection to the infrastructured network. If the IP address of the device changes, the gateway manager reconfigures the gateway. If the connection to the infrastructured network is lost, the gateway is implicitly stopped.

3.5.4 Service Discovery Service

The service discovery service enables gateway discovery in the wireless ad-hoc network. As there is no network infrastructure, the service discovery service must be distributed. The gateway manager uses the service discovery service for two purposes:

- When the gateway component acts as a gateway, the gateway manager uses the service discovery service to advertise the gateway to the gateway clients. When the gateway component stops acting as a gateway, it stops advertising the gateway.
- When the gateway component acts as a gateway client, the gateway manager uses the service discovery service to discover available gateways.

The service discovery service is based on an existing implementation specified in the SESSI project [61]. It describes a service discovery service that uses the Service Location Protocol Version 2 (SLPv2) [21]. It also extends the original protocol by adding security.

The gateway advertises itself by using an SLP URL [21]. The gateway client uses the same URL to discover gateways. The URL is defined as follows:

service:gateway.sessi::://**IP_ADDRESS**

Here, the suffix .sessi defines the naming authority. **IP_ADDRESS** is the IP address of the gateway. The IP address enables gateway discovery without DNS, but this requires that the URL is changed when the IP address of the gateway changes.

Security

The service discovery service must be able to provide mutual authentication, mutual authorization, communication integrity, and communication confidentiality within the service discovery protocol. The service discovery service supports multiple security levels. It uses the SESSI authentication and authorization module to manage the credentials.

3.5.5 DNS Proxy

The DNS proxy provides a DNS service that can resolve host names and addresses in an infrastructured network and in a wireless ad-hoc network within the link-local scope. Both the gateway and the gateway client use the same DNS proxy; however, they are configured differently. The deployment of the DNS proxies is shown in Figure 3.8.

To resolve host names and addresses in an infrastructured network, an application use the local DNS proxy of the gateway client which in turn is uses the DNS proxy of the gateway. Further, the DNS proxy of the gateway server uses the DNS server of the infrastructured network. In addition, the DNS proxies use multicast DNS (mDNS) [13] to resolve host names and addresses in the wireless ad-hoc network.

The DNS proxy distinguishes queries between the wireless ad-hoc network and the infrastructured network by applying the following rule:

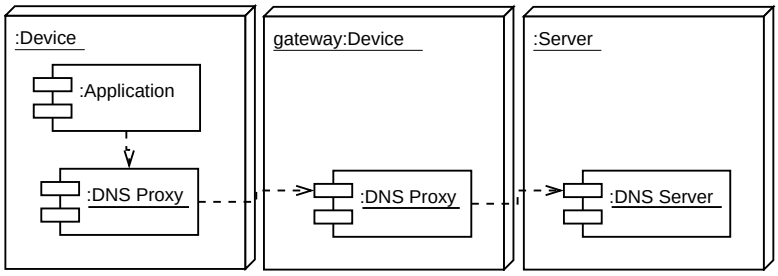

Figure 3.8: The deployment of the DNS proxies.

1. The query is resolved locally in the wireless ad-hoc network with mDNS if either of the following conditions is met:
   - The IP address is a link-local address within the range of 169.254.0.0/16.
   - The host name ends with the suffix **.local**.
2. Otherwise, the query is resolved externally with DNS.

Security

IPsec protects the DNS protocol between the gateway and the gateway client. The DNS proxy of the gateway client is configured to use the DNS proxy of the gateway server through the external IP address of the gateway which makes the DNS protocol go through the IPsec tunnel. However, the security of the DNS protocol between the gateway server and the DNS server is not changed. In contrast, mDNS does not provide any security, and IPsec cannot protect mDNS because it relies on multicast addresses.

3.5.6 SMTP Server

The gateway enables the use of an external STMP server. The deployment of the SMTP server is shown in Figure 3.9. Other devices in the wireless ad-hoc network can connect to the SMTP service in the gateway. However, the gateway does not implement an SMTP server; instead, it redirects the SMTP connection to the external SMTP server.

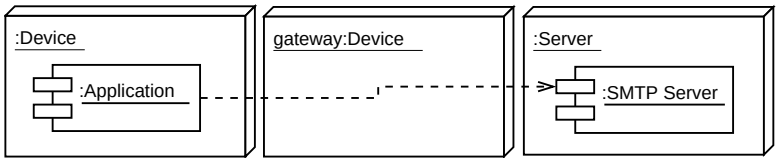

Figure 3.9: The deployment of the SMTP server.

Security

IPsec can protect the SMTP protocol between the gateway and the gateway client. The security of the SMTP traffic between the gateway and the external SMTP server is not modified.

3.5.7 IPsec

IPsec can provide communication security between the gateway and the gateway client. To accomplish this, the IPsec component must be active in both. This results in creating virtual interfaces. The virtual interfaces are shown in Figure 3.7.

The device uses its virtual interface to access the infrastructured network. It sends all packets not in the range of 169.254.0.0/16 to the IPsec interface. The gateway uses its virtual interface to communicate with the device, and it routes packets between the device and the infrastructured network.

Security

IPsec can provide mutual authentication, mutual authorization, communication integrity, and communication confidentiality between the gateway and the gateway client. However, it cannot guarantee the availability of the communication; thus, it is vulnerable to DoS attacks. The security properties are described as follows:

- Mutual authentication is based on public key cryptography. The RSA algorithm with a 1024 bit modulus is used.
- Mutual authorization is done along with the authentication. Only the authorized keys are available to the IPsec component.

- Multiple security levels are supported. In addition, when the primary intention is to prevent unauthorized use of the gateway, a fast but weak integrity algorithm can be used to enable sufficient communication integrity and reasonable performance.

3.5.8 NAT

Because the devices do not have valid addresses to access the infrastructured network, the gateway uses NAT to share its IP address with the other devices. More specifically, the NAT variant used is the Network Address and Port Translation (NAPT) [57]. It can modify the packets that are sent to or received from the external interface. It ensures that the outgoing packets have the source address of the gateway, and it may change the source port number to make it unique for each connection. For each connection, it associates the original values with the new values. If there are no unused source port numbers available, no association is created and the packet is dropped. In addition, NAPT processes incoming packets according to these associations. It checks whether the new values of any of the associations matches the destination address and the destination port of the incoming packet. If it finds a match, it changes the destination address and the destination port to the original values and forwards the packet to the wireless ad-hoc network.

Security

NAT enables access from the wireless ad-hoc network to the infrastructured network, but not vice versa. Although NAT does not provide any security to the gateway, it prevents access from the infrastructured network to the wireless ad-hoc network.

3.5.9 SESSI Authentication and Authorization Module

The SESSI authentication and authorization module maintains the credentials for all applications. An existing implementation is specified in the SESSI project [61]. It is used by the service discovery service by and the gateway manager. The gateway uses it to determine which devices can use it, and the gateway client uses it to determine which devices can act as a gateway to it. To distinguish the gateway and the gateway client from other applications, the following application identifiers are defined:

- gateway: **fi.hut.tml.sessi:gateway:1.0**
- gateway client: **fi.hut.tml.sessi:gateway-client:1.0**

In addition, using the gateway includes a required security level. As both gateway and gateway can require a security level, the higher security level is used. The detailed operation of the SESSI authentication and authorization module is described in the SESSI project [61].

Security

The SESSI authentication and authorization module is operated locally in the device. The private keys never leave the SESSI authentication and authorization module.

3.5.10 IP Address Configuration

In the wireless ad-hoc network, the devices configure IP addresses by using Dynamic Configuration of IPv4 Link-Local Addresses [12]. This allows the devices to configure and maintain unique IP addresses within the link-local scope in which all devices can communicate directly. If the device cannot maintain its IP address, it tries to obtain a new address. This can disable the communication that uses the old IP address.

When the device boots, the operating system starts the IP address configuration along with the networking components. The IP address configuration is continuous independent process that is active when the network interface is active. When the device shuts down, the IP address configuration is stopped along with other components.

Security

IP address configuration does not provide any security. An attacker can easily prevent all devices from obtaining an IP address. However, DoS attacks are outside the scope of this specification.

3.6 Functions and Features

3.6.1 Gateway Component States

Figure 3.10 shows the gateway component states and the transitions between them. The gateway component is only in one state at a time: in the disabled, networking, server, or client state. Initially, the gateway is in the disabled state. When the device boots, the gateway component enters the networking state that enables local communication in the wireless ad-hoc network. In this state, the gateway component may enter the gateway state or the gateway client state. Finally, the gateway enters the disabled state when the networking components are stopped.

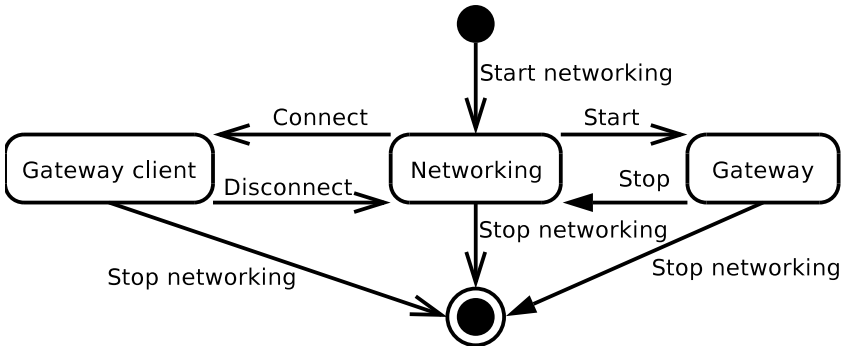

Figure 3.10: The gateway states.

Disabled

The gateway component is disabled, and the networking components are inactive.

Networking

The gateway component is disabled, but the networking components are active. This allows local communication in the wireless ad-hoc network.

Gateway client

The gateway component acts as a gateway client to a gateway. The device can communicate locally in a wireless ad-hoc network, and it can communicate with the infrastructured network through the gateway.

Gateway

The gateway component acts as a gateway to the gateway clients. This allows the gateway clients to communicate with the infrastructured network.

3.6.2 General Functions

The general functions are common to the gateway and the gateway client.

Start Networking

When the device starts, the operating system starts the networking components to enable communication with other devices. In addition, the following components are started along with other networking components:

- IP Address Configuration
- DNS Proxy
- Service Discovery Service

Stop Networking

When the device shuts down, the operating system stops the networking components. In addition, the following components are stopped along with other networking components:

- IPsec
- Service Discovery Service
- DNS Proxy
- IP Address Configuration

3.6.3 Gateway Functions

These functions are used when the gateway component acts as a gateway.

Start

This function starts the gateway component as a gateway. This makes the gateway available to the gateway clients and allows them to discover the gateway. It has the following steps:

1. The gateway checks that it is connected to the infrastructured network.
2. The gateway configures the DNS proxy to use the DNS server of the infrastructured network.
3. The gateway enables IPsec if security is needed.
4. The gateway enables NAT.
5. The gateway enables routing.
6. The gateway advertises itself by using the service discovery service.

Stop

This function stops the gateway component from acting as a gateway. This prevents gateway clients from discovering the gateway and disconnects all gateway clients. It has the following steps:

- The gateway server stops advertising itself by using the service discovery service.
- The gateway disables routing.
- The gateway disables NAT.
- The gateway disables IPsec if security was needed.
- The gateway the DNS proxy to use the DNS server of the infrastructured network.

3.6.4 Gateway Client Functions

These functions are used when the gateway component acts as a gateway client.

Discover

This function discovers and lists the available gateways by using the service discovery service.

Connect

This function connects connects the gateway client to the specified gateway for accessing the infrastructured network. The local communication in the wireless ad-hoc network is not affected. It has the following steps:

1. The gateway client sets the default gateway to the selected gateway if needed.
2. The gateway client starts IPsec if security is needed.
3. The gateway client configures the DNS proxy to use that of the gateway to resolve host names and addresses in the infrastructured network.

Disconnect

This function disconnects the gateway client from the gateway. This disables communication through the gateway. The local communication in the wireless ad-hoc network is not affected. This function has the following steps:

- The gateway stops IPsec.
- The gateway configures the DNS proxy to resolve host names and addresses in the wireless ad-hoc network only.

3.7 Interfaces

3.7.1 Command Line User Interface

The gateway is controlled by executing a program with command line arguments. This program acts according to the specified command line arguments and terminates when the command is finished. This program can be executed from a command interpreter, and it can also be executed from another program or script. The program should be along the search path of the operating system.

Gateway Functions

The gateway provides the following functions:

Start

gateway start

Description

Start the gateway. This allows gateway clients to connect to the gateway and to communicate with the infrastructured network through the gateway.

Arguments

- **start** specifies that the gateway is started.

Return Values

- 0: Success: the gateway was successfully started.
- Other: Error: The gateway could not be started.

Stop

gateway stop

Description

Stop the gateway. This disconnects the connected gateway clients and stops them from communicating with the infrastructured network through the gateway.

Arguments

- **stop** specifies that the gateway is stopped.

Return Values

- 0: Success: the gateway was successfully stopped.
- Other: Error: The gateway could not be stopped.

Gateway Client Function

The gateway client provides the following functions:

Discover

gateway discover

Description

Discover the available gateways. This function discovers all available gateways by using the service discovery service. It discovers only those gateways that this device is authorized to use. It displays a list of zero or more gateways. In addition, it displays whether this device authorizes the gateway to provide the gateway service. Each line contains the following information:

- **Server** is the fully qualified domain name of the gateway server.
- **Authorized** specifies whether this device authorizes the gateway to provide the gateway service. The value is either **Yes** or **No**.

Arguments

- **discover** specifies that the available gateways are discovered.

Return Values

- 0: Success: the available gateways were successfully discovered.
- Other: Error: The available gateways could not be discovered.

Connect

gateway connect **gateway**

Description

Connect the gateway client to the specified gateway for accessing the infrastructured network. The **gateway** argument must specify an active gateway that is either known in advance or discovered by using the **Discover** function. The gateway client can connect to the gateway only works if the gateway authorizes the gateway client and the gateway client authorizes the gateway.

Arguments

- **connect** specifies that the gateway client connects to the specified gateway.
- **gateway** is the fully qualified domain name of the gateway.

Return Values

- 0: Success: the gateway client successfully connected to the gateway.
- Other: Error: the gateway client could not connect to the gateway.

Disconnect

gateway disconnect

Description

Disconnect the gateway client from the connected gateway. This disables communication with the infrastructured network through the gateway.

Arguments

- **disconnect** specifies that the gateway client disconnects from the connected gateway.

Return Values

- 0: Success: the gateway client successfully disconnected from the gateway.
- Other: Error: the gateway client could not disconnect from the gateway.

3.8 Gateway Design

This section contains the design of the components. It describes how these components are implemented. It outlines the implementation classes, functions and variables. It also describes which tools and libraries are used.

3.8.1 Software development

Design Principles

- Simplicity: The software is as simple as possible.
- Minimum Effort: The software is as easy to implement as possible.
- Independence: The dependency on tools, modules and libraries is minimized.

Operating System

The software is intended to be run on Linux. The kernel version should be as new as possible because some components may require functionality that is unavailable in earlier kernel versions.

Programming Language

The software is written in the C++ language, but it provides a C APIs that can also be used from programs written in C. Therefore, all C APIs should use the C naming conventions, that is, the following construct should appear around the C API definitions:

```
#ifdef __cplusplus
extern ‘‘C’’ {
#endif

/* The C API definitions go here... */

#ifdef __cplusplus
}
#endif
```

Compiler

The gcc C/C++ compiler version 3.3.3 or higher is used.

3.8.2 Gateway Manager

The gateway manager controls the gateway components. The design of the gateway manager is given in Figure 3.11. The gateway manager provides a main program that uses distinct classes to control and monitor other gateway components. Next, the classes of the gateway manager are introduced.

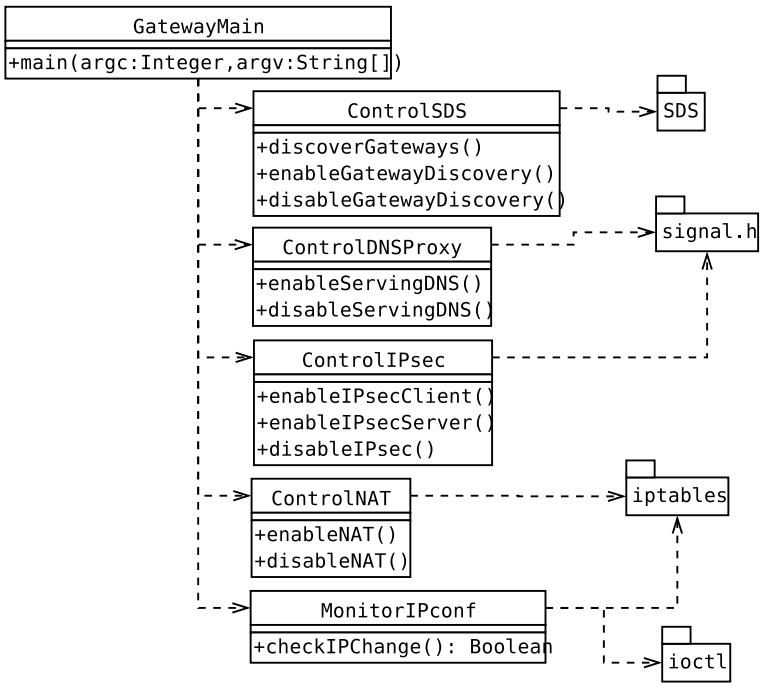

Figure 3.11: The gateway manager.

ControlSDS

ControlSDS controls the service discovery service. It provides the following functions:

- discoverGateways: Discover available gateways in the wireless ad-hoc network.

- enableGatewayDiscovery: Enable gateway discovery. This allows gateway client to discover the gateway.
- disableGatewayDiscovery: Disable gateway discovery. This prevents gateway clients from discovering the gateway.

In addition, if the IP address of the gateway changes, gateway discovery may need to be disabled and enabled to update the advertised IP address.

ControlDNSProxy

This class controls the DNS proxy. It provides the following functions:

- enableServingDNS: Enable the DNS proxy to use the specified DNS proxies or servers.
- disableServingDNS: Disable the use of the serving DNS proxies or servers.

If the IP address of the serving proxy changes, the DNS proxy must be restarted or reconfigured. This may also be necessary it the IP address of the DNS proxy itself changes. If the DNS proxy must be restarted, it may be temporarily unavailable.

ControlIPsec

This class controls IPsec. It provides the following functions:

- enableIPsecClient: Configure and enable IPsec for the gateway client.
- enableIPsecServer: Configure and enable IPsec for the gateway.
- disableIPsec: Disable IPsec for the gateway or the gateway client.

Before IPsec is enabled, the following settings must be configured:

- IP address of the gateway
- the keys of the gateway and the keys of the gateway clients

Moreover, if the IP address of the gateway or the gateway client changes, IPsec must be restarted or reconfigured.

ControlNAT

This class controls NAT. It provides the following functions:

- enableNAT: Enable NAT.
- disableNAT: Disable NAT.

NAT is applied only to the external interface of the gateway by using iptables [40]. As the gateway may have a dynamic IP address for its external interface, NAT can be implemented by using MASQUERADE [40] that drops all network address and port translations if the external IP address changes.

MonitorIPconf

This class monitors IP address configuration. It provides the following functions:

- checkIPChange: Check if the IP address has changed.

Checking the IP change can be implemented in two alternative ways:

1. The function uses iptables to monitor the messages related to obtaining and claiming an IP address. This allows the gateway manager to take action immediately, but this solution is dependent on the IP address configuration protocol.
2. The function periodically checks the IP address by using the ioctl function available on UNIX platforms. This results in a delay, but this solution is independent of the IP address configuration protocol.

3.8.3 Service Discovery Service

Because the service discovery service is based on an existing implementation, the design is not included in this document. The details are available in the SESSI project documentation [61].

3.8.4 DNS Proxy

The design of the DNS proxy is given in Figure 3.12. An application resolves host names and addresses by using the resolver library which is an integral part of BIND [9]. The resolver library uses the local DNS proxy that can be based on any software that can act as a DNS proxy. The DNS proxy can be provided by one of the following implementations:

- named, which is an integral part of BIND
- pdnsd [55], which is a lightweight DNS server
- djbdns [8], which is another DNS server

Moreover, the DNS proxy is modified to use mDNS to resolve local host names and addresses in the wireless ad-hoc network. The mDNS implementation is is based on Rendezvous [5]. It provides a mDNS client and a mDNS responder that implements a mDNS server. The mDNS client is integrated integrated with the DNS proxy. The mDNS responder can be an independent component, or it can be integrated with the DNS proxy.

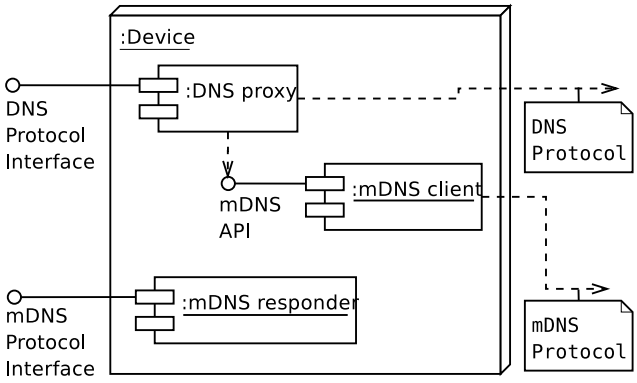

Figure 3.12: The design of the DNS proxy.

In addition, when the device boots, the DNS proxy is started along with other networking components.

3.8.5 SMTP

The gateway redirects incoming SMTP connections to the external SMTP server by using destination NAT (DNAT) [40].

3.8.6 IPsec

When the gateway acts as a client or a server, it uses IPsec that enables secure communication through the WLAN interface. IPsec can be provided in one of the following ways:

- IPsec is provided by the Linux kernel, KAME tools, and an IKE daemon. [56]
- IPsec is provided by using a FreeS/WAN implementation with X.509 certificate support [58]

3.8.7 NAT

NAT is provided by the kernel. It can be enabled by using iptables.

3.8.8 SESSI Authentication and Authorization Module

Because the SESSI authentication and authorization module is based on an existing implementation, the design is not included in this document. The SESSI authentication and authorization module is described in detail in the SESSI project documentation [61].

3.8.9 IP Address Configuration

The IP address configuration is done by zcip [24]. It is an independent program. It is started when the interface is brought up, and it is running as long as the interface is active. Finally, when the operating system shuts down, zcip is stopped.

3.9 Implementation

This section describes what needs to be implemented.

Mandatory features

The following features are mandatory:

1. All specified features (except mDNS and SMTP).

Optional features

1. mDNS
2. SMTP
3. supporting multiple security levels (*None*, *Authentication*, and *Confidentiality*)

3.9.1 Open Issues

Currently, there are no open issues.

Chapter 4

Results

In this chapter, I present the evaluation results of the gateway implementation.

4.1 Gateway Implementation

This section describes the implemented functionality component by component and verifies the implementation against the requirements.

4.1.1 Gateway Manager

The gateway manager was not implemented. Instead, the operating system starts the required services.

4.1.2 Service Discovery Service

The service discovery service was not needed because the implementation relies currently on a fixed configuration.

4.1.3 DNS Proxy

The DNS proxy is based on pdnsd [55]. The integration of mDNS and pdnsd was successful, but caching link-local addresses and host names does not work yet. The mDNS responder is a separate component, and it did not need to be modified.

4.1.4 SMTP Server

The gateway provides an external SMTP service in the ad-hoc network by redirecting incoming SMTP connections to an external server with destination NAT. In addition, the gateway client provides a local SMTP service that redirects the SMTP connection to the gateway.

4.1.5 IPsec

IPsec is based on FreeS/WAN [62] because it allows the kernel to do NAT and IPsec at the same time. As FreeS/WAN source package supported the kernels up to the 2.4 series only, I chose a compatible kernel and applied the FreeS/WAN patch to it. This configuration can provide secure communication between the gateway client and the gateway. However, as it protects all communication, it does not work as described in the specification.

Experimenting the native IPsec of the kernel version 2.6.8.1 with KAME tools indicated that the kernel cannot do native IPsec and NAT at the same time. Experimenting the native IPsec and NAT with Openswan [44] gave similar results.

4.1.6 NAT

NAT worked perfectly; however, the native IPsec of the Linux kernel did not work with NAT.

4.1.7 SESSI Authentication and Authorization Module

The gateway was not integrated with the SESSI authentication and authorization module because the gateway currently relies on a fixed configuration. Instead, the AA information was present in the IPsec configuration file. The file contains the certificates of the authorized users and the IP addresses required to establish IPsec tunnels. In a real implementation, the SESSI Authentication and Accounting Module is responsible for storing the certificates.

4.1.8 IP Address Configuration

IP address configuration worked perfectly. By default, zcip [24] generates the IP address from the Media Access Control (MAC) address. Therefore, it attempts to configure the same address every time. However, it can also generate the IP address at random.

4.1.9 Verifying the Implementation against Requirements

The gateway meets almost all functional requirements given in Table 3.1. As the gateway has access to an infrastructured network with DNS, it conforms to requirements A1 and A2. However, as it does not provide gateway discovery required by A3. Requirement A5 is met because the gateway does not need to configure the gateway client. Requirements A6 and A7 are met because the gateway enables access to the infrastructured network with DNS. Using IPsec provides mutual authentication, mutual authentication, communication integrity and communication confidentiality between the gateway client and the gateway, and thus it meets requirements A4 and A8. The gateway also meets the additional functional requirements. It provides the SMTP service required by B1. Requirement B2 is met as multiple gateways can coexist in the same ad-hoc network because the gateway need not configure the network.

Moreover, the gateway conforms to the non-functional requirements shown in Table 3.3. Because the gateway is independent of the link-layer technology, it conforms to requirement N1, which the only requirement in the table.

4.2 Test Environment

The test environment is shown in Figure 4.1 There are two PCs in the test environment, the first is the gateway client and the second is the gateway. The gateway PC communicates with the external network through the default gateway. The ad-hoc network is based on Ethernet, and only IPv4 is used. The PCs are connected by using a crossover cable. The most important properties of the test hardware are given in Table 4.1. In addition, the gateway computer is also connected to a LAN, and it has a global IP address.

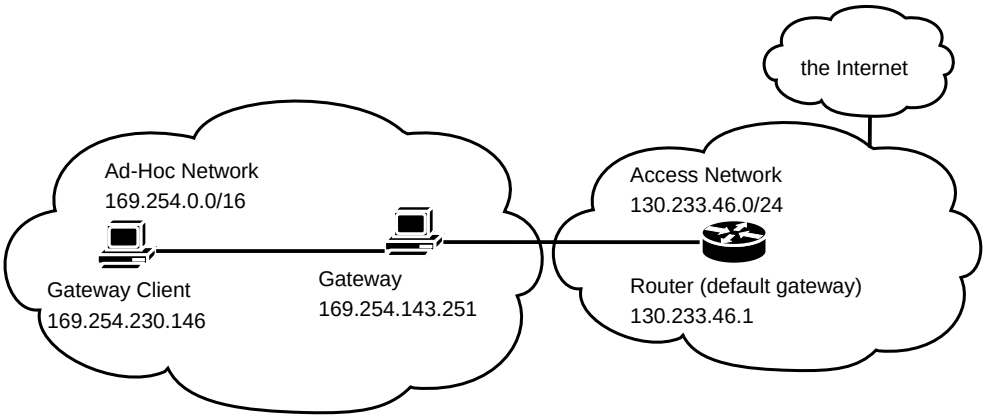

Figure 4.1: The test environment.

Property	**GW Client**	**Gateway**
CPU	Celeron 433 MHz	Pentium II 350 MHz
FSB	66 MHz	100 MHz
Memory	128 MB	128 MB
Network	1 x 100baseTx-FD	2 x 100baseTx-FD

Table 4.1: Test hardware.

4.3 Tests

4.3.1 SMTP Test

In the SMTP test, the gateway client establishes a connection to the gateway which in turn redirects the connection to an external SMTP server. This test is an instance of redirecting a TCP connection to another host. In the gateway client, the telnet command is used to connect to the SMTP port of the gateway. The test is successful if the external SMTP server immediately reports its name as specified by the SMTP protocol.

4.3.2 HTTP Test

The HTTP test repeatedly downloads a large file from an external HTTP server and measures the performance. This test also shows that NAT and the global DNS work. This test is done by executing a script that downloads

the file by using lynx and displays the elapsed time in seconds after each download. The following parameters were used:

- Number of iterations: 10
- File size: 80444620 bytes

This test contains three parts. The first part tests the performance of the gateway only. The second part tests the performance available to the gateway client on the *None* level. Finally, the third part tests the tests the performance available to the gateway client on the *Authentication* and *Confidentiality* levels.

There are two different test setups. The first setup does not provide any security; hence, it implements the security level *None*. The second setup uses IPsec to implement the security levels *Authentication* and *Confidentiality*.

4.4 Test Results

This section presents the results of the SMTP and HTTP tests.

SMTP Test

On the *None* level, the gateway successfully redirected the SMTP connection to the external SMTP server. However, on the *Authentication* and *Confidentiality* levels, FreeS/WAN intercepted the communication, and the gateway failed to redirect to the external SMTP server; instead, it attempted to use the SMTP server of the gateway.

HTTP Test

The gateway passed the HTTP tests. The first test measured the performance of the gateway only. Downloading the file usually took from 17 to 23 seconds; however, once it took 67 seconds. The second test measured the performance available to the gateway client on the *None* level. Downloading the file took from 21 to 34 seconds. Finally, the third test measured the performance available to the gateway client on the *Authentication* and *Confidentiality* levels. Downloading the file took from 69 to 72 seconds.

4.4.1 Security

The gateway implementation was able to support multiple security levels. The *None* level provided the best performance. In contrast, using the *Authentication* and *Confidentiality* levels decreased the performance. Because FreeS/WAN configuration does allow choosing the algorithms when IKE is used, FreeS/WAN used the 3DES [42] encryption algorithm and the HMAC-MD5-96 [38] integrity algorithm.

Chapter 5

Discussion

5.1 Analyzing the Gateway Implementation

In this section, I analyze the gateway implementation.

5.1.1 The Gateway Implementation

The gateway implementation provides a secure generic solution for accessing the infrastructured network from the ad-hoc network, and it provides multiple security levels that enable different security levels for different network environments. It works with most frequently used applications. It does not require any changes to the external network. Because it is independent of the network technology, it can be used with any network technologies that enable IP based communication. For example, WLAN can be based on IEEE 802.11 standard in the wireless ad-hoc network, and the gateway can use 3G functionality to access the infrastructured network. However, all functionality described in the gateway specification was not implemented due to the limited time available. Therefore, the gateway implementation only shows that the components work together. It relies on a fixed configuration only, and it does not use the service discovery service or the SESSI authentication and authorization module.

5.1.2 Development Process

Implementing the components took more time than expected. Implementing the DNS proxy required choosing the most appropriate DNS proxy for the

implementation, modifying mDNS software and integrating it to the selected DNS proxy. In addition, enabling the desired IPsec functionality required trying several Linux kernel and IPsec software combinations. Due to the limited time available for the implementation, the gateway implementation only shows that the components work together.

5.1.3 Components

Next, I discuss the gateway components described in the gateway specification including the components that are not used in the gateway implementation. The discussion includes the implemented functionality and alternative approaches that were not implemented.

Gateway manager

Currently, the operating system starts the gateway components in a single fixed configuration. These components are based on existing services. Existing services cannot usually reconfigure themselves when the IP addresses change. This problem can be solved by using the gateway manager that restarts or reconfigures the services when the IP addresses change. The gateway manager provides a centralized approach for managing changes in the ad-hoc network. However, when the services are restarted and reconfigured, they may be temporarily unavailable. This may cause errors to the applications.

Alternatively, the operating system starts the services that independently monitor the network and reconfigure themselves when needed. These services provide a decentralized approach for managing changes in the ad-hoc network. This can enable more seamless operation in the ad-hoc network and also in the infrastructured network. For example, zcip independently monitors the network and maintains the IP address. The services should be lightweight services that are designed and built from scratch for ad-hoc networks. On the other hand, it can be difficult or impossible to modify existing software suitable for dynamic ad-hoc networks.

The ad-hoc environment can also be made static in the network layer by exposing fixed IP addresses to the applications and by using dynamic IP addresses for transport. This can be accomplished by encapsulating fixed IP addresses inside a tunnel that uses dynamic IP addresses for transport. Alternatively, the IP stack can translate between fixed addresses and dynamic addresses. Both solutions require keeping the mapping of the fixed

and dynamic addresses up to date. Similar approaches are used in mobility solutions such as mobile IP and mobileNAT [10].

Service Discovery Service

The original idea was to use ICMP router discovery messages for service discovery. This allows the gateway clients to actively search for the available gateways, and it allows the gateway clients to passively discover the available gateways. ICMP router discovery messages also allow the hosts to discover available routers before obtaining IP addresses, but this does not provide any security. In contrast, SEcure Neighbor Discovery (SEND) [7] can authenticate the messages, but it is designed for IPv6 only. Nevertheless, IPsec can provide security for the gateway when the gateway is used even if the service discovery is insecure.

As the hosts immediately negotiate IP addresses when the network interface is brought up, they can use IP based service discovery. The gateway specification suggests using the secure service discovery service implemented in the SESSI project. Because SDS uses the SESSI authentication and authorization module, it can allow only authenticated and authorized gateway clients to discover the gateway.

DNS Proxy

The DNS proxy enables the DNS service in the ad-hoc network and in the external network without any changes to existing applications. It provides the DNS service that can use either the DNS protocol or the mDNS protocol. The DNS proxy does not provide any security. IPsec can protect external DNS queries in the ad-hoc network. In addition, if the gateway can use Domain Name System Security Extensions [18], they can also provide security to the gateway clients. However, IPsec cannot protect the mDNS protocol because mDNS uses multicasting.

In the ad-hoc network, the DNS proxy allows resolving host names and addresses of all hosts that are reachable from the ad-hoc network, but it does not provide the DNS service to the external network. This is similar to the network topology in which the external network is reachable from the ad-hoc network, but the ad-hoc network is not reachable from the ad-hoc network because NAT prevents access from the external network to the ad-hoc network. Consequently, as both the network topology and the DNS proxy have internal and external domain, they are in line with each other.

Currently, the DNS proxy implementation fails to provide caching of local host names and addresses in the ad-hoc network. Because caching of local host names and addresses is disabled, the DNS proxy always uses mDNS which ensures the freshness of the data. Because all hosts in the ad-hoc network have the DNS proxy that implements caching, cached host names and addresses can be instantly resolved without using the network. This can speed up the applications and save resources. On the other hand, using multiple caches slows down resolving host names and addresses not found in the cache. However, when the IP addresses in the ad-hoc network change, the local host names and addresses in the DNS proxy may become invalid. To prevent using invalid data, the DNS proxy can use short Time To Live (TTL) values or disable caching for local host names and addresses. It can also monitor the network and update the cache. In addition, if the IP address used by the DNS proxy changes or the IP address of the serving DNS proxy changes, the DNS proxy must reconfigure itself independently, or it must be restarted or reconfigured.

SMTP

Although the SMTP protocol does not limit access to the SMTP servers, many organizations allow users to access the SMTP service from the internal network only. Consequently, when a user roams in a foreign network, he or she may have to use the internal SMTP service of the foreign network. In contrast, the mailbox servers based on the Internet Message Access Protocol (IMAP) [15] are usually accessible from anywhere.

When used with IMAP, the implemented SMTP service allows sending and receiving email by using a single fixed configuration. The SMTP service is always available in the loopback address, and the IMAP server resides in a fixed global address.

Using FreeS/WAN prevents the gateway from redirecting incoming SMTP connections to the external SMTP server. Although redirecting a TCP connection to another server is a generic solution, it only works with protocols compatible with NAT and only without security. Instead, the gateway should provide an SMTP server that only relays electronic mail to the external server.

IPsec

IPsec provides mutual authentication and authorization between the gateway and the gateway client by using IKE before the actual communication takes place. This allows authentication by using public key cryptography. Further, only authorized keys are present in the IPsec configuration. IKE also provides keys for IPsec protocols.

The gateway implementation uses IPsec with ESP. ESP can provide communication integrity and confidentiality simultaneously. Either or both can be disabled by using null algorithms. ESP can also work through NAT because ESP does not protect the integrity of the encapsulating IP addresses and port numbers that are modified by NAT. A solution for using IPsec through NAT is described in [4, 25, 35]. As developers have become aware of the advantages of ESP, it is the only available option in some IPsec implementations.

AH can provide communication integrity only, and it can also protect the integrity of the encapsulating IP addresses and port numbers. It does not work with NAT because NAT modifies the encapsulating IP addresses and port numbers. Using AH with ESP can provide communication integrity and confidentiality. However, ESP can alone provide almost identical functionality. ESP can also provide only communication integrity when it is used with null encryption [20]. Finally, some IPsec implementations (e.g. FreeS/WAN) no longer support AH.

To enable secure communication with external hosts and with the gateway, the gateway implementation establishes an IPsec tunnel between the gateway and the gateway client. They use their IP addresses to transport the ESP protocol in the ad-hoc network. As the gateway communicates with the gateway or an external host through the gateway, the IP address of the gateway client and the IP address of the communicating party appear in the IP header of the encapsulated protocol.

The gateway implementation does not use IPsec in transport mode because it provides secure communication between two hosts only. Although using IP Encapsulation within IP [45] or Generic Routing Encapsulation (GRE) [19] can provide a tunnel that can be protected by using IPsec in transport mode, this adds to overhead and requires additional configuration.

Using IPsec for secure communication significantly decreases the performance and adds to resource consumption. This results from the IPsec protocol overhead and from integrity and encryption algorithms. Demanding applications, e.g. multimedia applications, may not be able deliver adequate performance with IPsec. In addition, if the communication is also protected on the ap-

plication level, it is protected twice which may lead to excessive resource consumption.

The gateway implementation uses FreeS/WAN because it allows the kernel to do IPsec and NAT at the same time. Using FreeS/WAN requires support from the kernel. The most reliable option is to integrate the IPsec support in the kernel because it ensures that all functionality is available. However, FreeS/WAN did not support the latest Linux kernels; the newest supported kernel was the 2.4 series. In contrast, the native IPsec of the 2.6 series kernels cannot do IPsec and NAT at the same time. Because the gateway implementation relies on NAT, the gateway implementation cannot use the native IPsec of the Linux kernel before this incompatibility is fixed in newer kernel versions. However, if newer kernels can do IPsec and NAT at the same time, this is the preferred solution. If this happens, FreeS/WAN can be replaced with Openswan [44] that uses the native IPsec of the Linux kernel because both are configured in the same way.

Currently, the gateway implementation is not integrated with the SESSI authentication and authorization module; instead, the IPsec configuration defines the authorized connections. However, in a real implementation, the SESSI authentication and authorization module can be used with IPsec in two ways: generating IPsec configuration from the SESSI authentication and authorization module or making IPsec to obtain the keys from the SESSI authentication and authorization module directly. The former is easy to implement, but it may fail to refresh the configuration unless IPsec is restarted or reconfigured. The latter is more difficult to implement, but it may or may not be a able to react to changing configurations.

5.1.4 Security

The gateway implementation can provide mutual authentication, mutual authorization, communication integrity, and communication confidentiality between the gateway and the gateway client, but it cannot protect against DoS attacks. An attacker can send forged data that can interfere with the communication or prevent all communication by doing RF jamming. The gateway provides security independently of the external network, but this security applies to the local communication between the gateway and the gateway client only. As accessing different infrastructured networks can include different level of security, the gateway can provide an appropriate security level for communication with the infrastructured network.

The gateway implementation supports various security levels: *None*, *Authen-*

tication, and *Confidentiality*. These levels are discussed in detail below.

None does not provide any security, but it provides the best performance. *None* allows any host in the ad-hoc network to access the gateway. However, *None* can be used in a secure environment. A wireless ad-hoc network based on the IEEE 802.11 standard [26] can be protected by using WPA (described in [63]) with a shared secret. Although this enables communication only within a group of devices sharing the secret key, it may enable strong encryption by using dedicated hardware, which provides excellent trade-off between security and resource consumption.

Authentication provides mutual authentication, mutual authorization, and communication integrity between the gateway and the gateway client by using IPsec, but it cannot protect any multicast or broadcast communication. When FreeS/WAN uses IKE, FreeS/WAN always provides both communication integrity and communication confidentiality; therefore, FreeS/WAN also provides communication confidentiality already on this level.

Confidentiality provides the same functionality as *Authentication*, but *Confidentiality* guarantees that communication confidentiality is also provided.

5.1.5 Privacy

To protect the privacy of the user, the confidentiality of the sensitive information must be provided, and the user's identity must not be disclosed to unauthorized parties. Applications can disclose various information on the user including the user's identity. In particular, authenticating to a service discloses the user's identity to the service provider. In addition, an adversary can eavesdrop application protocols or actively use applications to disclose sensitive information. For example, an adversary can use mDNS to discover the host name of the device either by eavesdropping the mDNS protocol or by actively using mDNS.

The identity of the user can also be disclosed by combining a fixed MAC address or IP address with other available information. The network interface has a fixed MAC address, and MAC addresses are present in every transmitted packet. If zcip generates the IP address from the MAC address, this may result in generating a fixed IP address. Although MAC addresses and link-local IP addresses are used only in the local link, they may be visible in the application protocols used in the external network.

Privacy can only be provided if it is not violated in any way. This requires that applications that disclose sensitive information are not used. In partic-

ular, the device must not use the mDNS server. The user must not authenticate to untrusted services. The device must also assign its MAC address at random when the interface is brought up, and zcip can generate the IP address either from the MAC address or at random. Consequently, the adversary can no longer rely on fixed MAC addresses.

5.2 Analyzing the Test Results

In this section, I analyze the test results.

5.2.1 SMTP Test

The SMTP test shows that the gateway can redirect the SMTP connection to another host on the *None* level. This test also proves that the DNS proxies can resolve local addresses in the ad-hoc network by using mDNS. However, the test fails on the *Authentication* and *Confidentiality* levels. As FreeS/WAN intercepts the communication, the kernel fails to redirect the SMTP connection to another SMTP server.

5.2.2 HTTP Test

This test measures the performance of the gateway, and it also proves that the DNS proxies can resolve addresses in the infrastructured network. On the *None* level, the test results indicate that the bandwidth available to the gateway client is slightly lower than the bandwidth available to the gateway client when there is no other communication. This is explained by the one hop longer route used by the gateway client. Moreover, the test results indicate considerable variation in performance. As the network technology is very fast, the performance depends on external conditions such as on the load of the HTTP server. Finally, on the *Authentication* and *Confidentiality* levels, the performance is approximately from two to three times lower than on the *None* level. This is explained by the protocol overhead and the algorithms, namely 3DES and HMAC-MD5-96.

5.2.3 Simulating WLAN with Ethernet

A link-local network based on Ethernet is conceptually similar to a wireless ad-hoc network in which all hosts communicate in the same link. The test

environment used a crossover Ethernet cable that enables link-local communication between two hosts only. This limitation can be overcome by using straight Ethernet cables and a hub. Even 16-port hubs are available. However, in practical wireless ad-hoc networks, some devices may be too far from each other to communicate directly. Consequently, the communication is not link-local. This can make the protocols that require link-local communication fail. For example, zcip may fail to negotiate unique link-local addresses. This problem can be solved by using protocols that work beyond the link-local scope.

5.3 Using NAT

NAT provides a generic solution for accessing external services from the private network. It enables IP based access from a private network to an external network, but not from the external network to the private network. This allows the hosts in the private network to access the external network by using IP to transport protocols that are compatible with NAT, but external hosts cannot access the private network in this way.

Consequently, NAT limits using client-server and peer-to-peer protocols. If the server resides in the private network, the client-server protocol can used in the private network only. On the other hand, if the server resides in the external network, the protocol can be used from anywhere. In addition, client-server protocols cannot be used to access the private network from the external network. This problem can be solved if the direction of the connection can be reversed. However, this may require significant changes to the protocols.

Most frequently used protocols can be used through NAT. NAT enables infrastructural services such as DNS and most frequently used applications such as WWW, but it does not work with all protocols. The protocol fails to work through NAT it the protocol does not allow NAT to modify the IP addresses and port numbers. There are two possible protocol-specific solutions to make the protocol work through NAT: making NAT compatible with the application protocol and making the application protocol compatible with NAT. In the former solution, the NAT implementation recognizes the application-level protocol and modifies the data to make it compatible with the modified addresses and port numbers, for example, the Linux kernel can support using FTP [53] through NAT. In the latter solution, the applications are made compatible with NAT. However, these solutions are protocol-specific, and they do not solve all problems.

Alternatively, if the application protocol supports proxies or another similar mechanism, the gateway may allow using the protocol between the private network and the external network without network layer connectivity. Instead, the proxy operates in the application layer and relays data between the hosts. This can allow global connectivity on the service level.

The Linux NAT implementation is managed with iptables. The gateway enables NAPT by using the MASQUERADE feature on the external interface. The MASQUERADE feature enables NAPT with a dynamic address. When the IP address of the external interface changes, the MASQUERADE feature forgets all connections through NAT because they do not work anyway, and starts accepting new connections by using the new IP address. Consequently, it provides continuous operation with a dynamic IP address. Alternatively, NAPT can be enabled by using source NAT (SNAT) [40], but this works with a fixed IP address only.

In addition, a connection can be redirected to another IP address by using DNAT. The gateway implementation uses DNAT to provide the SMTP service. This can provide a generic solution for providing services in the gateway when the services reside in the external network. The gateway clients need not know the IP address of the services. However, this only works on the *None* level. On the *Authentication* and *Confidentiality* levels, FreeS/WAN intercepts the communication and prevents using DNAT. Consequently, redirecting incoming connections does not provide a generic solution that works on all security levels. In contrast, an application-specific proxy works with all security levels, but it does not provide a generic solution for all applications.

Chapter 6

Conclusion

The gateway implementation provides a secure generic solution for accessing the infrastructured network from the ad-hoc network, and it provides multiple security levels for different network environments. It works with most frequently used applications. It does not require any changes to the external network, and it can be used with any network technologies that enable IP based communication. For example, WLAN can be based on IEEE 802.11 standard in the wireless ad-hoc network, and the gateway can use 3G functionality to access the infrastructured network.

The gateway manager provides a centralized approach for managing changes in the ad-hoc network because it can reconfigure or restart services based on existing implementations. However, building and designing new services from scratch for ad-hoc networks can enable more seamless operation as the services can independently manage changes in the ad-hoc network. Alternatively, static IP addresses can be provided to existing implementations in the network layer.

If IP addresses are available, the IP based service discovery service can be used to discover both infrastructural services and application-specific services. Consequently, only one service discovery service is needed.

The implemented DNS proxy enables resolving host names and addresses in the wireless ad-hoc network and in the external network without any changes to existing applications. Because the addresses may change in the wireless ad-hoc network, the DNS proxy must use small TTL values or disable caching for local host names and addresses. Alternatively, the DNS proxy must monitor the network and keep the cache up to date. Although IPsec can protect the DNS protocol, it cannot protect the mDNS protocol that uses multicasting.

IPsec can provide mutual authentication, mutual authorization, communication integrity and communication confidentiality between the gateway and the gateway client, but it cannot guarantee the availability of the communication.

ESP is the preferred IPsec protocol. When used with null encryption, ESP provides the same functionality as AH does. Because ESP does not protect the IP header, it can work with NAT. In addition, ESP is the only available option in some IPsec implementations.

IPsec protocol overhead and cryptographic algorithms decrease the performance and add to resource consumption. In the test environment, the communication was from two to three times slower with IPsec than without IPsec. IPsec may prevent mobile devices from delivering acceptable performance for demanding applications.

The native IPsec of the 2.6 series of the Linux kernels does not work with NAT. If NAT is needed and the Linux kernel cannot do IPsec and NAT at the same time, using FreeS/WAN with a supported kernel is the preferred solution. However, if future Linux kernels are able to do IPsec and NAT at the same time, this is the preferred solution. If this happens, FreeS/WAN users may be able to switch to Openswan that uses the native IPsec of the Linux kernel because both are configured in the same way.

Using WPA with the security level *None* provides an alternative to using IPsec especially for mobile devices. When used with a shared secret, WPA enables communication in a closed communication group only. In addition, WPA is often implemented by using dedicated hardware that consumes less resources than encryption done in software.

If privacy is provided, it must be provided on all levels. The user's identity must not be disclosed to unauthorized parties. The user must not use untrusted applications that disclose sensitive information on the user, and the user must not authenticate to untrusted services. In addition, if the IP and MAC addresses are generated at random, an adversary cannot use fixed addresses to disclose the user's identity.

A link-local wireless ad-hoc network can be simulated with Ethernet. However, many practical ad-hoc networks do not enable link-local communication. Therefore, the protocols tested in a link-local network based on Ethernet may not work.

NAT provides a generic solution for accessing external services from the private network, and it works with most frequently used applications. However, it prevents external hosts from accessing the private network, and it re-

mains incompatible with some applications. Alternatively, using application-specific proxies on the service level can enable global connectivity in the application layer without network-layer connectivity. Moreover, using NAT with application-specific proxies can enable even more services than NAT alone can.

The gateway can also be extended to use private addresses in the wireless ad-hoc networks and to use tunnels between the networks. Here, the gateways are used to build a large private network in which NAT is not needed. Also, infrastructured networks with private or global IP addresses can join the private network. Nevertheless, other infrastructured networks are accessed through NAT. Although the gateway implementation only provides half of this functionality, it is one step towards global IP connectivity.

Bibliography

[1] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Service and System Aspects; 3G Security; Wireless Local Area Network (WLAN) interworking security. TS 33.234 V6.2.1, 3GPP, September 2004. http://www.3gpp.org/ftp/Specs/archive/33_series/33.234/33234-621.zip.

[2] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3GPP system to Wireless Local Area Network (WLAN) interworking; System description. TS 23.234 V6.2.0, 3GPP, September 2004. http://www.3gpp.org/ftp/Specs/archive/23_series/23.234/23234-620.zip.

[3] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and Levkowetz, H. Extensible Authentication Protocol (EAP). RFC 3748, IETF, June 2004. http://www.ietf.org/rfc/rfc3748.txt.

[4] Aboba, B., and Dixon, W. IPsec-Network Address Translation (nat) Compatibility Requirements. Tech. rep., IETF, March 2004. http://www.ietf.org/rfc/rfc3715.txt.

[5] Apple Computer, Inc. Rendezvous. http://www.apple.com/macosx/features/rendezvous/, 2004. Referred: 26 Nov 2004.

[6] Arkko, J., and Haverinen, H. Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA). Internet draft, IETF, October 2004. Expires: 25 April 2005 http://www.ietf.org/internet-drafts/draft-arkko-pppext-eap-aka-13.txt.

[7] Arkko, J., Kempf, J., Sommerfeld, B., Zill, B., and Nikander, P. SEcure Neighbor Discovery (SEND). Internet draft, IETF,

July 2004. Expires: 15 January 2005. http://www.ietf.org/internet-drafts/draft-ietf-send-ndopt-06.txt.

[8] Bernstein, D. djbdns: Domain Name System tools. http://cr.yp.to/djbdns.html. Referred: 26 Nov 2004.

[9] BIND9.NET. DNS, BIND, DHCP, LDAP and Directory Services. http://www.bind9.net/, 2004. Referred: 26 Nov 2004.

[10] Buddhikot, M., Hari, A., Singh, K., and Miller, S. MobileNAT: A New Technique for Mobility Across Heterogeneous Address Spaces. In *Proceedings of the 1st ACM international workshop on Wireless mobile applications and services on WLAN hotspots* (2003), ACM Press, pp. 75 – 84.

[11] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and Arkko, J. Diameter Base Protocol. RFC RFC3588, IETF, September 2003. http://www.ietf.org/rfc/rfc3588.txt.

[12] Cheshire, S., Aboba, B., and Guttman, E. Dynamic Configuration of IPv4 Link-Local Addresses. Internet draft, IETF, July 2004. Expires: 2 January 2005 http://www.ietf.org/internet-drafts/draft-ietf-zeroconf-ipv4-linklocal-17.txt.

[13] Cheshire, S., and Krochmal, M. Multicast DNS. Internet draft, Apple Computer, Inc., February 2004. Expired: 14 August 2004. http://files.multicastdns.org/draft-cheshire-dnsext-multicastdns.txt.

[14] Conta, A., and Deering, S. Internet Control Message Protocol (ICMPv6) for the Internet Protocol version 6 (ipv6) specification. RFC 2463, IETF, December 1998. http://www.ietf.org/rfc/rfc2463.txt.

[15] Crispin, M. INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1. RFC 2060, IETF, December 1996. http://www.ietf.org/rfc/rfc2060.txt.

[16] Deering, S. ICMP Router Discovery Messages. RFC 1256, IETF, September 1991. http://www.ietf.org/rfc/rfc1256.txt.

[17] Deering, S., and Hinden, R. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460, IETF, December 1998. http://www.ietf.org/rfc/rfc2460.txt.

[18] Eastlake, D. Domain Name System Security Extensions. RFC 2535, IETF, March 1999. http://www.ietf.org/rfc/rfc2535.txt.

[19] Farinacci, D., Li, T., Hanks, S., Meyer, D., and Traina, P. Generic Routing Encapsulation (GRE). RFC 2784, IETF, March 2000. http://www.ietf.org/rfc/rfc2784.txt.

[20] Glenn, R., and Kent, S. The NULL Encryption Algorithm and Its Use With IPsec. RFC 2410, IETF, November 1998. http://www.ietf.org/rfc/rfc2410.txt.

[21] Guttman, E., Perkins, C., Veizades, J., and Day, M. Service Location Protocol, Version 2. RFC 2608, IETF, June 1999. http://ietf.org/rfc/rfc2608.txt.

[22] Harkins, D., and Carrel, D. The Internet Key Exchange (IKE). RFC 2409, IETF, November 1998. http://www.ietf.org/rfc/rfc2409.txt.

[23] Haverinen, H., and Salowey, J. Extensible Authentication Protocol Method for GSM Subscriber Identity Modules (EAP-SIM). Internet draft, IETF, October 2004. Expires: 25 April 2005 http://www.ietf.org/internet-drafts/draft-haverinen-pppext-eap-sim-14.txt.

[24] Hertzog, R. Overview of zcip source package. http://packages.qa.debian.org/z/zcip.html, 2004. Referred: 26 Nov 2004.

[25] Huttunen, A., Swander, B., Volpe, V., DiBurro, L., and Stenberg, M. UDP Encapsulation of IPsec ESP Packets. Internet draft, IETF, May 2004. Expired: 3 November 2004. http://www.ietf.org/internet-drafts/draft-ietf-ipsec-udp-encaps-09.txt.

[26] IEEE. *IEEE Std 802.11 1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Std 802.11 1999 ed. IEEE, 1999.

[27] IEEE. *IEEE Std 802.11b-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band*, IEEE Std 802.11b-1999 ed. IEEE, 1999.

[28] IEEE. *IEEE Std 802.1X-2001, Port-Based Network Access Control*, IEEE Std 802.1X-2001 ed. IEEE, 2001.

[29] IEEE. *IEEE Std 802.3-2002, Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*, IEEE Std 802.3-2002 ed. IEEE, 2002.

[30] IEEE. *IEEE Std 802.11i-2004, Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications - Amendment 6: Medium Access Control (MAC) Security Enhancements*, IEEE Std 802.11i-2004 ed. IEEE, 2004.

[31] ISO. *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*, ISO/IEC 7498-1:1994 ed., 1994.

[32] Kaufman, C. Internet Key Exchange (IKEv2) Protocol. Internet draft, IETF, September 2004. Expires: March 2005. http://www.ietf.org/internet-drafts/draft-ietf-ipsec-ikev2-17.txt.

[33] Kent, S., and Atkinson, R. IP Authentication Header. RFC 2402, IETF, November 1998. http://www.ietf.org/rfc/rfc2402.txt.

[34] Kent, S., and Atkinson, R. IP Encapsulating Security Payload (ESP). RFC 2406, IETF, November 1998. http://www.ietf.org/rfc/rfc2406.txt.

[35] Kivinen, T., Huttunen, A., Swander, B., and Volpe, V. Negotiation of NAT-Traversal in the IKE. Internet draft, IETF, February 2004. Expired: 10 July 2004. http://www.ietf.org/internet-drafts/draft-ietf-ipsec-nat-t-ike-08.txt.

[36] Klensin, J. Simple Mail Transfer Protocol. RFC 2821, IETF, April 2001. http://www.ietf.org/rfc/rfc2821.txt.

[37] Leech, M., Ganis, M., Lee, Y., Kuris, R., Koblas, D., and Jones, L. SOCKS Protocol Version 5. RFC 1928, IETF, March 1996. http://www.ietf.org/rfc/rfc1928.txt.

[38] Madson, C., and Glenn, R. The Use of HMAC-MD5-96 within ESP and AH. RFC 2403, IETF, November 1998. http://www.ietf.org/rfc/rfc2403.txt.

[39] Narten, T., Nordmark, E., and Simpson, W. Neighbor Discovery for IP version 6 (IPv6). Request for Comments 2461, IETF, December 1998. http://www.ietf.org/rfc/rfc2461.txt.

[40] Netfilter. netfilter/iptables project homepage. http://www.netfilter.org/, 2004. Referred: 26 Nov 2004.

[41] Nilsson, A., Perkins, C. E., Tuominen, A. J., Wakikawa, R., and Malinen, J. T. AODV and IPv6 internet access for ad hoc networks. *ACM SIGMOBILE Mobile Computing and Communications Review 6*, 3 (July 2002), 102–103.

[42] NIST. *Data Encryption Standard (DES)*, FIPS PUB 46-3 ed. NIST, October 1999. http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf.

[43] NIST. *Advanced Encryption Standard (AES)*, FIPS PUB 197 ed. NIST, November 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[44] Openswan. Openswan. http://www.openswan.org/, 2004. Referred: 26 Nov 2004.

[45] Perkins, C. IP Encapsulation within IP. RFC 2003, IETF, October 1996. http://www.ietf.org/rfc/rfc2003.txt.

[46] Perkins, C. IP Mobility Support. RFC 2002, IETF, October 1996. http://www.ietf.org/rfc/rfc2002.txt.

[47] Perkins, C., Belding-Royer, E., and Das, S. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561, IETF, July 2003. http://www.ietf.org/rfc/rfc3561.txt.

[48] Plummer, D. C. An Ethernet Address Resolution Protocol or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware. RFC 826, IETF, November 1982. http://www.ietf.org/rfc/rfc826.txt.

[49] Postel, J. User Datagram Protocol. RFC 798, IETF, August 1980. http://www.ietf.org/rfc/rfc768.txt.

[50] Postel, J. Internet Control Message Protocol. RFC 792, IETF, September 1981. http://www.ietf.org/rfc/rfc792.txt.

[51] Postel, J. Internet Protocol. RFC 791, IETF, September 1981. http://www.ietf.org/rfc/rfc0791.txt.

[52] Postel, J. Transmission Control Protocol. RFC 793, IETF, September 1981. http://www.ietf.org/rfc/rfc793.txt.

[53] Postel, J., and Reynolds, J. FILE TRANSFER PROTOCOL (FTP). RFC 959, IETF, October 1985. http://www.ietf.org/rfc/rfc959.txt.

[54] Rigney, C., Willens, S., Rubens, A., and Simpson, W. Remote Authentication Dial In User Service (RADIUS). Rfc 2865, IETF, June 2000. http://www.ietf.org/rfc/rfc2865.txt.

[55] Rombouts, P. pdnsd maintenance page. http://www.phys.uu.nl/ rombouts/pdnsd.html. Referred: 26 Nov 2004.

[56] Spenneberg, R. Linux Kernel 2.5/2.6 using KAME-tools. http://www.ipsec-howto.org/x247.html, 2004. Referred: 26 Nov 2004.

[57] Srisuresh, P., and Egevang, K. Traditional IP Network Address Translator (Traditional NAT). RFC 3022, IETF, January 2001. http://www.ietf.org/rfc/rfc3022.txt.

[58] Steffen, A. Linux FreeS/WAN. http://www.strongsec.com/freeswan/, July 2004. Referred: 26 Nov 2004.

[59] Striegel, A., Ramanujan, R., and Bonney, J. A Protocol Independent Internet Gateway for Ad Hoc Wireless Networks. In *26th Annual IEEE Conference on Local Computer Networks (LCN'01)* (November 2001), IEEE, pp. 92 – 101.

[60] Thayer, R., Doraswamy, N., and Glenn, R. IP Security Document Roadmap. RFC 2411, IETF, November 1998. http://www.ietf.org/rfc/rfc2411.txt.

[61] UHE, HUT and TUT. *SESSI Interface and Functional Specification*, July 2004. Will be published online 2005. http://www.tml.hut.fi/Research/SESSI/.

[62] Unofficial FreeS/WAN Support. Unofficial FreeS/WAN Support + Download Site. http://www.freeswan.ca/, 2004. Referred: 26 Nov 2004.

[63] Wi-Fi Alliance. *Wi-Fi Protected Access: Strong, standards-based interoperable security for today's Wi-Fi networks.* Wi-Fi Alliance, Apr 2003. White paper. http://www.wi-fi.org/OpenSection/pdf/Whitepaper_Wi-Fi_Security4-29-03.pdf.

Appendix A

Detailed Information

A.1 Implementation Details

Enabling zcip

The gateway and the gateway client use the following configuration to enable zcip on the ad-hoc network interface. Because zcip creates an invalid default route to the specified device, the default route is immediately deleted. However, the default route that the gateway uses to access the external network must not be deleted. In this example, eth0 is configured by using zcip that is accessed by specifying the full path.

```
File /etc/network/interfaces:
auto eth0
iface eth0 inet manual
        up /root/Software/src/zcip-4-5/zcip -i eth0; route del default;
```

Enabling pdnsd

The gateway uses the following pdnsd configuration.

```
File /usr/local/etc/pdnsd.conf:
// $Id: pdnsd.conf.in,v 1.4 2000/11/11 20:32:58 thomas Exp $

global {
        perm_cache=512;
```

```
        cache_dir="/var/cache/pdnsd";
        max_ttl=604800;
        run_as="nobody";
        paranoid=on;
        server_port=53;
        server_ip="169.254.143.251";
}

server {
        ip="130.233.47.34";
        timeout=30;
        interval=30;
        uptest=none;
        ping_timeout=50;
        purge_cache=off;
}

source {
        ttl=86400;
        owner="localhost.";
#       serve_aliases=on;
        file="/etc/hosts";
}
```

The gateway client uses the following pdnsd configuration:

```
File /usr/local/etc/pdnsd.conf:
// $Id: pdnsd.conf.in,v 1.4 2000/11/11 20:32:58 thomas Exp $

global {
        perm_cache=512;
        cache_dir="/var/cache/pdnsd";
        max_ttl=604800;
        run_as="nobody";
        paranoid=on;
        server_port=53;
        server_ip="127.0.0.1";
}

server {
        ip="169.254.143.251";
```

```
        timeout=30;
        interval=30;
        uptest=none;
        ping_timeout=50;
        purge_cache=off;
}

source {
        ttl=86400;
        owner="localhost.";
#       serve_aliases=on;
        file="/etc/hosts";
}
```

The gateway uses the following resolver configuration:

```
File /etc/resolv.conf:
search local tml.hut.fi
nameserver 169.254.143.251
```

The gateway client uses the following resolver configuration:

```
File /etc/resolv.conf:
search local
nameserver 127.0.0.1
```

The script that starts, stops, and restarts pdnsd is added. It allows the operating system to start and stop pdnsd along with other services. In addition, named must be disabled because it uses the same port as pdnsd does. The following commands are executed only once:

```
cp pdnsd /etc/init.d/pdnsd
ln -s /etc/init.d/pdnsd /etc/rc2.d/S11pdnsd
ln -s /etc/init.d/pdnsd /etc/rc2.d/K34pdnsd
rm /etc/rc*.d/*bind9
```

Enabling mDNSResponder

The mDNSResponder is added to the system in the same way as above.

Configuration

The following script provides a different configuration for the gateway and the gateway client:

```
File TestSetup:
#!/bin/sh

SMTP=130.233.228.92
GATEWAY=169.254.143.251

case "$1" in
(client)
# disable routing
echo 0 > /proc/sys/net/ipv4/ip_forward
# clear
iptables -t nat -F POSTROUTING
iptables -t nat -F PREROUTING
iptables -t nat -F OUTPUT
# enable SMTP
iptables -t nat -A OUTPUT -p tcp --dport 25 \
-j DNAT --to $GATEWAY
exit 0
;;

(gateway)
# enable routing
echo 1 > /proc/sys/net/ipv4/ip_forward
# clear
iptables -t nat -F PREROUTING
iptables -t nat -F POSTROUTING
iptables -t nat -F OUTPUT
# enable MASQUERADE
iptables -t nat -A POSTROUTING -j MASQUERADE
# enable SMTP
iptables -t nat -A PREROUTING -p tcp --dport 25 -i eth0 \
-j DNAT --to $SMTP
iptables -t nat -A OUTPUT -p tcp --dport 25 \
-j DNAT --to $SMTP
exit 0
;;
```

```
(*)
echo "Error - unknown role"
exit 1
esac
```

IPsec Configuration

IPsec is not used on the *None* level, and the following IPsec configuration is used on the *Authentication* and *Confidentiality* levels:

```
File /etc/ipsec.conf:
# /etc/ipsec.conf - FreeS/WAN IPsec configuration file
# RCSID $Id: ipsec.conf.in,v 1.11 2003/06/13 23:28:41 sam Exp $

# This file:  /usr/share/doc/freeswan/ipsec.conf-sample
#
# Manual:     ipsec.conf.5
#
# Help:
# http://www.strongsec.com/freeswan/install.htm

version 2.0 # conforms to second version of ipsec.conf specification

# basic configuration
config setup
# Debug-logging controls:  "none" for (almost) none, "all" for lots.
# klipsdebug=all
# plutodebug=all
# crlcheckinterval=600
# strictcrlpolicy=yes
interfaces=ipsec0=eth0

conn %default
# rightrsasigkey=%cert
# leftrsasigkey=%cert
keyingtries=1

# OE policy groups are disabled by default
conn block
auto=ignore
```

```
conn clear
auto=ignore

conn private
auto=ignore

conn private-or-clear
auto=ignore

conn clear-or-private
auto=ignore

conn packetdefault
auto=ignore

# Add connections here.

conn t
leftid=@gwc.local
rightid=@gws.local
left=169.254.230.146
right=169.254.143.251
leftsubnet=
rightsubnet=0.0.0.0/0
leftnexthop=
rightnexthop=
auto=start
authby=rsasig
leftrsasigkey=0sAQNpc0ogtEzfZH...
rightrsasigkey=0sAQOIjRuUciaBv6...
#Disable Opportunistic Encryption
include /etc/ipsec.d/examples/no_oe.conf
```

IPsec is enabled on the gateway by executing the following command:

```
ipsec setup --start
```

IPsec is enabled on the gateway client by executing two commands: the first enables IPsec and the second enables communication to the gateway. This is accomplished by executing the following commands:

```
ipsec setup --start
route add -host gws dev ipsec0
```

IPsec is disabled by executing the following command:

```
ipsec setup --stop
```

A.2 HTTP Test

The HTTP test repeatedly downloaded a large compressed file. The size of the file was 80444620 bytes. The following script was used to execute the HTTP test:

```
File test-http:
#!/bin/sh
i=0
n=10
while [[ $i != $n ]]
do
i=$(($i+1))
s=‘date +%s‘
lynx -source \
http://www.tml.hut.fi/~eal/linux-2.6.8.1-installed.tgz > /dev/null
e=‘date +%s‘
echo "Round $i - $(($e - $s)) seconds"
done
```

A.3 HTTP Test Results

Testing the Gateway Only

```
Round 1 - 17 seconds
Round 2 - 22 seconds
Round 3 - 21 seconds
Round 4 - 23 seconds
Round 5 - 20 seconds
Round 6 - 20 seconds
```

```
Round 7 - 21 seconds
Round 8 - 19 seconds
Round 9 - 20 seconds
Round 10 - 67 seconds
```

Testing the Gateway Client without IPsec

```
Round 1 - 34 seconds
Round 2 - 37 seconds
Round 3 - 52 seconds
Round 4 - 35 seconds
Round 5 - 27 seconds
Round 6 - 22 seconds
Round 7 - 21 seconds
Round 8 - 25 seconds
Round 9 - 32 seconds
Round 10 - 29 seconds
```

Testing the Gateway Client with IPsec

```
Round 1 - 70 seconds
Round 2 - 72 seconds
Round 3 - 71 seconds
Round 4 - 69 seconds
Round 5 - 70 seconds
Round 6 - 69 seconds
Round 7 - 69 seconds
Round 8 - 69 seconds
Round 9 - 69 seconds
Round 10 - 70 seconds
```