

Middleware for adaptive network connectivity

ROLAND WALTERSSON



**KTH Information and
Communication Technology**

Master of Science Thesis
Stockholm, Sweden 2006

COS/CCS 2006-10

Middleware for adaptive network connectivity

Roland Waltersson

Master's Thesis

Department of Communication Systems

Royal Institute of Technology

Stockholm, Sweden



Abstract

As the coverage of modern wireless technologies expands, today's mobile phones and PDAs often have a range of heterogeneous networks to which they may connect. It would benefit mobile applications to use the network which best corresponds to its need. However, making the handovers between networks transparent to allow communication sessions to survive is not trivial as the TCP/IP suite, used by most networking applications today, was initially not designed with mobility in mind.

A Vinnova founded research project at Saab (together with associates¹) has found that it could prove advantageous to monitor network quality together with the application's needs and make intelligent decisions regarding what network to use. An algorithm for network classification and evaluation has been proposed.

This thesis examined prerequisites and methods for implementing adaptive network connectivity through transparent handovers for mobile devices, resulting in a tentative model to demonstrate the mentioned research results. The prototype, implemented as a user space middleware, utilizes UDP encapsulation and a per-packet basis link evaluation, resulting in small overhead and fast context adaptation. Link monitoring ensures that server and clients are constantly updated about network quality and availability.

The prototype yielded transparent handovers between networks, with short handover delays, at the cost of reduced performance for fast networks.

¹ Blekinge Institute of Technology, Swedish National Testing and Research Institute, and Swedish Road Administration

Sammanfattning

Täckningen av trådlösa nätverk ökar konstant, och dagens mobiltelefoner och handdatorer har ofta ett antal olika nätverk de kan ansluta till. Det skulle vara fördelaktigt för mobila applikationer att använda det nätverk som bäst passar dess behov. Att göra övergångarna mellan dessa nätverk transparenta så att pågående kommunikationssessioner kan fortgå är dock inte helt trivialt, då TCP/IP protokollen från början inte var tänkta för mobila enheter.

Ett av Vinnova finansierat forskningsprojekt utfört på Saab med flera¹ har undersökt möjligheterna att övervaka kvaliteten på olika trådlösa nätverk samt kraven från applikationerna, och med detta som grund utföra intelligenta beslut om vilket nätverk som ska utnyttjas. En algoritm för att utföra dessa beslut har också föreslagits.

Detta examensarbete undersökte möjligheterna för att implementera adaptiv nätverksroaming genom transparenta övergångar för mobila enheter, och resulterade i en demonstrator, som även kan visa resultaten från den nämnda forskningen. Prototypen, implementerad som en "middleware", använder UDP tunnling och en per-paket nätverksutvärdering, vilket resulterade i liten overhead och snabb anpassning till nya kommunikationssituationer. En länkövervakare såg till att server och klienter alltid var uppdaterade om kvaliteten och tillgängligheten av olika nätverk.

Prototypen gav totalt transparenta övergångar mellan nätverk med relativt korta fördröjningar, med nackdelen av viss reducerad prestanda för snabba nätverk.

¹ Blekinges Tekniska Högskola, Sveriges Provnings- och Forskningsinstitut, samt Vägverket

Preface

I would like to thank Gerald Maguire, my examiner and supervisor at the Royal Institute of Technology, for his support and encouragement throughout this thesis. Many were the times he pointed me in the right direction.

At Saab Communication I owe Peter Lindberg, my supervisor, gratitude for his interest in my work and his guidance. Furthermore I would like to thank Lars Leigard and the rest of the people at the unit who had to endure my questions.

I would also like to thank the people over at BTH; Stefan Chevul, Markus Fiedler, and Lennart Isaksson, for their feedback on this report and valuable advice.

Finally, I hope you will enjoy reading this report as much as I did writing it.

Roland Waltersson, Växjö, Sweden, April 2006

Table of contents

1	Introduction	1
1.1	Background	1
1.2	Objectives	1
1.3	Methodology	2
1.4	Thesis outline	2
1.4.1	<i>Part 1: Background</i>	2
1.4.2	<i>Part 2: Research</i>	2
1.4.3	<i>Part 3: Design, implementation and evaluation</i>	3
2	Background	4
2.1	The PIITSA project	4
2.1.1	<i>Purpose and goal</i>	4
2.1.2	<i>ITS services</i>	5
2.1.3	<i>Application-perceived throughput</i>	5
2.1.4	<i>Roaming strategy</i>	6
2.1.5	<i>This thesis part of PIITSA</i>	6
2.2	GSM/GPRS	7
2.2.1	<i>GSM technology</i>	7
2.2.2	<i>GPRS technology</i>	8
2.3	UMTS	9
2.3.1	<i>Network architecture</i>	10
2.3.2	<i>Radio interface</i>	11
2.3.3	<i>UMTS in Sweden</i>	11
2.4	Digital Audio Broadcasting	11
2.4.1	<i>Sending IP packets with DAB</i>	12
2.4.2	<i>DAB in Sweden</i>	12
2.5	WLAN	12
2.5.1	<i>WLAN architecture</i>	13
2.5.2	<i>WLAN standards</i>	14
2.6	Mobile IP	14
2.6.1	<i>Basic concepts</i>	15
2.6.2	<i>Registration</i>	15
2.6.3	<i>Care of address and co-located care of address</i>	16
2.6.4	<i>IP tunnelling</i>	16
2.6.5	<i>Triangular routing and reverse tunnelling</i>	16
2.6.6	<i>Mobile IP shortcomings</i>	17
3	Intelligent Transport Systems and Services	18
3.1	Introduction	18
3.2	Areas of ITS	19
3.2.1	<i>Electronic payment</i>	19
3.2.2	<i>Traveller Information</i>	19
3.2.3	<i>Crash prevention</i>	20
3.2.4	<i>Incident Management</i>	20
3.2.5	<i>Emergency Management</i>	20
3.2.6	<i>Transit Management</i>	21
3.2.7	<i>Collision Avoidance Systems and Collision Notification Systems</i> ..	21
3.3	Traveller Information in Sweden	21
3.3.1	<i>Traveller Information sources</i>	21

3.3.2	<i>Information broadcasting example: RDS-TMC</i>	22
3.3.3	<i>Future ITS strategy</i>	23
4	Operating system and programming language	24
4.1	Important attributes	24
4.1.1	<i>Application Programming Interface</i>	24
4.1.2	<i>Supply of devices</i>	24
4.1.3	<i>Future prospect</i>	24
4.2	Operating systems	24
4.2.1	<i>Symbian OS with Java</i>	24
4.2.2	<i>Windows Mobile 2003/5.0 with .NET CF</i>	26
4.3	Conclusion	27
5	Hardware	29
5.1	Important attributes	29
5.2	Devices.....	29
5.2.1	<i>QTek 9000</i>	29
5.2.2	<i>HP iPAQ HW6515 Mobile Messenger</i>	30
5.2.3	<i>Sony Ericsson P990</i>	30
5.2.4	<i>Nokia N91</i>	31
5.3	Conclusion	31
6	Wireless network roaming solutions	32
6.1	Columbitech Wireless VPN™	32
6.1.1	<i>Architecture</i>	32
6.1.2	<i>Mobility</i>	33
6.1.3	<i>Security</i>	34
6.1.4	<i>Roaming</i>	34
6.2	Birdstep Intelligent Mobile IP Client.....	34
6.2.1	<i>Architecture</i>	35
6.2.2	<i>Mobility</i>	35
6.2.3	<i>Security</i>	36
6.2.4	<i>Roaming</i>	36
6.3	Resilient Mobile Sockets	36
6.3.1	<i>Architecture</i>	36
7	Design alternatives and considerations	38
7.1	Requirements	38
7.2	Mobility.....	39
7.2.1	<i>Seamless roaming through third-party Mobile IP</i>	39
7.2.2	<i>Seamless roaming through a virtual interface</i>	41
7.3	Link monitoring	43
7.4	Security	44
8	Implementation	45
8.1	Platform and programming language.....	45
8.2	Overview	45
8.3	Mobility.....	46
8.3.1	<i>The TAP device</i>	46
8.3.2	<i>UDP tunnelling</i>	47
8.3.3	<i>Example of operation</i>	47
8.4	NSB Controller	48
8.4.1	<i>Sending and receiving data</i>	49

8.4.2	<i>Executing handovers</i>	49
8.5	The Translator and address allocation	49
8.5.1	<i>Address allocation</i>	50
8.5.2	<i>Translation</i>	50
8.6	Link monitoring	50
8.6.1	<i>Monitoring through control messages</i>	51
8.6.2	<i>WMI monitoring</i>	51
8.7	Control messages	51
8.8	Network selection	52
8.9	Security	53
9	Evaluation	54
9.1	The test-bed.....	54
9.2	Results.....	55
9.2.1	<i>File transfer performance</i>	55
9.2.2	<i>Handover delays</i>	56
9.2.3	<i>Results from NETSTAT control messages</i>	58
9.2.4	<i>Memory consumption</i>	59
9.2.5	<i>WMI performance</i>	59
9.3	Discussion	59
9.4	Future work.....	60
9.4.1	<i>Improvements</i>	60
9.4.2	<i>Porting</i>	61
10	References	62
11	Abbreviations and acronyms	66

List of figures

Figure 1: The GSM network architecture	8
Figure 2: GPRS and GSM architecture combined	9
Figure 3: UMTS network architecture. Note the similarities to the GSM/GPRS architecture.....	11
Figure 4: WLAN architecture	14
Figure 5: Mobile IP traffic example.....	15
Figure 6: IP tunnelling in Mobile IP	16
Figure 7: Variable speed limits is a typical ITS application	18
Figure 8: This map from the website trafiken.nu shows an example of a Traveller Information service [45].....	20
Figure 9: Display from a device running the Symbian OS from Symbian Ltd. [31].	25
Figure 10: The Sun Java API structure	26
Figure 11: Windows Mobile 2005	27
Figure 12: The application connections are terminated locally and redirected over a single TCP connection via the VPN Server.	33
Figure 13: RMS architecture. The internal socket is a normal UDP socket	37
Figure 14: The NSB - User Application relationship	38
Figure 15: The third-party MIP HA between client and server illustrated with triangular routing.....	40
Figure 16: Insertion of UDP header (b) on IP-in-IP protocol (a) to avoid NAT problem	41
Figure 17: Session (application) based mobility. Dual sockets are used, one internal and one external. The external connection (which could be TCP or UDP) is allowed to go down or change.....	43
Figure 18: Overview of the NSB's blocks.....	45
Figure 19: The TAP device catches all client and server application traffic and forwards it to user space, where NSB listens.....	47
Figure 20: A TCP packet encapsulated in an UDP packet (Network – and Transport layer protocols displayed).....	47
Figure 21: The UDP packet travelling over the Internet.....	48
Figure 22: The TCP packet is unchanged even though the packet is sent on another network.....	48
Figure 23: The NSB seen as a multiplexer.	49
Figure 24: The server translation table.	50
Figure 25: The experimental test-bed.	54

List of tables

Table 1: Smartphones and PDAs of interest today.	31
Table 2: The NSB control messages.	52
Table 3: Average NSB upload (UL) and download (DL) speeds compared to speeds without NSB.	55
Table 4: Handover latency at client side during upload, using a backup network.	56
Table 5: Handover latency at server side during download, using a backup network.	57
Table 6: Handover latency at client side during upload, using no backup network.	57
Table 7: Handover latency at server side during download, using no backup network.	57
Table 8: Average statistics as reported by NETSTAT (measured on 50 messages) during general Internet browsing.	58
Table 9: Average statistics as reported by NETSTAT (measured on 50 messages) during file downloads.	58

1 Introduction

This chapter describes the background, objectives, methods, and outline of the thesis.

1.1 Background

Saab, Blekinge Institute of Technology (BTH), Swedish National Testing and Research Institute, and the Swedish Road Administration (Vägverket) are jointly conducting the project “Personal Information in Intelligent Transport Systems through Seamless communication and Autonomous decision”, or PIITSA for short. The project aims to define functions for decision making and seamless communication to facilitate the implementation of Intelligent Transportation System and Services (ITS) applications, that is, transport – and traffic related applications that utilize modern information technology.

One goal of the PIITSA project is to define a module for handling seamless communication through vertical handovers between heterogeneous commercial networks such as DAB, GPRS/UMTS, and WLAN, to be used for ITS applications on mobile devices. Today in such applications the traffic flow towards the user is typically greater than the traffic flow from the user, and there is a need for a (mobile) asymmetric communication network. Previous studies within the PIITSA project have shown that DAB may be used as a one-way, broadcast channel to the users with good data transmission rates, while GPRS/UMTS or WLAN could be used for bidirectional communication. To intelligently switch between these technologies, a Network Selection Box (NSB) was defined [19].

The NSB is supposed to handle the handovers between networks, and is transparent to the user. Moreover, when more than one network is available, applications should have the possibility of influencing the choice of network by stating for example what kind of service it uses. The NSB monitors QoS parameters constantly and makes intelligent decisions about the appropriate network to use. For this, a tentative roaming strategy was defined [19].

In fall 2005, the conceptual ideas underlying the NSB and its decision making policy was maturing. The final phase of the project is to demonstrate the NSB generally and specifically the roaming strategy at work. The objective of this master thesis is to examine how such a demonstrator could be designed and evaluated, aiming at a scalable design which could later evolve into a full-fledged application.

1.2 Objectives

The objectives of this master thesis can be summarized as follows:

- Analyse today’s market to decide what platform, developing tool, and hardware that could be used to implement an application as such mentioned above.
- Conduct a background study of related projects and the ITS area generally.

- Design a tentative platform for the NSB, with a focus on an operational demonstrator for PIITSA.
- Implement and evaluate the demonstrator. The demonstrator should be implemented in such a way that it may be used as a platform for future extensions, possibly even a full application.

1.3 Methodology

The first objective of the thesis was to decide on what platform the example application should be implemented upon. As the information technology market evolves incessantly, the main source of information proved to be the Internet and magazine articles, combined with interviews with manufacturers. The Internet is sometimes an unreliable media, and as the sources often were subjective they were carefully cross-checked against other sources. The information gathered at this stage was qualitative, secondary data, meaning written data not aimed directly at this study.

Quickly, the research was narrowed down to a number of platforms judged to be interesting for this thesis. The parameters of the hardware and software were then compared separately, and their importance for the final decision evaluated.

Another part of this thesis consisted of the design of a demonstration application. In order to make rational decisions regarding the structure of the application, a study of similar projects were conducted. The source of information was once again mainly articles and whitepapers found on the Internet, and to some degree studies of open source code.

The design and implementation of the tentative NSB model consisted of practical programming in C# and C, using the results gained from prior studies.

Lastly, the solution was evaluated, with regard to among other things efficiency and handover delays. Quantitative data was gathered through a network monitor and packet analyzer software. This primary data was compiled and evaluated with regard to the parameters of interest for the final evaluation.

1.4 Thesis outline

This thesis is divided into three parts, as follows.

1.4.1 Part 1: Background

The first part of this thesis, presented in chapters 2 and 3, contains relevant background information about the wireless technologies proposed to be used in the solution, i.e. WLAN, DAB, GPRS, and UMTS, and a closer look at ITS with a focus on Swedish advancements in this area. An overview of Mobile IP is also presented.

1.4.2 Part 2: Research

The second part, chapters 4, 5, and 6, contains the results obtained from the analysis conducted *prior* to the design phase. The first two of these chapters takes the form of a market research where two different operating systems and four mobile devices are compared. Chapter 6 examines two commercial solutions and one research model available today for network roaming at the application layer.

1.4.3 Part 3: Design, implementation and evaluation

The third part of the thesis is presented in chapter 7, 8, and 9. Chapter seven briefly describes two substantially different ways in which the NSB could be designed. Chapter 8 offers a more detailed account of the actual implementation, where one of the mentioned designs was used. In Chapter 9, the NSB implementation is evaluated.

2 Background

This chapter contains general background information about the PIITSA project, and the wireless technologies: GPRS, UMTS, WLAN, and DAB. It also discusses Mobile IP technology, which is a technology that aims to make IP addresses independent of the current network attachment point. Intelligent Transport Systems and Services (ITS) is discussed in Chapter 3.

2.1 The PIITSA project

Saab, Blekinge Institute of Technology, Swedish National Testing and Research Institute, and the Swedish Road Administration (Vägverket) together are conducting the project PIITSA, Personal Information in Intelligent Transportation Systems Through Seamless communication and Autonomous decisions.

PIITSA is a three year R&D project, aiming at defining functions for communication and communications policies used by ITS applications for mobile users. The project started in January 2004, and ends in 2006 with a conceptual demonstration of the results obtained. The project is mainly founded by Vinnova, and Saab has the role of project leader. The following chapter is based on the project outline in [19].

2.1.1 Purpose and goal

The ITS area is evolving rapidly today. The Internet is providing people with information about traffic conditions, public transportation, trip booking et cetera. Such information is particularly useful when delivered to the mobile user, and to achieve this, wireless communication must be utilized.

Sweden has a well developed wireless infrastructure, and generally provides good coverage with multiple wireless technologies. However, Sweden is a sparsely populated country, and remote locations may lack modern wireless technologies.

The project's purpose is to define the required functions in a mobile user's terminal to deal with the changing of communication possibilities experienced when the terminal changes position. The term Network Selection Box (NSB), is used to describe the function that handles the network selection and handovers when conditions changes or when the user or application changes its communication preferences. These handovers should be transparent to the end-user.

The objectives of PIITSA are:

- Define a mechanism to handle seamless communication between an information source and a mobile user. Asymmetric communication may be utilized. Commercial networks like DAB, GPRS, UMTS, and WLAN should be used.
- Define the interface between the NSB and the user application.
- Demonstrate the NSB during 2006.

To achieve these goals, the tasks include:

- Analysis of the possibilities to use IP over DAB,

- Performance evaluation of these networks,
- Investigation of how mobility support for IP could be implemented,
- Development of a “roaming-strategy” which chooses the best available network according to service demands and measured link qualities,
- and examination of possible user applications, such as a travel planner, traffic information, etc.

It should be mentioned that there is some controversy over whether DAB is of any interest at all, considering the government’s recent decision to stop further development. However, other technologies like Digital Video Broadcast (DVB) still look promising and are in many respects similar to DAB, and therefore the author of this thesis has chosen to include a discussion about the DAB technology and DAB measurements.

The following chapters describe some results obtained within the PIITSA project.

2.1.2 ITS services

Before the measurements and the work on a roaming strategy could start, it had to be considered what sort of communication requirements an ITS applications typically introduces. Obviously, identifying such requirements only becomes interesting when there are multiple networks available; in some cases, the user will only have GSM coverage (or no network coverage at all), and then the decision is easy to make.

When there are multiple networks available, the NSB must choose the one that best meets the application’s needs. These needs depend on what kind of service the application currently uses. Five generic services were defined [21]:

1. Public Streaming Service (PSS): A broadcast service dedicated to the public. Users can subscribe to a channel to receive information.
2. Individual Messaging Service (IMS): A unicast message from the user to the server, e.g. reporting of an accident. Generally small data amounts.
3. Backwards Streaming Service (BSS): A kind of IMS, but with information being sent on a regular basis.
4. Selective Streaming Service (SSS): A multicast service to a group of users. Information is sent to these users without being requested for. May be a form of location-based PSS.
5. Personal Interactive Service (PIS): A service initiated by the user, who then receives dedicated information, e.g. requesting a time table or requesting a recommended route through a town.

2.1.3 Application-perceived throughput

One objective of PIITSA is to examine the performance of the available commercial networks. An important attribute to examine is the application-perceived throughput. Throughput is often measured at data link layer. A long delay may be disturbing for the user, and if long delays occur frequently, the user

may lose interest in the service. In this case it is important to look at the throughput at the application layer, which includes the processing and unwrapping time.

On a DAB link, the throughput was found to be 120 Kbps when using a 128 Kbps link with a packet size of 540 bytes, provided that the receiver did not move. In a GPRS uplink, packets were received at 15 to 20 Kbps on a stationary host. Via the GPRS downlink, a huge variation was found; throughputs between 2 to 38 Kbps were measured. On a UMTS uplink, a 66 Kbps throughput was observed (with a maximum of 135 Kbps) while in the downlink scenario, the throughput reached 360 Kbps (with a maximum of 384 Kbps) [39].

2.1.4 Roaming strategy

As an activity within PIITSA, researchers at Blekinge Institute of Technology examined how to define a roaming strategy for seamless network switching, where the main parameters are cost, type of service, and application-perceived throughputs [13].

The study found that in order to choose the best network one has to look at what service will be used. These Five services were defined in the previous section.

Earlier measurements of DAB, GPRS, UMTS, and WLAN were analysed with the following parameters in mind:

- Initial Delays (ID): The setup time for a connection. The uplink process for GPRS was found to have the shortest ID.
- Link Capacity (LC): The throughput of the link. WLAN inarguably provides the highest throughput, even though multiple WLAN units may create an interference problem if improperly configured.
- Directional Losses (DL): The packet loss. The UMTS and GPRS downlinks have high packet loss, while their uplink loss rate is very low.

Having defined the parameters, they were compared using the AHP algorithm (Analytic Hierarchy Process). AHP is a theoretical and mathematical model for decision making, and its full definition is out of the scope for this thesis. The basic idea is that you compare two parameters at a time, rate their relative importance to each other, and finally obtain a full matrix with relative rankings.

The research concluded that further pre-processing and classifying of QoS data has to be done before any decision can be done. One method for pre-processing and classifying is to use Fuzzy Set modelling.

The research also highlighted the problem that some network adapters already implement their own roaming strategy, which we would be unable to control. It is therefore important to examine the devices before choosing, for example, a combined GPRS/UMTS modem.

2.1.5 This thesis part of PIITSA

My thesis work started when the project was in its final stage. Many of the tasks mentioned in Section 2.1.1 have already been accomplished, however the design and implementation of a demonstrator needed to be done before the end of 2006.

To achieve this, the results obtained from earlier studies within PIITSA, open source software from third parties, and emulation of functions that will take a long time to develop by myself will be used.

2.2 GSM/GPRS

General Packet Radio Service (GPRS), is a service extending the circuit-switched Global System for Mobile Communication (GSM) technology by introducing packet based communication to GSM. The purpose was to improve wireless data transmission rates for mobile users. The technology is sometimes described as 2.5G, somewhere in between the second generation (2G) and the third generation (3G) telecommunication systems. While GSM offers a maximum data rate of 9.6 Kbps, GPRS offers a theoretical maximum data rate of 171 Kbps. As GPRS is a GSM service, the GSM technology is first briefly explained.

2.2.1 GSM technology

GSM was established in 1982 as a European standard for telecommunication. Today, the technology is deployed all over the world. Four companies operate the GSM networks in Sweden, namely TeliaSonera, Tele2/Comviq, Telenor, and Vodafone (formerly Europolitan) [27].

Initially, GSM was dedicated primarily to voice traffic, which is characterized by moderate data volume rates in a continuous flow. Therefore, initially GSM was constructed as a circuit-switched network, with a predetermined set of resources allocated for each connection.

2.2.1.1 The radio link

The GSM radio network is divided into cells, which consists of a base station antenna with a certain coverage area. The coverage areas are smaller in urban areas, so that traffic volumes is distributed over multiple cells. When a mobile terminal moves out of coverage from one cell to another, a handover should occur.

The mobile equipment (the phone) communicates with the base station using either the 900 MHz or 1800 MHz bands in Europe. Each band uses two sub bands to achieve duplex communication, each 25 MHz wide. Resource sharing is reached through a combination of Time – and Frequency Division Multiple Access (TDMA/FDMA). The carrier frequencies are separated by 200 KHz, giving 124 channels are that multiplexed together using FDMA. However, one base station only uses some of these channels, to avoid interference with adjacent base stations. Furthermore, each of these channels is divided into 8 timeslots using TDMA [40], [8].

2.2.1.2 Network architecture

The GSM network is composed of three entities, the Mobile Node (MN), the Base Station Subsystem (BSS), and the Network Subsystem, see Figure 1 below.

The Mobile Node consists of the phone which contains a Subscriber Identity Module (SIM) card. The SIM, which is implemented using a small smart card, contains an identification number for the subscriber and a secret key for authentication. This allows the user to change terminal and still be able to make and receive calls, provided of course that the SIM card is inserted into the new terminal. The SIM card also contains some storage space for names & numbers and Short Message Service (SMS) messages.

The BSS is composed of the Base Transceiver Station (BTS) and a Base Station Controller (BSC). The former houses the radio equipment that communicates with the MN. The latter is a communication hub connected to a number of BTS, and handles, among other things, call routing and handovers between a set of transceivers.

The Network Subsystem's central component is the Mobile services Switching Center (MSC). The MSC interconnects all the operators BSCs, and handles call routing, location updating, handovers between BSCs, etc. The BSCs are connected to the MSC, typically via a wireless link. The MSC contains two databases; the Home Location Register and the Visitor Location Register. The Visitor Location Register keeps track of all phones currently under the coverage of the BTSs. The Home Location Register keeps track of where phones that have their home at this MSC, currently are. They are both used to route call sessions between the BSCs. Multiple MSCs may be connected through leased links, PSTN, or ISDN [40].

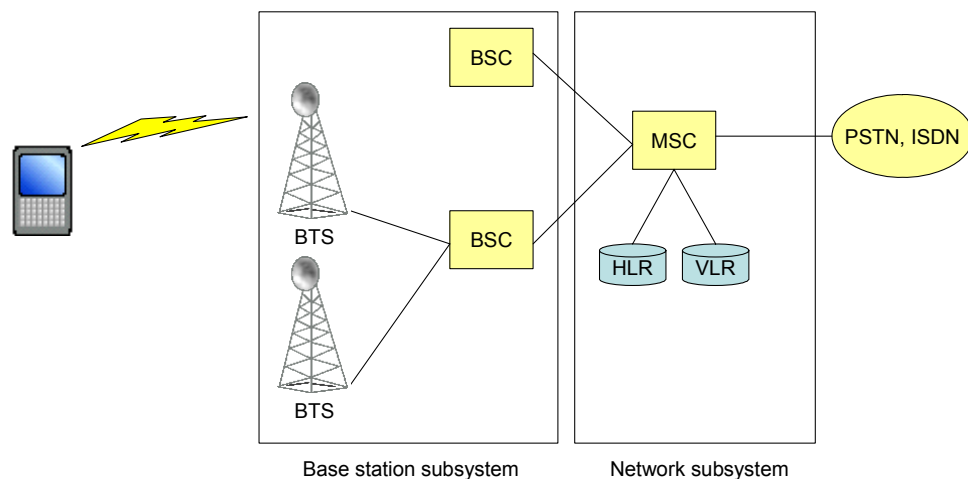


Figure 1: The GSM network architecture

2.2.2 GPRS technology

Data transmission is often characterized as being “bursty”, that is, high data volumes that need to be transmitted quickly at short, irregular intervals. As GSM was designed as a circuit switched network, it is not optimal for bursty traffic. GPRS offers a way to establish and maintain a logical connection without using bandwidth, while high data rates may be reached when needed by allocating a variable amount of bandwidth.

2.2.2.1 Packet switching

By utilizing packet switching, resources are only used when needed. The packets are sent independently over the network, and, if needed, reassembled when they reach their destination. As mentioned, the GSM radio link uses TDMA. GPRS uses as many of these time slots as it needs, up to eight in theory. Generally the operator wants to give voice traffic preference, so GPRS can often use only 3-4 timeslots, which leads to a practical data rate of about 40 Kbps. Also, the devices themselves may limit the maximum number of time slots used for GPRS [9].

As GPRS is packet switched (and only consumes bandwidth during actual data transmission) it can act as if being “always connected”, and is sometimes referred to as Mobile Internet. Every phone could in theory have an own IP address, just like a computer. When using GPRS, the user pays for the amount of data sent and received, rather than the time connected (as during calls). This further supports the illusion of being “always connected”.

2.2.2.2 GPRS network

The GPRS network extends the GSM networks, and the relation is illustrated in Figure 2.

The Gateway GPRS Support Node (GGSN) provides a gateway between the GPRS network and other data networks, most notably Internet. The GGSN is responsible for authentication, location management, and traffic volume measurement for subscriber billing. It encapsulates outgoing GPRS packets to other protocols, like IP or X.25, and vice versa. The GGSN also keeps track of where to send incoming packets using the Home Location Register and Visitor Location Register.

The Serving GPRS Support Node (SGSN) acts much like the MSC in GSM networks, and controls session management and GPRS mobility management such as handovers. To route packets, it uses a Home Location Register and a Visitor Location Register, just like the GSM network. To conclude, the SGSN and GGSN are essentially routers, forwarding traffic to/from the MN [2].

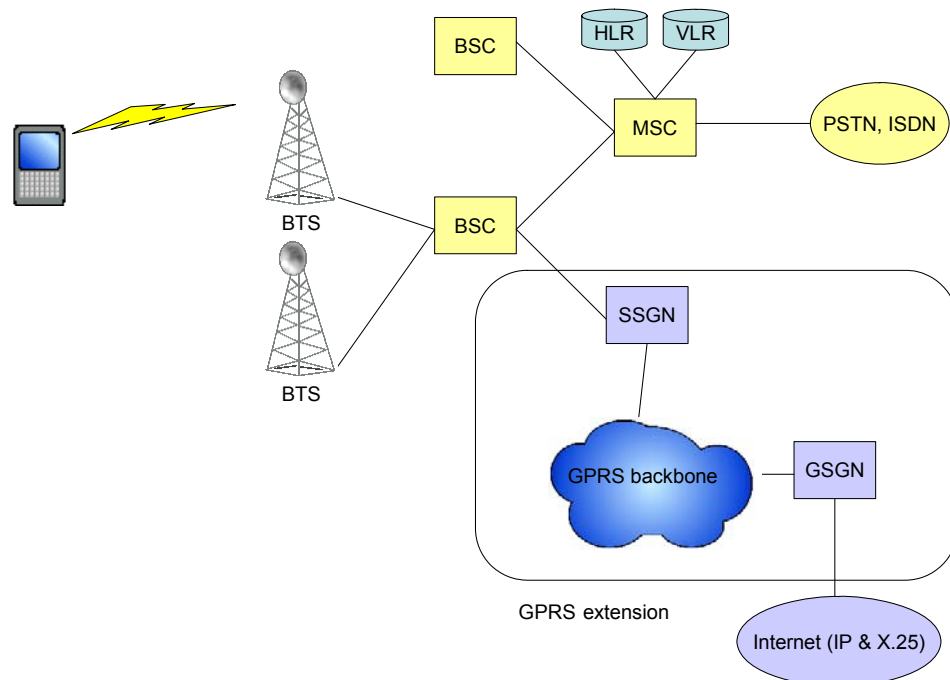


Figure 2: GPRS and GSM architecture combined

2.3 UMTS

Universal Mobile Telecommunications Service, UMTS, is a standard for third generation (3G) cellular telecommunication and offers services like speech and

information transfer services at much higher rates than GSM/GPRS. The offered data rates are:

- 144 Kbps for rural outdoor use
- 384 Kbps for pedestrian or urban outdoor use
- 1920 Kbps for indoor or fixed-environment use

UMTS networks services have different QoS classes for four types of traffic:

- Conversational class (voice, video, telephony, video gaming)
- Streaming class (multimedia, video on demand, web cast)
- Interactive class (web browsing, network gaming, database access)
- Background class (email, SMS, downloading)

For example, the background class demands error free transmission, but has almost no restrictions regarding delay. For conversational services, low delay and low delay variation is demanded [49].

2.3.1 Network architecture

UMTS is composed of three domains; the Core Network (CN), the UMTS Terrestrial Radio Access Network (UTRAN), and the User Equipment (UE), see Figure 3.

The task of the CN is to provide switching, routing, and transit for traffic. The architecture is based on the GSM/GPRS architecture, where packet data transfers and calls are treated in different ways. The GGSN and SGSN works in the same way as for GPRS, and the MSC works just like its GSM equivalent.

The UTRAN domain provides a new air interface access. The antenna is now called a Node-B, and its controller is called the Radio Network Controller (RNC). It roughly corresponds to the GSM parts, BTS and BSC. The Node-B is responsible for air interface transmission, WCDMA coding, and error handling. The RNC controls the handovers, segmentation/reassembly, and channel allocation, among other things.

The UE, meaning the phone, contains a USIM card, similar to the GSM SIM card. It includes security functions, user authentication, support for one or more user profiles, etc. Most important, it contains the subscriber's identity number, just as the SIM card did. The phone can operate in either circuit-switched or packet-switched mode, or both at the same time [49].

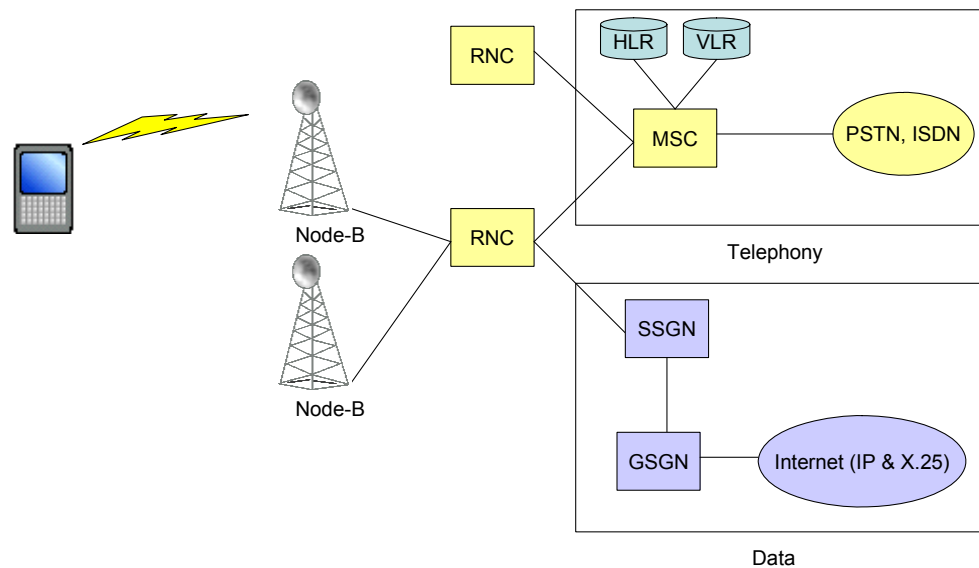


Figure 3: UMTS network architecture. Note the similarities to the GSM/GPRS architecture.

2.3.2 Radio interface

UTRAN uses Wideband CDMA, WCDMA, as its air interface. It employs a spread spectrum transmission, together with Frequency Division Duplex. The uplink and downlink channels are spaced apart by 5 MHz. The uplink uses frequencies from 1920 MHz to 1980 MHz, while the downlink uses frequencies from 2110 MHz to 2170 MHz. To address a specific user, the transmission is spread over a wide spectrum by multiplying the data with a certain spreading code. The codes are orthogonal to avoid interference with each other. The code is then dispersed at the receiver with the appropriate spreading code, to extract the data [37].

2.3.3 UMTS in Sweden

The Swedish 3G licenses were distributed in a beauty contest by Post – och Telestyrelsen (PTS) in 2000 to Tele2, Europolitan (today Vodafone), Hi3G (3) and Orange. UMTS now covers about 85 % of the population, which means Sweden has the best 3G coverage in Europe. However, if one looks at the land area covered by 3G, Sweden does not rank that high [34].

2.4 Digital Audio Broadcasting

Digital Audio Broadcasting, DAB, was developed by the Eureka 147 project, an EU initiative started in 1988. The technology is a major advancement of the FM radio. It offers improved sound quality and targeted data and information services [60].

Digital radio is transmitted in blocks of frequencies called channels. One channel is able to carry stereo and mono radio channels as well as services such as text and data. One channel can transfer six stereo programs at 192 Kbps.

The mode used to transmit DAB is COFDM, Coded Orthogonal Frequency Division Multiplex, which is a technique that among other things aims at reducing noise. The data is carried by many sub carriers at a low rate. At low transmission rates, multipath distortion becomes less significant – as the problem shifts to inter-symbol interference. Interference between the modulated carriers is prevented by

employing orthogonality, as in UMTS. In short, a set of signals is orthogonal if the spacing of the carrier frequencies is the reciprocal of the symbol duration. The individual carriers are modulated using differential QPSK [3].

In order to reduce the errors further, an error-correcting algorithm is used. To make errors random, the transmitted data is spread out across all the carriers and interleaved in time. Also, the signal is coded using convolutional coding [3].

2.4.1 Sending IP packets with DAB

If DAB should be used in PIITSA, it is imperative that IP packets can be carried effectively by DAB. The problem has been examined earlier, and it was found to be possible to use IP and any internet transport protocol over DAB. This is achieved by encapsulating the IP datagram in an MSC_DAB (the Main Service Channel, which is the portion of the DAB packet that carries data) on DAB transport level. Furthermore, it was found that DAB could be used as a unidirectional channel to the mobile user with pretty good throughput, even though some problems occurred with TCP packets due to long delays [27].

2.4.2 DAB in Sweden

Broadcasting with DAB started in 1995 in Sweden. Originally, DAB was supposed to cover 85 % of the population of Sweden. The attitude among potential customers towards DAB turned out to be quite luke warm, however, and the government decided it could not be justified economically to reach the original goal. Today, only Stockholm, Göteborg, Malmö, and Luleå have DAB coverage, which corresponds to about 35 % of the population. Only 7000 people listen to digital radio each week [43].

State-owned Sveriges Radio is the only provider of digital stations, and they currently run 6 stations. Teracom AB is responsible for building and maintaining the network. They too are state-owned [44].

In the UK, DAB now covers 80 % of the population, and is quite a popular media. Other countries, like Canada, Ireland, Finland, and Germany have stopped broadcasting DAB due to the lack of consumer interest. In Sweden, the DAB expansion has cost 400 million SEK so far, which means that DAB has cost about 57 000 SEK per active user [11].

The Swedish government recently decided not to spend any more funding on extending or developing DAB. In effect, this means the DAB project has failed.

2.5 WLAN

A wireless LAN (WLAN) often referred to as WiFi, is a wireless data transmission system designed to provide network access or communication between computing devices by using radio waves or light rather than a cable infrastructure. WLANs are typically found in public areas such as airports and train stations, but also in office buildings and in homes that wants to share an Internet connection.

The IEEE 802.11 was ratified as a standard for wireless LANs in 1997. IEEE 802.11 operates at the two lowest layers in the OSI model, the physical layer and the media access and control sub layer (belonging to the link layer). This means that any transport protocol, such as TCP or UDP, and IP at the network layer, will

run on 802.11 networks. Obviously, the greatest benefit of WLAN is that it provides data communication to a device while not being restricted by a physical cable. Another benefit is the low cost to set up a wireless network, due to cheap equipment and the fact that no cables have to be installed [48].

Many WLAN standards use the license-free ISM bands at 2.4 or 5 GHz, which offers a 83 MHz spectrum for any kind of wireless communication. Thus they may face unwanted interference. Spread spectrum or OFDM technology are used, as they provide a reliable way to send information [48].

2.5.1 WLAN architecture

The smallest WLAN is when two mobile stations communicate directly. When connected in this way the WLAN is referred to as an ad-hoc network.

A number of mobile stations may also be connected to an Access Point (AP), often found in for example cafés, airports, offices, homes, etc. When connected in this way, we have a Basic Service Set (BSS).

The AP may then be connected further via an ordinary wired LAN to other APs, to the Internet, or any other wide-area network. Such a complex network is called an Extended Service Set (ESS), see Figure 4. The 802.11 and the related IEEE 802.2 standard guarantees that a WLAN can connect to an outside wired network. Further, 802.11 provides a set of functions to support mobility. A typical coverage radius for an outdoor AP is 150 to 300 meters.

If a user moves from the coverage of one AP to another, and the APs are connected in an ESS, a BSS-transition occurs. If a user moves between two APs not connected in an ESS, an ESS-transition occurs. The Distribution System Services defined in 802.11 handles the association and reassociation processes, which registers and deregisters the mobile station with a new AP. This often means that when a mobile station moves between one subnetwork to another then the mobile station's IP address will change. The 802.11 standard does not define how seamless mobility should be supported (although it defines tools for it, via the association and reassociation processes). A common approach is to use Mobile IP, explained in Section 2.6.

The Station Services, also defined by 802.11, provides authentication and privacy. The WLAN can be configured to accept all hosts (open authentication) or only hosts possessing a certain key (shared key authentication). In this case, each user needs a special key to be able to log on the network. The key is used together with the Wired Equivalent Privacy (WEP) algorithm. For privacy, WEP also encrypts the traffic between mobile stations and the AP. For the system to be secure, the key must be updated quite often. Unfortunately, WEP has proved to provide inadequate security, and many WLANs today use additional software encryption, or WPA (WiFi Protected Access), a security system vastly superior to WEP [48].

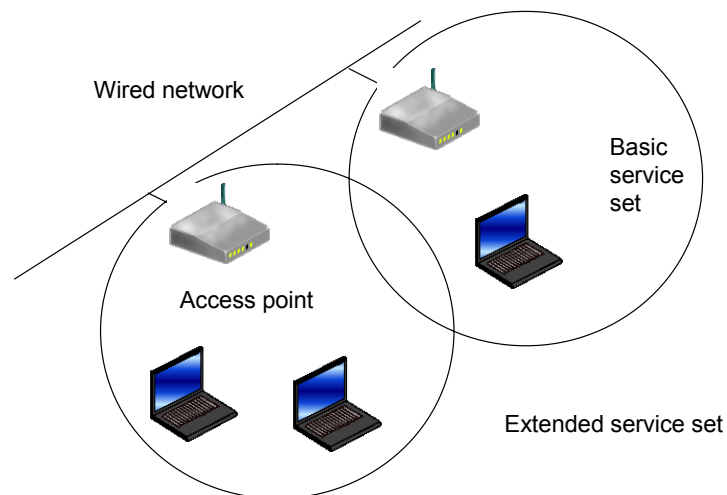


Figure 4: WLAN architecture

2.5.2 WLAN standards

Many WLAN standards have been proposed and used, but the most widely used today are 802.11b, and 802.11g.

2.5.2.1 802.11b

The b-standard, which actually was introduced before the a-standard supports speeds of up to 11 Mbps. It communicates through DSSS (Direct Sequence Spread Spectrum, which spreads the data through bit coding) at the physical layer. One disadvantage of 802.11b is that the frequency band is crowded, and subject to interference from other networking technologies, microwave ovens, 2.4GHz cordless phones (a huge market), and Bluetooth. Another problem is the lack of available channels, which may cause interference with adjacent WLAN networks.

2.5.2.2 802.11g

This standard also operates in the 2.4 GHz band, to provide compability with the 802.11b standard. Its modulation technique is compatible with the 802.11b, which support data rates of 11 Mbps. However, 802.11g also offers data rates at 54 Mbps. 802.11b and 802.11g also have in common that both only provide three interference-free channels, and neither are very scalable.

2.6 Mobile IP

Mobile IP, as defined by IETF [32], is a proposed standard to solve the problems associated with the fact that a mobile node, be it a laptop, PDA, or mobile phone, will change its IP address whenever it changes its network attachment point. The purpose of Mobile IP is to make this change transparent to network applications, so as not to interrupt for example an ongoing file transfer or a streaming service.

Most applications will loose their communication session if the IP address changes. An exception is HTTP, used for web browsing, which may start a new TCP session for each page request.

2.6.1 Basic concepts

Mobile IP consists of three or four entities; the Home Agent (HA), the Mobile Node (MN), the Correspondent Node (CN), and (sometimes) a Foreign Agent (FA). The CN is another computer with whom the MN communicates. The MN is some sort of mobile terminal, for example a laptop. The HA is situated in the home network where the MN is thought to reside most of its time. If MN enters another network, it is assigned a care of address (CoA), which is the IP address assigned to the MN in the new network. As soon as this new address is acquired, the MN registers it with the HA, so that the HA is updated about MN's actual (CoA) IP address. This means that the MN has two IP addresses, one home address and one foreign address.

Now imagine the CN, for example a server, wants to send data to the MN. It only knows about the MN's home address, so the data is sent to the HA. When the packet arrives, the HA looks up the last registered CoA address for the MN, then it forwards the packet using IP tunnelling (see below). The address is either the address of a FA, or, if no FA is present in the network, the MN's address on that network. In the latter case the address is called a co-located care of address. The FA (or MN directly) sends packets back to the CN, either directly or through the HA, as explained later.

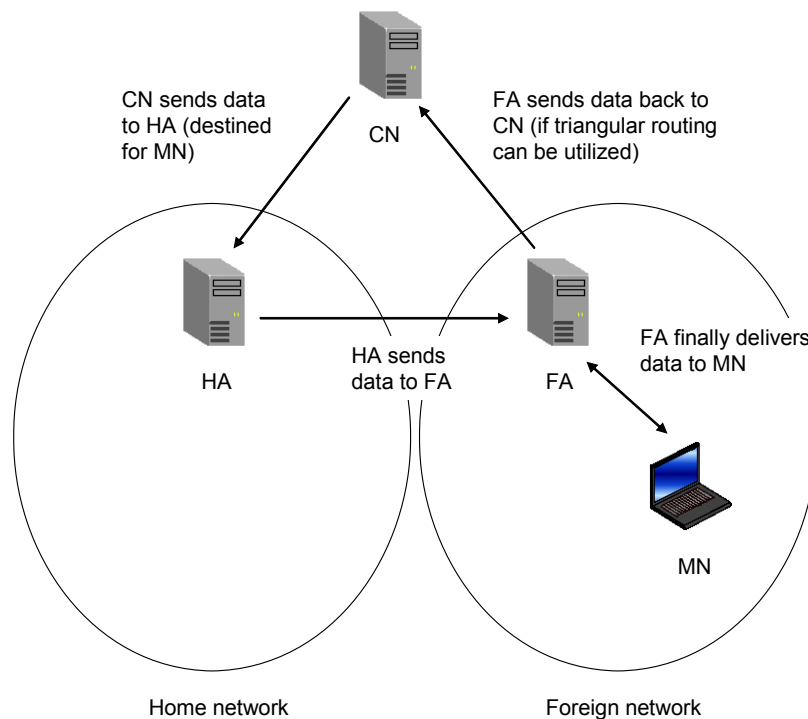


Figure 5: Mobile IP traffic example

2.6.2 Registration

The MN discovers its new IP address by listening for periodically transmitted ICMP router advertisements from the HA/FA. The message contains information about available IP addresses, if a FA or HA is available and some other information. The MN can also explicitly ask for this information through a router solicitation message.

The new IP address is then registered with its HA through a registration request. The HA confirms when it has updated its routing tables.

2.6.3 Care of address and co-located care of address

Figure 5 shows the data flow when a FA is used. Often, there is no FA available as it would require the service provider to have hardware installed at many different networks. The MN itself is then responsible for unpacking and handling the tunneled packets received from its HA, and to register the new IP addresses at the HA. In this case, the care of address is said to be co-located, that is pointing directly at the MN. The concept is just the same as if a FA was available, and placed physically at the MN.

2.6.4 IP tunnelling

IP tunnelling is the process where an IP packet from the CN is encapsulated into another IP packet containing the actual IP address of the MN, creating a so-called IP-within-IP packet, see Figure 6. The flow can be described as follows:

1. The CN sends data destined for a MN.
2. The packet is encapsulated within another IP packet, where the FA address is set as the destination and HA's address as source address.
3. Upon reception at the MN (or FA), the original IP header is kept, while the outer one is dropped.

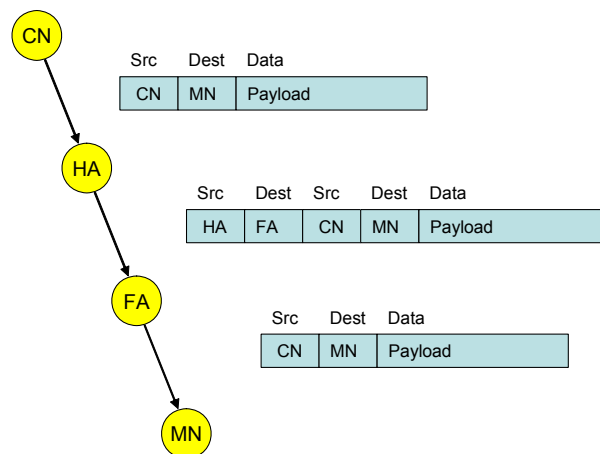


Figure 6: IP tunnelling in Mobile IP

2.6.5 Triangular routing and reverse tunnelling

When MN sends data back to the CN, it may spoof the source address of the IP packet to its home address. This step is essential as the CN should send packets to the MN's home address, not to the MN directly.

However, some networks use *ingress filtering*, a security measure which drops packets where the source address do not match the actual source computer. To solve this problem, packets may first be sent back to the HA and then to the CN. Thus we no longer have the triangular routing case (as in Figure 5), but instead routing via *reverse tunnelling*.

2.6.6 Mobile IP shortcomings

Mobile IP is characterized by considerable delays during handoffs, especially when the MN moves frequently. This may cause TCP to trigger a timeout, and thus an application may not be able to continue a connection – if not designed for such eventualities.

Another problem that generates long delays is the fact that reverse tunnelling is often used with Mobile IP. This means that even if the CN is just one hop away, the packets from MN must be routed back to its HA, and then to the CN.

Large delays are especially unacceptable with streaming services, such as video or VoIP. Solutions like Cellular IP and Mobile IPv6 has been proposed, but they are not yet common on the market.

3 Intelligent Transport Systems and Services

This chapter introduces Intelligent Transport Systems and Services (ITS), and more closely examines some areas that are of interest for the PIITSA project. The intention is to familiarize the reader with the problems involved with ITS, so that the need for reliable wireless communication can be understood.

3.1 Introduction

ITS is the collective name for a broad range of solutions and concepts aiming to improve transportation in any way. Or, as ITS Canada puts it:

“ITS: The application of advanced and emerging technologies (computers, sensors, control, communications, and electronic devices) in transportation to save lives, time, money, energy and the environment.” [15]

The definition is quite comprehensive, and different countries and authorities interpret it in slightly different ways. However the main thought is that to battle congestion and safety problems, the roads must be used more efficiently instead of just expanding the road infrastructure. In order to achieve this, the private sector have to cooperate with authorities, and a clear set of standards for building road infrastructure must be defined [14], [16].

Lately, much research has been conducted on ITS in universities and companies worldwide. ITS applications have also been implemented to some extent. A good example in Sweden is the variable speed limits that are currently being evaluated. In this case, speed limits are controlled by road conditions and detection of congestion. Vägverket has defined a national strategy for ITS applications between 2006 and 2009 [52]. This is related to the fact that Sweden is going to host the ITS World Congress in Stockholm 2009.



Figure 7: Variable speed limits is a typical ITS application

Although much money is spent on research and implementation of ITS solutions, there remain some problems. One problem is that the benefits of ITS is not yet commonly recognized. Another is the degree of co-operation between companies and authorities. The prerequisites for such co-operation vary greatly between different nations. A third problem is the privacy of the individual. Many ITS applications require a network of cameras monitoring the infrastructure, identification of vehicles, trackers in vehicles, etc. This is violating personal integrity according to some people [14].

ITS applications are generally divided into two main branches; intelligent infrastructure and intelligent vehicles. These branches are divided further, as explained below.

3.2 Areas of ITS

Intelligent Infrastructure is the name for all ITS applications not directly associated with the vehicle. The US Department of Transportation has divided Intelligent Infrastructure into the following areas: Electronic Payment, Traveller Information, Crash Prevention, Incident Management, Emergency Management, Transit Management, and others. Areas that sort under Intelligent Vehicles include: Collision Avoidance Systems, Collision Notification Systems and Driver Assistance Systems. Some of these ITS areas are further explained below [16], [50].

3.2.1 Electronic payment

To pay for maintenance, many highways in Europe today have tolls. Stockholm introduced congestion pricing on a trial basis in the beginning of 2006. To avoid congestion at toll plazas, electronic toll collection is sometimes used. Systems may consist of a vehicle-mounted transponder so that readers can register vehicle's passing a zone border. The owner of the vehicle is then charged with the cost of passing into that zone. This is a typical example of ITS applications that are already in wide use.

Another example of electronic payment is the ability for a traveller to book and pay for a parking spot in advance. This could be done through a PDA or a stationary computer, perhaps via a website. These systems are not yet widely in use commercially.

3.2.2 Traveller Information

This is a wide area covering both pre-trip and en-route information to the traveller. The information could include: road conditions, incidents and congestions on the planned route, navigation, disturbances in the public transportation system, etc. In the US, many states provide a 511 number that travellers can call to get current information about road conditions and mass transit. In Sweden, Stockholm's Lokaltrafik (SL) provides trip planning and disturbance reports through their website, while Eniro provides navigation support on their site. En-route traffic information has already existed a long time in the form of FM radio broadcasts targeted to a particular area.

The challenge is to merge all such separate systems into a single system, supporting multimodal travelling both inside cities and in rural areas. As the traveller wants updated information while on the move, such a system would benefit from being available on a PDA or a mobile phone. As this subject closely relates to PIITSA, it will be discussed in detail later.



Figure 8: This map from the website trafikenu shows an example of a Traveller Information service [47].

3.2.3 Crash prevention

Crash prevention is the collective name for all systems that detects road and roadside hazards and warns nearby vehicles via for example a mobile terminal. These systems typically utilize some sort of wireless communication.

For example, if a driver approaches a sharp curve at high speed, he could be advised to lower his speed. Infrared cameras could detect animals and warn the vehicle's driver. Roadside lightning or in-vehicle signals could make the driver aware that a vehicle is approaching an intersection, or that a train will soon pass through a rail crossing.

3.2.4 Incident Management

To minimize the time it takes for rescue services to reach the scene of an accident, an Incident Management system is already under development in many countries.

The first step is surveillance of the road to detect the incidents. The system may consist of roadside cameras (still or full-motion) and acoustic roadway detectors.

To locate the vehicles involved in the incident, an Automated Vehicle Location (AVL) system is used. The basic idea is that all vehicles are equipped with a tracker, so that they can be located electronically.

Dynamic Message Signs are another ITS application already in use (mounted for example on the E4/E20 just south of Stockholm to inform about roadway conditions). These signs could be used to spread information about a nearby accident. The information could also be sent directly to the computers of approaching vehicles.

3.2.5 Emergency Management

This area is related to Incident Management. For example, another way of detecting accidents is for the vehicles to deploy Automated Collision Notification systems (ACN) through sensors and wireless communication. Telemedicine

systems provide a link between responding ambulances and emergency medical facilities, enabling doctors to advise emergency medical personnel regarding treatment of patients en-route to the hospital.

All kinds of sensors and systems that warn for large-scale emergencies including natural disasters (hurricanes, earthquakes, floods, winter storms, etc.) and man-made disasters (nuclear power plant accidents, terrorism, bombing, etc.) are called Early Warning Systems and falls under Emergency Management.

3.2.6 Transit Management

Transit Management includes all ITS solutions that relates to mass transportation. One example is in-vehicle and facility surveillance in the form of cameras and microphones (implemented in the subway and recently in night-service buses in Stockholm), which produces images and audio that may be sent to a transit management centre. Another security-related example is Remote Disabling Systems – that can disable a vehicle remotely.

Fleet Management is a sub area utilizing for example AVL to position a mass transit company's units, and dispatch information about delays to travellers. Such information could also be used for planning of future routes or timetable changes.

Systems that allow automatic collection and reporting of vehicle maintenance information are an element of Fleet Management as well. The information could be uploaded at the end of a run, or while in service via wireless communication.

3.2.7 Collision Avoidance Systems and Collision Notification Systems

Collision Avoidance Systems are an example of Intelligent Vehicles systems. The basic idea is that the vehicle uses sensors to detect and warn of obstacles, rollovers, impending forward collisions, and potential rear impacts. The driver is notified via the in-vehicle computer, perhaps by an audio signal.

While Collision Notification Systems where also a part of the Emergency Management area, in this case the individual vehicle is the focus. ACN is, as mentioned above, one way to notify a central control point or nearby vehicles that an accident has occurred.

3.3 Traveller Information in Sweden

Vägverket has sponsored ITS related research since 1995. At the time of writing, a national strategy for ITS research until 2009 exists. Between 1999 and 2002, 119 R&D projects were conducted at a cost of over 200 million Swedish kronor [22].

Since the development of common traveller information data sources and standards is a prerequisite for a full application (like the one this thesis will demonstrate) to work, it is of interest to examine what measures have been taken in Sweden so far, and what plans Vägverket and the private sector has for the future.

3.3.1 Traveller Information sources

The Vägverket has seven regional Traffic Information Centrals (TICs). They collect information from a variety of sources, among others the police and SOS Alarm (112), through sensors and cameras, and from tips from private persons. The TICs then decides which of this information to distribute, through for

example the Traffic Message Channels, the Internet, or radio. TRISS is the system Vägverket uses to collect, store, and secure the quality of this information. The TRISS FTP server resides in Borlänge. Generally, the information in TRISS can be categorized into: Accidents, Road conditions, Obstacles, and Information [53], [51].

Nationell Vägdatabas (National Road Database) is a database, run by among others Vägverket, developed following a directive from the government to provide updated, quality assured information about the entire road network. It contains information about road works, obstacles, and permanent restrictions (such as axle load restrictions). It is supposed to support community planning and travel and transport information, and tourist information [29]. Both TRISS and Nationell Vägdatabas are accessible for companies and authorities within the transportation area.

3.3.2 Information broadcasting example: RDS-TMC

RDS-TMC is short for Radio Data System, Traffic Message Channel. It is a system designed to broadcast weather and traffic information to vehicles over the FM radio band. In order to receive TMC messages, the vehicle must be equipped with a TMC receiver, which is typically a navigation system or a radio with a display. The information may be displayed on a screen or spoken in the form of a recording, and advanced navigation systems can change the suggested route based on TMC messages automatically [46].

TMC is a European standard and travellers get the messages in their own language, regardless of what country they currently are located in. The standard also ensures a certain quality of the information, and compability with the other countries. One such quality parameter is that the service must be available 24/7, and that it at least covers the TERN-network (Trans European Road Network). The TMC Forum is responsible for development and administration of the system, and they are financed by ERTICO [54].

In Sweden, Vägverket is responsible for the RDS-TMC service, while the private sector often is responsible for the service in other countries. The service covers the main roads and some bigger roads in cities.

The TMC message, received via the FM antenna, consists of an event and a position, and a time stamp. The message is coded in the Alert C protocol (Advice and problem Location of European Road Traffic), which is translated by each country [54]. A simple TMC message could be “218 8379 1 3”, which means

- 218 (event) – accident, congested traffic 4 km.
- 8379 (location) – code for intersection between Rv80 and Rv70 in Rättvik (geocodes so that GPS receivers may identify the location).
- 1 (recommendation) – avoid area.
- 3 (duration) – one hour.

The FM radio broadcast system may seem like an old technology. However, one important benefit is that virtually all cars equip a FM receiver. TMC Forum is looking at other channels, such as GSM/CDMA, to distribute the information.

3.3.3 Future ITS strategy

As mentioned earlier, Sweden has defined a ITS strategy to 2009, when the country will host the ITS World Congress. To achieve a wider use of ITS, Vägverket has defined five areas to focus on:

1. Improved traffic safety,
2. More effective commuter traffic,
3. Support for more effective industry transports,
4. Quality-ensured road – and traffic information,
5. Effective and credible work with ITS [52].

To fulfil the second goal, Vägverket wants to augment the collaboration with public transportation to be able to evaluate and conclude the information in a better way. Vägverket will also evaluate their own traveller information services (SMS and WAP). They also want to give the responsibility of RDS-TMC to an external service provider, to reduce expenses.

To fulfil the fourth goal, Vägverket wants to employ its own traffic reporters, rather than using Sveriges Radio's reporters. They want to have at least 5 service providers for travelling times on different road sections, and will use 2500 probes to provide this information. The cooperation with police, radio, and rescue services should also work better [52].

4 Operating system and programming language

With today's large number of portable terminals such as PDAs, smart phones, tablet PCs, etc. comes a range of operating systems as well. Widely used are for example the Symbian OS from Symbian Ltd. (a descendant of EPOC), Microsoft's Windows CE and Windows, PalmOS from PalmSource Inc., and Linux. One objective of this thesis was to find the platform best suited to implement the demonstrator (and perhaps an example application as well). In order to do this, one must first identify what is desired with respect to the operating system and programming language [59].

It should be noted, that while this study was interesting for future development of the demonstrator, the author of this thesis choose to implement the prototype NSB on a desktop computer. The reasons for this are explained in Chapter 8.

4.1 Important attributes

Three attributes will dominate the choice of platform, and they are presented here.

4.1.1 Application Programming Interface

Even though the operating system itself may come only with a set of libraries and no real API, third parties may have developed APIs specifically for an operating system or even a series of devices. A well-developed API will greatly promote code re-usability, and reduce the time a programmer would have to spend writing native code and allow the programmer to concentrate on the user interface and functionality.

4.1.2 Supply of devices

One important aspect to study when comparing different platforms is what devices are available in the market today using this platform. The choice of platform is tightly coupled with what hardware is to be used. Chapter 5 examines available devices on the market.

4.1.3 Future prospect

Harder to predict, but still important is where this platform may be in a couple of years. If there is a risk that the platform will disappear from the market, obviously this is a poor candidate as it could be difficult to port an application to another platform.

4.2 Operating systems

After a brief examination, the analysis was focused on two wide-spread technologies:

- Symbian OS with Java, used on for example Nokia and SonyEricsson phones, and
- Microsoft's Windows Mobile 2003/ 5.0, also known as PocketPC and used by for example HP and QTek, with .NET as the development environment.

4.2.1 Symbian OS with Java

Symbian OS is the platform found on Nokia's and SonyEricsson's smart phones, among others. Symbian Ltd. is owned by Ericsson, Panasonic, Nokia, Samsung,

Siemens AG, and Sony Ericsson. The latest version available is Symbian OS v9.1, introduced early in 2005. The user interface is shown in Figure 9.

The operating system deploys a kernel which handles scheduling and memory management, but not the file system. The scheduling works in such a way, that when no application is running, the CPU simply switches off after scheduling a wakeup. Symbian OS also deploys an advanced system for managing memory, to prevent memory leaks and to minimize memory consumption.

There are multiple user interfaces to Symbian OS, such as UIQ and Nokia's Series 60, Series 80, etc. Each comes with its own SDK [26].



Figure 9: Display from a device running the Symbian OS from Symbian Ltd. [33].

4.2.1.1 API

Unfortunately, no Standard Development Kit (SDK) exists for Symbian OS. Instead, companies such as SonyEricsson and Nokia provide their own APIs, written in Java and C++. The SUN Java Micro Edition, J2ME, supplies its own API, and thus many favour the combination of the Symbian OS with Java. Important to note is that today there exist two different frameworks for J2ME. One is the Connected Device Configuration, CDC, and the other one is Connected Limited Device Configuration (CLDC). Both were developed for handheld devices, but CLDC is a subset of CDC and lacks some functionality, such as native code invocation and some security-related functions. Figure 10 shows the relationship between J2SE and J2ME.

Third-party APIs also exists. For example, Nokia offers packages for accessing video and multimedia, wireless network communications, and enhanced UIs as an extension to J2ME [41].

Java is supposed to be platform-independent, as is J2ME. However, all the optional third-party extensions for mobile devices make this statement more or less untrue for J2ME. Platform-independence could be achieved, but one must put a lot of effort into the standardization process. The predicted use of native code in the NSB would also reduce platform-independence.

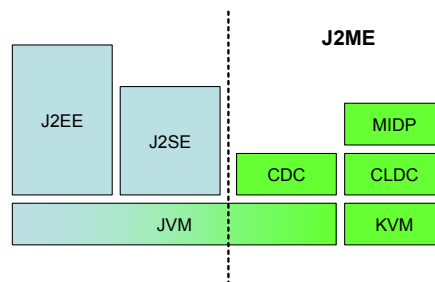


Figure 10: The Sun Java API structure

4.2.1.2 Supply of devices

As mentioned earlier, Nokia and SonyEricsson phones use Symbian OS, so there are many phones on the market which could be considered for ITS applications. One advantage is that because all these devices are primarily phones, all models include internal modems and all have Bluetooth capabilities. However, internal GPS receivers are not yet common.

4.2.1.3 Future prospects

As Symbian Ltd. is owned by Nokia and SonyEricsson among others, they will surely continue to use Symbian OS for a while, and the platform still dominate the market. However, .NET is becoming increasingly popular for development on both stationary units and mobile ones, maybe as the Windows environment is familiar to many. Analysis predict that Symbian will lose market share over the next five years. If this means that in time Windows Mobile will be the biggest operating system, this speaks against using Symbian OS as a platform [1].

4.2.2 Windows Mobile 2003/5.0 with .NET CF

Microsoft Windows Mobile, sometimes interchangeably called PocketPC, is found on HP's iPAQ series, QTek's handhelds, and in Dell and Acer's devices, to mention only a few. The OS is a derivative of Microsoft's Windows CE, and supports the .NET Compact Framework, which is a subset of the .NET Framework. The .NET CF can be seen as Microsoft's equivalent of Sun's J2ME [23].

The system is, just as Symbian OS, designed to run with limited memory. The kernel runs in less than one megabyte of memory. Just as in Microsoft's Windows NT, the fundamental unit of execution is a thread. According to Microsoft, Windows Mobile 5.0 is optimized for lower power consumption than its predecessors. The user interface is illustrated in Figure 11.



Figure 11: Windows Mobile 2005

4.2.2.1 API

The current release is Windows Mobile 5.0, introduced in May 2005. Windows Mobile 5.0 supports development in .NET CF 2.0 and Visual Studio 2005. .NET CF 2.0 introduces some handy APIs, for example messaging and telephony functions, GPS functions, extended graphic libraries, etc. Functions for easily controlling serial ports is also be included. Microsoft has also included extended support for interoperability between managed and native code [23].

Compared with J2ME, .NET CF has better support for calling native methods, partly because Microsoft is the sole company that controls both .NET CF and the Windows OS. The P/Invoke methods introduce a way to call functions in the Win32 native libraries (although far from easy to use for the programmer) [61].

4.2.2.2 Supply of devices

Currently, Dell and QTek are two companies that include Windows Mobile 5.0 on their PDAs. The HP iPAQ series will surely follow soon. There is an ample supply of different brands on the market. One drawback is that few yet come with internal UMTS modems.

4.2.2.3 Future prospects

As described above, the future of Windows-based PDAs look bright. Even though devices with Windows Mobile 2003 are still being released, 5.0 will soon be standard. To find suitable hardware by the end of 2006 should not present a problem.

4.3 Conclusion

Symbian OS and Windows Mobile 5.0 are both good candidates for development on handheld devices. Symbian is currently the biggest. Windows Mobile has good future prospects, but lacks a range of products as manufacturers are changing to Windows Mobile 2005.

The set of available APIs for the systems are comparable. Both .NET CF and CDC J2ME allow invocation of native code, and both systems allow managed code access to some internal hardware. As the application will probably utilize GPS, Windows Mobile 5.0's API is interesting.

The development tool to use for all .NET programming is Visual Studio. VS 2005 beta was just released, and the final version will support .NET CF 2.0. This is an environment where the screen and buttons of a number of devices can be simulated. On the Java side, most J2SE tools like NetBeans, JBuilder, Eclipse, etc. supply similar functionality for J2ME. Consequently, finding a good IDE should not present a problem when choosing between Symbian and Windows Mobile.

As both operating systems seem equally capable of hosting the application, one should examine the available devices before making a decision. This is discussed in chapter 5. However, due to future prospects and how the systems manage native code, the author of this thesis would recommend the application to be developed in the .NET CF 2.0 environment on a Windows Mobile 5.0 device. As a programming language, C# or Visual Basic .NET is interesting, as they both avoid some old-fashioned concepts (like explicit pointers) in the C++ language. If a Symbian OS device with superior technical attributes is found, however, one should not hesitate choose it as a platform.

5 Hardware

An ITS application is typically located inside a vehicle or carried by a pedestrian. For pedestrians, the portability of the terminal is of greatest importance. As discussed earlier, the ideal device for the final application would be a smartphone or a PDA, as they combine extensive functionality with a familiar user interface and good portability. In this chapter, four devices are examined to highlight what is available on the market today. Two Windows-based PDAs and two Symbian-based smartphones are examined.

5.1 Important attributes

The NSB prototype should ideally support four different techniques for communication; GPRS, UMTS, WLAN, and DAB. The ultimate choice would be a device with all these techniques integrated, but currently no such device exists. No handheld device has a DAB receiver integrated. Also, some devices have a combined GPRS/UMTS modem, which automatically switches to GPRS when UMTS becomes unavailable. As this would bypass our decision making algorithm, such modems must be avoided.

For many ITS applications, the device must be equipped with a GPS. This may not be needed for the demonstrator, depending on what functions are implemented.

The hardware that is not integrated in the device could be connected via Bluetooth or a SD port. To have many units connected this way does not present a problem for a demonstration, but would not be very handy in real applications. Presumably, one will have to make a compromise here.

The operating system is, as discussed earlier, an important attribute that will be examined.

Other attributes, such as battery life time, robustness, etc. are harder to judge by reading the data sheets, and would have to be measured experimentally. Without conducting any experiments, we may conclude that battery time will be very limited in any handheld device, if all networks mentioned earlier are used.

5.2 Devices

The devices which were examined are a QTek 9000, a HP iPAQ HW6515, a Sony Ericsson P990, and a Nokia N91. They are interesting as they are either new on the market or soon to be released and thus give a good overview of today's market, and also as they integrate many of the desired features.

5.2.1 QTek 9000

QTek is the European name for digital assistants manufactured by High Tech Computers (HTC) in Taiwan. They are distributed in Sweden by BrightPoint Inc. QTek enjoys a good reputation among users, and are generally fast in adopting to new technologies.

QTek's new flagship, the QTek 9000, is the first Windows Mobile device with 3G support. It was released early in 2006, and will surely compete with Nokia's and Sony Ericsson's new products.

In addition to implementing GSM/GPRS and UMTS, QTek 9000 also supports the IEEE 802.11b standard. It has a 3.6" screen and a keyboard.

Its external interfaces include one SDIO/MMC card slot, and a Bluetooth device, enough to connect a range of devices.

QTek was early to deploy Windows Mobile 5.0 on their devices, and QTek 9000 is no exception.

All in all, QTek 9000 is an interesting device. A GPS and a DAB receiver would have to be connected for a demonstrator. As the product is not yet on the market, however, teething troubles are yet to be discovered [35].

5.2.2 HP iPAQ HW6515 Mobile Messenger

HP iPAQ series has been around for quite some time, and established a solid presence on the PDA scene. Their latest release, the HP iPAQ HW6515 Mobile Messenger was released back in July 2005, and lacks the UMTS support promised in the other examined devices. It also lacks WLAN support. However, it implements a Tom-Tom GPS receiver, which is an interesting feature. The GPS receiver has been tested with good results [25].

The screen size is 3", and the lower part of the device contains a keyboard. It weights half of what the QTek 9000 weights, 165 g, and is a bit smaller.

The external interfaces of the iPAQ HW6515 pretty much mirrors the QTek. It has a SDIO slot and Bluetooth.

In the time of writing, the HW6515 ships with Windows Mobile 2003, and no upgrade to 5.0 will be available [25].

The HP iPAQ is interesting because of its internal GPS. However, it lacks WLAN, 3G and DAB support, which means many external devices will have to be connected for the NSB to work. This will make the unit a lot bulkier. It should be considered how important an internal GPS is.

5.2.3 Sony Ericsson P990

After much secretiveness, Sony Ericsson finally unveiled its first 3G compliant smartphone P990, to be released in early 2006. It features WLAN compability, Bluetooth, support for 2 GB memory with a Memory Stick, and of course GSM/GPRS capabilities.

The phone has a screen size of 2.8", slightly less than the iPAQ, and deploys a keyboard beneath the flip-down keypad.

The P990 runs on Symbian OS v 9.1. It supports the Java ME platform through both CDC and CLDC, which is a great benefit for developers. It comes with a range of other APIs as well, for example MIDP 2.0 (JSR-118), Bluetooth (JSR-82), Mobile Media API (JSR-135), etc. The P990 uses the UIQ 3.0 user interface, a platform that facilitates development according to Sony Ericsson.

All in all, the P990 phone seems as a potential platform for this thesis, especially since CDC is included. The support for huge storing capabilities may be of use if it is decided that databases or maps should be locally cached.

5.2.4 Nokia N91

The Nokia N91 is very similar to the Sony Ericsson P990 from our point of view. It supplies WLAN support, GSM/GPRS and UMTS modem, and Bluetooth. N91 supports 4 GB of storage capacity, which is 2 GB more than the P990.

Nokia N91 also runs on Symbian OS v9.1. One great drawback is that it only supports CLDC, and thus it does not allow for native code invocation. This makes it less attractive than the Sony Ericsson P990.

5.3 Conclusion

The result from the analysis is summarized in Table 1 [35], [12], [42], [28]..





	QTek 9000	HP iPAQ HW6515	Sony Ericsson P990	Nokia N91
				
Comm.	GSM, GPRS, UMTS, WLAN	GSM, GPRS, EDGE	GSM, GPRS, UMTS, WLAN	GSM, GPRS, UMTS, WLAN
GPS	No	Yes	No	No
Operating system	Windows Mobile 5.0	Windows Mobile 2003	Symbian OS v9.1 (CDC)	Symbian OS v9.1 (CLDC)
Bluetooth	Yes	Yes	Yes	Yes
SD	Yes	Yes	No	No
Internal memory	128 MB ROM, 64 MB RAM	64 MB ROM, 64 MB RAM,	80 MB	-
Release date	Early 2006	July 2005	2006	2006

Table 1: Smartphones and PDAs of interest today.

Most of the phones examined could work as a platform for the demonstrator. The HP is less suitable as it lacks both WLAN and UMTS support. HP has yet not announced any plans to release a 3G phone, but it will surely be released during 2006. The Nokia phone is less suitable than the Sony Ericsson, as it only supports CLDC.

In summary, the QTek 9000 is a good choice if Windows Mobile 5.0 is preferred as an operating system, while the Sony Ericsson P990 is a good choice if Symbian OS is chosen.

6 Wireless network roaming solutions

This chapter examines two commercially available solutions for wireless roaming, namely Columbitech's Wireless VPN™ [55] and Birdstep's Intelligent Mobile IP Client [56], [4]. A research project using extended sockets is also examined [17].

The commercial solutions both provide Virtual Private Network (VPN) functionality in addition to mobility. A VPN enables users to be connected to a corporate or organisation intranet, when in reality they are working from a remote location. This is achieved by establishing secure tunnels, which means data is encrypted before it is sent over the insecure Internet, and decrypted at the organisation's gateway. Such a tunnel is normally established through a user authentication. As remote users often are confined to wireless technologies, implementing roaming support is a natural part of a VPN solution.

VPN is not needed for the NSB. However, the solutions are well documented and provide good insight into mobility and network roaming in general. Other products that provide similar functionality are NetMotion's Mobility XE, IP Unplugged, and Viatore's Mobile IP VPN.

6.1 Columbitech Wireless VPN™

Columbitech's Wireless VPN is a VPN solution providing secure remote access to corporate data, focused on solving the problems involved with wireless access. Data is transferred via an encrypted and authenticated WTLS session over a public network to a VPN server residing on the corporate network. The VPN sessions are supposed to provide totally seamless handovers and also allow resumption of for example an interrupted file transfer after a period in an area without any network coverage. Columbitech's Wireless VPN runs on most Windows desktop distributions, and to some degree on handheld devices.

The solution operates at the session layer, which is the fifth layer in the OSI model and does not utilize Mobile IP, which operates at the IP layer. The VPN is transparent to the application layer. Meanwhile, by residing on top of a transport layer, some shortcomings of the TCP protocol have been overcome.

6.1.1 Architecture

The Wireless VPN solution consists of a Client, and an Enterprise Server, with an optional so-called Gatekeeper.

The Enterprise Server terminates all of the VPN connections and resides inside a firewall. The clients can connect directly to the Enterprise Server or through the Gatekeeper. When a client connects to the server, an authentication process is started (see section 6.1.3). The client is assigned an IP address from a pool of IP addresses. This is the address that will be visible to hosts within the corporation, hiding the temporary physical IP addresses of the client. The VPN server is the only component that cares about the client's actual IP addresses, and neither end of the application connection knows about its existence. The server also includes a set of administrative tools and a certificate manager.

The Gatekeeper is an additional server component, residing on a corporate DMZ, that is, between one firewall to the Internet and another firewall leading into the corporation (and the Enterprise Server). The purpose of the Gatekeeper is to

simplify firewall configuration, enable load balancing, and increase security by taking care of authentication outside the firewall.

The client is implemented as a virtual network interface, which is assigned an IP address by the server. All traffic is then forced through this interface, so that encryption and session layer compression can be applied before sending the packets. The client also includes a GUI monitor, enabling the user to either choose a specific network interface to use or allow the network interface be chosen automatically.

6.1.2 Mobility

In [57], some of the inherent problems with IP Mobility are pointed out, and it is explained how these problems are bypassed by the Wireless VPN by placing mobility support at the session layer.

The Mobile IP solution, as explained in Section 2.6, hides the change of IP addresses by intercepting all traffic to the MN using a Home Agent – enabling the MN to use a static IP addresses. This HA keeps track of the current IP address of the client and forwards the packets to that address by using IP-in-IP encapsulation. This hides the IP address change from the transport layer, which allows a TCP session to survive. Columbitech's approach, on the other hand, makes no attempt to keep transport level connections alive during roaming. The client's TCP connection is terminated locally at the virtual NIC, making the application think it is always connected. Another (real) TCP connection, for example over GPRS, connects the applications via the VPN Server. This relationship is illustrated in Figure 12. So, if a connection is broken, the VPN Server initiates a new real connection transparently to the client and server. The WTLS session, which is used as the session layer protocol, is able to resume without needing a new handshake.

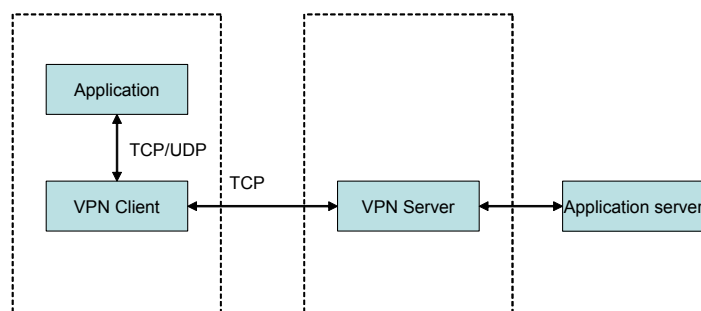


Figure 12: The application connections are terminated locally and redirected over a single TCP connection via the VPN Server.

This approach solves two problems. The first is TCP's sensitivity to large delay variations. If a TCP connection suddenly starts to use a lower capacity link, the risk is that a number of timeouts will be triggered, from which the TCP connection may take long to recover. Unlike when using MIP, the underlying TCP session does not need to survive as explained earlier.

The other problem is the so-called NAT-problem. Commercial service providers often use Network Address Translation servers in order to augment its public IP address space. Providing only its own well-known public IP address outwardly, a

NAT server uses the destination port number included in the TCP header to identify from which host the packet was sent from, and routes the incoming packets to the correct machine. A problem arises as NAT servers expect transport level information rather than IPSec headers following the IP header as is the case in a VPN application (although some NATs today are IPSec-aware). A similar problem exists for Mobile IP, as there is no transport protocol between the two IP headers in IP-in-IP encapsulation. This causes the NAT server to discard the packets. Columbitech's WTLS header is placed after the transport protocol header, solving this problem. For Mobile IP, there are other solutions to this problem, as explained later.

6.1.3 Security

As mentioned, WTLS is used to encrypt, authenticate, and validate transmitted data. WTLS is a wireless implementation of TLS, which is an enhanced version of SSL 3.0. To provide support for low bandwidth devices, WTLS adds data and header compression to the TLS protocol. Unlike TLS, which uses a stream based design, WTLS uses a design that is better fitted for packet based transmissions [30].

The Wireless VPN uses DES (56 bit), 3DES (112 bit), and AES (256 bit) for symmetric data encryption, and RSA (up to 15360 bit) for asymmetric encryption during the key exchange. This is a typical use of security protocols, as while the asymmetric encryption methods are superior when it comes to security, they are too processor intensive to be used on the payload data. For hashing and signing, the MD5 (128 bit) and SHA (up to 512 bit) algorithms are used.

If a Gatekeeper is used it could be configured to request client certificate authentication or one-time-password authentication prior to a VPN session. The VPN server could then ask for a Windows NT or RADIUS password. Furthermore, the Enterprise server includes a firewall to filter and inspect packets.

6.1.4 Roaming

The session layer is responsible for handling roaming between different networks in WVPN. A so-called *Session Resume* lets the user reconnect after a network change or failure without further identification processing. If data was being transferred, a *Transaction Recovery* is supposed to pick up data from where it was interrupted.

If in automatic mode, the client constantly scans the networks and chooses the best available one. If no network is available, the virtual network interface will still seem to be connected and to accept traffic. When a new connection is found, the buffered data is redirected to that network interface. If the virtual NIC buffer becomes full, standard TCP mechanisms are used to halt the data flow until a connection is found. The delays during handoffs associated with TCP are not eliminated, but the TCP session will not fail because of the them.

6.2 Birdstep Intelligent Mobile IP Client

The Birdstep Intelligent Mobile IP Client addresses the issue of wireless mobility in quite another way than Columbitech's solution, namely by applying a Mobile IP solution together with a (separate) VPN solution, thereby creating what they call a Mobile VPN. Birdstep merely supplies the client that handles network

handovers, while it relies on third party solutions for securing the connection (through VPN) and routing the data (through a Mobile IP implementation).

Birdstep's solution can be run in either Standard mode or Universal mode. In Standard mode, mobility is provided between public networks and works outside any enterprise firewall. A third party VPN is required. In Universal mode, mobility is provided both in public networks *and* on the enterprise intranet. The VPN is switched on and off depending on whether the user is connected to the intranet or not. Unlike the Columbitech solution, a virtual network adapter is never used.

6.2.1 Architecture

Birdstep Intelligent Mobile IP Client makes use of the basic Mobile IP concepts to allow network roaming on a secure VPN tunnel. It relies on third party VPN software and Mobile IP Home Agents. Birdstep supports Mobile IP from HP, Cisco, Sun, and Linux, and VPNs from Nortel, CheckPoint, Cisco and Microsoft. An overview of Mobile IP can be found in Section 2.6.

By relying so heavily on third-party server components, it is incorrect to speak about a system architecture, as the client is the only software added by Birdstep. The client provides authentication support, network performance monitoring, automatic handovers, and network interface detection.

Birdstep uses different solutions depending on whether the user is accessing the enterprise from the outside or from the inside (of the local network and firewall). If the user is located outside the enterprise and using an insecure public network via for example a dial-up connection, the solution is called "VPN overlaid Mobile IP". In this case, the HA is typically located in a corporate DMZ. The VPN session is not altered by the fact that Mobile IP is used, thus providing the security needed. The VPN concentrator is also placed in the DMZ. The advantage of this solution is that a VPN tunnel survives a handoff, and that any Mobile IP solution can be used.

On the other hand, when the user resides inside the corporate network, the "Mobile IP inside VPN" approach is used. The HA is placed on the enterprise intranet, and Mobile IP only works between different access networks within the intranet. The VPN terminate the IPsec tunnel in the corporate DMZ. This means that for internal access, the VPN is inactive, which reduces overhead to some degree.

6.2.2 Mobility

Mobility is provided by the Mobile IP implementation, and Birdstep is compatible with several major MIP solutions. The fact that an IP header follows directly after another IP header instead of a TCP/UDP header as expected by the NAT server causes NAT-problems, as discussed earlier. To overcome this, NAT traversal as described in the IETF draft is supported [56]. To summarize, NAT traversal detects the presence of a NAT server, and modifies the packets in such a way that they are accepted and forwarded by the NAT server.

Birdstep also supports reverse tunnelling to overcome the problems associated with ingress filtering, and co-located care of addresses to compensate for the lack of a Foreign Agent.

6.2.3 Security

Mobile IP itself only covers some security aspects, namely the following:

- Authentication of registration messages,
- preventing hi-jacking of traffic,
- ensures integrity of the binding list, and
- accommodates any authentication algorithm.

However, data security is not part of Mobile IP. Instead, Birdstep relies upon the VPN to provide appropriate data security, which is typically obtained through IPSec or some SSL implementation (as in OpenVPN).

6.2.4 Roaming

The Birdstep client monitors the performance of the available connections, and makes handover decisions based on that information. If necessary, the client performs a PPP dial-up to access the corporate network with for example GSM and ISDN. Always-connected networks like GPRS and UMTS are initiated as soon as the client is started.

6.3 Resilient Mobile Sockets

Resilient Mobile Sockets, RMS, is a transport layer solution to mobility proposed by the licentiate thesis [17]. Similar research has been conducted earlier; see for example [36] and [38]. Extending a socket, in this case the UDP socket, means that existing applications must be modified to use RMS. Most user applications today use either the Windows Socket 2 API or the Berkeley Socket API. The proposal does not deal with security issues.

6.3.1 Architecture

RMS is an UDP extension focused on providing mobility for real-time media applications such as Voice over IP (VoIP). The idea is to extend UDP to the RMS socket, which provides additional functionality for mobility. For example, RMS does address translation, and buffers data while no network connection is available. Just as Columbitech's solution, a failing (internal) socket is removed and replaced with a new one as soon as it is available.

RMS also provides a handover manager, which decided when a handover should take place. It also supports more than one open connection at a time. The handover manager is completely separated from the rest of the socket, making it reconfigurable.

The packet translator contains a database containing both home location and current location of each connected end-point. Using this database, the packet's IP addresses are re-stamped to be directed to the current IP addresses. This hides changes of location to the application layer. If a client-server relationship is assumed, it is enough that the server to keep an updated database with the client's current location. However, RMS is designed to also be used for client-to-client communication. The architecture is shown in Figure 13.

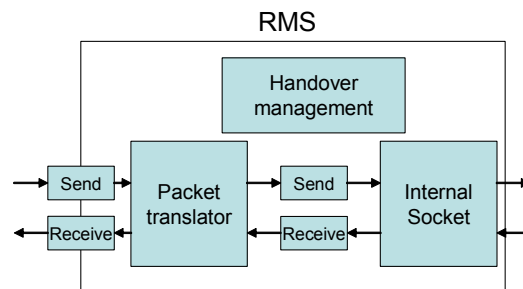


Figure 13: RMS architecture. The internal socket is a normal UDP socket

RMS is able to distinguish between data destined for the end applications and data containing control information for RMS. Control information can for example be the exchange of location information, most notably the client sending its new current IP address to the server. Such control messages are marked with a magic number to be recognized.

The RMS socket was found to introduce quite short handover delays. The main disadvantage of the RMS socket is obviously that the applications have to be aware of the socket extension in order to utilize it.

7 Design alternatives and considerations

The NSB is the core of the PIITSA project. It is responsible for handover control, the roaming strategies, and monitoring QoS parameters. To begin with, the requirements on the NSB are analyzed. Next, this chapter presents two possible ways in which the NSB demonstrator could be implemented based on these requirements. Both alternatives are inspired from the studies accounted for in Chapter 6.

The objective of this thesis is to demonstrate the research results obtained in the PIITSA project, not to develop a full-fledged solution. As far as possible, though, future development (which could be the next step in PIITSA) should be able to use the demonstrator as a platform.

7.1 Requirements

From earlier studies within PIITSA, a picture of how the NSB would interact with an user application has formed. The envisioned relationship between a user application and NSB can be seen in Figure 14. We can identify the difference between the server side NSB and the client side NSB, mainly due to the fact that DAB is a one-way service. Furthermore, the NSB should be able to be controlled by the user on both server and client side.

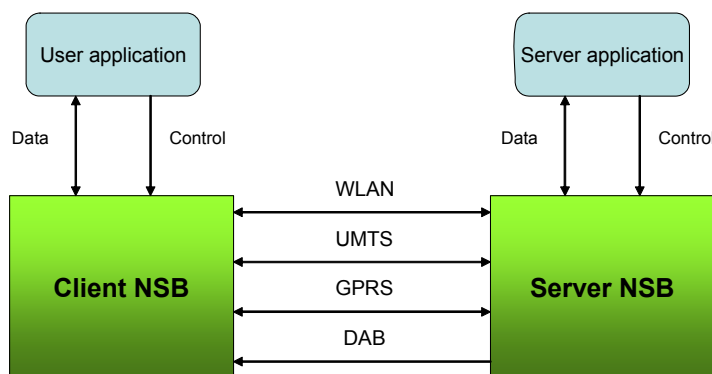


Figure 14: The NSB - User Application relationship

These basic requirements are imposed on the prototype NSB:

1. The NSB should support roaming between (ideally) the GPRS, UMTS, WLAN, and DAB networks, transparently to the user [19].
2. The NSB should be controllable, both through a user interface and through requests issued by the application.
3. Network quality parameters should be monitored, to be used in the decision-making algorithm [13].
4. The prototype should contain a (basic) decision-making unit whose task is to select the appropriate network connection based on the user/application preferences and network quality [13].

5. The five generic services (explained in Section 2.1.2) should be supported [21].

With these requirements in mind, the design issues can be isolated and examined. The four main issues to consider are the following:

- On what device and OS to implement the demonstrator (this issue was addressed earlier)? Is it possible to port the software at a later stage?
- How to provide support for handovers and mobility which meets the demands of the five generic services defined by Fiedler et al. in [21]?
- How to provide adequate security for the client and server?
- How to communicate with the user, alternatively with the user application?

The previous chapters describe the analyses conducted so far. Several different platforms were examined to understand the potential of a handheld device to host a NSB application. Next, other solutions with similar functionality to the NSB were examined.

The insight gained during the preliminary study enables us to focus on a smaller number of design alternatives. The following sections presents potential NSB designs judged to be viable, that is, meeting the requirements stated above. First, the mobility issue is handled, and two design alternatives are proposed. Link monitoring and security is then covered briefly. As the roaming policy unit for PIITSA is being developed by BTH as described in Section 2.1.4, that part of the NSB is not treated here. For another example of research on policy based roaming, see [24].

7.2 Mobility

The greatest problem to solve is, as have been pointed out earlier, the issue of providing seamless handovers between the networks when the clients IP address changes. This section examines two substantially different ways of achieving this.

7.2.1 Seamless roaming through third-party Mobile IP

Inspired by Birdstep Intelligent MIP, where the server operates with a third party Mobile IP implementation, this design alternative also focuses on the client software leaving the mobility issue in the hands of an existing solution. The client NSB takes care of when to switch networks, user preferences, the security etc. A robust MIP implementation and a solution to the NAT problem have to be found. See Section 2.6 for a general description of Mobile IP.

7.2.1.1 Basic concepts

The client software consists of two parts: the NSB and the MN software, which is a part of MIP. No FA is used; the MN knows the co-located care of address. The HA may be connected via a wired LAN to the internet, as can the server. The MN is responsible for among other things communication to and from HA, and forwards packets to the application. The Mobile IP software at the MN also changes the source address to its home source address before the messages are sent to the server.

The NSB is oblivious to what MIP implementation is used, as seen in Figure 15. On the client side, the NSB lets the user switch manually or automatically between the networks available, dialling modem connections when necessary. All applications, including web browsers, can be used with the NSB as connectivity is always available. Even file transfers and streaming services may resume after a handover. One problem is that all traffic passes through the HA, and this can generate bad performance.

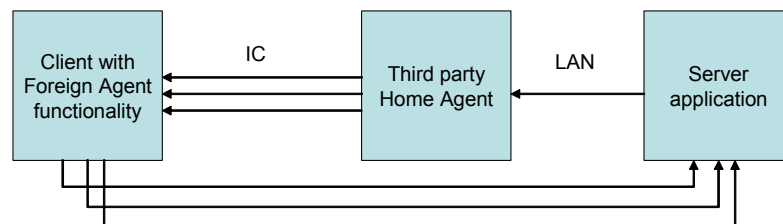


Figure 15: The third-party MIP HA between client and server illustrated with triangular routing.

7.2.1.2 Mobile IP solutions

Dynamics from Helsinki Institute of Technology (HUT), is an open-source software developed in 2001. It is no longer maintained, but is still a popular alternative as it remains one of the few free IPv4 MIP implementations. Dynamics includes software for both HA and FA, and runs on Linux. The MN software is partially ported to Windows 98/2000 (but requiring among other things that Cygwin is installed). One drawback of Dynamics is that it does not implement NAT traversal, see the following section [7].

Another free MIP distribution is Linux Mobile IP from MosquitoNet Mobile Computing Group. Linux MIP does not provide a FA; otherwise, it is very similar to the HUT solution. NAT traversal is not supported, and just as Dynamics, the solution is adapted for Linux Redhat. No porting to Windows for the MN exists, which makes this distribution quite unusable for our purpose [20].

Cisco has developed MIP for their Cisco IOS operating system, complete with HA, FA, and an MN with support for RFC 3519 Mobile IP NAT Traversal and a bunch of other RFCs as well. It supports co-located care of address, and the HA runs on most Cisco routers. One benefit of the Cisco MIP implementation is that extensive documentation is available and the well-known IOS command-line interface for can be used configurations. A drawback is that it is not free [6].

There are a number of other implementations as well. One problem is that many solutions have connectivity management built into the client, which do not suit us well since the roaming policy should be developed within the PIITSA project.

7.2.1.3 Solutions to the NAT-problem

The NAT issues associated with MIP have been discussed earlier (see section 6.1.2) and debated extensively. The need to implement NAT compability with MIP is obvious, as virtually all wireless network operators use NAT servers. Vodafone offers public, static IP addresses to users (that is, outside their NAT

server). However, our solution cannot be limited to one service provider if it is to be commercialized at some point.

One solution to the NAT problem is to simply insert a new transport protocol header between the two IP packets, thus creating an UDP tunnel upon the first IP protocol for the data traffic, as shown in Figure 16. It is preferred to use UDP as the second transport protocol, as double TCP protocols may reduce performance and create congestion problems. Also, an UDP header is a lot smaller than a TCP header.

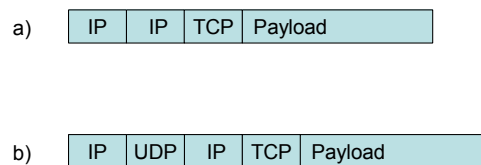


Figure 16: Insertion of UDP header (b) on IP-in-IP protocol (a) to avoid NAT problem

This is basically the approach proposed by IETF in RFC 3519 [18], but they suggest adding some extra control features which may or may not turn out to be needed in our solution. The proposed IETF solution is quite extensive as it is formulated to support GRE tunnelling and minimal IP encapsulation in addition to IP-in-IP encapsulations.

We are faced with the alternatives of either developing UDP tunnels similar to how they are specified in the RFC from scratch, or using a commercial MIP with NAT traversal implemented, such as Cisco's Mobile IP software. Another possibility is to use a Vodafone subscription with a public IP address during the demonstration, and develop support for NAT traversal at a later stage.

7.2.1.4 Advantages

- Seamless mobility is achieved as TCP sessions survive network handovers.
- No extensive development would need to be done.

7.2.1.5 Disadvantages

- Additional software may have to be purchased or developed, as none of the free MIP solutions really meet our demands.
- Applications may crash if no network coverage at all is present.
- Mobile IP can provide bad routing performance as packets have to pass the home agent.

7.2.2 Seamless roaming through a virtual interface

Some companies that supply wireless VPN solutions, like Columbitech's WVPN dealt with in section 6.1, and Netmotion's Mobility XE, do not use Mobile IP; but instead utilize session based mobility, and custom protocols residing above the transport and network layers (but not fully at the application layer). They argue

that this approach supplies application session persistence even when the user moves through coverage gaps. This second design alternative examines how this could be implemented in our solution.

7.2.2.1 Basic concept

One problem when using MIP arises if an application tries to send or receive data while no network coverage at all exists, as the TCP protocol will then assume the network congested and in time terminate the connection, many times resulting in the application crashing [58].

As described in chapter 6, Columbitech uses a virtual NIC where the TCP connection is locally terminated (the internal connection), and a second (external) TCP connection to the VPN server over the air interface. This hides failing TCP sessions to the application. If the external connection goes down, a new one is initiated. On the server, a “Translator” maps the current physical address of the client with a virtual address gained through for example a DHCP request at server start-up. The server application’s internal TCP connection is terminated at this virtual address, so that the server experiences a static client IP address.

The research by Xun et al. [36] describe their experiences with split TCP sockets. They identify the problems involved when the receiving buffer of the internal TCP connection fills up due to the external TCP connection being down (due to lack of network coverage or a handover with great delay). This must be managed using TCP control mechanisms. Moreover, the report uses the term *virtual port*, which is essentially the same a virtual NIC. A *Virtual Port Protocol* is proposed, which initiates connections and resynchronizes data transfers after reconnections.

By employing a similar solution, no third party MIP would be needed. The greatest challenge would be to design the protocol which halts and recovers TCP sessions. Unlike standard solutions like MIP with VPN, documentation is hard to find as there are no obvious way to design session persistence. The following problems would have to be solved:

- Set up two TCP connections, one internal (to a virtual NIC or virtual port) and one external connection, that is the physical link. Or, instead of terminating the TCP tunnel locally, packets could be tunneled in another protocol over the actual interfaces (see full detail about this in chapter 8).
- Develop the Translator Server, who operates between the client and server and associates the current physical IP of the client with a constant, virtual address. Neither the client nor the server should be aware of the Translator.
- Develop mechanisms for buffering TCP packets and halting TCP sendings when buffer is full.
- Develop a protocol for maintaining two TCP connections (or a tunnel).

Figure 17 shows what the NSB could look like implemented this way.

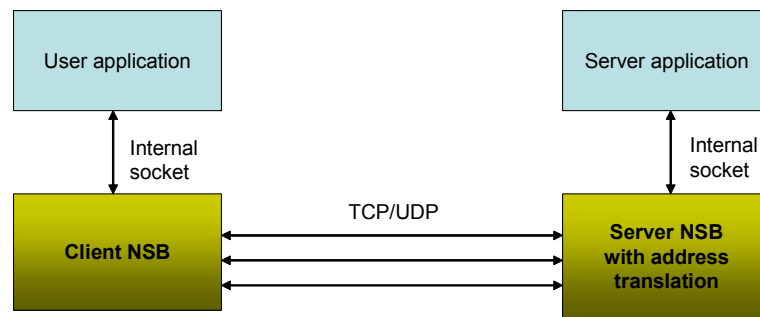


Figure 17: Session (application) based mobility. Dual sockets are used, one internal and one external. The external connection (which could be TCP or UDP) is allowed to go down or change.

The problem with this solution is to “collect” all incoming and outgoing network traffic, and to be able to treat the data the way you want (either by sending it through another socket or by tunnelling it). One way is to use a virtual network adapter, which would have to be programmed as a driver (like Columbitech’s WVPN). Applications should not have to be aware of the NSB to use it.

The other way is to develop a custom socket, that forwards the data to an NSB at application layer (like the RMS socket). This would require all applications that want to use the NSB to be aware of its existence, and is in this sense not an attractive alternative.

7.2.2.2 Advantages

- True application session persistence is achieved even between network coverage gaps.
- No third party software needed.

7.2.2.3 Disadvantages

- The application would have to be NSB-aware, or
- Kernel code would have to be written/utilized to implement a virtual network interface.

7.3 Link monitoring

Collection and evaluation of QoS parameters has been discussed widely in the Internet community, and is far from a simple task. The subject will be treated only briefly, as it is out of the scope for this thesis to develop a full QoS reporting unit.

There are mainly four parameters involved when we talk about quality of service, namely bandwidth, jitter, delay and reliability [8]. *Bandwidth* is the number of bits transferred from one side to the other per second (or to the other side and back, the round-trip time). *Delay* is the average time it takes a bit to travel from source to destination. *Jitter* is the standard deviation of the delay. *Reliability* is measured in the average packet loss in percent. All of these parameters should be measured, as they are essential data for the decision making unit [13].

We can justify the need of active monitoring of QoS parameters. Especially with wireless networks, throughput, delay, jitter and packet loss vary considerably over time. An example is GPRS which uses a different number of timeslots depending on the network load. Thus, we cannot merely refer to a database with static parameters for each network, and consequently the networks must be monitored constantly.

However, there are times, as at the initial start-up of the NSB, that there will be no measurements available on the network connections. It is therefore imperative to have access to a static database with estimated network characteristics for each network, which can be used as a temporary base for decision making. Extensive research has been conducted on average link qualities, and these results could be used to create the local database, together with current cost parameters.

A greater problem to solve is how to measure the link quality. One way is to send control packets over the networks, for example time-stamped ICMP packets (like the ping utility). In this way, round-trip time delay and packet loss may be measured. One drawback is the bandwidth cost. Imagine a server side NSB pinging 1000 clients with two network connections each, every second. A ping will generate two ICMP packets; one request and one reply. An ICMP packet is at least 42 bytes long (with 0 bytes data). This would generate a load at the server's internet connection of 1.4 Mbps. To avoid this amount of extra traffic, the control messages could be piggy-backed on ordinary data packets.

7.4 Security

The solutions examined in Chapter 6 both use advanced VPN implementations that utilize VPN tunnels to the corporate intranet as the users roam between networks. We have seen that typically a VPN gateway is located in the enterprise DMZ for this purpose, and that RADIUS servers are sometimes used for Authentication, Authorization, and Accounting (AAA) tasks.

These solutions are designed for corporations with an intranet to which users should have constant, secure access to. Furthermore, the solutions aim at securing every possible application.

The security demands on our application are not that heavy, especially not on the demonstration stage. For one thing, if a web browser is used the HTTPS protocol can be utilized, or if a custom application that needs security is built, security can be implemented in that application. We probably have no intranet to protect. Regardless of how the NSB is implemented, this can be achieved both in a custom application and in a web browser without any additional software. Conclusively, the NSB may rely on the user application to provide adequate security for the current service.

8 Implementation

Chapter 7 proposed three ways to implement different degrees of mobility in the NSB, and also examined the problems of security and link monitoring. The suggestions, especially those for mobility, were based on studies of existing commercial and non-commercial solutions, presented in chapter 6. This chapter presents the implementation of mobility and basic link monitoring found to be adequate but yet extendable enough to fit a demonstration of the PIITSA project.

The chosen approach handles mobility at the application layer, and is in that way most similar to the second design alternative. A virtual network interface is used to provide a virtual static IP address for the applications, while UDP tunnelling is used to send the packets over the actual interface.

8.1 Platform and programming language

The NSB Client and Server were both implemented on desktop computers, running Windows XP. The reason for this was that Windows XP contains some features that greatly facilitated development, for example the WMI interface. Also, the TAP device (see Section 8.3.1) does not run on the embedded versions of Windows. The possibility to port the NSB to a mobile device is considered in Section 9.4.2.

C# .NET was used as the main programming language. Some OS dependent code was written in C, such as the communication with the TAP device.

8.2 Overview

The NSB is composed of three major units (four on the server), as shown in Figure 18.

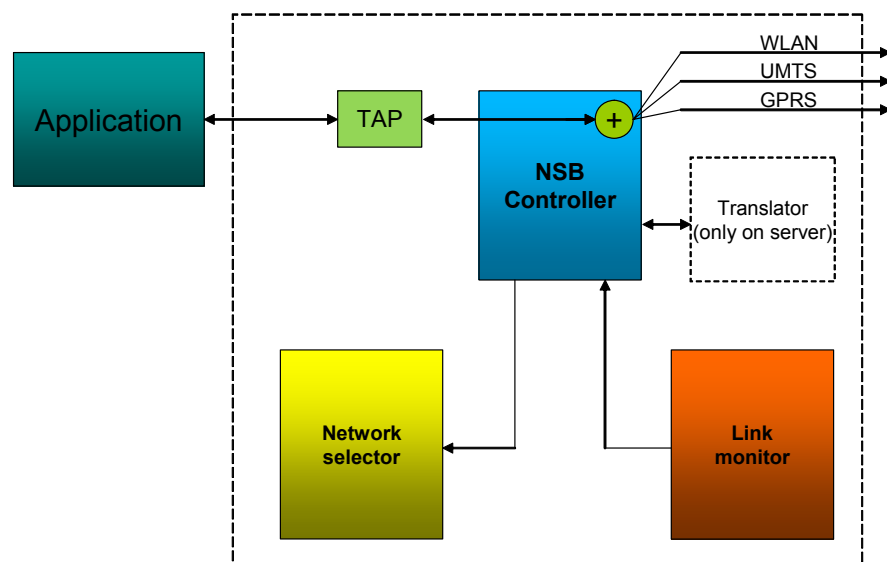


Figure 18: Overview of the NSB's blocks.

The *NSB Controller* is the main block that controls and communicates with the peripheral units. It is also responsible for sending and receiving data,

communicating with the virtual network interface (the TAP driver), and executing handovers.

The *Link Monitor* is an active unit that is responsible for monitoring all network connections, and to tell the NSB about changes in the network's availability. It also monitor and store information about the quality of each network.

The *Network Selector* is a passive unit that contains methods to determine the best network to use for the moment. It relies among other things on information gathered by the Link Monitor.

The *Translator* is a unit that resides only on the server. It keeps a database containing information about all connected clients and their available networks. It also handles virtual address allocation.

8.3 Mobility

To make handovers transparent to user applications, we must hide the inevitable change of IP addresses, as discussed earlier. To achieve this, a virtual network interface on both client and server is used. The IP addresses of these interfaces (here called virtual IP addresses) will stay constant, as explained in the following section.

8.3.1 The TAP device

The TAP device can be defined as a virtual network interface, and consists of a driver running in kernel mode. The TAP driver used in this implementation is from the open source VPN project OpenVPN , developed by James Yonan, which in turn used code from CIPE-Win32 by Damion K. Wilson.

In most respects, the TAP device works as any network interface. It will show up in Windows as just another network connection, to which an IP address may be assigned. Just as a network adapter, it also keeps a local cache of ARP addresses. By allocating a low metric to the TAP device, we ensure that any application will bind its sockets to the TAP, unless if they specifically states another network interface.

As with most drivers, you can read and write to the TAP as if it were a file. The read operation blocks execution until data from an application bound to the TAP is available. On the other hand, any socket bound to the TAP will receive all data written to it through a write operation. Thus, the following can be achieved to transfer packets between two TAPs:

1. Read on the TAP. When an application issues a `send()` on a socket bound to the TAP, our read operation will return data to user space, in this case the NSB.
2. Choose what network adapter to *really* use, and send data over this link (using UDP tunnelling, see below).
3. On the server side, data is received through a real network adapter. The data is immediately written to the TAP device.
4. The server application socket listening on the TAP will receive the data.

For applications, the NSB will be transparent. Figure 19 shows the Application – TAP – NSB relationship.

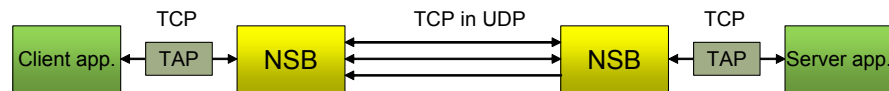


Figure 19: The TAP device catches all client and server application traffic and forwards it to user space, where NSB listens.

8.3.2 UDP tunnelling

Tunnelling is the process of encapsulating one protocol into another. Before sending the packets over Internet via the real network interfaces, the NSB encapsulates the packets (read from the TAP device) into an UDP packet. UDP has small overhead and requires no handshake procedure before transmission can start. If the packet coming from the application (the “real” packet) is an ordinary TCP/IP packet, this protocol will handle reliability. As far as the UDP protocol is concerned, the whole TCP/IP packet is payload, as shown in Figure 20.

On the receiver side, the UDP header is removed, leaving the payload, which happens to be another (TCP/IP) packet. Packets arriving this way should be destined to the TAP, and is therefore written to the device. All sockets bound to the TAP IP address will receive this data.

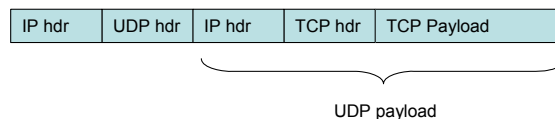


Figure 20: A TCP packet encapsulated in an UDP packet (Network – and Transport layer protocols displayed).

8.3.3 Example of operation

Let us imagine that the server application is a FTP server, while the client application is a FTP client. FTP software uses the File Transfer Protocol, which in turn uses TCP, to reliably transfer files. Furthermore, let’s assume the server TAP to has an IP address of 10.8.0.1, while the client TAP has 10.8.0.2. The client, which is a handheld device, has two real network adapters; a GPRS modem with address 13.2.4.44, and a WLAN card with address 155.10.10.5. The server has a public, static IP address of 189.5.5.3.

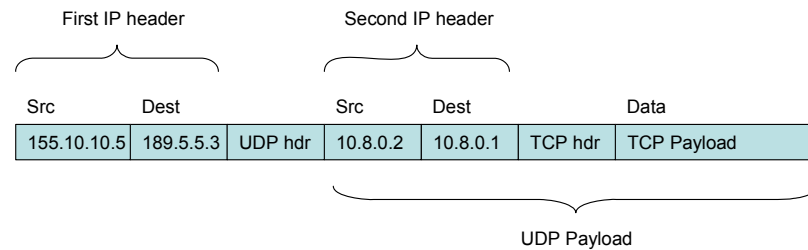


Figure 21: The UDP packet travelling over the Internet.

Now, the client requests to upload a file to the FTP server, which is accepted and a transfer starts. The first TCP packet is caught by the TAP and pushed to user space, as NSB has issued a blocking read on the driver. The NSB now has a complete TCP packet (with IP header and Ethernet frame) with a destination address of 10.8.0.1 and a source address of 10.8.0.2. It then encapsulates it into a UDP packet, and sends it over the best link (WLAN). The UDP packet has a destination address of 189.5.5.3 (the NSB knows about the server's real IP address) and a source address of 155.10.10.5 (the WLAN card). The destination port number is the NSB. See Figure 21.

The packet travels over the Internet and arrives at the server NSB at 155.10.10.5. The UDP header is dropped, and leaving the raw TCP packet. As the destination is 10.8.0.1, the servers' virtual network interface, the NSB decide to write the packet to the TAP device. The FTP server, bound to the TAP device, receives its first packet.

After a while, the user moves out of WLAN coverage and this connection is lost. Mechanisms at the client discover this, and from now on all packets are sent over the GPRS modem, still with UDP encapsulation but with a different source address (13.2.4.44). However, the *TCP packets* arriving at the FTP server still have a source address of 10.8.0.2, so the file transfer survives, as seen in Figure 22. Although some packets were lost before the client discovered the loss of WLAN coverage, TCP is used to handle lost packets and simply requests retransmission of these packets.

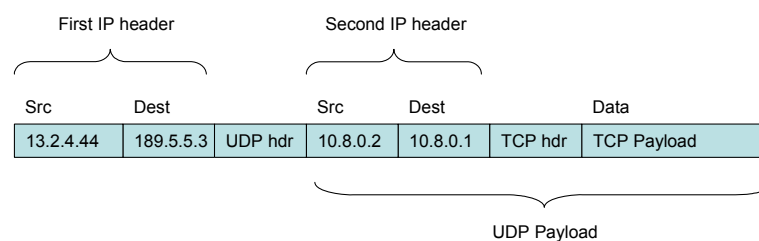


Figure 22: The TCP packet is unchanged even though the packet is sent on another network.

8.4 NSB Controller

The NSB Controller is the main part of the NSB. It communicates with the rest of the units, and can be viewed as the “executive unit”. It is responsible for sending

packets, receiving packets, send and react to control messages, and executing handovers

8.4.1 Sending and receiving data

The NSB uses blocking, synchronous send and receive operations, running in different threads. On the client, one socket is created for each network. All sockets are UDP sockets, and control messages are also sent through these sockets.

To assure that packets are sent on the intended interface, a route is created in the routing table for each added network, with the same metric value as the other networks. In this way, Windows will not route packets in an undesirable way.

Prior to sending each packets, the NSB server refers to the tables in the Translator and the `evaluateSend()` method in the network selector to find a good network to use. The NSB client uses a similar table and the `evaluateSend()` method. In this way, the NSB can be seen as a multiplexer, as seen in Figure 23.

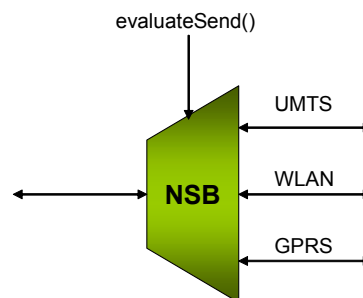


Figure 23: The NSB seen as a multiplexer.

8.4.2 Executing handovers

As the NSB makes a decision about what network to send on for each packet, as explained above, no “handovers” in the normal sense occurs. Still, if the only network goes down, another one must be connected. For this purpose, the NSB uses methods in the Network Selector to find the most suitable network to connect to.

Even if a backup network exists, it will take some time for the NSB to discover the loss of the first network. We may therefore speak of handovers when the NSB is *forced* to change which network to send over. The delays associated with such handovers are examined in the evaluation (Section 9.2.2).

8.5 The Translator and address allocation

The server must be able to route traffic to several clients, each with their own set of networks. To achieve this, the server implements a function called the Translator. The Translator is responsible for allocating virtual IP addresses to clients and to, for each client, keep a list with available networks that packets may be sent over.

8.5.1 Address allocation

When a client connects to the server NSB, the first thing it does is to request a virtual IP address that it may use for its TAP device. This is done through a control message. The Translator assigns a free IP address from an address pool and sends the address via another control message. In this sense the server operates much like a DHCP server, although the address lease times are generally infinite. The client will keep this address until it disconnects or crashes.

8.5.2 Translation

As soon as a client acquires a virtual IP, an entry for that client is added at the server Translator. The virtual IP is immediately associated with an initial network, which is the network that was used by the client to transfer the control message.

As new networks become available and go down, posts are added and deleted from the Translation table. After a while, the table could look something like Figure 24.

Virtual IP	Net IP	Net	Quality
10.8.0.3	153.22.4.6	WLAN	78
	45.0.0.233	GPRS	13
	122.43.32.2	DAB	24
10.8.0.4	83.222.11.5	GPRS	7
10.8.0.5	74.99.99.2	UMTS	56
	77.44.3.4	WLAN	96

Figure 24: The server translation table.

This lets the server choose what network to use when communicating with the client. For example, if the server wants to send a packet to 10.8.0.5, it could use WLAN, and would then create an UDP packet with a destination address of 77.44.3.4. The Quality column actually corresponds to collections of data gained from measurements done by the Link Monitor.

The importance of keeping the translation table updated is obvious. This is achieved by a combination of control messages and through constant link monitoring, as explained in the following sections. UDP itself has no way of telling if the remote endpoint is still connected, as it does not implement reliability. It is therefore inevitable that during a handover some packets are sent to a disconnected network interface before the disconnection is discovered. As explained earlier, this seldom presents any problem given the link monitor discovers the new condition within a reasonable time.

8.6 Link monitoring

The link monitor is an active unit, i.e. it runs in a separate thread. In addition to checking if a network is down, it also registers Round Trip Time (RTT) and packet loss for each network. To this end, bouncing control message is used. The

client also uses the Windows Management Instrumentation (WMI) service to detect loss of network connection.

8.6.1 Monitoring through control messages

Both client and server use control messages in order to constantly be updated about the network situation. These control messages works pretty much like a normal “ping”. A control packet (NETSTAT) is time stamped by the sender (which could be either client or server) and sent to the receiver, who answers with an ACKNETSTAT message. When the sender receives this message, the timestamp is compared with the current time, thus measuring RTT. If no ACKNETSTAT message has been received for a predefined period of time, it is considered lost, and the network quality level is degraded.

In addition, the NETSTAT and ACKNETSTAT packets contains information about how many packets that were sent by the sender and received at the receiver side in the last time period. In this way, packet loss may be measured, which is essential information about a link’s quality.

8.6.2 WMI monitoring

By using Windows Management Instrumentation, system information may be retrieved and edited by calling an internal Windows database, either locally or remotely. Every network interface has a separate entry in this database, and this fact is used by the NSB to detect changes in the network connection status. This service is only used by the client, where network connections are expected to go up and down periodically.

If, for example, a network connection is disabled by the user, the server would normally have to wait for some missing NETSTAT messages before deciding that this network was really not connected. Instead, if there is another network available, the client sends this information to the server as soon as it is discovered through the use of WMI. This usually only takes about one second.

The client performance is also increased through the use of WMI. Some network problems (like the loss of an IP address which happens for example when a modem is disconnected) would normally be discovered first when the client tries to send a packet. By discovering these abnormalities earlier, connection attempts and so on can be done before the network is really needed.

8.7 Control messages

In the previous two sections, the NSB’s dependency on control messages has been highlighted. Table 2 presents a list of the control messages with a short explanation for each one of them.

Name	To – From	Explanation
REGISTER	Client – Server	Register with server and get a virtual IP.
ACKREGISTER	Server – Client	Answer to register request, contains virtual IP.
ADDNET	Client – Server	A new network is available.
REMOVENET	Client – Server	A network was lost.
IAMALIVE	Client – Server	Client is up and wants to keep his virtual IP.
BYE	Client – Server	Client logging off, release virtual IP.
NETSTAT	Both ways	Network statistics sent by the monitor.
ACKNETSTAT	Both ways	Answer on NETSTAT message.
APPPREF	Application – Both	Message for controlling NSB settings from applications.

Table 2: The NSB control messages.

The control messages use the same UDP sockets as other traffic (except APPREF, which uses a separate port). This means that the NSB itself must handle reliability for critical control messages.

The two control messages REGISTER and ACKREGISTER are used for address allocation. The client sends a REGISTER request together with information about his first network. The server answers with an ACKREGISTER message that contains the virtual IP the client will use, and creates a post in its Translator table.

ADDNET and REMOVENET are sent by the client when a new network is connected or when a connection is lost. ADDNET contains the virtual IP of the client, and information about the new network. It is important that this message reaches the server, as the server can't use a network that is not registered with its Translator. The REMOVENET message is not that critical, as the server's monitor will discover a network loss anyway within time. It is sent to speed up the server's adoption to the new situation.

IAMALIVE is used by the client to hold on to a virtual IP address. The server expects all clients to send this message periodically, otherwise it assumes that the client has crashed without notification and releases the client's virtual IP so that others can use it. When the client NSB shuts down normally, it sends a BYE message to tell the server that it does not need its virtual IP anymore.

The NETSTAT and ACKNETSTAT messages are used for link monitoring, and were explained in detail in Section 8.6.

The application communicates with the NSB through APPREF messages. They contain information about what kind of service the application is currently using (Streaming, Messaging or Interactive) and if automatic network selection should be turned on. This information is used by the network selector to find the most suitable network to use.

8.8 Network selection

The roaming strategy that decides what network to connect to and what network to send data over is being developed by BTH, and is yet to be integrated into the

prototype NSB. So far, the tentative NSB uses a simple, passive interface with two methods, `evaluateConnect()` and `evaluateSend()`.

On the client side, if no network connection exists, the NSB tries to connect to the first connection in the list. If a connection attempt fails, a back off timer forces the NSB to wait for a while before trying again. The server side normally never needs to connect to a network, so `evaluateConnect()` is not used.

On both client and server, `evaluateSend()` is called every time a packet is about to be sent. This makes the NSB adapt to new conditions as quickly as is possible. The prototype evaluations works in the following way:

For each service the applications is using (Messaging, Streaming, Interactive), each network has a priority. For example, the Streaming service may be associated with the following list of priorities: WLAN (1), UMTS (2), and GPRS (3). Then, network availability is checked by looking at link monitoring history. We may find that WLAN is not connected at all, and that UMTS recently has failed to answer some NETSTAT control messages. Now, WLAN is out of the question, and there is a risk that the user recently moved out of UMTS coverage. Therefore, the `evaluateSend()` method will decide that GPRS is currently the best link to send packets over, albeit it initially had the lowest priority.

The importance of fine tuning both the link monitor and network selector becomes obvious. The NSB should not try to send a lot of packets on a disconnected network, but it should not assume a network to be unusable too early either. Occasional large delays can occur on any network. As seen in the evaluation, this fine tuning is important to get as short handover delays as possible.

8.9 Security

The NSB model does not implement any security, in accordance with the previous discussion regarding this issue. However, the TAP driver is taken from the OpenVPN project, so implementing a user space security protocol, like Secure Socket Layers (SSL), should not present any greater difficulty. OpenVPN itself uses OpenSSL, which is an open source implementation of SSL. Applying a security protocol always adds overhead, which in turn reduces performance, and this should be taken in consideration when deciding where and when to add encryption.

9 Evaluation

This chapter presents the evaluation methods and the results from the NSB prototype evaluation. A discussion is presented, as well as ideas for improvements and future work.

9.1 The test-bed

The purpose of the test-bed was to provide the means to perform measurements on and evaluate networks in a realistic environment. The complete test-bed is shown in Figure 25.

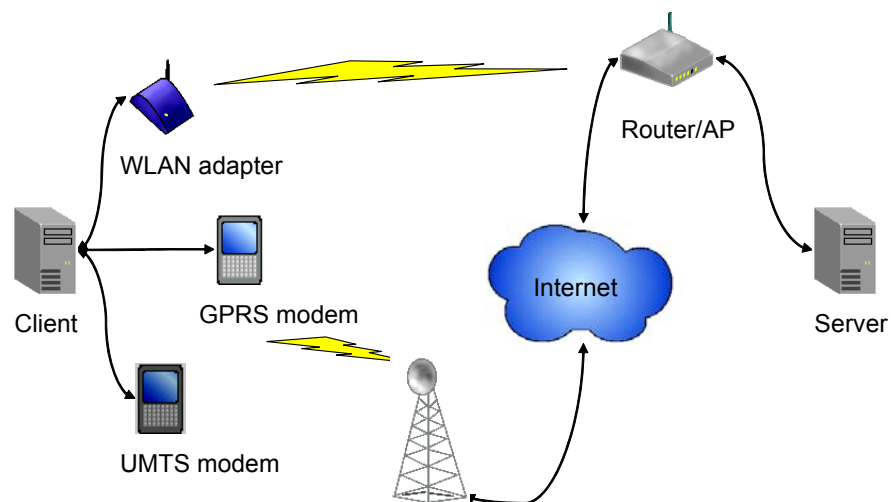


Figure 25: The experimental test-bed.

The desktop computer operating as the server was an Intel Pentium III, 860 MHz unit with 256 MB of RAM memory, and another unit with identical specification was used as client. Both machines ran Windows XP SP 2. The server used its internal 3Com 3C920 Fast Ethernet network adapter to connect to the router (see below).

The WLAN connection consisted of a NetGear MA 101 wireless USB adapter, with a NetGear MR 314 11 Mbps wireless router acting as access point. As the router used NAT, port forwarding was used to make the server accessible for the clients.

The GPRS connection used a Tele2/Comviq subscription, with a SonyEricsson T630 cellular phone acting as modem. The phone was connected to the client via a Belkin Bluetooth v1.1 USB adapter.

The UMTS link utilized a Vodafone Mobile Connect 3G/GPRS PC-card, and was connected via a Quatech PC Card reader to a PCI slot. GPRS roaming was disabled through AT commands.

As a packet analyzer, the open source software Ethereal v0.10.13 was used. To test file transfers, GoldenFTP Server was used, together with GlobalSCAPE's CuteFTP Home FTP client.

A drawback with the test-bed was the fact that since the WLAN network was actually a local network behind the server, the packets sent through the WLAN interface never needed to travel over the Internet. This yielded lower delays than what would have been expected in a real-world scenario.

9.2 Results

During the evaluation, performance (throughput), handover delays, and some other attributes were examined. Throughput and delays were tested during FTP file transfers, as a file transfer assures a bidirectional, constant flow of packets at the highest possible speed, thus putting the NSB under heavy stress.

The first attribute, performance, refers to the average upload and download speed in bytes. The middleware logic would inevitably have effect on transfer speeds, and the impact of this effect was examined.

The handover delays, that is, the time between the last packet is transmitted on one network until the first packet is transmitted on another network when connection to the first network is lost, can be categorized into the following types:

- *Handover with a backup network*, where two networks are available but one is lost or deselected due to various reasons.
- *Handover without backup network*, where only one networks is available and lost due to various reasons, meaning another network must first be connected.

For both types of handovers, *detection of loss of connectivity* is included in the stated delay. As mentioned before, detection was achieved through either WMI checkups, by the socket API reporting a send failed (on the client side), or absence of control message replies. The detection of the network connectivity loss often contributed to the major part of the delay.

In addition, typical RTT and packet losses as reported by the NETSTAT control messages were examined, and the effects this had on the NSB performance.

9.2.1 File transfer performance

Table 3 present the results from measurements on file transfer performance, with and without the NSB. The file used for the experiment was a 3.04 MB Windows executable. The file was transferred 10 times on each network, in each direction.

Network	NSB speed, UL	NSB speed, DL	Speed w/o NSB, UL	Speed w/o NSB, DL
GPRS	2.5 KB/s	4.7 KB/s	2.8 KB/s	5 KB/s
UMTS	12.5 KB/s	16.7 KB/s	15 KB/s	21 KB/s
WLAN	197 KB/s	219 KB/s	306 KB/s	338 KB/s

Table 3: Average NSB upload (UL) and download (DL) speeds compared to speeds without NSB.

In the case of WLAN, the fastest network, the NSB reduced the speed by 35 % for both uploads and downloads.

For UMTS, the download speed were reduced by about 20 %, and only 17 % for uploads.

For GPRS, download speed were reduced by 6 %, and upload speed reduced by 11 %.

We observe that performance is affected greatly for fast networks, but not very much for slow networks. This implies that the NSB has problems to simply deal with the large number of bytes transmitted on fast networks. The possible reasons for this are discussed in Section 9.3.

9.2.2 Handover delays

The following sections present measurements done of handover delay during file transfers. To reduce the number of tests, handovers were tested between WLAN and only one of the connections, namely UMTS. The delays should not be different when switching to GPRS, as dialling time virtually is the same for both.

The handover delays are measured at the client side during an upload, and at the server side during a download. This means that the delays are always measured at the sender side.

To initiate a handover from WLAN, the network was either manually unplugged (at the WLAN adapter), disabled through the “Windows Network Connections” panel, or by actively changing the NSB settings through the control port.

To initiate a handover from UMTS, the network was disconnected from the Windows “Status” panel, or by actively changing the NSB setting through the control port.

Each type of measurement on the handovers was tested 10 times.

9.2.2.1 Handovers with backup network

During the handovers with a backup network, the NSB had access to both a WLAN network and an UMTS connection. Table 4 presents the delays for a handover during an upload, that is, when the client is sending packets.

Networks	Reason for handover	Handover delay, avg.	Handover delay, min. – max.
WLAN to UMTS	Unplugged	1.1 s	0.2 - 2.0 s
WLAN to UMTS	Disabled	1.1 s	0.2 – 2.1 s
WLAN to UMTS	Selected	0.0 s	0.0 s
UMTS to WLAN	Disconnected	1.3 s	0.1 – 2.7 s
UMTS to WLAN	Selected	0.0 s	0.0 s

Table 4: Handover latency at client side during upload, using a backup network.

Respectively, Table 5 presents delays during a file download.

Networks	Reason for handover	Handover delay, avg.	Handover delay, min. – max.
WLAN to UMTS	Unplugged	1.1 s	0.4 – 2.1 s
WLAN to UMTS	Disabled	1.1 s	0.3 – 2.1 s
WLAN to UMTS	Selected	0.0 s	0.0 s
UMTS to WLAN	Disconnected	1.4 s	0.1 – 2.8 s
UMTS to WLAN	Selected	0.0 s	0.0 s

Table 5: Handover latency at server side during download, using a backup network.

The time to detect a loss of connection is normally just a bit longer for a download than for an upload. The reason is that the client can detect loss of connection through the operating system or socket API and act immediately, while the server must wait either for a REMOVENET message sent over the operational network that tells the server about the connection loss, or for an unacknowledged NETSTAT message. Of course, the file transfer cannot resume full speed until both sides uses the new network, as the sender side also expects acknowledgements.

The delays times also vary considerably. This depends on how the client discovers the network loss; either through WMI or by trying to send a packet over a lost network (then getting an exception from the Socket API). Generally, the socket exception arrives earlier than the WMI notification during file transfers.

9.2.2.2 Handovers without backup network

When initiating a handover without any backup, only one network was available. This means that the NSB first had to detect the network loss, then make a decision regarding what network to connect to, and finally execute the connection. When switching to the modem, *the time for dialling* is included in the delay. When switching to WLAN, it is assumed that the computer is under the coverage of an unprotected WLAN network, but that the NSB has not yet registered this network. Table 6 shows the delays during file upload.

Networks	Reason for handover	Handover delay, avg.	Handover delay, min. – max.
WLAN to UMTS	Unplugged	10.8 s	7.5 – 14.6 s
WLAN to UMTS	Disabled	11.2 s	8.3 – 14.7 s
UMTS to WLAN	Disconnected	4.2 s	3.9 – 4.5 s

Table 6: Handover latency at client side during upload, using no backup network.

Table 7 presents the delays for the download case.

Networks	Reason for handover	Handover delay, avg.	Handover delay, min. – max.
WLAN to UMTS	Unplugged	12.5 s	11.1 – 14.8 s
WLAN to UMTS	Disabled	12.8 s	11.5 – 15.3 s
UMTS to WLAN	Disconnected	5.0 s	4.6 – 5.4 s

Table 7: Handover latency at server side during download, using no backup network.

Compared to when a backup network existed, delays are pretty long. The longer delays partly arise from the time it takes to dial the UMTS connection.

Another reason is that during a handover without a backup network, the client is unable to send a REMOVENET control message to the server over another network. The server must therefore detect a network disconnection through unacknowledged NETSTAT messages. As mentioned, the delay times are greatly influenced by the network selector settings, that is, how many NETSTATs that could be missed before the network is considered unusable, and how often the NETSTAT messages are sent.

During the tests, NETSTAT messages was sent each 1.5 seconds, and the link was considered unusable if 5 consecutive messages were missed. Most of the times, the ADDNET control message sent when a new network is connected will arrive before this time period expires, and as the new link has not missed any NETSTAT messages, the server will consider this network to be better and switch to it long before actually removing the old (disconnected) network.

9.2.3 Results from NETSTAT control messages

The NETSTAT control messages reports Round Trip Time, packet loss and unacknowledged NETSTAT messages to the NSB. As many functions in the NSB are directly dependent this information, this section presents average NETSTAT values. The measurements, presented in Table 8 and Table 9, were conducted at NETSTATs sent from the server side.

Network	RTT	Packet loss
WLAN	0 ms	< 1 %
UMTS	215 ms	< 1 %
GPRS	870 ms	< 1 %

Table 8: Average statistics as reported by NETSTAT (measured on 50 messages) during general Internet browsing.

Network	RTT	Packet loss
WLAN	5 ms	< 1 %
UMTS	543 ms	< 1 %
GPRS	4121 ms	< 1 %

Table 9: Average statistics as reported by NETSTAT (measured on 50 messages) during file downloads.

The Round Trip Times deteriorates considerably when transferring a file. Especially interesting to note is the astronomical RTT for the GPRS link. With so long delays, it becomes much harder for the server side NSB to detect when the GPRS link is actually down, as it forces the NSB to be more patient with the GPRS link. This in turn leads to that if the GPRS link really goes down, it will take a long time before the NSB (server side) detects this.

In an attempt overcome this problem, the Differentiated Services field (formerly the Type Of Service field) in the IP header was modified to give higher priority to NETSTAT packets than normal traffic. This only led to a very small improvement of RTT. Furthermore, Windows XP do not let users change the Differentiated Services field without adding a new post to the registry. Access to the registry is often restricted to an administrator, so this solution was not considered viable.

9.2.4 Memory consumption

The NSB Server consumes about 17 MB of memory, and as it will most likely run on a server or desktop computer, memory usage is not really an issue. The NSB Client uses 22 MB of memory, which may present a problem on some smartphones. Lower memory consumption should be possible to achieve if the code is rewritten more carefully with the .NET compact framework

9.2.5 WMI performance

WMI is used on the client side to quickly detect when a network connection goes down or is disabled. WMI runs as a service and is actually a (remote) connection to a database containing information about among other things machine hardware. This makes access to WMI information relatively slow. Checking for connectivity on a network adapter takes approximately one second. In addition, calling WMI is CPU heavy and it slows down file transfer speeds greatly if done too often.

9.3 Discussion

From the performance tests, we see that the NSB reduces file transfer speeds dramatically for fast networks, by 35 percent for WLAN. The bottleneck has not been identified, but tests conducted on OpenVPN from where the TAP driver originates [31] show similar results, which may suggest that the problem lies within the driver.

Initially, it was believed that the network evaluation done before sending each packet was a bottleneck, but this possibility was eliminated after crosschecking against transfer speeds without network evaluation. As the packet size during file transfers was around the MTU (about 1500 bytes), the 20 byte UDP header only stands for 1.3 % of extra overhead, and hence this was not the problem either.

When using the WLAN, considerable fragmentation was detected due to the extra headers added, making the packet larger than the WLAN card's MTU. This certainly contributed to the reduced performance.

If we take into consideration that the main application of the NSB will be for travel information services, where messages are typically small, the reduced performance should not present a major problem. On the other hand, the UDP header's relative size to the transmitted data becomes larger.

It is hard to make the handover delays shorter than they are. It has been shown that connectivity loss detection often corresponds to the major part of the delay. Sending control messages more often could reduce detection time in some cases, but it would also have the effect that networks could incorrectly be considered bad or down too early.

The handover delays can be compared to other solutions. [9] presents measurements on handovers between GPRS and WLAN using Dynamics Mobile IP from HUT. Assuming that both WLAN and GPRS connected (as in the first delay measurements in this thesis), the handover delays were 0.5 - 1 s from GPRS to WLAN, and about 3 s for the reversed case.

In addition, [24] accounts for measurements with the same Mobile IP solution when switching between two WLAN base stations. The average time for this was found to be 4.8 s.

Measurements were also conducted on the RMS socket solution. The average handover latency, when switching between two WLANs by disabling one connection took about 0.7 s. If instead one access point was powered off (an experiment this thesis did not examine), the handover latency rose to about 12 s [17].

It should be noted that if TCP is used, a handover introduces more performance effects than just the actual delay time. To avoid congesting a network, TCP adjusts the data rates to a pace at which packets have been acknowledged by the receiver. This is called flow control, and it means that after some seconds without any network connection, or when switching from a slow to a fast network, it will take some time until the new link operates at full capacity.

As explained above, the use of WMI slows down data rates and is CPU heavy (and therefore also battery consuming). Even though loss of network connectivity is discovered fast, it is doubtful whether the extra seconds it takes for normal monitoring to report network unavailability are so bad that the use of WMI can be justified.

To conclude, the main objectives of this thesis, being the design, implementation and evaluation of a transparent function implementing mobility for wireless networks, were reached. Even though further development could do much to improve the application (see the following section), the tentative model properly shows the main concepts of how to achieve network mobility. The application is also extendable as each function, such as the link monitor and network selector, is isolated from the rest of the NSB. This is important if it is to be used for further research in network evaluation.

9.4 Future work

There were some development that had to be omitted from this thesis, mainly because of lack of time. Some ideas for future development is presented here.

9.4.1 Improvements

There are many things that the author of this thesis would have liked to implement. Primarily, in order to have full control and insight into the TAP driver, it would be preferable to develop a driver from scratch, custom-made for the NSB. In this way, perhaps it would also be easier to find why the NSB slows down the networks.

An interesting thing to try is to implement packet buffers in the NSB. Currently, packets that the application tries to send when no network is available, or during a handover, are lost. Perhaps performance would be increased if instead they were placed in a queue that was flushed when a new network were available.

A bigger project would be to enable server farming with load balancing, that is, letting multiple machines act as NSBs and take care of some clients each. This would be essential if we expect thousands of clients to connect to the NSB.

Yet another useful feature would be the possibility, as on Columbitech's Wireless VPN, to bridge Ethernet networks via the NSB. This could be useful for corporations when their staff is working out of office and connected via a wireless link. If this feature is implemented it would be interesting to enable another server

to operate as a DHCP server and allocate addresses to clients both on the corporate intranet *and* to clients connecting via the NSB.

9.4.2 Porting

For the moment, the NSB has been tested on Windows XP. The limiting factor is the TAP device (that only runs Windows XP/2000) and some of the WMI classes that are only implemented on Windows XP.

Porting to a handheld, or at least a tablet PC, is a tempting prospect for demonstration purposes. Modifications would have to be done to the TAP device, and the dependency of WMI would have to be reduced.

10 References

- [1] ABI Research, <http://www.abiresearch.com/home.jsp>, 2005-10-20
- [2] Agilent Technology, *Understanding General Packet Radio Service*, <http://literature.agilent.com/litweb/pdf/5988-2598EN.pdf>, 2005-11-07
- [3] Baily, Stephen (1999), *A technical overview of digital radio*, BBC Research and development
- [4] Birdstep, *Introducing Birdstep Intelligent Mobile IP Client v2.0 Universal Edition*, Birdstep Technology ASA
- [5] Cisco Systems, Inc., *Overview of GSM, GPRS, and UMTS*, http://www.cisco.com/univercd/cc/td/doc/product/wireless/moblwrls/cmxx/mmg_sg/cmxxgsm.htm, 2005-11-07
- [6] Cisco Systems, Mobile IP support for RFC 3519 NAT Traversal http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123newft/123t/123t_8/gtnatmip.htm, 2005-11-29
- [7] Dynamics Mobile IP at Sourceforge.net, <http://dynamics.sourceforge.net/>, 2005-11-30
- [8] Forouzan, A. Berhouz (2004), *Data communications and networking*, 3rd edition, McGraw-Hill
- [9] Graf, Marcus & Mörstam, Mattias (2002), *Sömlös övergång mellan GPRS och WLAN*, Lunds Tekniska Högskola
- [10] GSM World, *What is GPRS?*, <http://www.gsmworld.com/technology/gprs/intro.shtml>, 2005-11-07
- [11] Hernadi, Alexandra (2005), *Bara 7 000 hör på dab-radio*, SvD 2005-05-15
- [12] HP, *HP produkter & tjänster*, <http://welcome.hp.com/country/se/sv/prodserv.html>, 2005-10-27
- [13] Isaksson, Lennart, et al. (2005-09-27), *Roaming Strategy for PIITSA WP 2.3*, Blekinge Institute of Technology
- [14] ITS America, <http://www.itsa.org/whatits.html>, 2005-11-01
- [15] ITS Canada, <http://www.itscanada.ca/english/aboutits.htm>, 2005-11-01
- [16] ITS on Wikipedia, http://en.wikipedia.org/wiki/Intelligent_Transportation_System, 2005-11-01
- [17] Kristiansson, Johan (2004), *Licentiate Thesis: Creating Always-Best-Connected Multimedia Applications for the 4th Generation Wireless Systems*, Luleå University of Technology

- [18] Levkowetz, H. (2003), *Mobile IP Traversal of Network Address Translation (NAT) Devices*, RFC 3519
- [19] Lindberg, Peter (2004), *PIITSA:0006A Övergripande projektbeskrivning*, SAAB Communication
- [20] Linux Mobile IP at Stanford University, <http://mosquitonet.stanford.edu/mip/>, 2005-12-02
- [21] M. Fiedler, S. Chevul, L. Isaksson, P. Lindberg, and J. Karlsson (2005), *Generic Communication Requirements of ITS-Related Mobile Services as Basis for Seamless Communication*, Next Generation Internet Works, 2005, 18-20 April 2005 Pages 426-433
- [22] Marton, Maria & Jenstav, Marika (2003), *Översyn av ITS – FoU 1997-2002*, Vägverket Sektion ITS
- [23] Microsoft Mobile Developer Center, <http://msdn.microsoft.com/mobility/windowsmobile/>, 2005-10-21
- [24] Mola, Guilo (2004), *Interactions of Vertical Handoffs with 802.11b wireless LANs: Handoff Policy*, <ftp://ftp.it.kth.se/Reports/DEGREE-PROJECT-REPORTS/040303-Mola-final-with-cover.pdf>, 2006-05-10
- [25] MsMobiles, Review of HP iPAQ Mobile Messenger hw6515, <http://msmobiles.com/news.php/4011.html>, 2005-10-27
- [26] My-Symbian.com, <http://my-symbian.com/main/index.php>, 2005-10-20
- [27] Näslund, Erik (2004), *Associating DAB and GPRS to provide an asymmetric communication network*, University of Linköping
- [28] Nokia, *Nokia N91*, <http://www.nokia.se/phones/n91>, 2005-10-27
- [29] NVDB, <http://www3.vv.se/nvdb/index.asp>, 2005-11-04
- [30] Open Mobile Alliance, <http://www.openmobilealliance.org/>, 2005-11-18
- [31] *Open VPN on Windows notes*, <http://openvpn.net/INSTALL-win32.html>, 2006-04-02
- [32] Perkins, C. (2002), *IP Mobility Support for IPv4*, RFC 3344
- [33] Pocket Info, <http://www.pocketinfo.nl/>, 2005-10-21
- [34] Post – och telestyrelsen, *Faktablad 3G i Sverige*, <http://www.pts.se/Archive/Documents/SE/3G%20i%20Sverige.pdf>, 2005-11-08
- [35] Qtek, *Qtek 9000*, <http://www.myqtek.se/sweden/produkter/9000.aspx>, 2005-10-27
- [36] Qu, Xun, et al. (1997), *A Mobile TCP Socket*, Joint Computer Science Technical Report Series, The Australian National University

- [37] Radio-Electronics.com, *Introduction to UMTS/WCDMA*, http://www.radio-electronics.com/info/cellulartelecomms/umts/umts_wcdma_radio.php, 2005-11-07
- [38] Riegel, M. & Tuexen, M., *Mobile SCTP, IETF Internet Draft*, August 2002
- [39] S. Chevul, J. Karlsson, L. Isaksson, M. Fiedler, P. Lindberg, and L. Sandén (2005), *Measurements of Application-Perceived Throughput in DAB, GPRS, UMTS and WLAN Environments*, In Proceedings of RVK, June, 2005, Linköping, Sweden
- [40] Scourias, John, *Overview of the Global System for Mobile Communications*, <http://ccnga.uwaterloo.ca/~jscouria/GSM/gsmreport.html>, 2005-11-07
- [41] Series 80 Developer Platform 2.0 SDK for Symbian OS – For MIDP – User’s Guide, Forum Nokia 2005
- [42] Sony Ericsson, *Sony Ericsson unveil UMTS P990 Smartphone*, http://www.sonyericsson.com/spg.jsp?cc=global&lc=en&ver=4001&template=pc3_1_1&zone=pc&lm=pc3&prid=3982, 2005-10-27
- [43] Svergies Radio, *Digitalradion*, <http://www.sr.se/cgi-bin/mall/index.asp?programID=2205&nyheter=>, 2005-11-08
- [44] Teracom AB, <http://www.teracom.se>, 2005-11-08
- [45] The TCP/IP Guide, <http://www.tcpipguide.com>, 2006-03-22
- [46] TMC Forum, http://www.tmcforum.com/en/about_tmc/what_is_tmc/, 2005-11-04
- [47] Trafiken.nu, <http://trafiken.nu>, 2005-11-02
- [48] Tutorial Reports, *Wireless LAN (WiFi) Tutorial*, <http://www.tutorial-reports.com/wireless/wlanwifi/>, 2005-11-10
- [49] UMTS World, *Overview of the Universal Mobile Telecommunication System*, <http://www.umtsworld.com/technology/overview.htm>, 2005-11-07
- [50] US Department of Transportation, *What is ITS?*, http://www.its.dot.gov/its_overview.htm, 2005-11-01
- [51] Vägverket (2003), *Personlig Pendlarinformation*, Vägverket sektion ITS
- [52] Vägverket (2005), *Nationell ITS strategi 2006-2009*, Vägverket
- [53] Vägverket, *Hur information samlas in*, http://www.vv.se/templates/page3_13698.aspx, 2005-11-03
- [54] Vägverket, *RDS-TMC*, http://www.vv.se/templates/page3_13606.aspx, 2005-11-04

- [55] *Whitepaper: Columbitech Wireless VPN™ Technical Description*, Columbitech (2004)
- [56] *Whitepaper: Birdstep Intelligent Mobile IP Client v2.0, Universal Edition*, Birdstep Technology ASA (2002)
- [57] *Whitepaper: IP Mobility versus Session Mobility*, Columbitech (2001)
- [58] *Whitepaper: Wireless LANs: The essentials for saving your sanity*, Netmotion Wireless, Inc. (2005)
- [59] Wikipedia, <http://en.wikipedia.org>, 2005-10-20
- [60] World DAB, *About DAB*, <http://www.worlddab.org/about.aspx>, 2005-11-08
- [61] Yuan, Michael Juanto, *Let the mobile games begin*, www.javaworld.com, 2005-10-25

11 Abbreviations and acronyms

3G	Third Generation
AAA	Authentication, Authorization and Accounting
ACN	Automated Collision Notification
AHP	Analytical Hierarchical Process
AP	Access Point
API	Application Programming Interface
AVL	Automated Vehicle Location
BSC	Base Station Controller
BSS	Basic Service Set
BT	Bluetooth
BTS	Base Transceiver Station
CDC	Connected Device Configuration
CDMA	Code Division Multiple Access
CLDC	Connected Limited Device Configuration
CN	Core Network
COFDM	Coded Orthogonal Frequency Division Multiplex
CPU	Central Processing Unit
DAB	Digital Audio Broadcasting
DES	Data Encryption Standard
DHCP	Dynamic Host Configuration Protocol
DMZ	Demilitarized Zone
EDGE	Enhanced Data rates for GSM Evolution
ESS	Extended Service Set
FA	Foreign Agent
FDMA	Frequency Division Multiple Access
FTP	File Transfer Protocol
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
GPS	Global Positioning System
GRE	Generic Routing Encapsulation
GSM	Global System for Mobile communications
GUI	Graphical User Interface
HA	Home Agent
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HUT	Helsinki University of Technology
IC	Interaction Channel
ICMP	Internet Control Message Protocol
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force

IP	Internet Protocol
ISDN	Integrated Services Digital Network
ISM	Industrial, Scientific and Medical
ITS	Intelligent Transportation Systems and Services
JNI	Java Native Interface
JSR	Java Specification Requests
JVM	Java Virtual Machine
MIDP	Mobile Information Device Profile
MIP	Mobile IP
MN	Mobile Node
MSC	Mobile services Switching Center
NAT	Network Address Translation
NIC	Network Interface Card
NSB	Network Selection Box
NVDB	Nationell VägDataBas (National Road DataBase)
OS	Operating System
PC	Personal Computer
PDA	Personal Digital Assistant
PPP	Point to Point Protocol
PSK	Phase-Shift Keying
PSTN	Public Switched Telephone Network
PTS	Post – och Telestyrelsen
QoS	Quality of Service
QPSK	Quadrature Phase-Shift Keying
RADIUS	Remote Authentication Dial In User Service
RAM	Random Access Memory
RDS-TMC	Radio Data System – Traffic Message Channel
RFC	Request For Comment
RNC	Radio Network Controller
RTT	Round Trip Time
SD	Secure Digital
SDIO	Secure Digital Input Output
SDK	Standard Development Kit
SGSN	Serving GPRS Support Node
SIM	Subscriber Identity Module
SMS	Short Message Service
SRA	Swedish Road Administration
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TERN	Trans European Road Network
TSL	Transport Secure Layer

UDP	User Datagram Protocol
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
USB	Universal Serial Bus
USIM	Universal Subscriber Identity Module
UTRAN	UMTS Terrestrial Radio Access Network
VPN	Virtual Private Network
VS	(Microsoft) Visual Studio
WCDMA	Wideband Code Division Multiple Access
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Network
WMI	Windows Management Instrumentation
WPA	WiFi Protected Access
WTLS	Wireless Transport Layer Security
XML	Extensible Markup Language

