

Mobile WLAN access point for the ETH shuttle bus

Report**Author(s):**

Cedraschi, Nicolas; Grob, Daniel; Lenders, Vincent; Farkas, Károly; Plattner, Bernhard

Publication date:

2003-03

Permanent link:

<https://doi.org/https://doi.org/10.3929/ethz-a-004517891>

Rights / license:

[In Copyright - Non-Commercial Use Permitted](#)

Originally published in:

TIK Report 163

*N. Cedraschi, D. Grob, V. Lenders,
K. Farkas and B. Plattner*

*Mobile WLAN Access Point for the ETH
Shuttle Bus*

*TIK-Report
Nr. 163, March 2003*

N. Cedraschi, D. Grob, V. Lenders, K. Farkas and B. Plattner
Mobile WLAN Access Point for the ETH Shuttle Bus
March 2003
Version 1
TIK-Report Nr. 163

Computer Engineering and Networks Laboratory,
Swiss Federal Institute of Technology (ETH) Zurich

Institut für Technische Informatik und Kommunikationsnetze,
Eidgenössische Technische Hochschule Zürich

Gloriastrasse 35, ETH-Zentrum, CH-8092 Zürich, Switzerland

Acknowledgements

This report documents the work on a semester thesis accomplished at the *Computer Engineering and Networks Laboratory (TIK)* of the *Swiss Federal Institute of Technology Zurich (ETHZ)* during autumn 2002 and an ensuing project until February 2003.

The authors would like to thank their tutors Vincent Lenders and Károly Farkas, PhD students at TIK, for their dedicated and competent support in various aspects and Armin Brunner, Derk-Jan Valenkamp and Karl Auer of the *Informatikdienste* of the ETH Zürich for their helpful assistance and engagement. Finally, we thank Prof. Dr. Plattner for his assistant inputs.

Moreover, we would like to thank Martin Oesterle (Nokia Switzerland) and Ingo Wendler (Sunrise) and their companies for their adjuvant support and their outstanding cooperativity. We would also like to mention *Fujitsu-Siemens* who contributed two laptops at favorable conditions.

We benefited from an excellent infrastructure provided by TIK. Thus, we were allowed to focus on the important tasks.

Zurich, 7th of March 2003

Daniel Grob, Nicolas Cedraschi

Abstract

The goal of this semester thesis and the ensuing project was to develop a mobile *Access Point* for *Wireless LAN (IEEE 802.11b)*, applicable in the shuttle bus that connects the two campi ETH Zentrum and ETH Hönggerberg.

A proper *Wide Area Wireless Technology* was evaluated and an according interface was set up. This interface and a software-based Access Point were implemented on an embedded device (Set Top Box), operated by an embedded Linux (LEAF) which routes the traffic between the two interfaces. The system characteristics were tested and evaluated.

The access to the WLAN must be granted for users with a n.ethz account and should be transparent, i.e. authentication and access procedure are the same as on the fixed Access Points in the ETH WLAN. There are two authentication concepts:

- *Old Access Concept*: The user authenticates via SSH connection on a validation server. The validation server unlocks the user's IP address on the gateway firewall to the ETH LAN.
- *New Access Concept*: The user authenticates on a VPN-server and establishes a VPN-tunnel to the latter by using a dedicated VPN-software, whence he can access the ETH LAN and thence the Internet.

To realize the Access Point, two different system concepts were implemented and evaluated:

- *System Concept I* was implemented as a first prototype. It supports the New Access Concept only.
- *System Concept II* supports the Old Access Concept as well, yet was more sophisticated and therefore more difficult to realize. Its implementation was the main motivation for the ensuing project.

The labour and the investigations within the scope of this project realized a mobile Access Point that supports both of the above mentioned access concepts, but also revealed difficulties and limitations. The Wide Area Wireless Technology (GPRS) that connects the bus to the ETH WLAN, forms a bottleneck concerning data rate and delay.

Kurzfassung

Das Ziel dieser Semesterarbeit und des anschliessenden Folgeprojektes war es, einen mobilen Access Point für Wireless LAN (IEEE 802.11b) zu entwickeln, der im Pendelbus, der zwischen den Campi ETH Zentrum und ETH Hönggerberg verkehrt, eingesetzt werden kann.

Eine geeignete Technologie zur drahtlosen Verbindung zum ETH Netzwerk wurde evaluiert und eine entsprechende Schnittstelle aufgesetzt. Diese Schnittstelle und ein Software-Access-Point wurden als eingebettetes System auf einer Set Top Box realisiert. Das System wird von einem embedded Linux (LEAF) betrieben, das die zwei Schnittstellen verbindet und deren Verkehr routet. Die Systemeigenschaften wurden getestet und evaluiert.

Der Zugang zu diesem Dienst soll für User mit einem n.ethz Account möglich sein und gleich ablaufen wie bei fest installierten Access Points an der ETH. Dafür stehen dem User zwei Authentisierungsverfahren zur Verfügung:

- *Altes Authentisierungsverfahren:* Der User authentiziert sich über eine SSH Verbindung auf einem Validierungsserver. Dieser schaltet die IP des Users auf einer Gateway-Firewall frei.
- *Neues Authentisierungsverfahren:* Der User authentiziert sich bei einem VPN-Server und baut mit Hilfe einer VPN-Software einen VPN-Tunnel zu diesem auf, von wo er Zugang zum ETH LAN und zum Internet erhält.

Für die Realisierung des Access Points wurden zwei Systemkonzepte entworfen und implementiert:

- *Systemkonzept I* wurde als erster Prototyp implementiert; es unterstützt nur das neue Authentisierungsverfahren.
- *Systemkonzept II* bietet auch das alte Authentisierungsverfahren an, ist jedoch deutlich schwieriger zu realisieren. Dessen Implementierung war die Hauptmotivation für das erwähnte Folgeprojekt.

Die im Rahmen dieses Projektes durchgeführten Untersuchungen haben gezeigt, dass mit bestehender Technologie ein mobiler Access Point realisiert werden kann, der beide Authentisierungsverfahren unterstützt. Es hat sich aber auch herausgestellt, dass die für die Verbindung zwischen Bus und Internet in Frage kommenden drahtlosen Technologien (hier GPRS) den Flaschenhals bezüglich Datenraten und Verzögerungszeit bilden.

List of Acronyms

| | |
|--------------|---|
| AP | Access Point |
| ETSI | European Telecommunication Standard Institute |
| GSM | Global System for Mobile communication |
| GPRS | General Packet Radio Service |
| HSCSD | High Speed Circuit Switched Data |
| IEEE | Institute of Electrical and Electronics Engineers |
| LAN | Local Area Network |
| LEAF | Linux Embedded Appliance Firewall |
| PPP | Point-to-Point Protocol |
| SSH | Secure SHell |
| STB | Set Top Box |
| UMTS | Universal Mobile Telecommunication System |
| VPN | Virtual Private Network |
| WLAN | Wireless LAN |

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 1.1 | Motivation | 3 |
| 2 | Technology Review | 5 |
| 2.1 | Wireless LAN | 5 |
| 2.1.1 | WLAN, the 802.11 Standard | 6 |
| 2.1.2 | Wireless LAN Concept of ETH World | 10 |
| 2.2 | Wide Area Wireless Technology | 13 |
| 2.2.1 | Comparison | 14 |
| 2.2.2 | General Packet Radio System GPRS | 15 |
| 3 | Related Works & Technologies | 19 |
| 3.1 | Related Works | 19 |
| 3.2 | Related Technologies | 20 |
| 3.2.1 | MobileIP | 20 |
| 3.2.2 | CellularIP | 20 |
| 3.3 | Wireless LAN Business Models | 21 |
| 4 | The Access Point | 23 |
| 4.1 | Requirements | 23 |
| 4.2 | The AP Hardware | 24 |
| 4.2.1 | Wide Area Wireless Interface | 24 |
| 4.2.2 | The WLAN Interface | 25 |
| 4.3 | The AP Software | 25 |
| 4.4 | System Concept I | 27 |
| 4.5 | System Concept II | 29 |
| 4.5.1 | SSH-PPP-VPN | 29 |
| 4.5.2 | IPsec | 32 |

| | | |
|----------|---|-----------|
| 4.5.3 | Conclusions | 35 |
| 5 | Embedding of the Access Point | 37 |
| 5.1 | Porting System to Box | 37 |
| 5.2 | Implementation into the Bus | 37 |
| 5.2.1 | Automation | 37 |
| 5.2.2 | Monitoring & Logging | 39 |
| 5.2.3 | External Antenna | 39 |
| 6 | Evaluation & Testing | 41 |
| 6.1 | Signal Strength | 42 |
| 6.2 | Data Rate | 43 |
| 6.3 | Request/Response Time | 44 |
| 7 | Conclusions & Further Perspectives | 47 |
| 7.1 | Results | 47 |
| 7.1.1 | Wide Area Wireless Technology | 47 |
| 7.1.2 | Access Concepts | 48 |
| 7.1.3 | Testing | 48 |
| 7.1.4 | Porting to STB | 48 |
| 7.2 | Further Perspective | 49 |
| 7.2.1 | Cascading of several GPRS connections | 49 |
| 7.2.2 | Implementing a Proxy Server | 50 |
| 7.2.3 | Installation in the Bus | 50 |
| A | Used Hardware | 51 |
| A.1 | WLAN Access Point | 51 |
| A.1.1 | PC Card | 51 |
| A.1.2 | Configuration Prototype | 52 |
| A.1.3 | Configuration Set Top Box | 53 |
| A.2 | GPRS Modem | 54 |
| A.2.1 | PC Card | 54 |
| A.2.2 | Configuration Prototype | 54 |
| A.2.3 | Configuration Set Top Box | 54 |
| A.2.4 | Dial-up Scripts | 55 |
| A.3 | Set Top Box | 56 |
| A.3.1 | Features | 56 |

| | | |
|----------|-------------------------------------|-----------|
| B | Used Software | 59 |
| B.1 | Software Prototype | 59 |
| B.1.1 | Firewall & NAT | 60 |
| B.2 | Software Set Top Box | 60 |
| B.2.1 | IPsec Configuration | 60 |
| B.2.2 | Firewall & NAT | 60 |
| B.3 | Router Configuration | 61 |
| | | |
| C | Scripts & Configurations | 63 |
| C.1 | Firewall & NAT | 63 |
| C.1.1 | System Concept I | 63 |
| C.1.2 | System Concept II | 67 |
| C.2 | Automation Script | 69 |
| C.3 | The GPRS Connection | 71 |
| C.3.1 | Minicom Script | 71 |
| C.3.2 | PPP Configuration | 72 |
| C.3.3 | Sunrise Configuration | 72 |
| C.3.4 | Swisscom Configuration | 72 |
| C.4 | Router Configuration | 73 |
| C.5 | IPsec Configuration | 74 |

List of Figures

| | | |
|-----|--|----|
| 1.1 | The Access Point (AP) on the ETH Shuttle Bus Connects to the ETH Network over GPRS | 3 |
| 2.1 | 802.11 Standards within the ISO Standard | 6 |
| 2.2 | The Different Modes of 802.11 | 7 |
| 2.3 | The Wireless LAN Concept of ETH World | 11 |
| 2.4 | Wide Area Cellular Network Evolution | 13 |
| 2.5 | GPRS Network | 17 |
| 3.1 | MobileIP | 20 |
| 3.2 | CellularIP | 21 |
| 4.1 | System Concept I | 27 |
| 4.2 | System Concept II, Old Access Concept | 29 |
| 4.3 | System Concept II, New Access Concept | 30 |
| 4.4 | VPN over PPP and SSH | 30 |
| 4.5 | IPsec packet in transport mode | 33 |
| 4.6 | IPsec packet in tunnel mode | 33 |
| 4.7 | IPsec packet with additional UDP encapsulation | 34 |
| 5.1 | State Flow of the mobile Access Point automation | 38 |
| 6.1 | Route between the two Campi | 41 |
| 6.2 | GSM Signal Strength during the Bus Ride | 42 |
| 6.3 | Data Rate of TCP over GPRS. Measured with Netperf | 43 |
| 6.4 | Request/Response Time. Measured with Ping | 44 |
| A.1 | WPC11 by Linksys | 51 |
| A.2 | GlobeTrotter Universal Tri-band GPRS/GSM PC-Radio Card | 54 |

A.3 Settop Box STB3036N, Allwell 56

Chapter 1

Introduction

These days, the *Internet* enters a new stage of expansion as more and more Internet-enabled devices are being deployed in various contexts. The common perspective foresees that millions of various devices and machines are going to be connected to the Internet, building its capillaries.

With the mingling of computers and telephones into sophisticated portable devices and the progress in wireless communication technology, the Internet disperses from its current realm of classical wired desktop applications (WWW and e-mail) to mobile location-independent applications on hand-held devices, that allow to provide location based realtime information, e.g. train schedules or weather forecasts¹.

A first step towards mobile networking (Internet access) has already been taken with the adapting of mobile communication technologies like *Global System for Mobile Communications (GSM)* or *General Packet Radio System (GPRS)*. However, the data rates of GSM/GPRS are not competitive to the IEEE 802 standards, e.g. the Ethernet. The *Universal Mobile Telecommunication System (UMTS)* is announced as competitive to these standards and as a general solution for all mobile applications. The promised performances for UMTS concerning data rates and coverage are very ambitious and although researchers and telecommunication corporations all over the world have made huge efforts to overcome all technical difficulties, the actual implementation of the technical specification currently appears to be too expensive and lacks useful applications for the broad market so that its launch has been postponed.

¹http://mobile.sunrise.ch/wap/wap_lcl.htm

In the recent years various wireless LAN technologies have been introduced, e.g. the *802.11* (see Section 2.1) standards by the IEEE or *HiperLAN* by the ETSI . These are about to partly fill the gap that the postponed UMTS leaves open. Within the frame of this report the expression *WLAN* (Wireless Large Area Network) hence refers to the IEEE 802.11b standard. The original purpose is to give users mobility within a restricted area and to get rid of the rather bothersome wiring. The usage of the licence free *Industrial, Scientific, Medicinal (ISM)* frequency bands as transmission media makes these technologies applicable for a broad deployment (commercial or non-commercial).

Although WLAN does not offer the same degree of mobility as UMTS or GPRS, it offers a number of benefits such as its comparatively high data rates and its low setup costs that make it a serious competitor to mobile communication technologies, like UMTS, especially in urban environments and highly frequented locations, so-called *Hot Spots*.

1.1 Motivation

Within the scope of the ETH World project [2] as a step towards creating "a universal virtual communication and cooperation platform" called *virtual campus*, it is scheduled to integrate a WLAN infrastructure to provide access to the ETHZ infostructure and thence to the Internet. The deployment of this network is making good progress, so that currently most of the buildings of the campi (ETH Zentrum and ETH Hönggerberg) are covered by a network of IEEE 802.11b *Access Points (AP)*. However, the area between the two campi is not covered by AP. Therefore, taking the shuttle bus to change campus inevitably results in connection loss.

The goal of this thesis is to provide WLAN connectivity on the bus just as on the campi. The concept is to set up an AP for the bus, that is connected to the WLAN of the ETH over a GPRS connection (see Figure 1.1). This connection has to be transparent, i.e. the user on the bus should be able to connect to the ETH WLAN subnet, as if he was on one of the campi. A WLAN-GPRS bridge has to be developed (and deployed) that provides transparent access to the ETH WLAN subnet. From this subnet the user can then access the entire ETH LAN and thence the Internet.

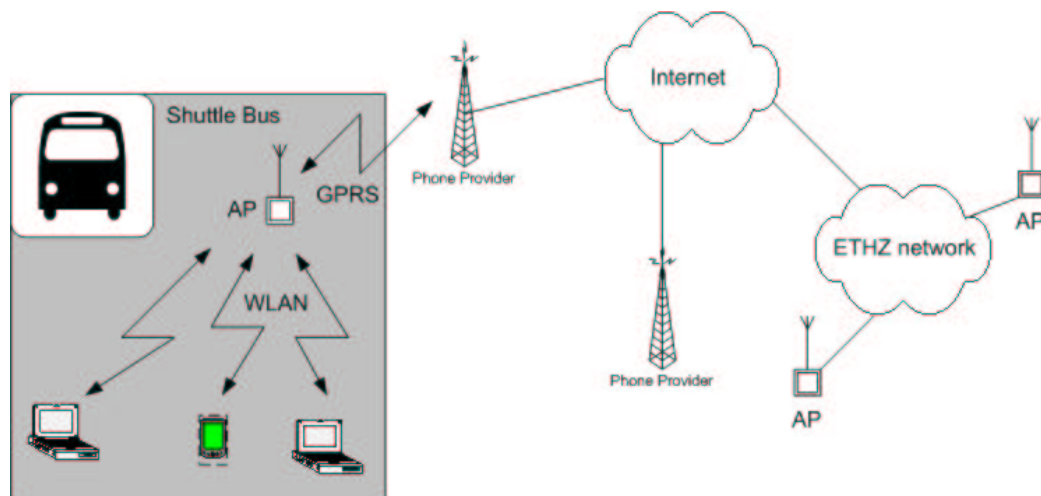


Figure 1.1: The Access Point (AP) on the ETH Shuttle Bus Connects to the ETH Network over GPRS

Chapter 2

Technology Review

2.1 Wireless LAN

WLAN [3, 4] is a flexible data communication system implemented to extend or substitute a wired LAN within a building or a campus. Using electromagnetic waves rather than a cable infrastructure, it minimizes the need for wired connections and therefore drastically reduces the cost-intensive pulling of cables through walls and ceilings. Moreover, systems can be configured in a variety of topologies to meet the needs of specific applications and installations. Topologies are easily changed and range from peer-to-peer networks, suitable for a small number of users, to full infrastructure networks.

Due to considerable progresses in the fields of radio transmission and fast integrated electronics, WLAN has seen a remarkable performance increase concerning the data rate. It is now already competitive to its older wired predecessor, the 10Mbit-Ethernet. WLAN gives way for new applications adding a new flexibility to networks.

Today's working environment is characterized by an increasingly mobile workforce. Users are equipped with notebook computers and spend more of their time working in teams that cross functional, organizational and geographic boundaries. WLAN systems provide LAN users with seamless access to real-time information within a campus, regardless of location or hardware configuration.

2.1.1 WLAN, the 802.11 Standard

The IEEE 802 committee has established the 802 standards that have driven the LAN industry for the past two decades. In 1997, after seven years of work, the IEEE published 802.11, the first internationally sanctioned standard for WLAN. In September 1999 they ratified the 802.11b "High Rate" amendment to the standard, which added two higher data rates (5.5 and 11 Mbps) to 802.11.

Like all IEEE 802 standards, the 802.11 standards focus on the bottom levels of the ISO communication standard, the physical layer and data link layer (see Figure 2.1). The basic architecture, features, and services of 802.11b are defined by the original 802.11 standard. The 802.11b specification affects only the physical layer, improving data rates and providing more robust connectivity.

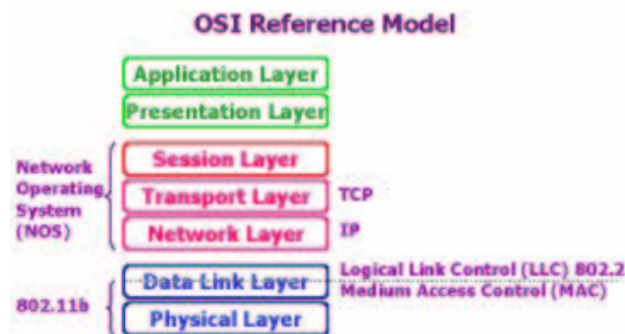


Figure 2.1: 802.11 Standards within the ISO Standard

Operating Modes

802.11 defines two pieces of equipment, a *wireless station*, which is usually a mobile device equipped with a wireless *Network Interface Card (NIC)*, and an AP, which acts as a bridge between the wireless and the wired network. An AP usually consists of a radio, a wired network interface (as defined e.g. in IEEE 802.3), and bridging software conforming to the 802.1d bridging standard. The AP acts as *Base Station (BS)* for the wireless network, aggregating access for multiple wireless stations onto the wired network.

The 802.11 standard defines two modes: *Infrastructure* mode and *Ad hoc*

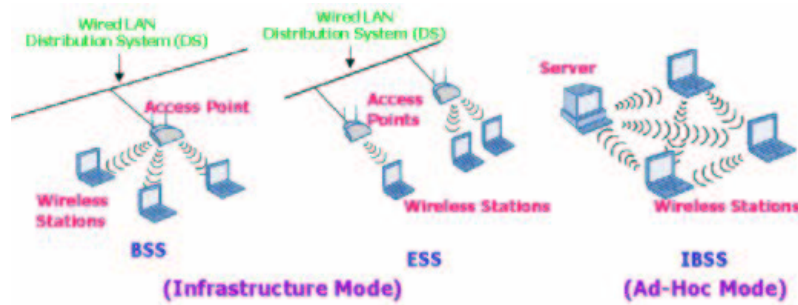


Figure 2.2: The Different Modes of 802.11

mode (see Figure 2.2). In the infrastructure mode, the wireless network consists of at least one AP connected to the wired network infrastructure and a set of wireless clients. This configuration is called *Basic Service Set (BSS)*. An *Extended Service Set (ESS)* is a set of two or more BSSs forming a single subnetwork. Since most corporate WLANs require access to the wired LAN for services they will operate on *Infrastructure* mode. The *Ad hoc* mode (also called peer-to-peer mode or *Independent Basic Service Set (IBSS)*) is simply a set of wireless stations that communicate directly with one another without using an AP or any connection to a wired network.

The Physical Layer

The three physical layers originally defined in the 802.11 standard included two spread-spectrum radio techniques and a diffuse infrared specification. Spread-spectrum techniques, in addition to increase reliability, boost throughput, and allow many unrelated products to share the spectrum without explicit corporation and with minimal interference.

The original 802.11 wireless standard defines data rates of 1 Mbps using *Frequency Hopping Spread Spectrum (FHSS)* or *Direct Sequence Spread Spectrum (DSSS)*. It is important to note that FHSS and DSSS are fundamentally different data transfer mechanisms and will not interoperate with one another.

Using FHSS, the 2.4 GHz band is divided into 75 1-MHz subchannels. Each conversation between a sender and a receiver within the 802.11 network occurs over a different hopping pattern, and the patterns are designed to minimize the chance of two senders using the same subchannel simultaneously. In contrast, the DSSS technique divides the 2.4 GHz band into 14 22-MHz

channels. Adjacent channels overlap one another partially, with three of the 14 being completely non-overlapping. Data is sent across one of these 22 channels.

The key contribution of the 802.11b addition to the WLAN standard was to standardize the physical layer support of two new speeds, 5.5 Mbps and 11 Mbps. To accomplish this, DSSS had to be selected as the sole physical layer technique for the standard.

To support very noisy environments as well as spatial range, 802.11b WLAN use *Dynamic Rate Shifting (DRS)*, allowing data rates to be automatically adjusted to compensate for the changing nature of the radio channel.

The Data Link Layer

The *Data Link Layer (DLL)* of 802.11 consists of two sublayers: *Logical Link Control (LLC)* and *Medium Access Control (MAC)*. 802.11 uses the same 802.2 LLC and 48-bit addressing as other 802 LANs, allowing for very simple bridging from wireless to wired 802 LANs, but the MAC is unique to WLAN.

The 802.11 MAC is very similar in concept to 802.3, in that it is designed to support multiple users on a shared medium by having the sender sense the medium before accessing it. 802.3 Ethernet LAN use *Carrier Sense Multiple Access with Collision Detection (CSMA/CD)* as MAC.

In a 802.11 WLAN, collision detection is not possible due to antenna limitations; a station must be able to transmit and listen at the same time, therefore it can not hear a collision. To account for this difference, 802.11 uses a slightly modified protocol known as *Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)* or the *Distributed Coordination Function (DCF)*. CSMA/CA works as follows. A station wishing to transmit senses the air interface and, if no activity is detected, waits an additional, randomly selected period of time and then transmits if the medium is still free. If the packet is received intact, the receiving station issues an ACK frame that, once successfully received by the sender, completes the process. If the ACK frame is not detected by the sending station, a collision is assumed to have occurred and the data packet is transmitted again after waiting another random amount of time.

CSMA/CA thus provides a way of sharing access over the air. This explicit ACK mechanism also handles interference and other radio related problems very effectively. However, it does add some overhead to 802.11

that 802.3 does not have, so that an 802.11 LAN will always have a lower data rate than a wired LAN.

Another MAC-layer problem specific to wireless is the *hidden node* issue, in which two stations on the opposite sides of an access point can both sense activity from an AP, but not from each other, usually due to distance or an obstruction. To solve this problem, 802.11 specifies an optional *Request to Send/Clear to Send (RTS/CTS)* protocol at the MAC layer.

Finally, the 802.11 MAC layer provides two other robustness features: *Cyclic Redundancy Check (CRC)* and packet fragmentation. Each Packet has CRC checksum calculated and attached to ensure that the data is not corrupted in transit.

Association and Roaming

When an 802.11 client enters the range of one or more APs, it chooses an AP to associate with, based on signal strength and observed packet error rates. Once accepted by the AP, the client tunes in to the radio channel to which the AP is set. Periodically, it surveys all 802.11 channels in order to assess whether a different AP would provide it with better performance characteristics. If it determines that this is the case, it reassociates with the AP, tuning to the radio channel to which that AP is set. If two APs are in range of one another and are set to use the same or partially overlapping channels, they may cause some interference for one another, thus lowering the total available bandwidth in the area of overlap.

Security

802.11 provides MAC layer access control and an encryption mechanism, known as *Wired Equivalent Privacy (WEP)*, with the objective of providing WLANs security equivalent to their wired counterparts. For the access control, the ESSID (also known as WLAN Service Area ID) is configured into each AP and is required knowledge in order for a wireless client to associate with an AP. In addition, there is provision for a table of MAC addresses called an *Access Control List* to be included in the AP, restricting access to clients whose MAC addresses are on the list.

For data encryption, the standard provides for optional encryption using a 40-bit shared-key algorithm from RSA Data Security¹. Beyond Layer 2, 802.11

¹<http://www.rsasecurity.com>

WLANs support the same security standards supported by other 802 LAN for access control (such as network operating system logins) and encryption (such as IPsec or application-level encryption).

2.1.2 Wireless LAN Concept of ETH World

As mentioned in the introduction, the WLAN concept of ETH World [5] implies the deployment of a set of APs on the two campi. In the early stage of the ETH WLAN, there was only one subnet of public IPs for all WLAN users. This allowed roaming between the buildings, but as the number of users grew and more buildings were equipped with APs, the ETH WLAN subnet exceeded a critical size, which reduced its performance and made it increasingly difficult to administrate. Therefore, in a second stage the ETH WLAN has been divided into several subnets for different buildings with corresponding routers and DHCP relays (see Figure 2.3). However, as a consequence of this, roaming between the different buildings is no longer possible.

Currently over a hundred APs are deployed in most of the ETH buildings, all physically and virtually connected together to the ETH WLAN subnet and explicitly separated from the rest of the ETH LAN. The DHCP relays of the different buildings forward DHCP requests to a central DHCP server, which manages the IPs of all WLAN subnets.

If a user connects to one of these APs, he broadcasts a DHCP request into his subnet, which will be forwarded by the corresponding router to the central DHCP server. The DHCP server in turn assigns him an internal IP of the corresponding subnet. At this point the user can network with all the other WLAN clients, who are in the same subnet, but he cannot access the rest of the ETH network or the Internet. In order to do so, he needs to authenticate. This is done in two different ways, as depicted below in the following two sections. The first access concept is older and does not require any special software for users, but it brings some security issues with it. The second access concept is just about to be introduced and is planned to replace the first access concept in the long run, since it deploys a higher degree of security.

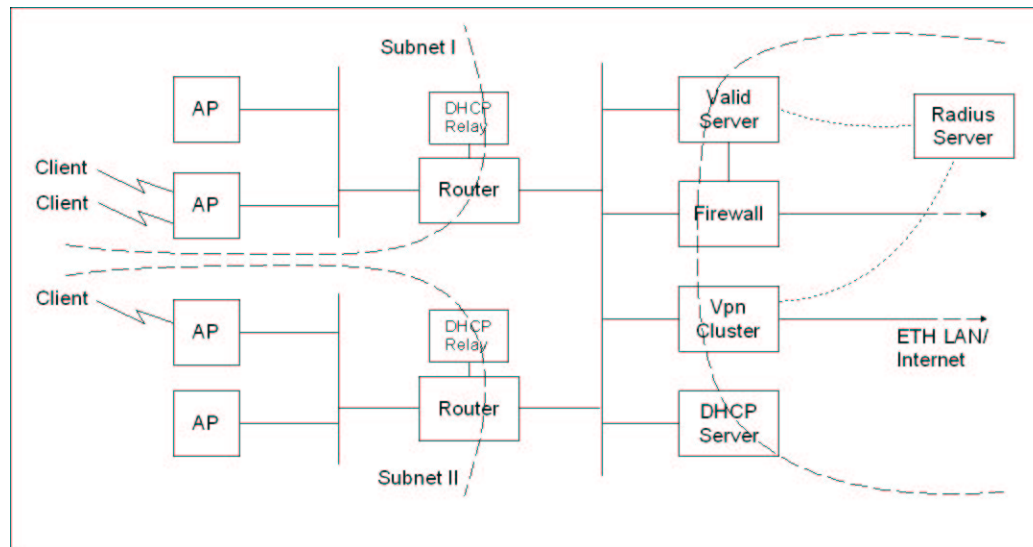


Figure 2.3: The Wireless LAN Concept of ETH World

The Old Access Concept

As mentioned above, the WLAN network of the ETH is separated from the rest of the ETH LAN or any other network. The only connection between the ETH WLAN subnet and the ETH LAN is a firewall. To get access, a user has to authenticate himself on the *Valid server*. If the authentication is successful, the user's IP is unlocked on the firewall and he is allowed to access the ETH LAN. The authentication is effectuated by a SSH or telnet login on the authentication server (Valid server).

This method introduces some security holes: First of all, the traffic of all other users in the same subnet can be sniffed and overheard over the air interface. The second problem is that the firewall does not re-lock the corresponding IP when a user logs out. It keeps the IP unlocked for at least 12 hours. So after an IP has been left it can be adopted by intruders to unauthenticated get access.

The New Access Concept

The new access concept is based on a *Virtual Private Network (VPN)*. VPN is a concept that allows a set of computer systems to communicate “securely” over a public network. Security features include encryption, strong authentication of remote users or hosts and mechanisms for hiding or masking information about the private network topology from potential attackers on the public network.

The ETH uses a software based VPN-application whose client software is downloaded and installed on the user machine. This software connects the user to a dedicated VPN-server in the WLAN network, that acts as gateway to the rest of the ETH LAN. After a successful authentication on this VPN-server, the client receives two public IPs for the two sides of the connection, and a VPN is set up between the client and the server. This method is more secure than the first one, since packets are encrypted by the client and can not be overheard over the transmission medium. It is as if the users device was physically wired to the VPN-server.

2.2 Wide Area Wireless Technology

The expression *Wide Area Wireless Technology* refers to wireless technologies providing ranges of more than one kilometer. In this section, available and upcoming technologies are presented and compared in a general overview. The most favorable system for this project is evaluated and described.

The impressive growth of cellular mobile telephony as well as of the number of Internet users promises an exciting potential for a technology that merges both: *cellular wireless data services*. Within the next few years, there will be an extensive demand for wireless data services.

There are several major *second-generation (2G)* digital cellular standards used throughout the world. The most widespread are the *Global System for Mobile Communication (GSM)*, the *Code Division Multiple Access (CDMA)* standard called *cdmaOne*, *Time Division Multiple Access (TDMA)*, and *Personal Digital Communication (PDC)* which is mainly used in Japan. In order to comply with the upcoming extensive demand for wireless data services, there will be a transition to 3G technologies that, in addition to voice services, will add support for *always on* packet data access and eventually, new multimedia types of wireless services. GPRS (2.5G) is a first step into this direction, but based and working on the same infrastructure as GSM. Figure 2.4 depicts the Wide Area Cellular Network evolution towards 3G.

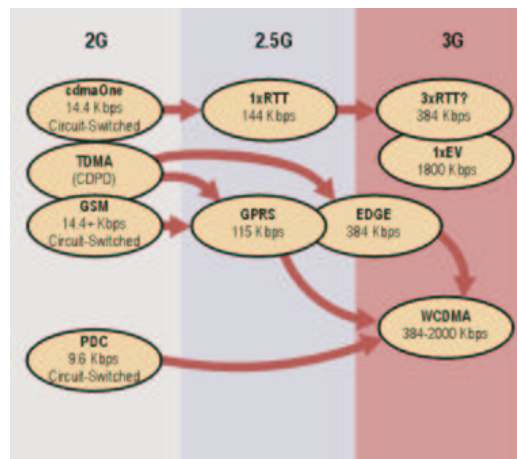


Figure 2.4: Wide Area Cellular Network Evolution

2.2.1 Comparison

Five Wide Area Wireless Technologies are compared²:

- *HSCSD: High Speed Circuit Switched Data* is an extension of GSM. It bundles GSM time slots, and thus achieves theoretical rates of up to 57,6 kbit/s (four timeslots of 14.4 kbit/s each). The obtained data rates are about 35-40 kbit/s. A HSCSD connection is billed per time unit. The network coverage corresponds to the coverage of the GSM network.
- *GPRS: General Packet Radio System* is described in detail in the next section. It is another extension of GSM that offers data rates of 30-50 kbit/s. A GPRS connection is billed per data unit. The network coverage corresponds to the coverage of the GSM network.
- *EDGE: Enhanced Data rates for GSM Evolution* is an upcoming evolution of GSM, allowing bit-rates of 48 kbit/s per time slot, i.e. 384 kbit/s in total. It is packet switched and requires relatively small changes to network hardware and software since it uses the same frame structure and bands as the existing GSM. At the moment no provider is planning on deploying EDGE in Switzerland in the near future.
- *UMTS: Universal Mobile Telecommunication System* is a wide band CDMA technology of *third-generation (3G)* mobile networks, introducing data rates of up to 2 Mbit/s under ideal circumstances, but realistic values are expected to be around 300-400 kbit/s. Its launch is ahead, but with much lower performances in the beginning. Swisscom is about to build a UMTS network which is going to operate with data rates of 64 kbit/s in a first phase.
- *Satellite Systems*: There are several technologies and providers that provide mobile connectivity over satellite, e.g. Iridium³. The performances for a handheld set are rather modest with 9.6 kbit/s. The technology is very expensive and not conceived for data transfer. There are satellite systems, which provide higher data rates, but these require expensive equipment, e.g. parabolic reflectors, that must be continuously re-directed. This technology exceeds the scope of the present project.

| Technology | Data Rate | Coverage | Billing |
|-------------------|-------------|--------------------|---------------|
| HSCSD | 57.6 kbit/s | good | per time unit |
| GPRS | 53.6 kbit/s | good | per data unit |
| EDGE | 384 kbit/s | none | per data unit |
| UMTS | 384 kbit/s | under construction | per data unit |
| Satellite Systems | 9.6 kbit/s | global | per time unit |

Table 2.1: Wide Area Wireless Technologies in Comparison

In Table 2.1 an overview of the properties of the above mentioned Wide Area Wireless Technologies is given. As seen in this table, the choice is reduced to either HSCSD or GPRS, since all the other technologies are either too expensive (Satellite Systems) or not (yet) deployed in Switzerland.

Among these two, GPRS meets the requirements for the mobile AP better, since it is packet switched, i.e. it only uses a channel, if there actually is data to transmit. This corresponds to the fluctuating traffic that the clients of the AP are expected to produce when surfing the Internet.

2.2.2 General Packet Radio System GPRS

GPRS [6, 7] is a bearer service for *Global System for Mobile Communication (GSM)* that greatly improves and simplifies wireless access to packet data networks, e.g. the Internet. It applies a packet radio principle to transfer user data packets in an efficient way between GSM mobile stations and external packet data networks. Packets are directly routed from the GPRS mobile stations to packet switched networks. Networks based on the *Internet Protocol (IP)* and X.25 networks are supported in the current version of GPRS.

Users of GPRS benefit from shorter access times and higher data rates. In conventional GSM, the connection setup takes several seconds and rates for data transmission are restricted to 9.6 kbit/s. In practice, GPRS offers

²This overview reflects the current state as of the Orbit '02 (October 2002, Basel)

³<http://www.iridium.com>

session establishment times below one second and ISDN-like data rates up to several ten kbit/s.

Moreover, GPRS packet transmission offers a more favorable billing for data traffic than circuit switched services, which is billed per time unit and is *always on*. The latter is unsuitable for applications with bursty traffic. The user pays for the entire airtime, even for idle periods when no packets are sent (e.g. when the user reads a Web page). For packet switched services, on the other hand, billing can be based on the amount of transmitted data.

To sum up, GPRS improves the utilization of the radio sources for data traffic, offers data based billing, higher transfer rates, shorter access times, and simplifies the access to packet data networks. A downside is that GPRS packets have lower priorities than speech packets, so the performance is dependent on the traffic load in the local cell.

GSM/GPRS Network Overview

GPRS uses the same physical layer as GSM, which uses a combination of *Time Division Multiple Access (TDMA)* and *Frequency Division Multiple Access (FDMA)* for medium multiplexing. Two frequency bands have been reserved for GSM operation: 890 - 915 MHz for uplink connections, and 935 - 960 MHz for the downlink connection. Each of these bands of 25 MHz width is divided into 124 single carrier channels of 200 kHz width with a gross data rate of 270 kb/s. A certain number of these frequency channels is allocated to a *Base Transceiver Station (BTS)*, i.e. to a cell. Each of these 200 kHz frequency channels carries eight TDMA channels by dividing each of them into eight time slots. The eight time slots in these TDMA channels form a TDMA frame. Each time slot of a TDMA frame lasts 156.25 bit times and, if used, contains a data burst. The time slot lasts $15/26$ ms = 576.9 μ s; so a frame takes 4.613 ms. The recurrence of one particular time slot defines a physical channel.

A GSM mobile station uses the same time slots in the uplink as in the downlink. The channel allocation in GPRS is different from the original GSM. GPRS allows a single *Mobile Station (MS)* to transmit on multiple slots of the same TDMA frame (multi slot operation). Therefore, the channel allocation is very flexible: one to eight time slots per TDMA frame can be allocated for one MS. Moreover, uplink and downlink are allocated separately, which efficiently supports asymmetric data traffic. Using 8 time slots

results in theoretical data rates of up to 171 kBit/s. However, GPRS packets have a lower priority assigned than GSM packets. Therefore, GPRS performance depends on the number of active GSM users in the same cell. The current GPRS devices are limited to use up to 4 time slots. This results in an actual data rate of about 30-50 kb/s. In conventional GSM, a channel is permanently allocated for a particular user during the entire call period (whether data is transmitted or not), whereas in GPRS the channels are only allocated when data packets are sent or received, and they are released after the transmission. For bursty traffic, this results in a more efficient usage of the scarce radio resources.

Internetworking with IP Networks

A GPRS network can be interconnected with an IP-based packet data network, such as the Internet or intranets. GPRS supports both IPv4 and IPv6. From outside, i.e. from an external IP network, the GPRS network looks like any other IP subnet. A special piece of equipment, the *Gateway GPRS Support Node (GGSN)* acts as an interface between the GPRS backbone network and the external packet data networks (see Figure 2.5). Each registered user who wants to exchange data packets with an IP network gets an IP address. The IP address is out of the address space of the GPRS operator.

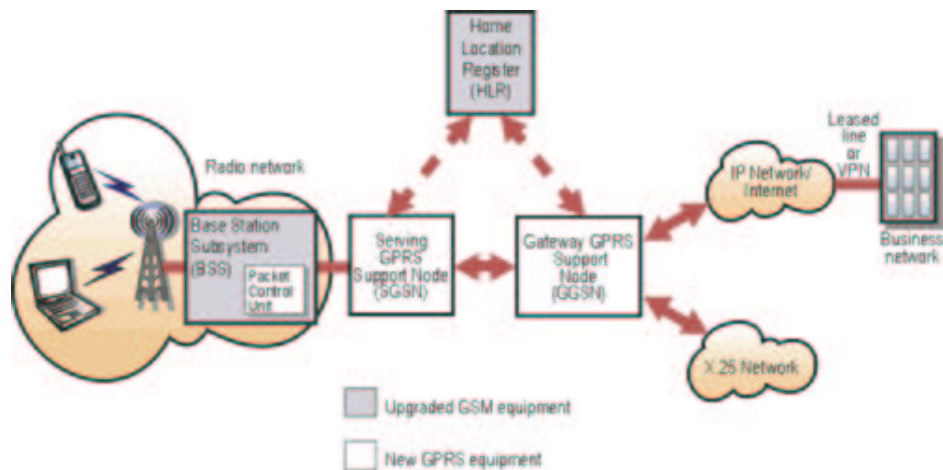


Figure 2.5: GPRS Network

Chapter 3

Related Works & Technologies

3.1 Related Works

Despite of intensive investigations on the Internet, only two comparable projects on mobile 802.11b APs were found.

A company that offers a comparable system is *Icomera*¹. Yet, information and technical details are sparse on their homepage. The *Icomera Train GatewayTM* system [8] consists of a hub on the train and the Train Gateway that is placed within the target network. These two support various wireless technologies (GPRS, Satellite etc.) to get connected. The choice of the *Wide Area Wireless Technology* is left to the client.

A different approach is taken by *Wireless Train System*² (WTS) with their *Wireless Train Service Architecture (WTSA)* concept, where the whole roadway of the train is covered by APs and repeaters along the track. The APs are connected to the Internet, e.g. via ADSL.

Both concepts are expensive, either because of the costly Wide Area Wireless Technology or an enormous infrastructural effort. *Icomera*'s solution emanates from a rather simple concept, but to offer comfortable data rates for a number of users, several wide area wireless connections must be cascaded to widen this bottleneck. These technologies are very expensive. GPRS for example is about CHF 0.10 per kByte traffic³. The solution of WTS is less cost intensive to run, but the costs of the equipment and especially their

¹<http://www.icomera.com>

²<http://www.wirelesstrain.net>

³<http://swisscom-mobile.ch/sp/FDAAAAAA-de.html>

setup and installation are very expensive, since a whole physical network has to be built along the track and the APs and repeaters have to be supplied with power.

3.2 Related Technologies

3.2.1 MobileIP

MobileIP [9] is an extension of the IP protocol. It deals with the problem of handling a large number of mobile stations moving fast between different radio cells (Handoff) by using two addresses: The *home address* and the *care-of address*. The home address is static, whereas the care-of address changes at each new point of attachment. Moreover, MobileIP defines two entities to provide mobility support: a *Home Agent* (HA) and a *Foreign Agent* (FA) (see Figure 3.1).

The *Mobile Station* (MS) sends packets to a host. On their way back to the MS, the answer packets of the host are routed to the HA, since the MS is attached to the foreign network with its care-of address and not its home address. The HA redirects the answer packets through an IP tunnel to the FA by adding a new header with the care-of address as destination. The FA unwraps these packets and forwards them to the MS.

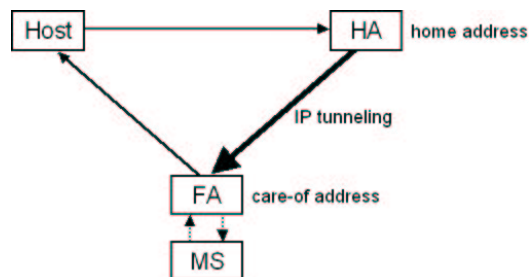


Figure 3.1: MobileIP

3.2.2 CellularIP

CellularIP [10] is a new protocol for mobile hosts that is optimized to provide access to a MobileIP enabled Internet with support of fast moving wireless hosts (see Figure 3.2). It inherits cellular systems principles for mobility

management, passive connectivity and handoff control. The central components of a CellularIP network are the *Base Station (BS)* and a gateway router. Mobile hosts attached to the network use the IP address of the gateway as their care-of address. Figure 3.2 illustrates the path of the packets addressed to a mobile host. The gateway “detunnels” packets and forwards them toward the BS. Inside the CellularIP network, mobile hosts are identified by their home addresses and data packets are routed without tunnelling or address translation.

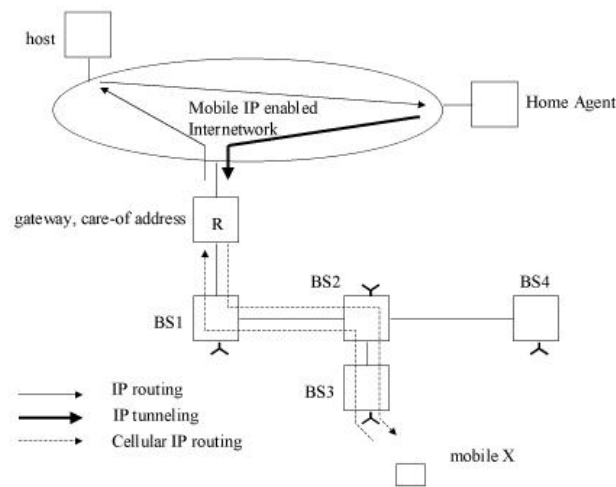


Figure 3.2: CellularIP

3.3 Wireless LAN Business Models

Currently a number of companies and universities provide wireless LAN to allow employees or students an ubiquitous internet access within their buildings. Moreover, a number of companies have started to setup APs in highly frequented public places, so-called *Hot Spots*. Among these companies are the classical Telecom companies like Swisscom⁴, but since the setup of a WLAN network is comparatively inexpensive and there is no licence fee for the usage of the frequencies, there are also new companies (e.g. Monzoon⁵)

⁴<http://www.swisscom-mobile.ch/sp/9DAAAAAA-de.html>

⁵<http://www.monzoon.ch>

entering this promising market. However, the WLAN providing business is still “under construction”, many problems lack elegant solutions e.g. billing and authentication.

Since the launch of UMTS has been delayed, there are now efforts to get the best out of the existing technologies. Nokia offers a PCMCIA-card (D211⁶) that combines both technologies (GPRS & WLAN) and allows seamless roaming between them.

⁶<http://www.nokia.com/phones/nokiad211>

Chapter 4

The Access Point

We now turn to the discussion of the “Mobile WLAN Access Point for the ETH Shuttle Bus” as it has been conceived and implemented within the scope of the thesis and the ensuing project.

The first section of this chapter specifies the requirements that the AP has to meet. The next two sections deal with the AP hardware and its interfaces and the AP software, followed by the presentation of the two system concepts that were developed and implemented during this project. The first system concept was implemented during the semester thesis and the second during its sequel.

4.1 Requirements

The AP has to meet various criteria:

- *Performance:* The AP should provide several users on the bus with wireless Internet access at comfortable data rates and with reasonable delay time.
- *Embedding:* The system should run on an embedded platform, more precisely, on a Set Top Box (STB), which is ideal for this kind of purposes, since it is small, compact, silent and has a low power consumption. The STB should be operated by an embedded Linux.
- *Transparency:* Although the AP is not physically connected to the ETH WLAN subnet, but routed through the Sunrise cellular network

and the Internet, the AP should act just like any regular AP on the campus to the user. Access and authentication should work like on the campus, though all traffic crosses the Internet before reaching the ETH network.

- *Security:* The AP should suffice the same security standards as any other AP of the ETH WLAN subnet.
- *Power Supply:* The AP should be embedded on the bus, i.e. it should not be depending on any dedicated power supply systems, but be integrated into the bus' power supply. Moreover, it should also work when the motor of the bus is turned off. Thus, it has to be equipped with a rechargeable battery that is charged while the motor of the bus is turned on.
- *Automatic Maintenance:* The AP must be fail-safe. An exception handling mechanism must cope with routine errors.

4.2 The AP Hardware

The prototype is based on a Fujitsu-Siemens Laptop, operated by Debian Linux (Kernel 2.4.19). Besides other interfaces, it has an Ethernet port, an integrated WLAN card and two PCMCIA slots.

For the usage on the bus, the system should be ported to the Set Top Box STB3036N (STB) by GCT Allwell¹ (See Appendix A.3). This box is an embedded PC, composed of standard PC components, with passive cooling. The processor is a GEODE GX1 (32-bit x86, with MMX compatible instruction set support). This STB provides one PCI slot, which can be equipped with a PCMCIA-Adapter (e.g. P222 by Elan Digital Systems²), which offers two PCMCIA slots. Moreover, it has an integrated Ethernet port and two IDE slots.

4.2.1 Wide Area Wireless Interface

As pointed out in Section 2.2, GPRS is chosen as Wide Area Wireless Technology. It meets the above-mentioned requirements best, since it is packet

¹<http://www.allwell.com.tw/>

²<http://www.elan-digital-systems.co.uk/adaptor/data.php>

switched and available on the route between the two campi.

As GPRS interface a GPRS/GSM card (*Globetrotter*, see Appendix A.2) by Option³ is used. Using one of the PCMCIA slots and the *serial_cs* kernel module, it is addressable like a serial device. For the ensuing project the card was supplemented with an external antenna, thus improving the signal susceptibility of the card and there with the data rate of the connection.

4.2.2 The WLAN Interface

To avoid the incorporation of a dedicated hardware AP, i.e. to keep the system compact, the AP interface is realized as a firmware/software AP. The HostAP Driver⁴ by Jouni Malinen, enables a commercial WLAN card, that is based on the Prism Chipset 2/2.5/3, to act as an AP. The HostAP drivers are loaded as Linux kernel modules. Moreover, HostAP supports a number of other features, e.g. AP bridging, monitor mode, and support for wireless tools.

4.3 The AP Software

The STB is operated by WISP, a distribution of an embedded Linux called *LEAF (Linux Embedded Appliance Firewall)*⁵, which is specialized on networking and wireless applications. LEAF is a slim, functional, non-graphical operating system based on a current Linux kernel, that provides the absolutely necessary features only. The various components and applications of LEAF are packed up and compressed into LRP-packages. It is thus very small and can be stored on a floppy disk or, as it is the case for the AP, a Compact Flash card.

At boot time, the LRP packages are unpacked and the operating system is assembled in the main memory, where it runs without any permanent memory or harddrive. There are additional LRP-Packages for a large variety of networking applications, e.g. IPSec, awk, wireless-tools and qmail. Packages for other applications can be created. All changes on the system that are to be permanent, have to be backed up in the according package on the media that stores the system, when not running. A more detailed description of

³<http://www.option.com>

⁴<http://hostap.epitest.fi>

⁵<http://leaf.sourceforge.net>

the configuration of the AP on the STB and the added packages are found in Appendices A and B.

Thus, routing and firewalling is provided by the kernel. Automation and maintenance routines are programmed as scripts. For further details see Chapter 5 and Appendix B

4.4 System Concept I

In the following sections two system concepts to achieve the aforementioned requirements are presented and investigated. The first of these was implemented and tested as part of the semester thesis, whereas the second was then only theoretically presented and discussed. System Concept II has been the main motivation for the ensuing project, during which it was implemented as well as ported to LEAF on the STB.

The first system concept is based on a firewall on the AP and supports the New Access Concept (see Section 2.1.2) of the ETH WLAN only. The setup of System Concept I is depicted in Figure 4.1.

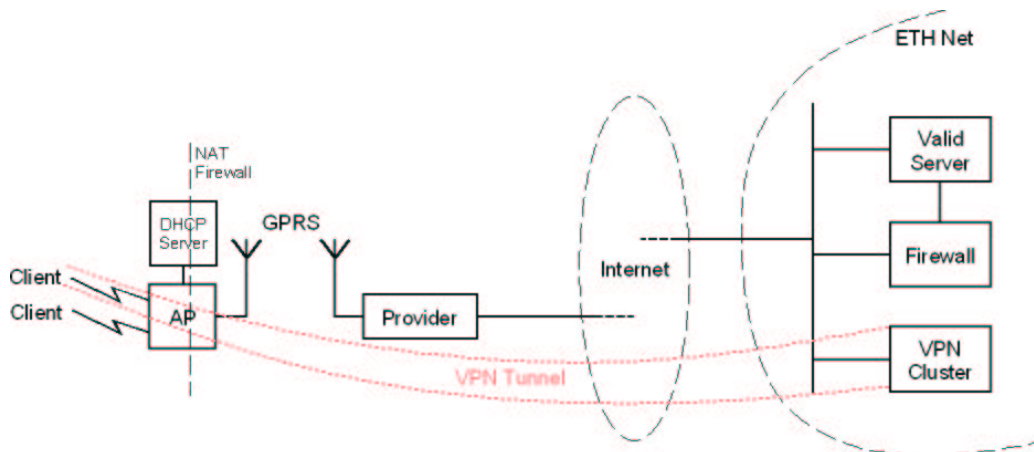


Figure 4.1: System Concept I

The mobile AP manages and operates a dedicated ETH-subnet of private IPs (172.30.199.0/24). The IP addresses are assigned by the DHCP-server on the AP. This subnet cannot be reached from outside, since these private IPs are not routed in the Internet. A *Network Address Translation (NAT)* gateway on the AP translates the private AP-subnet IPs to the IP of the GPRS point-to-point connection.

To access the ETH WLAN subnet, the client has to run a software *VPN-client*, provided by n.ethz⁶. The VPN-client connects to the VPN-server of

⁶<http://n.ethz.ch/>

the ETH and sets up an IPsec tunnel, through which the entire traffic of the client is routed. The address of the VPN-server is pre-configured in the VPN-software. Since the AP is connected to the ETH network via GPRS, i.e. it includes provider internal NAT, the VPN-client has to be configured to set up a TCP connection and NAT must be enabled. To avoid that a user accesses the Internet without passing the authentication on the VPN-server, the firewall (see Appendix C.1.1) allows traffic of the mobile AP to the following sites only:

- *VPN-server of ETH WLAN*: Gateway to the ETH WLAN, where all users must authenticate and set up a VPN-connection to get to the ETH LAN and thence to the Internet.
- The n.ethz homepage, where a client gets the VPN-client software.
- The *Domain Name Server* of the GPRS Provider. To enable the user to resolve the names of the VPN-server and the n.ethz homepage.

A dedicated IP-address (172.30.199.240) is reserved for maintenance reasons, and thus not assigned by the DHCP server.

Pros & Cons of this System Concept

- + The authentication happens on the VPN-server, which queries the central Radius-server. Therefore, it suffices to have a n.ethz account and the above-mentioned VPN-client software.
- + Smooth incorporation into the ETH WLAN without any modifications of the existing infrastructure.
- + Embedded system with no further components or outstations. It is thus easy to maintain.
- The Old Access Concept via valid server is not implemented, since the user traffic cannot be routed via GPRS connection and the ETH-firewall.

4.5 System Concept II

As pointed out in Section 2.1.2, the ETH WLAN is separated from the rest of the ETH network by a dynamic firewall or the VPN-server. To extend System Concept I and to enable the Old Access Concept as well, the entire traffic of all the AP clients must be explicitly routed into the ETH WLAN subnet by the AP. This is achieved by establishing a VPN-tunnel that connects the AP to a dedicated router within the ETH WLAN subnet (see Figures 4.2 and 4.3).

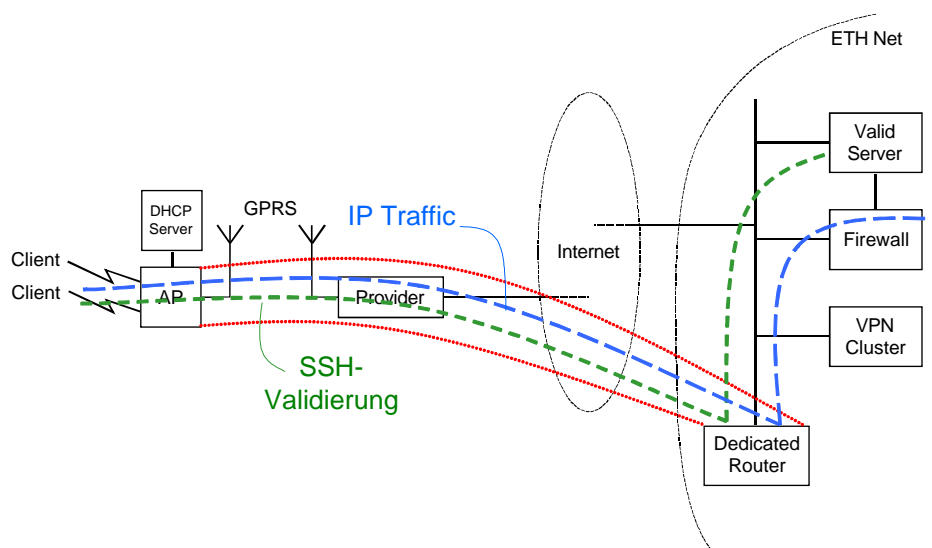


Figure 4.2: System Concept II, Old Access Concept

Thus, all traffic from the bus is bound for this router, whence a user is free to either authenticate on the Valid server (see Figure 4.2) or to set up his own VPN within the VPN-tunnel of the AP, and connect thus to the VPN-server (see Figure 4.3). Intensive investigations revealed two feasible concepts to set up such a VPN-tunnel. These are presented in the subsequent two sections.

4.5.1 SSH-PPP-VPN

A first approach is to establish a Secure SHell (SSH) connection (see Figure 4.4) on the GPRS connection. Through this connection a *Point-to-Point*

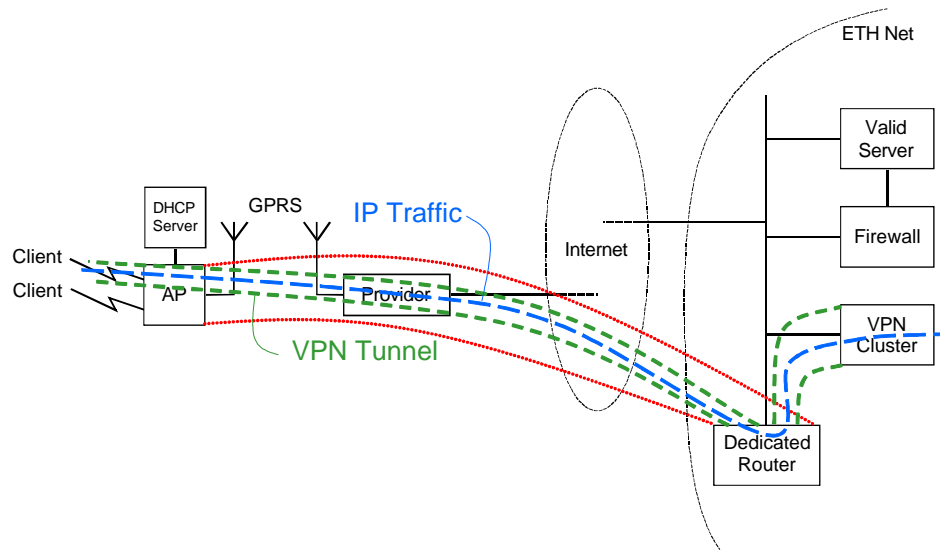


Figure 4.3: System Concept II, New Access Concept

Protocol (PPP) connection is routed. The TCP packets of this PPP connection are converted into an encrypted character stream. The traffic is thus tunneled through the Internet. This enables forwarding between different subnets [11].

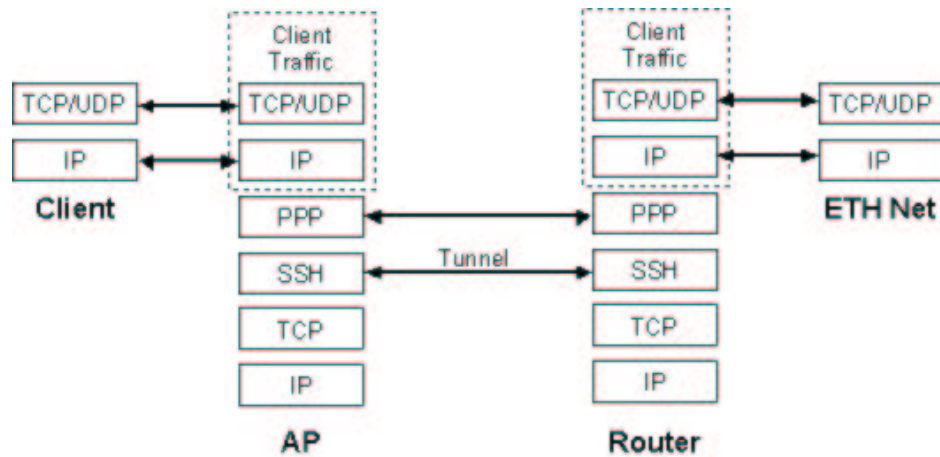


Figure 4.4: VPN over PPP and SSH

Pros & Cons

- + Supports both access concepts.
- + No mucking with firewalls If the SSH protocol traverses the connection, then PPP over SSH traverses as well.
- + PPP-SSH VPN's have no problems with dynamic IP addresses.
- If the SSH TCP connection is broken for any reason, the VPN goes down hard and takes all tunnelled TCP-connections with it.
- Works well with moderate loads over a reliable connection, but might cause some scalability problems. Has to be tested.
- Requires a dedicated router with SSH support within the ETH WLAN subnet.

4.5.2 IPsec

The second applicative VPN-tunnel concept employs IPsec⁷. IPsec is a security concept that combines 4 different transport protocols to ensure privacy on point-to-point connections across the Internet. It does so by using security services featuring various levels of encryption and authentication, namely the following two [12]:

- *Authentication Header (AH)* [13], which supports access control, connectionless message integrity, authentication and antireplay protection.
- *Encapsulating Security Payload (ESP)* [14], which supports access control, connectionless message integrity, authentication, antireplay protection and confidentiality.

The present setting requires ESP, since the IPsec AH protocol incorporates a cryptographic checksum including the IP addresses in the IP header. As masquerading changes those IP addresses and since the cryptographic checksum cannot be recalculated by the masquerading firewall, the masqueraded packets will fail the checksum test and will be discarded by the remote IPsec gateway. Therefore, IPsec VPNs that use the AH protocol cannot be successfully masqueraded. ESP with authentication can be masqueraded

Both AH and ESP support two transmission modes:

1. The *transport mode* mainly provides end-to-end protection, where the IP packet payload is encrypted. The decrypted package does not contain an IP header (see Figure 4.5) and is thus not routable in a subnet at the remote end of the IPsec connection. It is applicable for host-to-subnet connections only.
2. The *tunnel mode* encapsulates (encrypts) the entire IP packet (including the header) within a new IP-packet (see Figure 4.6) to ensure that no part of the original packet is visible or may be changed as it is forwarded through a network. The decrypted packet is therefore routable within the remote subnet. Thus, it is applicable for subnet-to-subnet connections.

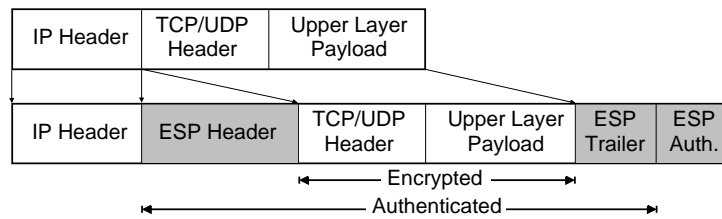


Figure 4.5: IPsec packet in transport mode

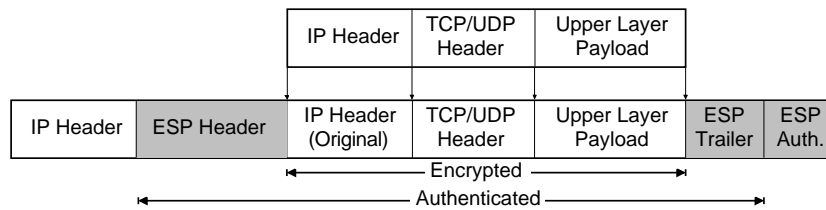


Figure 4.6: IPsec packet in tunnel mode

The present setting requires tunnel mode, as an entire subnet has to be routed through the VPN-tunnel.

NAT Implications

A major issue is the *Network Address Translation (NAT)* within the provider network. Each GPRS connection is associated to a provider-internal IP address. To cross the Internet it is either mapped 1-to-1 on an assigned public IP address (classical NAT) or together with all other connections on one public IP address (many-to-one), distinguished by different ports on transport layer. This method is either referred to as *Port Address Translation (PAT)* or *Network Address Port Translation (NAPT)*. As public IP addresses are a scarce resource, PAT is much more common than NAT.

In order to perform PAT, the ESP-packages need to be wrapped into transport layer packets (UDP/TCP) (see Figure 4.7), since the transport header contains the connection specific port number. For the tunnel mode, as used in this scenario, the IPsec packets are structured as depicted in Figure 4.6. The ESP packet is wrapped into an IP packet only and has therefore no assigned port number. Thus, PAT cannot be performed.

⁷http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113t/113t_3/ipsec.htm

Yet, the ESP header itself comprises a specific parameter to distinguish different IPsec connections (called *Security Associations (SA)*). The *Security Parameters Index (SPI)* is an arbitrary 32-bit value that, in combination with the destination IP address and security protocol (ESP), uniquely identifies the IPsec connection. This SPI is used to map different SA on one IP address.

A GPRS connection (without IPsec), when set up, gets an internal IP address within the provider network. On the providers gateway into the Internet, it is mapped with PAT on a public IP.

For a GPRS connection with IPsec tunnel, the provider needs an extra module on his gateway that performs ESP-mapping, using the connection-specific SPI to distinguish the connections.

Tests with the three local providers revealed, that only Sunrise and Orange feature the latter, whereas Swisscom just discards ESP-packages. A feasible way to solve this problem is by leasing a public IP address. Thereby, any NAT could be avoided, which also means a reduction of the round trip time (RTT) of up to 0.5 seconds.

A different approach to solve this issue is to encapsulate the ESP packets into UDP packets (called UDP-encapsulation [16]) (see Figure 4.7). The UDP header contains a port number, thus PAT is applicable. Freeswan is about to introduce a UDP-encapsulation feature. An according version is presently in test stage. It might be considered at a later time, since it would allow to have a third provider for a possible cascading of GPRS connections. Moreover, there are proprietary solutions that utilize UDP-encapsulation or similar concepts, as for example the aforementioned Cisco VPN-client (see Section 2.1.2) that is used with the New Access Concept.

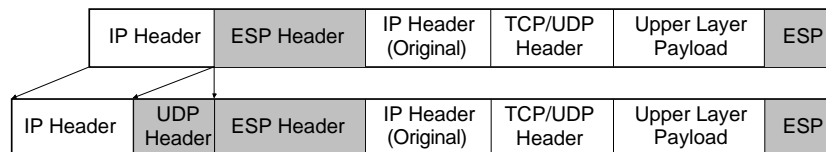


Figure 4.7: IPsec packet with additional UDP encapsulation

Pros & Cons

- + Supports both access concepts
- + Stable implementations and widely used configurations are available
- Requires special precautions for the provider internal NAT
- Requires a dedicated router within the ETH WLAN subnet
- generates additional overhead

4.5.3 Conclusions

The advantages of the IPsec concept outweigh those of the SSH-PPP-VPN concept, as similar settings using Freeswan are widely used and therefore supported. Thus, this concept was implemented on the STB.

System Concept II routes users on the bus directly into the ETH WLAN network, as if the mobile AP was physically connected to it. Thence, they are free to either authenticate on the Valid server and cross the firewall or build up a VPN-tunnel to the VPN-server.

The required transparency is created, but in return an additional router must be set up and maintained within the ETH WLAN. The addressed NAT implications were investigated and a working solution was found. For further details about the implementation see Appendix B.2.

By additionally setting up a *Generic Route Encapsulation (GRE)* tunnel through either of these tunnels (SSH-PPP-VPN or IPsec) dynamic routing (e.g. OSPF) is applicable.

In the present case, the routing is statical since the topology of the attached network is simple, and is performed on the dedicated router within the ETH WLAN,

Chapter 5

Embedding of the Access Point

5.1 Porting System to Box

The STB is described in detail in Appendix A.3. By using a PCI-PCMCIA adapter with two PCMCIA slots, the according cards for the WLAN and the GPRS interface as described in Sections A.1 and A.2 are integrated. The packages of the embedded Linux, (see Chapter 4.3), are stored on a 128Mbyte Compact Flash card. The Compact Flash drive is connected to the IDE-bus and is thus bootable. The rest of the hardware is basically analogous to the prototype and is set up accordingly. The Ethernet interface may be used to access the box (via SSH) for maintenance purposes.

5.2 Implementation into the Bus

5.2.1 Automation

When booted, the AP automatically sets up all interfaces, starts the DHCP daemon and sets up the GPRS connection and the IPsec-tunnel. Moreover, a periodically invoked cron job, checks the connection and the IPsec-tunnel and reconfigures and restarts them, in case they are not working properly. The state diagram in Figure 5.1 on the next page illustrates the state flow of this setup/check script (see Appendix C.2).

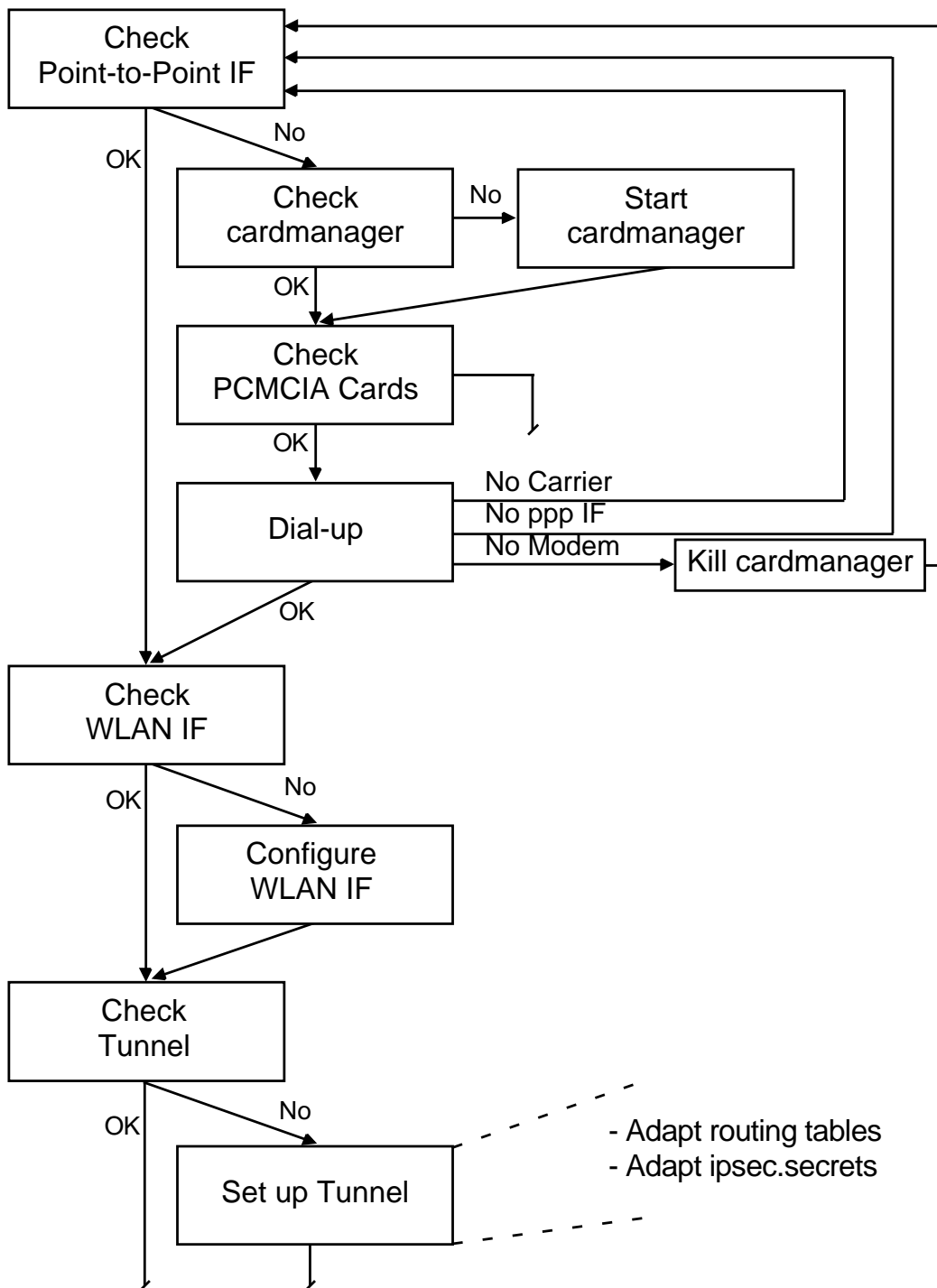


Figure 5.1: State Flow of the mobile Access Point automation

5.2.2 Monitoring & Logging

The AP is planned to send out mails with statistics about usage and traffic of the box. This has to happen after the GPRS connection is up, but before the IPsec-tunnel is up, since the AP as gateway is not part of the subnet and therefore cannot send any traffic through the tunnel. This also leaves some troubles to solve for a possible monitoring of the AP, since it cannot be pinged from the other side of the VPN.

5.2.3 External Antenna

The GPRS card is equipped with an external antenna, that notably improves the signal susceptibility . This helps to bridge possible coverage gaps on the route between Zentrum and Höggerberg.

Chapter 6

Evaluation & Testing

To evaluate the implemented prototype on the route between the two campi (see Figure 6.1), three metrics were selected to examine its performance. These are presented in the following sections and the received values are discussed. Since it is obvious that the deployed GPRS connection is the bottleneck for the whole system, the performance evaluation focuses on the GPRS connection.

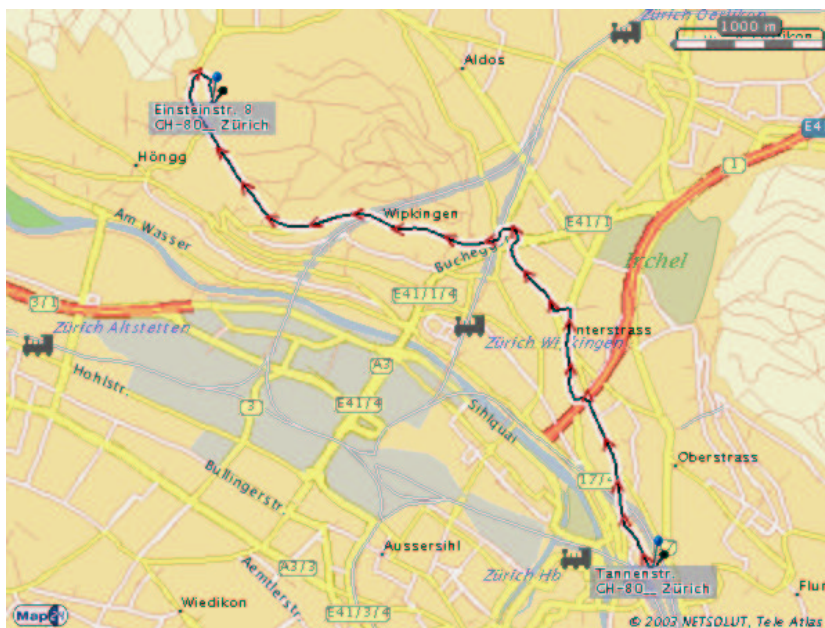


Figure 6.1: Route between the two Campi

The measurements were performed with the aforementioned GPRS modem (see Appendix A.2) and a Swisscom subscription, within the scope of the semester thesis, i.e. without the later on added antenna.

6.1 Signal Strength

In a first serie of tests the strength of the received signal on the route was measured. The GPRS card features a function (*at+csq*) that outputs the current signal strength in *dBm*. The range of the measurement runs from $-111dBm$ to $-51dBm$. *dBm* is converted to *mW* according to

$$P_{mW} = 10^{\left(\frac{P_{dBm}}{10}\right)}.$$

Figure 6.2 illustrates the mean values of the series. It illustrates that the values at the stations *Schaffhauserplatz* and *Weihensteig* are above average and that the signal strength gets weaker leaving the densely populated area (between *Im Wingert* and *Hönggerberg*) and stronger again approaching *ETH Hönggerberg*. These characteristics are confirmed by the data rates of Figure 6.3 in the subsequent Section.

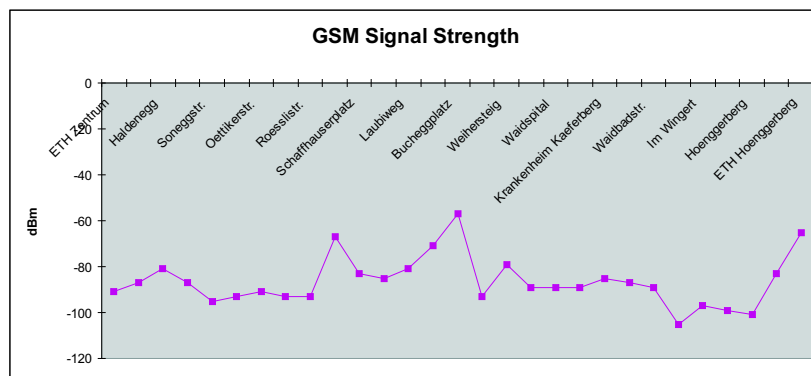


Figure 6.2: GSM Signal Strength during the Bus Ride

6.2 Data Rate

Figure 6.3 displays the data rates measured with *Netperf*¹. This program generates a TCP stream of 16kB messages to determine the data rates of a connection. The results are depicted in Figure 6.3 in 10kb/s. The resulting average is about 1.3kB/s. Moreover, there is a connection gap around *Hönggerberg*.

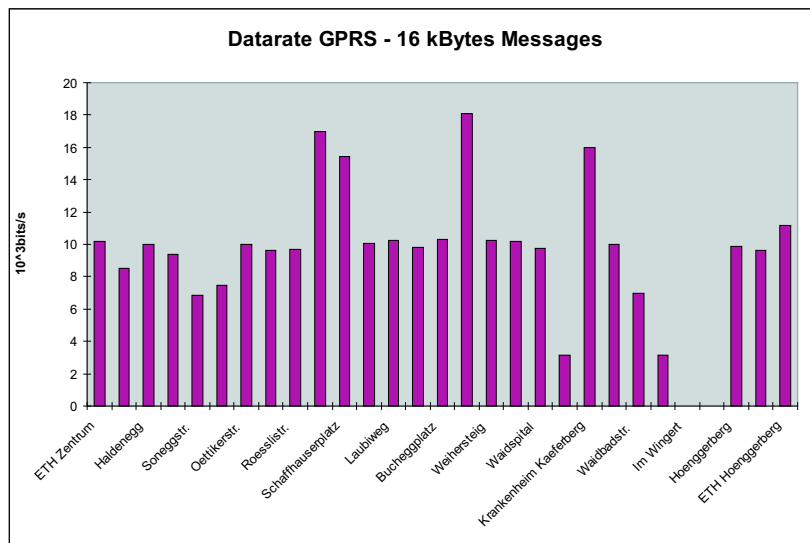


Figure 6.3: Data Rate of TCP over GPRS. Measured with Netperf

The Test with Netperf is problematic, since the results of the measurements are clearly below the expected values of 30 - 50 kbits/s. Tests showed that the data rate must be higher. The download of a test file with a regular browser resulted in a mean data rate of about 4kB/s. Lack of time prevented a closer examination of this discrepancy within the scope of this project.

¹<http://www.netperf.org/netperf/NetperfPage.html>

6.3 Request/Response Time

The round-trip time was measured with *ping*. The illustrated values (see Figure 6.4) are the mean of a serie of measurements. The measuring unit is *ms*. This results in a round-trip average latency in a range between one and two seconds.

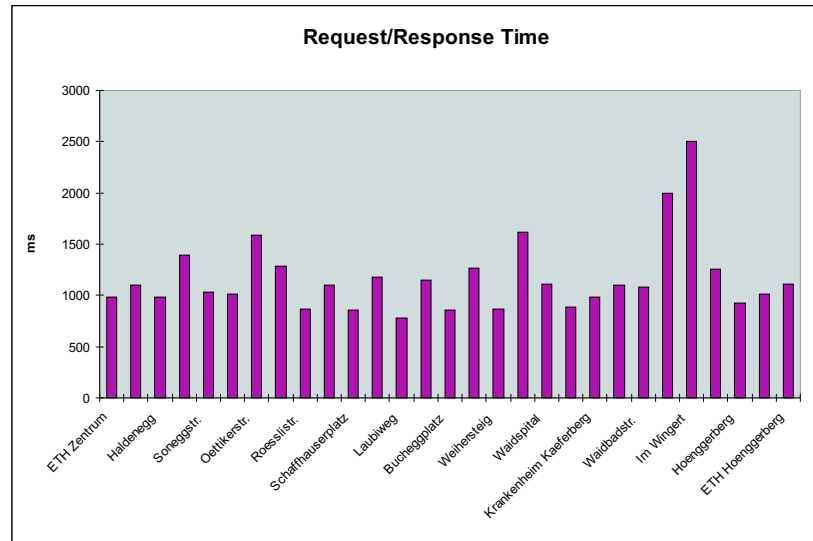


Figure 6.4: Request/Response Time. Measured with Ping

The major part of the time is spent within the GPRS network as seen in the following listing of a traceroute from the AP to a server within the ETH LAN. The first three columns show the results of three different measurements. Each value represents the time to the gateway (his address is noted in the last column) and back to the testing host.

Since all values are over $700ms$ the conclusion is drawn that about $600-700ms$ are spent within the GPRS network.

traceroute to pc-3298.ethz.ch [129.132.57.118], 30 hops max:

| | | | | |
|----|---------|---------|---------|---|
| 1 | 762 ms | 1069 ms | 900 ms | 10.141.200.254 |
| 2 | 935 ms | 1027 ms | 1051 ms | 10.141.200.254 |
| 3 | 1040 ms | 870 ms | 860 ms | 192.168.177.65 |
| 4 | 1260 ms | 1040 ms | 1027 ms | 172.25.145.2 |
| 5 | 1103 ms | 862 ms | 1419 ms | 172.25.145.9 |
| 6 | 983 ms | 867 ms | 871 ms | 192.168.177.85 |
| 7 | 1692 ms | 859 ms | 848 ms | 192.168.19.13 |
| 8 | 917 ms | 1051 ms | 859 ms | 138.188.101.249 |
| 9 | 1378 ms | 1047 ms | 1044 ms | 192.168.8.10 |
| 10 | 979 ms | 1050 ms | 860 ms | 192.168.10.1 |
| 11 | 1031 ms | 859 ms | 1388 ms | 194.209.131.132 |
| 12 | 886 ms | 1027 ms | 871 ms | 164.128.83.1 |
| 13 | 1006 ms | 859 ms | 848 ms | i71lzw-010-FastEthernet2-0.ip-plus.net [164.128.84.254] |
| 14 | 1039 ms | 1050 ms | 860 ms | i79zhh-015-ser5-1-1.ip-plus.net [164.128.33.201] |
| 15 | 721 ms | 863 ms | 1028 ms | zhb-005-GigEth8-0.ip-plus.net [164.128.33.33] |
| 16 | 861 ms | 862 ms | 848 ms | i79tix-005-GigEth1-2.ip-plus.net [164.128.34.146] |
| 17 | 1213 ms | 1028 ms | 1050 ms | Switch-1.ip-plus.net [164.128.33.118] |
| 18 | 861 ms | 863 ms | 848 ms | swiEZ2-G3-2.switch.ch [130.59.36.249] |
| 19 | 930 ms | 847 ms | 1051 ms | rou-eth-switch-1-giga-to-switch.ethz.ch [192.33.92.1] |
| 20 | 1185 ms | 1050 ms | 1040 ms | rou-etx-1-mega-transit.ethz.ch [129.132.99.79] |
| 21 | 841 ms | 848 ms | 1050 ms | pc-3298.ethz.ch [129.132.57.118] |

Chapter 7

Conclusions & Further Perspectives

The labour and the investigations within the scope of this thesis and the ensuing project, to build a mobile AP for the ETH Shuttle Bus, produced an embedded system that is applicable in the shuttle bus, but also revealed difficulties and limitations.

7.1 Results

A mobile AP for the Shuttle Bus is realizable with the existing technologies and infrastructure. But the Wide Area Wireless Technology, which determines the data rate and the data unit price, forms a considerable bottleneck. As for related works, only two related project were found (see Section 3.1). The subsequent sections represent a synopsis of the most important aspects of this project.

7.1.1 Wide Area Wireless Technology

The comparison of different *Wide Area Wireless Technologies* (see Section 2.2.1) proved GPRS to be the most appropriate technology for the present system, since it provides very good network coverage and is accounted per data unit. Nevertheless, data rates are limited to some ten kbit/s at a price of approximately 7 sFr. per Megabyte (prices by October 2002).

7.1.2 Access Concepts

An important functionality of the AP is the provision of transparent access to the ETH WLAN. It turned out that the New Access Concept (see Section 2.1.2) is easier to implement, as the user himself establishes a VPN-tunnel to a dedicated router, where he authenticates. Thus, no supplementary routing by the AP is required. In a first step, *System Concept I* (see Section 4.4) was successfully implemented and tested. It allows the users on the shuttle bus to connect to the VPN-Server via WLAN interface and thence access the ETH LAN.

To support both access concepts, a second concept was designed and evaluated within the scope of the ensuing project. This *System Concept II* (see Section 4.5) is based on an IPSec-tunnel to the dedicated ETH-router that is automatically set up by the Access Point at boot time. As this VPN-connection operates in tunnel mode (see Chapter 4.5.2), packages that cross the tunnel are fully routable. It thus virtually embeds the AP subnet into the ETH WLAN subnet.

7.1.3 Testing

The tests presented in Chapter 6 confirm that the *Wide Area Wireless Technology* is the bottleneck. With maximum data rates of approximately $4kB/s$, the performance is comparable to an analog modem.

The round-trip times are very high compared to other networking technologies. Thus, the system is not useful for terminal applications or streaming. The revealed coverage gap is not dramatic for applications other than the aforementioned. It is geographically limited and therefore neglectable for a best effort system as the Internet.

7.1.4 Porting to STB

System Concept II has been ported to the STB (see Sections 4.2 and 5), which deploys passive cooling and a Compact Flash card to store the operating system. As it operates without any movable components, it is less vulnerable to concussions, which is an important aspect as it is expected to run on a driving bus. The AP is operated by LEAF, an embedded Linux that is assembled in the RAM at boot time. It executes the various programs and scripts.

7.2 Further Perspective

A few things remain to be taken care of before users can actually connect to the AP on their way between the two campi. These are outlined in this section.

7.2.1 Cascading of several GPRS connections

An approach to improve the data rates of the AP and to speed up the connection is the cascading of multiple GPRS connections. This idea was theoretically examined.

Each time frame of a frequency channel within a GSM/GPRS frame consists of 8 time slots, two of which are reserved for control and management channels of the cell. The remaining 6 time slots are distributed among the GSM users and the GPRS users. As GSM is a real-time application GSM users have a higher priority on the first five time slots and the GPRS users on the last one, i.e. if there are five GSM users transmitting in a time frame (each of them has one time slot), all GPRS users have to share the remaining time slot.

Thus, two cascaded cards with a subscription to the same provider just inhibit each other mutually, since they have to share the same time slot. However, the operator can provide a second GPRS time slot per frame along the bus route, so that two GPRS connection should double the total data rate of the AP.

A second possibility is to use various providers, since they all work on different frequency channels and would therefore not inhibit each other. This solution should even triple the data rate, but would make the establishing and coordination of the GPRS connections with the present setting complicated and tedious. For both of this solutions additional GPRS cards and SIM cards were required and accordingly additional PCMCIA interfaces, i.e. the STB has to be replaced by a different embedded hardware.

Such a cascading of multiple GPRS connections has to be tested with additional hardware. For a start, a regular mobile phone may be utilized to cascade two GPRS connection and evaluate the bandwidth increase.

In case that several GPRS connections are deployed, channel balancing becomes an important issue that has to be investigated. There are tools for this kind of traffic adaption, e.g. the *Bandwidth Allocation Control Protocol*

(BACP)¹. These were examined, but not found to be applicable, as the operator has to be in control of both ends of the connections. The remote end in this case is the provider (Sunrise). Moreover, as GPRS is billed per data unit, the connections may be permanently on without causing any additional costs and hardly any additional traffic.

7.2.2 Implementing a Proxy Server

To avoid possible redundant downloading of much frequented web sides and to optimize the traffic volume, a proxy server should be implemented and deployed to cache popular web sites. Moreover, this proxy server would allow to set up an Internet gateway that automatically appears on the users browser when first started. This gateway should present possible sponsors and contain information and particularities of the mobile AP.

7.2.3 Installation in the Bus

Finally, the STB has to be embedded into the shuttle bus and supplied with power. Furthermore, the antenna should be positioned within the bus to allow the highest possible degree of signal susceptibility.

¹<http://www.networksorcery.com/enp/protocol/BACP.htm>

Appendix A

Used Hardware

A.1 WLAN Access Point

In this section the hardware and the configuration used for the WLAN interface are described.

A.1.1 PC Card

Instant Wireless Network PC Card WPC11, Linksys¹ (Figure A.1).

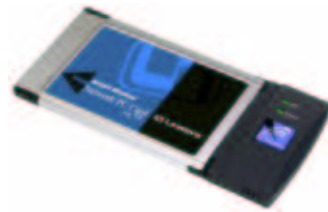


Figure A.1: WPC11 by Linksys

¹<http://www.linksys.com>

A.1.2 Configuration Prototype

Kernel Configuration for PCMCIA Support

PCMCIA/CardBus support: m(odule)
CardBus support: y(es)

Modules

yenta_socket.o : PCI-to-CardBus
ds.o : Driver Service
serial_cs.o : PCMCIA Serial Port Driver
hostap_cs.o : HostAP PCMCIA
hostap_pci.o : HostAP PCI

Mapping PCMCIA Card to Driver

in /etc/pcmcia/hostap_cs.conf

```
...  
...  
card "EMTAC A2424i 11Mbps WLAN Card"  
    manfid 0xc250, 0x0002  
#   cis "cis/Emtac.dat"  
    bind "hostap_cs"  
  
card "Linksys WPC11 11Mbps WLAN Card"  
    version "Instant Wireless ", " Network PC CARD", "Version 01.02"  
    bind "hostap_cs"  
  
card "Linksys WPC11 Ver 2.5 11Mbps WLAN Card"  
    manfid 0x0274, 0x1612  
    bind "hostap_cs"  
...  
...
```

Wireless Tools

The Wireless Extension² is a generic API allowing a driver to expose to the user space configuration and statistics specific to common Wireless LANs.

- `iwconfig` : Similar to `ifconfig`, but for wireless interfaces
- `iwevent` : Display Wireless Events
- `iwgetid` : Report ESSID, NWID or AP/Cell Address
- `iwlist` : Get Wireless statistics
- `iwpriv` : Configure optional parameters

Configuration

If the setup was successful, the output of `iwconfig` should look like the following listing or similar. The *ESSID* is set to *public* and the *Mode* to *Master*. From now on, the device can be configured like an ethernet device with `ifconfig`.

```
wlan0      IEEE 802.11-b  ESSID:"public"
           Mode:Master  Frequency:2.422GHz  Access Point: 00:03:2F:03:20:FF
           Bit Rate:11Mb/s  Tx-Power:-4 dBm  Sensitivity=1/3
           Retry min limit:8  RTS thr:off  Fragment thr:off
           Encryption key:off
           Power Management:off
           Link Quality:0  Signal level:0  Noise level:0
           Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
           Tx excessive retries:0  Invalid misc:0  Missed beacon:0
```

A.1.3 Configuration Set Top Box

The WLAN interface is configured with the WISP-Dist configuration utility `wdistconfig`. The first WLAN interface on the PCMCIA bus is called "netcs0".

²http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html

A.2 GPRS Modem

A.2.1 PC Card

GlobeTrotter Tri-band GPRS/GSM PCMCIA Modem by Option³ (Figure A.2).



Figure A.2: GlobeTrotter Universal Tri-band GPRS/GSM PC-Radio Card

A.2.2 Configuration Prototype

Kernel Configuration for Modem Support

PPP support : m (module)
PPP filtering : y (yes)
PPP support for sync tty ports : m

A.2.3 Configuration Set Top Box

additional packages: minicom.lrp
additional modules: 82365.o, serial_cs.o, ds.o

³<http://www.option.com>

A.2.4 Dial-up Scripts

To establish the ppp0 interface, a minicom script (see Section C.3.1) is invoked which in turn invokes *pon*. The minicom script registers on the GPRS network. The pon configuration is found in Section C.3.2 and the according scripts for the two providers in Sections C.3.4 (Swisscom) and C.3.3 (Sunrise).

A.3 Set Top Box

In this section the Set Top Box STB3036N (see Figure A.3) by Allwell⁴ is presented.

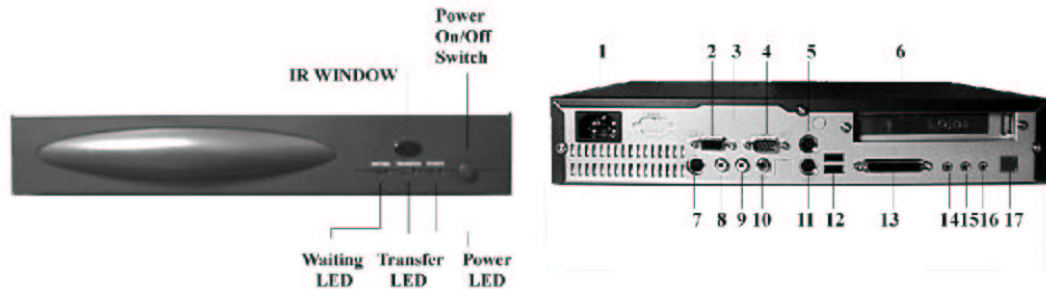


Figure A.3: Settop Box STB3036N, Allwell

Legend

| | | | |
|---|------------------|----|---------------|
| 1 | AC Power | 10 | Audio out R |
| 2 | VGA Port | 11 | Keyboard |
| 3 | Optional (Scart) | 12 | USB |
| 4 | RS-232 | 13 | Parallel port |
| 5 | PS/2 | 14 | Line out |
| 6 | Exp. Slot PCI | 15 | Line in |
| 7 | S-Video | 16 | Microphone |
| 8 | Composite Video | 17 | RJ-45 LAN |
| 9 | Audio out L | | |

A.3.1 Features

- Geode GX1 CPU 266 up to 333MHz 32-bit x86, with MMX compatible instruction set support
- Integrated Floating Point Unit (FPU)

⁴<http://www.allwell.com.tw>

- Two 168-pin DIMM sockets (max of 256MB)
- Support M-System DOC from 16-144MB
- Support 40 pin IDE interface Flash Module from 16-256MB
- 2 x Ultra DMA/33 for up to four IDE devices
- 10/100Mb Ethernet Controller
- EISA type slot support PCI and ISA add on
- One PCI 104 socket
- Board Size: Dimension 242 x 235 mm
- APM 1.2 compliant
- Award BIOS with APM and PnP
- Power Supply: 45W, 5V/3A, 12V/2A, -12V/0.3A, 90-264V Auto switching or
- Optional add-on: Smart card reader, Compact Flash card

Appendix B

Used Software

B.1 Software Prototype

Linux

Debian¹ Linux 3.0 with Kernel 2.4.19

HostAP

HostAP² driver for Intersil Prism2/2.5/3, Release 2002-10-12

Wireless Tools

Wireless Tools³ for Linux, Version 25

Cisco VPN-Client

VPN-Client⁴ for Windows and Linux, Release 3.6

¹<http://www.debian.org>

²<http://hostap.epitest.fi/>

³http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html

⁴<http://www.cisco.com/en/US/products/sw/secursw/ps2138/index.html>

B.1.1 Firewall & NAT

The script for firewall and NAT for the System Concept I as mentioned in 4.4, is found in Appendix C.1.1

B.2 Software Set Top Box

Linux

LEAF ⁵ WISP Distribution 2003-01-09(2470) using kernel 2.4.20 patched with FreeS/WAN-1.99.

Additional Packages:

| | |
|--------------|---|
| sshd.lrp | : SSH |
| minicom.lrp | : Minicom |
| apkg.lrp | : Packager for lrp packages |
| mawk.lrp | : Awk tools |
| ifconfig.lrp | : Ifconfig |
| ipsec.lrp | : IPsec |
| ap.lrp | : Scripts and Configuration files for the mobile AP |

The AP package contains the AP check/setup script, runAP (start script for AP), and the minicomscript (C.3.1).

The AP script (see Sections 5 and C.2) sets up the ppp0 interface. It is invoked firstly at boot time to establish the PPP connection and the tunnel and later on is periodically invoked by a cronjob to check and fix these.

B.2.1 IPsec Configuration

The IPsec configurations are straight forward and are found in Section C.5. Please note that the IPsec secret is changed by the AP script at each dial-up.

B.2.2 Firewall & NAT

The script for firewall and NAT for the System Concept II as mentioned in 4.5, is found in Appendix C.1.2

⁵<http://leaf.sourceforge.net>

B.3 Router Configuration

The Configuration of the dedicated router (see Section 4.5) within the ETH LAN is found in C.4. In this case, the router is a Cisco 2610. The configurations should be applicable for all routers supporting Cisco IOS and the applied security patches.

Appendix C

Scripts & Configurations

C.1 Firewall & NAT

C.1.1 System Concept I

```
#!/bin/sh
#
# rc.firewall-2.4-stronger
FWVER=0.80s

# Log:
# 0.80s - Addapted for mAP
# 0.73s - Added comments in the output section that DHCPd is optional
#         and changed the default settings to disabled
# 0.72s - Changed the filter from the INTNET to the INTIP to be
#         stateful; moved the command VARs to the top and made the
#         rest of the script to use them
# 0.70s - Added a disabled examples for allowing internal DHCP
#         and external WWW access to the server
# 0.63s - Added support for the IRC module
# 0.62s - Initial version based upon the basic 2.4.x rc.firewall
```

```
echo -e "\nLoading rc.firewall - version $FWVER..\n"

# The location of various iptables and other shell programs
IPTABLES=/sbin/iptables

LSMOD=/sbin/lsmmod
DEPMOD=/sbin/depmod
INSMOD=/sbin/insmod
GREP=/bin/grep
AWK=/usr/bin/awk
SED=/bin/sed
IFCONFIG=/sbin/ifconfig

#Setting the EXTERNAL and INTERNAL interfaces for the network
EXTIF="ppp0"
INTIF="wlan1"
echo " External Interface: $EXTIF"
echo " Internal Interface: $INTIF"
echo " ---"

# Determine the external IP
EXTIP="$IFCONFIG $EXTIF | $GREP 'inet addr' | $AWK '{print $2}' | \
$SED -e 's/.*://''"
echo " External IP: $EXTIP"
echo " ---"

# Assign the internal TCP/IP network and IP address
INTNET="172.30.199.0/24"
INTIP="172.30.199.1"
echo " Internal Network: $INTNET"
echo " Internal IP: $INTIP"
echo " ---"

# Setting a few other local variables
#
UNIVERSE="0.0.0.0/0"

#Swisscom DNS 1
SCDNS1="164.128.36.34"
echo " Swisscom DNS1: "$SCDNS1

#Swisscom DNS 2
SCDNS2="164.128.76.39"
echo " Swisscom DNS2: "$SCDNS2

#vpn-cluster ethz
VPN1="129.132.99.161"
VPN2="129.132.99.162"
VPN3="129.132.99.163"
VPN4="129.132.99.171"
```

```

echo " VPN-Cluster: "$VPN1
echo " VPN-Cluster: "$VPN2
echo " VPN-Cluster: "$VPN3
echo " VPN-Cluster: "$VPN4

#n-ethz.ch
NETHZ="129.132.97.10"
echo " n.ethz.ch: "$NETHZ

#Address for Maintenance
SERVICE="172.30.199.14"
echo " Service: "$SERVICE
echo " ---"

# Need to verify that all modules have all required dependencies
echo " - Verifying that all kernel modules are ok"
$DEPMOD -a

echo -en " Loading kernel modules: "

# NOTE: The following items are listed ONLY for informational reasons.
# There is no reason to manual load these modules unless your
# kernel is either mis-configured or you intentionally disabled
# the kernel module autoloader.
#

#Load the main body of the IPTABLES module - "ip_tables"
#
echo -en "ip_tables, "
#
if [ -z "$LSMOD | $GREP ip_tables | $AWK {'print $1'}" ]; then
    $INSMOD ip_tables
fi

#Load the stateful connection tracking framework - "ip_conntrack"
#
echo -en "ip_conntrack, "
#
if [ -z "$LSMOD | $GREP ip_conntrack | $AWK {'print $1'}" ]; then
    $INSMOD ip_conntrack
fi

#Load the FTP tracking mechanism for full FTP tracking
#
echo -e "ip_conntrack_ftp, "
#
if [ -z "$LSMOD | $GREP ip_conntrack_ftp | $AWK {'print $1'}" ]; then
    $INSMOD ip_conntrack_ftp
fi

#Load the IRC tracking mechanism for full IRC tracking

#
echo -en " ip_conntrack_irc, "
#
if [ -z "$LSMOD | $GREP ip_conntrack_irc | $AWK {'print $1'}" ]; then
    $INSMOD ip_conntrack_irc
fi

#Load the general IPTABLES NAT code - "iptable_nat"
# - Loaded automatically when MASQ functionality is turned on
#
# - Loaded manually to clean up kernel auto-loading timing issues
#
echo -en "iptable_nat, "
#
#Verify the module isn't loaded. If it is, skip it
#
if [ -z "$LSMOD | $GREP iptable_nat | $AWK {'print $1'}" ]; then
    $INSMOD iptable_nat
fi

#Loads the FTP NAT functionality into the core IPTABLES code
# Required to support non-PASV FTP.
#
echo -e "ip_nat_ftp"
#
if [ -z "$LSMOD | $GREP ip_nat_ftp | $AWK {'print $1'}" ]; then
    $INSMOD ip_nat_ftp
fi

echo " ---"

# Just to be complete, here is a list of the remaining kernel modules
# and their function. Please note that several modules should be only
# loaded by the correct master kernel module for proper operation.
# -----
#
# ipt_mark - this target marks a given packet for future action.
# This automatically loads the ipt_MARK module
#
# ipt_tcpmss - this target allows to manipulate the TCP MSS
# option for braintead remote firewalls.
# This automatically loads the ipt_TCPMSS module
#
# ipt_limit - this target allows for packets to be limited to
# to many hits per sec/min/hr
#
# ipt_multiport - this match allows for targets within a range
# of port numbers vs. listing each port individually
#
# ipt_state - this match allows to catch packets with various
# IP and TCP flags set/unset
#
# ipt_unclean - this match allows to catch packets that have invalid

```

```

# IP/TCP flags set
#
# iptable_filter - this module allows for packets to be DROPPed,
# REJECTed, or LOGged. This module automatically
# loads the following modules:
#
# ipt_LOG - this target allows for packets to be
# logged
#
# ipt_REJECT - this target DROPs the packet and returns
# a configurable ICMP packet back to the
# sender.
#
# iptable_mangle - this target allows for packets to be manipulated
# for things like the TCPMSS option, etc.

#Enable IP forwarding
echo " Enabling forwarding.."
echo "1" > /proc/sys/net/ipv4/ip_forward

echo " Enabling DynamicAddr.."
echo "1" > /proc/sys/net/ipv4/ip_dynaddr

echo " ---"

#####
# Enable IP forwarding and Masquerading
#

# Clearing any previous configuration
echo " Clearing any existing rules and setting default policy to DROP.."
$IPTABLES -F INPUT DROP
$IPTABLES -F INPUT
$IPTABLES -F OUTPUT DROP
$IPTABLES -F OUTPUT
$IPTABLES -F FORWARD DROP
$IPTABLES -F FORWARD
$IPTABLES -F -t nat

#Not needed and it will only load the unneeded kernel module
#$IPTABLES -F -t mangle
#
# Flush the user chain.. if it exists
if [ -n "$IPTABLES -L | $GREP drop-and-log-it" ]; then
    $IPTABLES -F drop-and-log-it
fi
#
# Delete all User-specified chains
$IPTABLES -X
#
# Reset all IPTABLES counters
$IPTABLES -Z

#Configuring specific CHAINS for later use in the ruleset
#
# NOTE: Without the --log-level set to "info", every single
# firewall hit will goto ALL vtys. This is a very big
# pain.
#
echo " Creating a DROP chain.."
$IPTABLES -N drop-and-log-it
$IPTABLES -A drop-and-log-it -j LOG --log-level info
$IPTABLES -A drop-and-log-it -j DROP

echo -e "\n - Loading INPUT rulesets"

#####
# INPUT: Incoming traffic from various interfaces. All rulesets are
# already flushed and set to a default policy of DROP.
#

# loopback interfaces are valid.
#
$IPTABLES -A INPUT -i lo -s $UNIVERSE -d $UNIVERSE -j ACCEPT

# local interface, local machines, going anywhere is valid
#
#$IPTABLES -A INPUT -i $INTIF -s $INTNET -d $UNIVERSE -j ACCEPT

# remote interface, claiming to be local machines, IP spoofing, get lost
#
$IPTABLES -A INPUT -i $EXTIF -s $INTNET -d $UNIVERSE -j drop-and-log-it

# external interface, from any source, for ICMP traffic is valid
#
# If you would like your machine to "ping" from the Internet,
# enable this next line
#
#$IPTABLES -A INPUT -i $EXTIF -p ICMP -s $UNIVERSE -d $EXTIP -j ACCEPT

# remote interface, any source, going to permanent PPP address is valid
#
#$IPTABLES -A INPUT -i $EXTIF -s $UNIVERSE -d $EXTIP -j ACCEPT

# Allow any related traffic coming back to the MASQ server in
#
$IPTABLES -A INPUT -i $EXTIF -s $UNIVERSE -d $EXTIP -m state --state \
ESTABLISHED,RELATED -j ACCEPT

#Maintenance Port SSH
$IPTABLES -A INPUT -i $INTIF -s $SERVICE -d $INTIP -j ACCEPT

```

```

# DHCPd
#
$IPTABLES -A INPUT -i $INTIF -p tcp --sport 68 --dport 67 -j ACCEPT
$IPTABLES -A INPUT -i $INTIF -p udp --sport 68 --dport 67 -j ACCEPT

# HTTPd - Enable the following lines if you run an EXTERNAL WWW server
#
#echo -e "          - Allowing EXTERNAL access to the WWW server"
#$IPTABLES -A INPUT -i $EXTIF -m state --state NEW,ESTABLISHED,RELATED \
#-p tcp -s $UNIVERSE -d $EXTIP --dport 80 -j ACCEPT

# Catch all rule, all other incoming is denied and logged.
#
$IPTABLES -A INPUT -s $UNIVERSE -d $UNIVERSE -j drop-and-log-it

echo -e "          - Loading OUTPUT rulesets"

#####
# OUTPUT: Outgoing traffic from various interfaces. All rulesets are
#         already flushed and set to a default policy of DROP.
#

# loopback interface is valid.
#
$IPTABLES -A OUTPUT -o lo -s $UNIVERSE -d $UNIVERSE -j ACCEPT

# local interfaces, any source going to local net is valid
#
$IPTABLES -A OUTPUT -o $INTIF -s $EXTIP -d $INTNET -j ACCEPT

# local interface, any source going to local net is valid
#
$IPTABLES -A OUTPUT -o $INTIF -s $INTIP -d $INTNET -j ACCEPT

# outgoing to local net on remote interface, stuffed routing, deny
#
$IPTABLES -A OUTPUT -o $EXTIF -s $UNIVERSE -d $INTNET -j drop-and-log-it

# anything else outgoing on remote interface is valid
#
$IPTABLES -A OUTPUT -o $EXTIF -s $EXTIP -d $UNIVERSE -j ACCEPT

# DHCPd
#
$IPTABLES -A OUTPUT -o $INTIF -p tcp -s $INTIP --sport 67 \
-d 255.255.255.255 --dport 68 -j ACCEPT

```

```

$IPTABLES -A OUTPUT -o $INTIF -p udp -s $INTIP --sport 67 \
-d 255.255.255.255 --dport 68 -j ACCEPT

# Catch all rule, all other outgoing is denied and logged.
#
$IPTABLES -A OUTPUT -s $UNIVERSE -d $UNIVERSE -j drop-and-log-it

echo -e "          - Loading FORWARD rulesets"

#####
# FORWARD: Enable Forwarding and thus IPMASQ
#

echo "          - FWD: Allow only existing/related connections in"
$IPTABLES -A FORWARD -i $EXTIF -o $INTIF -m state --state ESTABLISHED,RELATED \
-j ACCEPT
#$IPTABLES -A FORWARD -i $INTIF -o $EXTIF -j ACCEPT

# some addresses are open:
# Swisscom DNS
$IPTABLES -A FORWARD -i $INTIF -s $INTNET -d $SCDNS1 -j ACCEPT
$IPTABLES -A FORWARD -i $INTIF -s $INTNET -d $SCDNS2 -j ACCEPT

# vpn-cluster.ethz.ch
$IPTABLES -A FORWARD -i $INTIF -s $INTNET -d $VPN1 -j ACCEPT
$IPTABLES -A FORWARD -i $INTIF -s $INTNET -d $VPN2 -j ACCEPT
$IPTABLES -A FORWARD -i $INTIF -s $INTNET -d $VPN3 -j ACCEPT
$IPTABLES -A FORWARD -i $INTIF -s $INTNET -d $VPN4 -j ACCEPT

# n.ethz.ch
$IPTABLES -A FORWARD -i $INTIF -s $INTNET -d $NETHZ -j ACCEPT

# Catch all rule, all other forwarding is denied and logged.
#
$IPTABLES -A FORWARD -j drop-and-log-it

echo "          - NAT: Enabling SNAT (MASQUERADE) functionality on $EXTIF"
#
#More liberal form
$IPTABLES -t nat -A POSTROUTING -o $EXTIF -j MASQUERADE
#
#Stricter form
#$IPTABLES -t nat -A POSTROUTING -o $EXTIF -j SNAT --to $EXTIP

#####
echo -e "\nStronger rc.firewall-2.4 $FWVER done.\n"

```

C.1.2 System Concept II

```
#!/bin/sh
#
# rc.firewall-2.4-stronger
FWVER=0.81s

# Log:
# 0.81s - Adapted for Set Top Box
# 0.80s - Addapted for mAP
# 0.73s - Added comments in the output section that DHCPd is optional
# and changed the default settings to disabled
# 0.72s - Changed the filter from the INTNET to the INTIP to be
# stateful; moved the command VARs to the top and made the
# rest of the script to use them
# 0.70s - Added a disabled examples for allowing internal DHCP
# and external WWW access to the server
# 0.63s - Added support for the IRC module
# 0.62s - Initial version based upon the basic 2.4.x rc.firewall

echo -e "\nLoading rc.firewall - version $FWVER.\n"

# The location of various iptables and other shell programs
IPTABLES=/sbin/iptables

LSMOD=/sbin/lsmmod
INSMOD=/sbin/inssmod
GREP=/bin/grep
AWK=/usr/bin/awk
SED=/bin/sed
IFCONFIG=/sbin/ifconfig

#Setting the EXTERNAL and INTERNAL interfaces for the network
EXTIF="ppp0"
INTIF="netcs0"
echo " External Interface: $EXTIF"
echo " Internal Interface: $INTIF"
echo " ---"

# Determine the external IP
EXTIP="$(IFCONFIG $EXTIF | $GREP 'inet addr' | $AWK '{print $2}' | \
$SED -e 's/.*://' )"
echo " External IP: $EXTIP"
echo " ---"

# Assign the internal TCP/IP network and IP address
INTNET="172.30.199.0/24"
INTIP="172.30.199.1"
echo " Internal Network: $INTNET"
echo " Internal IP: $INTIP"

echo " ---"

# Setting a few other local variables
#
UNIVERSE="0.0.0.0/0"

#ETH DNS
DNS="129.132.250.220"

#Enable IP forwarding
echo " Enabling forwarding.."
echo "1" > /proc/sys/net/ipv4/ip_forward

echo " Enabling DynamicAddr.."
echo "1" > /proc/sys/net/ipv4/ip_dynaddr

$IPTABLES -P INPUT DROP
$IPTABLES -F INPUT
$IPTABLES -P OUTPUT DROP
$IPTABLES -F OUTPUT
$IPTABLES -P FORWARD DROP
$IPTABLES -F FORWARD

if [ -n "$IPTABLES -L | $GREP drop-and-log-it" ]; then
$IPTABLES -F drop-and-log-it
fi

$IPTABLES -X
$IPTABLES -Z

echo " Creating a DROP chain.."
$IPTABLES -N drop-and-log-it
$IPTABLES -A drop-and-log-it -j LOG --log-level info
$IPTABLES -A drop-and-log-it -j DROP

echo -e "\n - Loading INPUT rulesets"

# loopback interfaces are valid.
$IPTABLES -A INPUT -i lo -s $UNIVERSE -d $UNIVERSE -j ACCEPT

# local interface, local machines, going anywhere is valid
#$IPTABLES -A INPUT -i $INTIF -s $INTNET -d $UNIVERSE -j ACCEPT

# remote interface, claiming to be local machines, IP spoofing, get lost
$IPTABLES -A INPUT -i $EXTIF -s $INTNET -d $UNIVERSE -j drop-and-log-it

# remote interface, any source, going to permanent PPP address is valid
#$IPTABLES -A INPUT -i $EXTIF -s $UNIVERSE -d $EXTIP -j ACCEPT

#SSH
$IPTABLES -A INPUT -i $INTIF -p tcp --dport 22 -j ACCEPT
$IPTABLES -A INPUT -i $INTIF -p tcp --syn -j drop-and-log-it
```

```
# DHCPd
$IPTABLES -A INPUT -i $INTIF -p tcp --sport 68 --dport 67 -j ACCEPT
$IPTABLES -A INPUT -i $INTIF -p udp --sport 68 --dport 67 -j ACCEPT

# Catch all rule, all other incoming is denied and logged.
#$IPTABLES -A INPUT -s $UNIVERSE -d $UNIVERSE -j drop-and-log-it

echo -e " - Loading OUTPUT rulesets"

# loopback interface is valid.
$IPTABLES -A OUTPUT -o lo -s $UNIVERSE -d $UNIVERSE -j ACCEPT

# local interfaces, any source going to local net is valid
$IPTABLES -A OUTPUT -o $INTIF -s $EXTIP -d $INTNET -j ACCEPT

# local interface, any source going to local net is valid
$IPTABLES -A OUTPUT -o $INTIF -s $INTIP -d $INTNET -j ACCEPT

# outgoing to local net on remote interface, stuffed routing, deny
#$IPTABLES -A OUTPUT -o $EXTIF -s $UNIVERSE -d $INTNET -j drop-and-log-it

# anything else outgoing on remote interface is valid
$IPTABLES -A OUTPUT -o $EXTIF -s $EXTIP -d $UNIVERSE -j ACCEPT

# DHCPd
$IPTABLES -A OUTPUT -o $INTIF -p tcp -s $INTIP --sport 67 \
-d 255.255.255.255 --dport 68 -j ACCEPT
$IPTABLES -A OUTPUT -o $INTIF -p udp -s $INTIP --sport 67 \
-d 255.255.255.255 --dport 68 -j ACCEPT

# Catch all rule, all other outgoing is denied and logged.
$IPTABLES -A OUTPUT -s $UNIVERSE -d $UNIVERSE -j drop-and-log-it

#echo -e " - Loading FORWARD rulesets"

#echo " - FWD: Allow only existing/related connections in"
$IPTABLES -A FORWARD -i $EXTIF -o $INTIF -m state --state ESTABLISHED,RELATED \
-j ACCEPT
$IPTABLES -A FORWARD -i $INTIF -o $EXTIF -j ACCEPT

# some addresses are open:
# DNS
$IPTABLES -A FORWARD -i $INTIF -s $INTNET -d $DNS -j ACCEPT

# Catch all rule, all other forwarding is denied and logged.
$IPTABLES -A FORWARD -j drop-and-log-it

#####
echo -e "\nStronger rc.firewall-2.4-mAP $FWVER done.\n"
```

C.2 Automation Script

```
#!/bin/bash
# AP.sh

LOGDIR=/usr/MAPI/logs/AP.log
RPPPO=/proc/sys/net/ipv4/conf/ppp0/rp_filter
RPIPSECO=/proc/sys/net/ipv4/conf/ipsec0/rp_filter
IPSECRETS=/etc/ipsec.secrets

#mkdir /mnt/logs

log () {
echo $1 >> $LOGDIR
}

#-----
exit_prog () {
#umount /dev/hdc1 /mnt/logs
echo "Stopping AP starting routine..."
exit
}

#-----
check_cardmgr ()
{
if test -z "$(pidof cardmgr)"
then
echo "Cardmgr not running."
echo "Starting Cardmgr..."
log "$(date) : Starting Cardmgr..."
cardmgr
sleep 5
else
echo "Cardmgr running ..."
fi
}

#-----
check_slot (){
SLOTMSG=$(cardctl status $1 | awk '{print $1}')
ISWLAN=$(cardctl status $1|grep 5V)
ISGPRS=$(cardctl status $1|grep 3.3V)
if (test "$SLOTMSG" = no); then
echo "Slot $1: No card."
echo "Please insert card!"
log "$(date) : No card in slot $1. Bye!"
exit_prog
elif (test -n "$ISWLAN"); then
echo "Slot $1: WLAN"

elif (test -n "$ISGPRS"); then
```

```
echo "Slot $1: GPRS"
fi
}

#-----

check_gprs (){
echo "checking GPRS connection..."

if (test -z "$(ifconfig ppp0)"); then
echo "GPRS connection is not up!"
echo "Setting GPRS connection up..."
minicom -S /usr/MAPI/modem

MINICOMLOG=$(tail -n 1 /root/minicom.log | awk '{print $4}')
echo $MINICOMLOG
if (test "$MINICOMLOG" = Modem); then
log "Modem not accessible!"
kill $(pidof cardmgr)
if (test -e /var/run/cardmgr); then
rm /var/run/cardmgr.pid
fi
if (test -e /var/lib/pcmcia/stab); then
rm /var/lib/pcmcia/stab
fi
NM=$((NM+1))
return
elif (test "$MINICOMLOG" = Carrier); then
log "No Carrier!"
NC=$((NC+1))
return
elif (test "$MINICOMLOG" = connection); then
while (test -z "$(ifconfig|grep ppp0)"); do
sleep 2
i=$((i+1))
if (test $i -ge 15); then
log "Could not set up PPP connection"
NP=$((NP+1))
return
fi
done
return
fi
}

#-----

check_WLAN () {
#Config in /etc/network/interfaces

if (test -n "$(ifconfig netcs0)"); then
echo "WLAN interface is up"
```

```

else
    ifconfig netcs0 up
    iwconfig netcs0 essid demo mode master
    echo "Setting AP up..."
fi
}

#-----

set_configs () {
echo "0" > $RPPPP0
echo "0" > $RPPIPSECO
ip route add 129.132.254.93 via 11.0.0.1 dev ppp0
}

#-----

set_secret () {
echo "Adapting IPsec secrets..."
PtPIP='ifconfig | grep P-t-P | awk -F: '{print $2}' | awk -FP '{print $1}''

if (test -z "$PtPIP"); then
    echo "No ppp0!"
else
    echo "$PtPIP 129.132.254.93: PSK \"ethz-pix\" > $IPSECRETS
#echo "IPsec secret written"
fi

}

#-----

setup_tunnel () {
ipsec setup restart
sleep 2
ipsec auto --add mAP
ipsec auto --up mAP
}

#-----

#MAIN

#mount /dev/hdc2 /mnt/logs

if (test -z "'ipsec look|grep 0.0.0.0/1|grep ipsec0'"); then
log ""
log "*****"
log "'date' : Starting mAP..."
fi

NC=0
NM=0
NP=0

```

```

while (test -z "'ifconfig ppp0'"); do

    #echo "$NC"
    #echo "$NM"
    #echo "$NP"

    if (test $NC -ge 3); then
log "Setting up GPRS connection failed. Bye!"
exit_prog
    elif (test $NM -ge 3); then
log "Setting up GPRS connection failed. Bye!"
# shutdown -r now
exit_prog
    elif (test $NP -ge 3); then
log "Starting PPP daemon failed. Bye!"
# shutdown -r now
exit_prog
    fi

    check_cardmgr
    check_slot 1
    check_gprs
done

echo "PPP connection is up"

check_slot 0
check_WLAN

if (test -z "'ipsec look|grep 0.0.0.0/1|grep ipsec0'"); then
echo "Setting up IPsec tunnel..."
ip route add default via 11.0.0.1 dev ppp0
ip route change default via 11.0.0.1 dev ppp0
ipsec setup restart
set_configs
set_secret
setup_tunnel
log "'date' : Tunnel established"

else
echo "Tunnel is up"
fi

```

C.3 The GPRS Connection

C.3.1 Minicom Script

```
# Script to reset GPRS modem and enter pin
```

```
set i 0
set j 0
send "atz"
expect {
"OK" goto phase1
timeout 10 goto failed1
}
```

```
phase1:
```

```
print \nPhase1
inc j
if j > 5 goto failed1
send "at+cfun=0"
expect {
"OK" goto phase2
timeout 5 goto failed1
}
```

```
phase2:
print \nPhase2
send "at+cfun=1"
expect {
"OK" goto phase3
timeout 5 goto failed1
}
```

```
phase3:
sleep 4
print \nPhase3
send "at+cpin=\"XXXX\""
sleep 2
expect {
"OK" goto phase4
"ERROR" goto phase1
timeout 5 goto phase1
}
```

```
phase4:
print \nPhase4
inc i
if i > 10 goto failed2
send "at+cgreg?"
sleep 2
expect {
"0,1" goto dial
"0,0" goto phase4
```

```
}
```

```
dial:
! pon sunrise &
log PPP connection up
! echo "PPP connection up"
! killall -9 minicom
exit
```

```
failed1:
log No Modem
print No Modem
! killall -9 minicom
exit
```

```
failed2:
print No Carrier
log No Carrier
! killall -9 minicom
```

C.3.2 PPP Configuration

```
#/etc/ppp/pap-secrets
gprs * gprs
sunrise * gprs

#/etc/ppp/options
asynmap 0
noauth
crtcts
lock
show-password
noipdefault
-mn
+pap
proxyarp
lcp-echo-interval 30
lcp-echo-failure 4
noipx
```

C.3.3 Sunrise Configuration

Sunrise:

```
#/etc/ppp/peers/sunrise
noauth
crtcts
user sunrise
connect '/usr/sbin/chat -v -f /etc/ppp/chatdata.sunrise'
/dev/ttyS1
115200
noipdefault
defaultroute
bsdcomp 0,0
debug

#/etc/ppp/chatdata.sunrise
'ABORT' 'BUSY'
'ABORT' 'ERROR'
'ABORT' 'NO CARRIER'
'ABORT' 'NO DIALTONE'
'ABORT' 'Invalid Login'
'ABORT' 'Login incorrect'
'' 'ATZ'
'OK' 'AT+CGDCONT=1'
'OK' 'AT+CGDCONT=1,"IP","internet"'
'OK' 'ATD*99***1#'
'CONNECT'
```

C.3.4 Swisscom Configuration

```
#/etc/ppp/peers/swisscom
noauth
crtcts
user gprs
connect '/usr/sbin/chat -v -f /etc/ppp/chatdata.swisscom'
/dev/ttyS1
115200
noipdefault
defaultroute
bsdcomp 0,0
debug

#/etc/ppp/chatdata.swisscom
'ABORT' 'BUSY'
'ABORT' 'ERROR'
'ABORT' 'NO CARRIER'
'ABORT' 'NO DIALTONE'
'ABORT' 'Invalid Login'
'ABORT' 'Login incorrect'
'' 'ATZ'
'OK' 'AT+CGDCONT=1'
'OK' 'AT+CGDCONT=1,"IP","gprs.swisscom.ch"'
'OK' 'ATD*99***1#'
'CONNECT'
```

C.4 Router Configuration

```
!  
version 12.2  
service timestamps debug uptime  
service timestamps log uptime  
no service password-encryption  
!  
hostname rou-dock-test  
!  
logging buffered 4096 informational  
aaa new-model  
!  
ip subnet-zero  
!  
ip domain-name ethz.ch  
!  
!  
crypto isakmp policy 30  
  encr 3des  
  hash md5  
  authentication pre-share  
  group 2  
crypto isakmp key xxxxxx address 0.0.0.0 0.0.0.0  
!  
crypto ipsec transform-set trans-bus esp-3des esp-md5-hmac  
!  
crypto dynamic-map dyn-pix 20  
  set transform-set trans-bus  
  set pfs group2  
  match address 112  
!  
!  
crypto map map-pix local-address Loopback0  
crypto map map-pix 11 ipsec-isakmp dynamic dyn-pix  
!  
!  
interface Loopback0  
  ip address 129.132.254.93 255.255.255.255  
!  
interface FastEthernet0/0  
  description megatransit  
  ip address 129.132.99.121 255.255.255.192  
  ip policy route-map to_valid  
  duplex auto  
  speed auto  
  crypto map map-pix  
!  
router ospf 70  
  log-adjacency-changes  
  redistribute static metric 1000 metric-type 1 subnets route-map static-ospf  
  network 129.132.0.0 0.0.255.255 area 0.0.0.0  
!  
ip classless
```

```
ip route 172.30.199.0 255.255.255.0 FastEthernet0/0  
!  
ip access-list extended to_valid  
deny ip any 129.132.99.160 0.0.0.15  
deny ip any host 129.132.98.12  
deny ip any host 129.132.250.220  
deny ip any host 129.132.250.2  
deny ip any host 129.132.1.56  
deny ip any host 129.132.98.15  
deny ip any host 172.20.2.102  
deny tcp any host 129.132.178.197 eq www  
deny tcp any host 129.132.178.197 eq 443  
deny tcp any host 129.132.97.10 eq www  
deny tcp any host 129.132.97.10 eq 443  
deny tcp any host 129.132.200.207 eq www  
deny tcp any host 129.132.200.207 eq 443  
permit ip 172.30.199.0 0.0.0.255 any  
!  
access-list 11 permit 172.30.199.0  
!  
access-list 112 permit ip any 172.30.199.0 0.0.0.255  
!  
route-map to_valid permit 10  
  match ip address to_valid  
  set ip next-hop 129.132.99.66  
!  
route-map static-ospf permit 10  
  match ip address 11  
!  
!  
!  
!  
line con 0  
line aux 0  
line vty 0 4  
  autocommand access-enable host  
line vty 5 8  
  exec-timeout 0 0  
  rotary 1  
!  
!  
end  
  
rou-dock-test#
```

C.5 IPsec Configuration

```
# /etc/ipsec.conf - FreeS/WAN IPsec configuration file
config setup
    interfaces=%defaultroute
    klipsdebug=none
    plutodebug=none
    plutoload=%search
    plutostart=%search

conn mAP
    auto=add
    type=tunnel
    left=%defaultroute
    leftsubnet=172.30.199.0/24
    right=129.132.254.93
    rightsubnet=0.0.0.0/0
    rightnexthop=
    auth=esp
    esp=3des-md5-96
    authby=secret
    pfs=yes
    keyexchange=ike

# /etc/ipsec.secrets - FreeS/WAN IPsec secrets file
10.xxx.yyy.zzz 129.132.254.93: PSK "ethz-pix"
```

Bibliography

- [1] IEEE Std 802.11, The Institute of Electrical and Electronics Engineers Inc, 1997.
- [2] ETH World, <http://www.ethworld.ethz.ch>, 2002.
- [3] IEEE 802.11b Wireless LANs, Wireless Freedom at Ethernet Speeds, 3Com Corporation, 2000.
- [4] Introduction to Wireless LANs, WLANA Inc, 1999.
- [5] Wireless LAN Konzept ETH,
http://www.wireless.ethz.ch/files/ETHZ-WLAN-Konzept_2002.pdf,
April 2002.
- [6] Christian Bettstetter, Hans-Jörg Vögel, and Jürg Eberspächer. GSM Phase 2+ General Packet Radio Service GPRS: Architecture, Protocols and Air Interface, IEEE Communications Survey vol.2 no.3, 1999.
- [7] Alan Sicher and Randall Heaton. GPRS Technology Overview, White Paper, Dell, 2002.
- [8] Icomera Train Gateway v1.1,
<http://www.icomera.com/downloads/pdf/IcomeraTrainGateway-11.pdf>
- [9] Inge Gronboek. Cellular and Mobile IP: Overview and Enhancements, Project Paper, 1999.
- [10] Comet Group. CellularIP, Columbia University N.Y., 2002.
- [11] Scott Bronson. PPP-SSH MiniHOWTO,
<http://www.linux.org/docs/ldp/howto/mini/ppp-ssh/index.html>, 2002.

- [12] Christos Xenakis, Evangelos Gazis, and Lazaros Merakos. Secure Deployment in GPRS Mobile Networks, Proceedings, 2002.
- [13] S. Kent and R. Atkinson. IP Authentication Header, RFC 2402, Nov 1998.
- [14] S. Kent and R. Atkinson. IP Encapsulating Security Payload, RFC 2406, Nov 1998.
- [15] Bernard Adoba. IPsec-NAT Compatibility Requirements, Internet Draft, draft-ietf-ipsec-nat-reqts-02.txt, August 2002.
- [16] A. Huttunen. UDP Encapsulation of IPsec Packets, Internet Draft, draft-ietf-ipsec-udp-encaps-03.txt, June 2002.