

Umeå University
March 2005
Department of Computing Science
Master Thesis, 20 Credits
Martin Barsk
dva00mbk@cs.umu.se

A seamless vertical handover system prototype

Abstract

In the future Internet users probably want to roam between networks that are using different connection technologies. Users probably want to roam between the networks without any interruptions. A user may want to communicate with the best connection currently available, but when a user changes location the best connection may be broken. Another available connection should then takeover the connection, a vertical handover. A handover is called seamless if it can be done without having to restart the running applications. Vertical seamless handover is studied in this thesis and a prototype for solving the problem is presented. Some other proposals for performing vertical seamless handovers are also discussed. The main challenge when performing a vertical seamless handover is to keep the communication session alive while changing physical connection to the Internet. According to the requirements the problem was analyzed and a solution with a home address and a home network for a mobile node was the most appropriate.

The prototype has been tested in a public environment with satisfying results. Roaming between the WLAN technology IEEE 802.11, 3G and GPRS has been tested with the prototype although other types of connections may also work.

Acknowledgements

Thank you Tomas Røjmyr, my external supervisor at TietoEnator in Ursviken for making this Master thesis possible. Also thanks to all other employees at TietoEnator in Ursviken who have been involved in the project, you know who you are. Thank you Robert Brännström, at Luleå University of Technology for helping me with the tests at Campus Skellefteå. Thank you my internal supervisor at Umeå University, Thomas Nilsson, for helping me with the report. I definitely don't want to thank the thieves who stole my equipment twice in my office at TietoEnator.

Contents

1	Introduction	11
1.1	Thesis outline	11
1.2	Background	11
1.3	Purpose and motivation	12
1.4	Method	12
2	Brief background of WLAN (IEEE 802.11)	12
3	Brief background of GPRS	13
4	Mobile IP Overview	15
4.1	Terminology	15
5	Handover terminology	17
6	Network layer approaches	18
6.1	Mobility support in IPv6	18
7	Transport layer approaches	19
7.1	SCTP	19
7.2	TCP MH	20
8	Upper-layer approaches	21
8.1	Migrate approach	21
8.2	MSOCKS	22
8.3	Universal seamless handoff architecture	24
9	Discussion	24
10	Implementation	24
10.1	Requirements	24
10.2	Analysis	25
10.3	System overview	25
10.4	Environment	26
11	Mobile Node	27
11.1	Configuration	27
11.2	Script	27
11.3	Architecture	27
11.4	Routing	28
11.5	Algorithms	29
11.6	NAPT	32
11.7	Dynamic interfaces	32
12	Home Agent	32
12.1	Configuration	33
12.2	Architecture	33
12.3	Algorithms	33
12.4	ARP terminology	34
12.5	Capturing the mobile nodes incoming packets	35
13	Tunneling	35
14	Protocol	36
15	Handover cases	37
16	Tests	40
17	Conclusions	40
17.1	Limitations	40
18	Future work	41
19	References	42
A	Abbreviations	44
B	User Guide	46

List of Figures

Figure 1: Wireless LAN overview	13
Figure 2: Protocol architecture	13
Figure 3: Protocol architecture	14
Figure 4: GPRS overview	15
Figure 5: Mobile IP overview.....	16
Figure 6: Registration of a mobile node via the FA or directly with the HA	17
Figure 7: Protocol architecture	20
Figure 8: Vertical handover procedure with mSCTP	20
Figure 9: TCP-MH message overview	21
Figure 10: The Migrate session layer framework.....	22
Figure 11: Message exchange diagram for connection establishment between a MSOCKS client and correspondent host via a MSOCKS proxy.	23
Figure 12: System overview	26
Figure 13: Class diagram for mobile node.	28
Figure 14: Routing rules.....	28
Figure 15: Routing table wlan	29
Figure 16: Routing table gprs	29
Figure 17: Main routing table.....	29
Figure 18: Sending Registration request algorithm.....	30
Figure 19: Receiving registration reply algorithm	31
Figure 20: Algorithm when the registration reply timer expires.	32
Figure 21: Class diagram for the home agent.....	33
Figure 22: Receiving registration request algorithm	34
Figure 23: Tunnel configuration.....	35
Figure 24: Packet encapsulation.....	36
Figure 25: Registration request	37
Figure 26: Registration reply.....	37
Figure 27: Packet exchange when performing a handover manually.....	38
Figure 28: Packet exchange when WLAN interface loses its connection and a handover is performed.....	39
Figure 29: Packet exchange when WLAN interface is down and can't send a registration request. Handover to GPRS interface is performed.....	39

1 Introduction

The step beyond 3G is probably a multi access solution. Different access network will be used to meet the optimum cost and performance and the requirement to be always connected. In the future Internet users probably want to roam between different network technologies without any interruptions. Already today there exists several wireless access networks like GSM, GPRS, 3G, Bluetooth and WLAN. The goal of the thesis is to evaluate and prototype a multi access solution where the connection is kept while moving from one access type to another. Existing wireless network technologies can be divided into two categories: those that provide a low-bandwidth service over a wide area and those that provide a high bandwidth service over a narrow geographic area. Neither technology makes it possible to always be connected to a high-bandwidth and a wide-area connection. It would be interesting for a user with multiple wireless network interfaces to always use the best current wireless technology available. The user should be able to decide what is best for the moment and may prioritize different things like, bandwidth, battery consumption, security requirements or cost. The idea is that a mobile user should be able to switch from one network access to another depending on what's available and what needs the user currently has. The mobile user should always be able to choose the best fitting connection without having to restart all running applications. Example:

You are downloading a file from Internet to your laptop (mobile node) using a WLAN connection. The WLAN interface has received an IP address from a DHCP server [2] in the hotspot area you currently are located in. You are moving outside the WLAN coverage physically and lose the connection to the WLAN interface. Now you want your GPRS telephone to continue downloading the file without any interruption and active user actions. The GPRS connection has received a different IP address from a GGSN. After a while you return to the WLAN hotspot and your WLAN interface receives a new IP address. The download should now switch interface to the WLAN since it has better bandwidth than the GPRS connection. These handovers should do done automatically without any user actions.

1.1 Thesis outline

In the opening chapters overviews of different connection types can be read. Also the central part mobile IP and handoff terminology will be considered. In chapters 6 to 8 different techniques for vertical handovers are studied. Chapter 9 summarizes and discusses the in-depth study done in chapters 6 to 8. The design and implementation of the prototype can be read in chapter 10 to 14. Chapter 15 covers the different handover cases and chapter 16 describes how the tests were done. Finally chapter 17 and 18 covers the conclusions and future work of this Master Thesis.

1.2 Background

As mentioned the step beyond 3G is probably a multi access solution. Users would probably from time to time also like to connect to fixed broadband networks, WLANs and technologies such as Bluetooth associated to e.g. cable TV and DSL access points. Users also probably want to roam between different access networks with minimal user actions. The IETF Mobile IP standard was created for this but has many limitations. The main idea with Mobile IP is that a mobile node always should be reachable through a home IP address regardless of its point of attachment. Unfortunately today's Mobile IP doesn't support that a mobile node is registered to

the home network through multiple network interfaces at the same time. Another important limitation is that mobile IP doesn't support session continuity. Session continuity means that a user should be able to switch network interface (access technology) to the home network without having to restart the running applications. Switching access technology from one to another is called a vertical handover. If a handover can be done without having to restart the applications it's called a seamless handover.

1.3 Purpose and motivation

The goal of the thesis is to study different vertical handoff techniques and to implement an application prototype. The prototype application will run on a mobile node, a laptop, with a WLAN and a GPRS connection to the Internet. The user should be able to switch between the different connections without having to restart the running applications. The prototype application should work with existing applications and without any extensions to the applications. The prototype should work with no modifications in the infrastructure. So the application should be application and infrastructure transparent. The prototype should primarily manage to handle soft handovers and optional hard handovers.

The mobile node should always be reachable from a home IP address regardless of its current point of attachment or connection type(s). All communication sessions that the mobile node initiates will be done with the home address as source address.

1.4 Method

The work started with a requirement specification that were quite unspecific so the initial weeks were used to try out what could be done and what couldn't. Almost the whole implementation part of the project has been of experimental characteristics. Since it was hard to specify how this seamless vertical handoff problem would be solved the implementation part didn't get so extensive. The work in the beginning was also of mixed characteristics, literature studying and experimental implementation. However at the end a working implementation part was finished.

2 Brief background of WLAN (IEEE 802.11)

Wireless local area networks (WLAN) provides a high bandwidth service over a narrow geographic area and are typically restricted in their diameter to buildings, a campus or single rooms. The IEEE standard 802.11[12] specifies the most common family of WLANs. There are two types of WLANs, infrastructure and ad-hoc. Infrastructure networks often provide access to other networks such as Internet. Communication typically only takes place between the wireless nodes and an access point (AP). The stations and the access point that are within the same radio coverage form a basic service set (BSS). Several BSS:es may form one logical wireless network called extended service set (ESS) and is identified by a name (ESSID). So it is possible to reach Internet through a WLAN with a wireless node where the node is located within the radio coverage for the WLAN. An overview of an infrastructure WLAN (IEEE 802.11) that is bridged to the Internet is shown in figure 1. Ad-hoc networks don't need any infrastructure to work. Each wireless node can communicate directly with other nodes, so no access points are needed. The complexity of each node in an ad-hoc network is much higher than in an infrastructure. The IEEE 802.11 standard defines the physical and medium access control (MAC) layer. The 802.11

link layer is transparent to the IP layer together with upper part of the link layer called logical link control (LLC) [20]. The LLC layer provides an interface to the IP layer and covers the differences of the medium access control layers needed for the different media. Figure 2 shows the protocol architecture from a wireless node via an access point to a wired node.

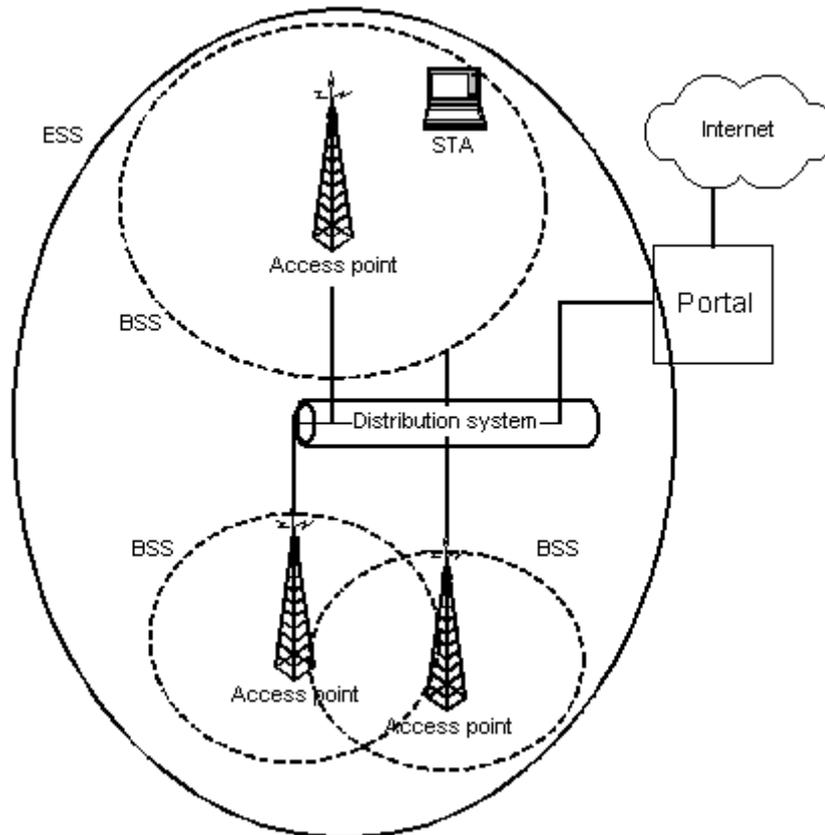


Figure 1: Wireless LAN (IEEE 802.11) overview

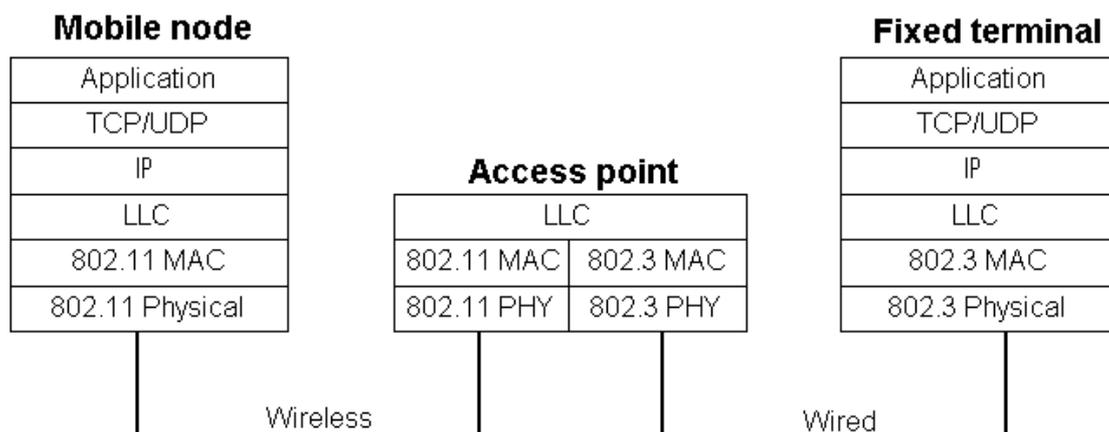


Figure 2: Protocol architecture

3 Brief background of GPRS

The general packet radio service (GPRS) [1] provides wireless packet transfer over a wide geographic area. GPRS is a service designed for the existing global system mobile communication (GSM) network. GPRS introduces two new network nodes:

The Service GPRS Support Node (SGSN) keeps track of the mobile phones location and performs security functions and access control. The Gateway GPRS Support Node (GGSN) is a gateway in GSM that allow mobile phones to access the public data network (PDN) or private IP networks. Data is transferred transparently between the mobile phone and the external data networks with encapsulation and tunnelling. The connection between the SGSN and the GGSN is enabled through a protocol named GPRS Tunnelling Protocol (GTP). Data packets are equipped with GPRS-specific protocol information and are transferred between a mobile phone and a GGSN. The connection between the GGSN and the PDN is enabled through the Internet Protocol (IP). To be able to connect to the Internet the mobile phone has to be assigned an IP address through the GPRS connection. The GGSN uses the Dynamic Host Configuration Protocol (DHCP) [2] to assign an IP address to a mobile phone. GPRS typically assigns private IP-addresses to mobile phones and uses Network Address Translator (NAT) [21] for mapping a private IP-address to a public at the GGSN. So it is not possible to reach the mobile phone from the Internet through a public IP-address. The mobile phone has to initiate the communication if a host from the Internet will be able to reach the mobile phone. A mobile phone dials up with a point-to-point (PPP) functionality. Figure 4 shows an overview over the GPRS entities. Figure 3 shows the protocol architecture from a mobile node to the IP network, Internet.

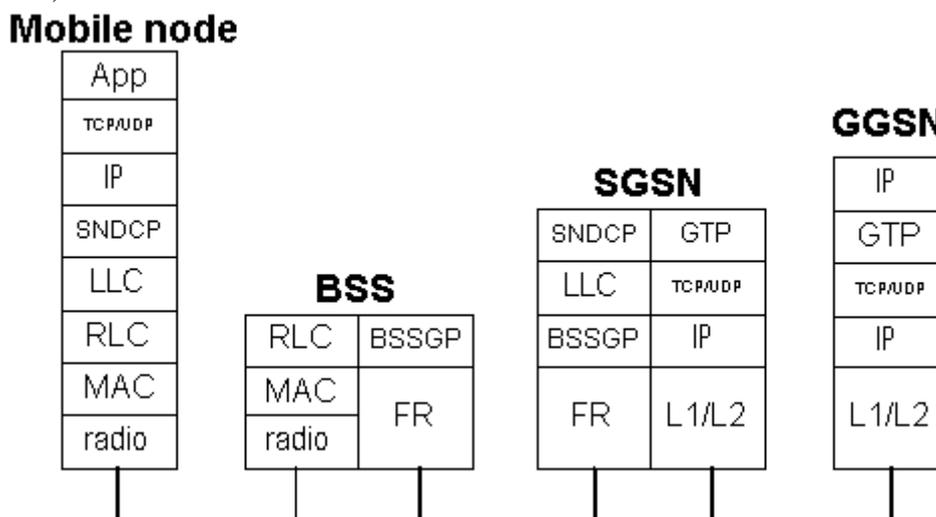


Figure 3: Protocol architecture

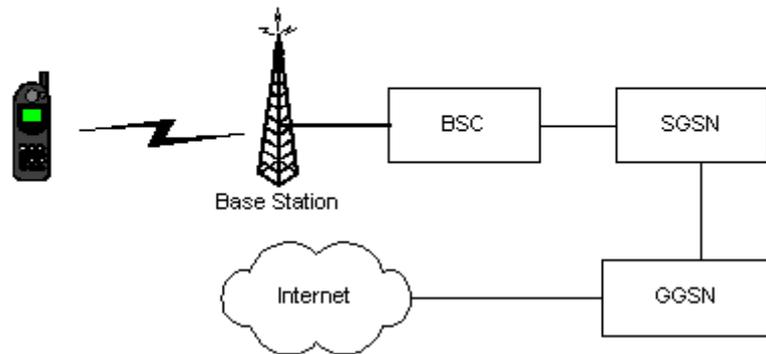


Figure 4: GPRS overview

4 Mobile IP Overview

Mobile IP [3] is an Internet protocol for allowing transparent routing of IP datagram's to mobile nodes in Internet. If you want to reach a mobile node wherever it is located it has to have a static home IP address. That is achieved with mobile IP. When a mobile node moves and attaches itself to another network, it obtains a new IP address. This is necessary as the IP routing mechanism rely on the topological information embedded in the IP address to deliver the data to the correct end-point. Mobile IP handles this by network agents. No modifications on the routers or end hosts are required. Each mobile node is identified by a static home network address from its home network, regardless of its current point of attachment.

4.1 Terminology

- Mobile node (MN): A mobile node with a static IP address. The mobile node can change its point of attachment to the Internet using mobile IP.
- Correspondent node (CN): Mobile nodes communication partner. The correspondent node can be fixed or a mobile node.
- Home network: The subnet the mobile nodes home IP address is belonging to.
- Foreign network: The current subnet the mobile node is visiting.
- Home Agent (HA): Is located in the home network. The home agent can be implemented on the router at the home network or at a regular node in the home network. The home agent can work as manager for the mobile node. With the manager solution the mobile node is always in a foreign network. When a mobile node is outside the home network the home agent receives all packets destined to the mobile node and tunnels them to the current location of the mobile node.
- Foreign Agent (FA): The foreign agent provides services to the mobile node during its visit to the foreign network. The foreign agent acts as the tunnel end-point, decapsulates incoming packets and forwards them to the mobile node. The foreign agent is typically implemented on the router at the foreign network. A foreign agent is not necessary needed; if the foreign agent is discarded the mobile node has to decapsulate the incoming packets itself. The mobile node is then co-located.

- Care-of address (COA): The care-of address defines the current IP address of the mobile node. All packets sent to the mobile node are sent to the home agent and tunneled to the care-of address. The care-of address is the tunnel endpoint. The care-of address can be located at two different points, at the foreign agent or at the mobile node directly. If the care-of address is located at the mobile node directly, the mobile node is then co-located.
- Tunnel: The path followed by a datagram while it is encapsulated. The model is that, while it is encapsulated, a datagram is routed to a knowledgeable decapsulating agent, which decapsulates the datagram and then correctly deliver it to its ultimate destination.

While a mobile node is away from its home network, it updates the home agent with information about its current IP address. The home agent receives all incoming packets destined to the mobile node, encapsulates and tunnels them to the mobile nodes current IP address (COA). When the mobile node wants to send packets to the correspondent node two options is available. The simpler way is to send the IP packet to the correspondent node with the mobile nodes home address as source address instead of the care-of address. But there are some problems with this option, many intranets only allow packets with topologically correct addresses to pass. Since the source address of the IP packet is changed, the address will not be topologically correct. The other alternative is to use reverse tunneling, when the mobile node wants to send a packet to the correspondent node it encapsulate and tunnels the packet to the home agent. The home agent then decapsulates the packet and forwards it to the correspondent node as when a packet is sent from the correspondent node to the mobile node. This is called reverse tunneling. Figure 5 shows the scenario when the mobile node is located in a foreign network and communicates via a foreign agent. The home agent and the foreign agent are acting as tunnel endpoints.

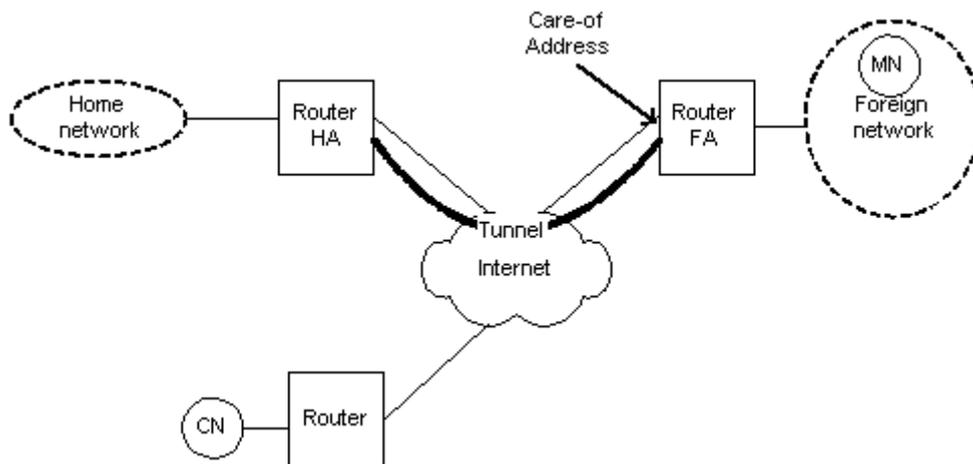


Figure 5: Mobile IP overview

When the mobile node is outside its home network it has to try to find a foreign network. Finding a foreign agent is done in two ways, either the agent sends out agent advertisement messages or the mobile node is sending out agent solicitation messages. The home and foreign agents advertise their presence by sending out advertisement messages periodically. A mobile node in a wireless subnet can receive an advertisement message either from a home or a foreign network. If a mobile node wants to find a wireless subnet but are not receiving any advertisement messages it

can send out solicitations messages. An agent responds to a solicitation message and the mobile node can receive a care-of address.

When a mobile node has found a subnet, either the home or a foreign network, it has to register at the agent. If the mobile node is in a foreign network the registration can be done in two ways, through a foreign agent or directly to the home agent. If the mobile node is using a foreign agent the registration goes through the foreign agent. The mobile node sends a registration request containing the care-of address to the foreign agent, which forwards the request to the home agent. The home agent sets up tunnel from the home agent to the foreign agent. The registration expires after negotiated lifetime, this is for avoiding mobility bindings which are no longer used. The home agent sends a reply message to the mobile node through the foreign agent after setting up the tunnel. If the mobile node not is using a foreign agent the registration message is sent directly to the home agent. All registration packets are sent using UDP as transport protocol. Figure 6 shows both register cases.

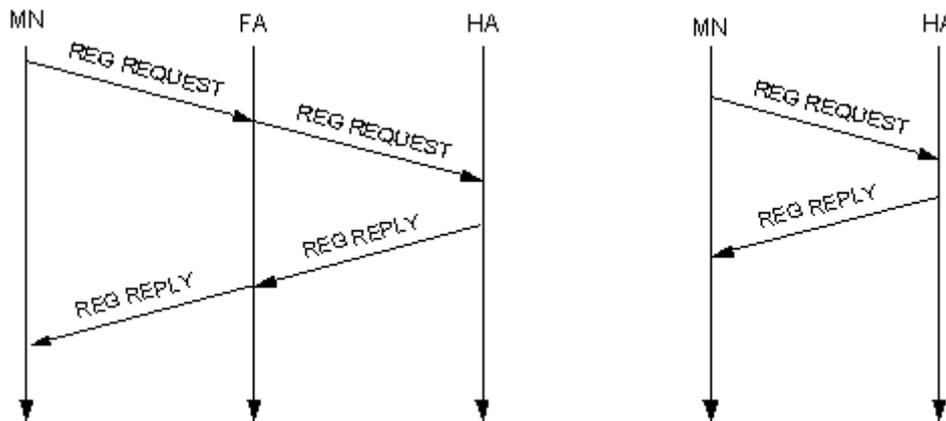


Figure 6: Registration of a mobile node via the FA or directly with the HA

5 Handover terminology

A handover occurs when a mobile nodes signal is passed from one base station to another. The handover occurs when a mobile node passes out of range that the cell can handle and into another cell's range.

Soft handover:

A soft handover occurs when a mobile node moves from one cell to another without losing the connection to the first one. The handover occurs when both cells have radio coverage. The mobile node is attached to the both networks simultaneously during the handover.

Hard handover:

A hard handover occurs when the old connection is broken before a new is established.

Seamless handover:

A seamless handover is a handover scheme that maintains the connectivity of all applications on the mobile device when the handover occurs.

Horizontal handover:

A horizontal handover is a handover between base stations that are using the same kind of wireless network interface.

Vertical handover: Vertical handover is a handover between two network access points using different connection technologies. Verticals handovers can be divided into two categories: upward and downward. An upward handover is a handover to larger cell size and a downward to a smaller cell size.

Seamless vertical handover:

A vertical handover is a handover between different network technologies. Since each network interface on a mobile node has its own IP address a vertical handover has to handle changing of IP addresses. Since a TCP communication session is identified by the tuple (source IP, source Port, destination IP, destination Port) an issue arises. When performing a vertical handover the mobile node has to switch physical IP address. The WLAN network interface and the GPRS network interface cannot use the same IP address. The challenging problem is how to keep the session alive while changing the physical connection interface (IP address). This topic has been studied widely and the solutions can be classified into three categories: Upper-layer approaches, new transport layer approaches and network layer approaches. Network layer approaches are typically based on mobile IP or mobile IPv6. These approaches require implementing agents on Internet for forwarding data. They require infrastructure modifications to work properly. New transport layer approaches require already existing applications to be rewritten. Since most existing applications are using either TCP or UDP as transport protocol they have to be rewritten. Upper-layer approaches implements a new layer above the transport layer. A session layer can make a session exist between two applications instead of between two hosts. Vertical handover solutions have to deal with two main issues: latency and packet loss.

6 Network layer approaches

Solving the vertical handover problem in the network layer is hard. Introducing a new Internet Protocol requires updates on the routers and other machines on the Internet. The new protocol IPv6 is intended to replace and probably will the today existing IPv4. IPv6 has support for mobile nodes and handles vertical handover. Another proposal for handling vertical handoffs is the S-MIP (A Seamless Handoff Architecture for Mobile IP) solution [17] S-MIP is built on a hierarchical approach. The hierarchical handover scheme in S-MIP separates mobility management in to micro and macro mobility management.

6.1 Mobility support in IPv6

IPv6 is a new version of IP and is intended to replace the current version IPv4 as Internet protocol. The length of the address has been increased from 32 bits to 128 bits. In mobile IPv6 [18] s in mobile IPv4 each mobile node is always identified by its home address, regardless of its current point of attachment to the Internet. When the mobile node is away from the home network the mobile node is associated with a care-of address (COA). Each time the mobile nodes move from one subnet to another the node will configure its COA with another COA belonging to the new subnet. The configuration can be done with DHCPv6 or PPPv6. Mobile IPv6 enables any IPv6 node to learn and cache the COA for a mobile node. This is for avoiding triangular

routing and to packet is sent using an IPv6 routing header instead of IPv6 encapsulation. A mobile node's association to a COA is known as binding and has a remaining lifetime. This is for other nodes to know how long to store the COA in the binding cache. When sending an IPv6 packet to any destination, a node checks its binding cache for an entry for the packet's destination address. If an address is found in the cache the packet is sent directly to the COA instead through a home agent where it has to get encapsulated. Mobile IPv6 introduces a set of new messages to achieve this, *Binding Update* and *Binding Acknowledgement*. After a mobile node has configured its COA, it must send a *Binding Update* to the HA and all corresponding nodes. The Binding Message contains the current COA for the mobile node. The recipients updates their binding cache and sends a *Binding Acknowledgement* if so was requested in the *Binding Update* message. The *Binding Update* message can be sent separate or together with any payload such TCP or UDP.

Although messages can be sent directly to the care-of address from a correspondent node to avoid triangular routing the mobile node can always be reached through its home address. So the movement of the mobile node is thus transparent to the transport and higher layers protocols.

7 Transport layer approaches

Today most applications are communicating with either TCP or UDP as transport protocol. Introducing a new transport protocol to solve the seamless vertical handover problem has its disadvantages. Unfortunately almost all today's existing applications have to be rewritten to support a new transport protocol. Two proposals for a new transport layer protocol are considered, Stream Control Transmission Protocol (SCTP) [4] and TCP Multi Homing (TCP MH) [7].

7.1 SCTP

Stream Control Transmission Protocol (SCTP) [4] offers a reliable delivery service for application over an IP network and is session-oriented. The most interesting feature of SCTP is multi-homing. An SCTP session can be established over multiple IP addresses. SCTP sends packets to a primary IP address, but can reroute packets to an alternative, secondary IP address if the primary IP address becomes unreachable. A SCTP session has a primary path between two SCTP hosts, but can also have multiple paths between the hosts. This type of session is defined as an association in SCTP. An SCTP association between two hosts A and B, is defined as:

$$\{[\text{IP addresses of A}] + [\text{port A}]\} + \{[\text{IP addresses of B}] + [\text{port B}]\}$$

In the base version of SCTP the endpoints exchange their IP addresses before the SCTP association is established and these addresses cannot be changed during the session. However mobile SCTP (mSCTP) [5] supports adding, deleting or changing IP addresses during an active session using Address Configuration (ASCONF) messages. SCTP supports the end-to-end principle [6]; anything that can be done in the end system should be done there. Since the transport layer is the lowest end-to-end layer in the Internet protocol stack, the vertical handover should be done there [6]. The end-to-end principle says that anything that can be done in the end system should be done there [6]. Figure 7 shows the protocol stack with mSCTP as transport protocol. The mSCTP is transparent to the IP layer. In figure 8 a message procedure is shown. The Mobile node communicates to the correspondent node through the

WLAN interface. The mobile node sends an ASCONF message and adds a GPRS IP address. Later the mobile node performs a handover and sends a ASCONF message which switches the primary IP address. Now all data to and from the mobile node are sent through the GPRS interface.

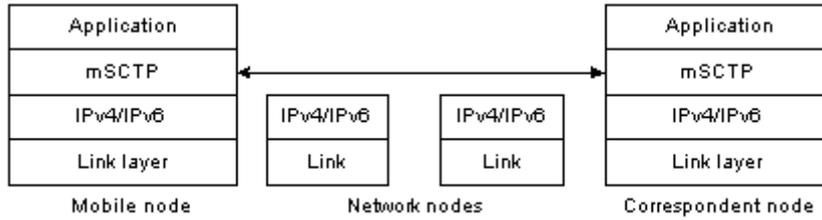


Figure 7: Protocol architecture

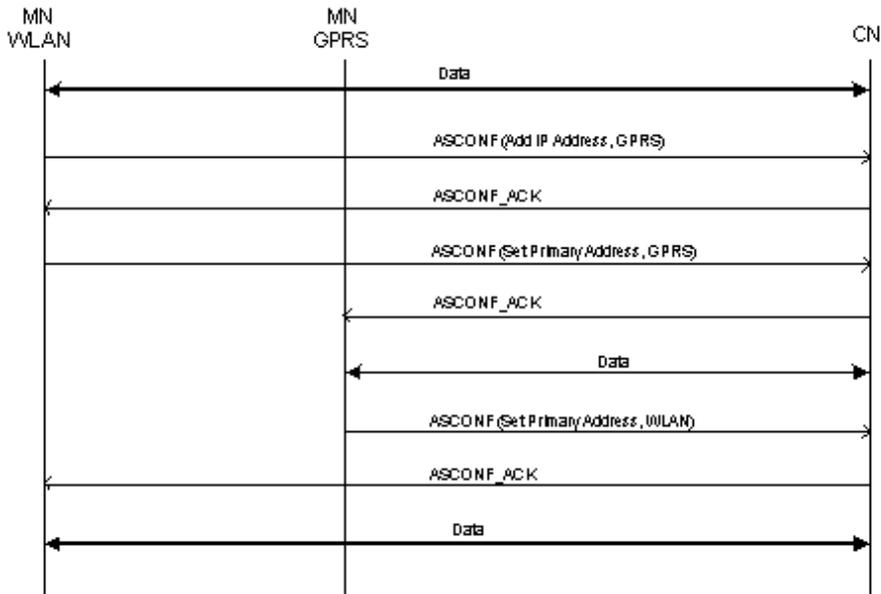


Figure 8: Vertical handover procedure with mSCTP

7.2 TCP MH

TCP Multi Homing (TCP MH) [7] resembles SCTP but TCP MH is just an extension of the existing TCP, not a complete new transport protocol. The existing TCP is only designed for communication between one local and one remote IP address. The TCP MH option makes it possible to handle multiple local and remote addresses during a TCP session. TCP MH provides multi-home feature to TCP without modification and dependence on any other elements in the Internet. Features as flow control, slow start, collision avoidance and fast retransmission in TCP are kept in TCP MH. A TCP MH session starts with a MH-Permit option in a SYN packet. If host accepts MH-permit a SYN-ACK with MH-Permit packet is returned and MH options can be used. A TCP session can be kept even though the source and/or the destination address changes. A session can also switch from IPv4 address to IPv6 address and vice versa. Hosts exchange their IP addresses with MH-Add-IPv4 or MH-Add-IPv6 options. After an endpoint has received a MH-Add option the endpoint register the new transmission path. There are also MH-Delete options for deleting addresses. Figure 9 shows a mobile node trying to establish a connection from the WLAN interface, but no response returns from the correspondent node. Later the mobile node switches interface to the GPRS connection and tries to connect with that. The connection is established after an ACK from the correspondent node to the mobile node. The three-

way handshake is finished after the correspondent node receives an ACK from the mobile node. When data is sent the mobile node uses the “MH-Add-IPv4” option, which tells the correspondent node that another IP address for the mobile node (IP for the WLAN interface) can be used. The correspondent node accepts that option and sends back an ACK. Later as seen in the picture the correspondent node sends data that doesn’t reach the mobile node and no ACK was received. The correspondent node then tries to send the data to the other IP address (WLAN) that is available for the session.

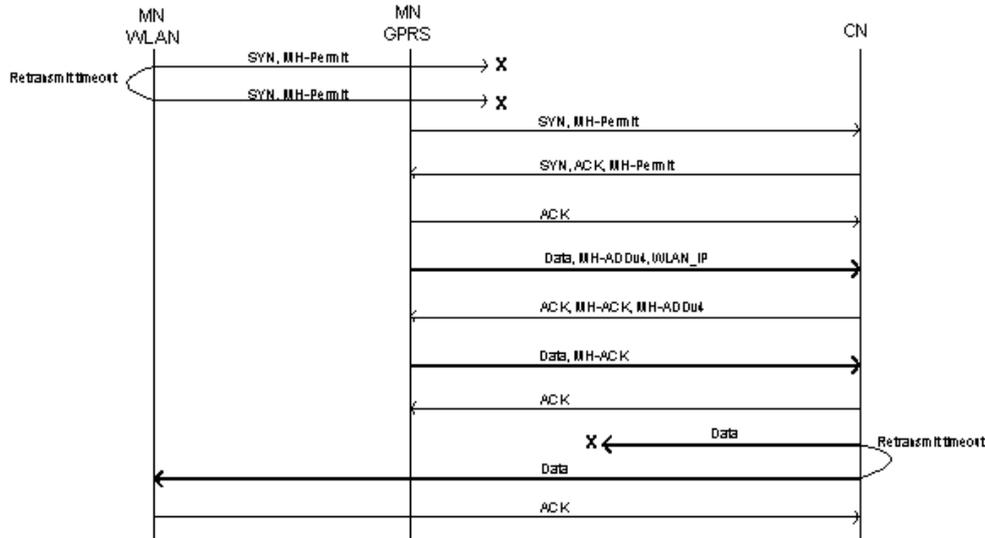


Figure 9: TCP-MH message overview

8 Upper-layer approaches

These approaches implement a new session layer above the transport layer. The application sessions will be transparent to the connection changes in the underlying layers. A session layer can make a session exist between two applications instead of between two hosts. The session then has to be identified in another way than the IP numbers and port numbers. Some session ID has to identify a session between two applications.

8.1 Migrate approach

The migrate approach [15, 16] is a session-oriented architecture end-to-end host mobility approach. The migrate approach propose a session layer. Sessions exists between application end points, and should survive changes in the transport and network layer protocol states. Once a session is established a locally unique token or Session ID identifies it. The authors of the migrate approach propose five important fundamental issues that has to be handled in Internet mobility.

1. Locate the mobile host: The desired end point must be located and mapped to an addressable destination.
2. Preserving communication: Once a session is established, communication should be able to handle changes in the network location of the end points.
3. Disconnecting gracefully: Disconnection should be rapidly detected.
4. Hibernating efficiently: If a host is unavailable for a period of time, the connection should be suspended and resources should be reallocated.

5. Reconnecting quickly: Communication peers should detect resumption of connectivity in timely manner. The system should be able to re-establish the connection without any extra effort.

Mobility support should be provided at the end hosts [6]. Many previous approaches like mobile IP rely on proxies. Proxy-based solutions have to deal with some performance issues. The proxies have to be well engineered and well located in the network to perform acceptably.

The selection of network end point and transport protocol remains under the application's control. Naming can abstract location details. The migrate approach provides a naming service, a mobile host isn't bound to a home IP address like in mobile IP. Instead the host is identified by a hostname. To locate mobile node hosts the widely deployed Domain Name System (DNS) [8] is used. Many applications resolve hostnames to an IP address at the beginning of a connection. No home agent is necessary as in mobile IP. When a mobile node changes its location and IP address, it sends a DNS update to one of the name servers. Since the session is identified with a session ID the session can remain from the new location. The session layer has to re-synchronize the session between the hosts. The session layer has to handle and save the state of the connection to be able to continue the session after a reconnection from another IP address. It is possible to hijack the connection with this solution, a secure key exchange part has to be implemented in the session layer. The migrate approach uses the Elliptic Curve Diffie-Hellman [10] for key exchanges. The same problem arises for the DNS update sent by the mobile node. The migrate approach uses the security of dynamic DNS updates in RFC 2137 [9]. Figure 10 shows the components of the Migrate architecture. The session layer has four interfaces: session establishment, connectivity status, policy decisions and application up-calls.

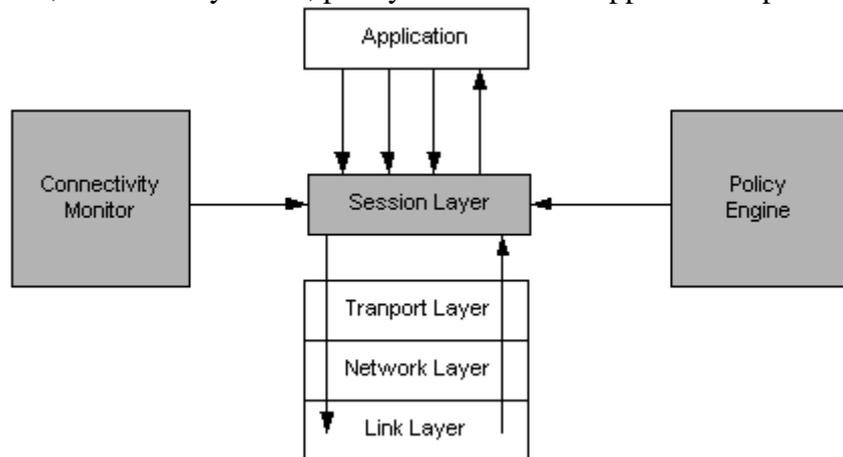


Figure 10: The Migrate session layer framework

8.2 MSOCKS

MSOCKS [13] is a proxy-based solution with a proxy inserted into the communication path between a mobile node and its correspondent host. MSOCKS is using a technique called TCP Splice [13]. TCP Splice preserves the end-to-end semantics as normal TCP connection. Normally in proxy-based solutions each session between a mobile node and a correspondent node is split into two separate TCP connections. TCP Splice allows a machine where two independent TCP connections terminate to splice the two connections together. The connection will form a single

end-to-end TCP connection between the endpoints of the two original connections with the proxy in the middle. The mobile node is communicating via a MSOCKS library that runs under the application. At the proxy a MSOCKS proxy process is running; an in-kernel modification on the proxy machine to provide the TCP Splice service. The correspondent host doesn't need any modifications. The MSOCKS protocol is built on top of the SOCKS protocol [14]. The MSOCKS library has similar functions like *bind*, *accept* and *connect* as SOCKS does. An identifier identifies the session between mobile host and proxy. MSOCKS also has a *reconnect* function, when a mobile host wants to change network interface (IP-address) it opens a new connection to the proxy and sends a reconnect message with the session identifier. The proxy unsplices the old mobile-to-proxy connection and splices in the new mobile-to-proxy connection. The end-to-end semantics of TCP are maintained together with TCP Splice. TCP Splice makes it appear to the endpoints of two separate TCP connections that those two connections are, in fact, one. Data can be lost with this solution; ACK'd data to the correspondent host but lost in the transmission to the mobile host is lost forever.

M SOCKS library is a layer between the application and the transport layer, it provides an interface to the application while internally using the normal TCP stack. To get this to work the applications has to use this library instead of the existing SOCKS or the existing application has to be recompiled. Figure 11 shows a message exchange diagram when a MSOCKS client tries to connect to a server on a correspondent host. The MSOCKS library function *Mconnect()* is used for making this split connection. *Mconnect()* first makes a connection to the proxy then it sends the server's address and port number to the proxy in a Connect message. The proxy connects to the desired server and splices the mobile client-proxy and proxy-server connections together. When the splice is set up the proxy finally sends an OK message back to the mobile client.

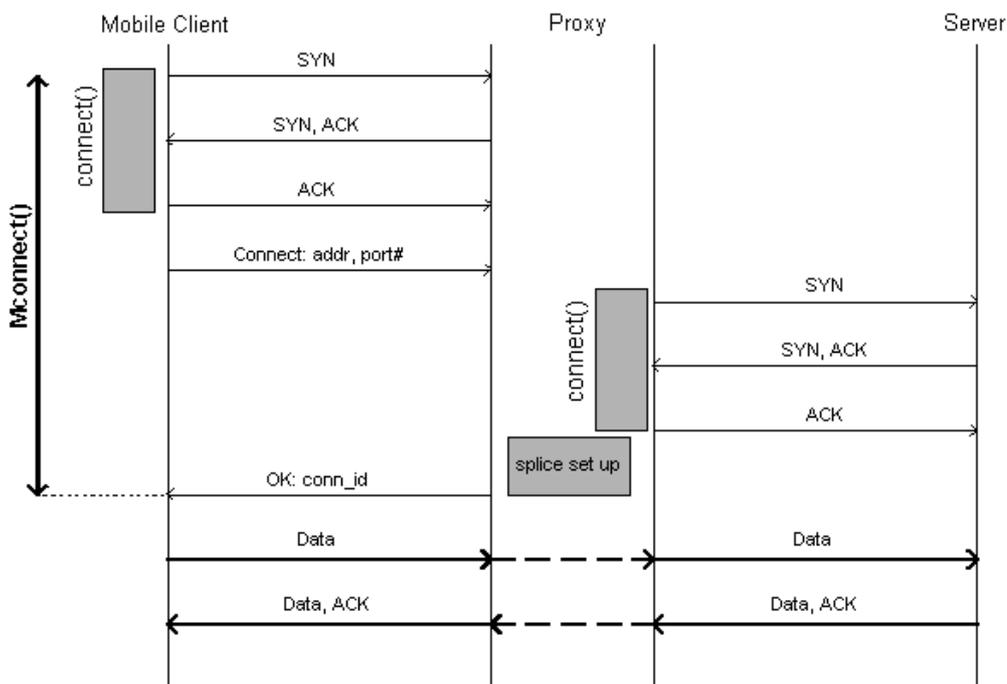


Figure 11: Message exchange diagram for connection establishment between a MSOCKS client and correspondent host via a MSOCKS proxy.

8.3 Universal seamless handoff architecture

Universal seamless handoff architecture (USHA) [11] achieves seamless handoff by following the middleware design. The USHA doesn't require any infrastructure modification, but the solution assumes that handoff only occurs on overlaid networks. The USHA network is composed of a handoff server (HS) and several mobile hosts (MH). The HS and the MH are communicating using an IP tunnel. Each MH maintains a tunnel to the HS. All applications are communication using the tunnel interface instead of any physical IP addresses available. All packets communicating via the tunnel are encapsulated and transmitted to the HS using the UDP protocol. The tunnel has two virtual and two physical IP addresses. The applications are communicating using the virtual addresses and the tunnel is using the physical addresses to communicate. When a handoff occurs the physical IP address is switched on the mobile host. A handoff client is responsible for switching the physical address of the virtual tunnel to a new interface. The end-to-end TCP semantics are kept since the session is bound to the virtual addresses and to the tunnel is using UDP.

9 Discussion

The different proposals for solving the seamless vertical handover problem are shown in table 1. Both the session and transport layer proposals require that the existing applications have to be rewritten. The network layer approaches require infrastructure modifications to work. So none of these approaches solves the seamless vertical handover problem in an easy way. The USHA solution cannot be categorized to any of these layers and doesn't require any infrastructure or application modifications. I think that a seamless vertical handover should be done in the transport layer. To use SCTP as transport protocol seems as much easier and better way to perform seamless vertical handover than using agents and tunnels. Unfortunately most of today's application either uses TCP or UDP as transport protocol. But in the future it may be a good idea to implement applications with a transport protocol that supports multiple IP addresses during a session. Another advantage is that if the problem is solved in transport layer no modification on the infrastructure is needed. Solving the handover with home agents or proxies will not scale as well. The home agent or the proxy may be the bottleneck in the system and the traffic to and from the mobile node are taking longer paths than necessary.

Table 1: Seamless vertical handover approaches

Session layer	Transport layer	Network layer
Migrate approach (10.1)	SCTP (9.1)	Mobile IPv6 (8.1)
MSOCKS (10.2)	TCP MH (9.2)	S-MIP (8)

10 Implementation

10.1 Requirements

The goal of the implementation is to create a basic vertical handover system prototype. A home agent and a mobile node application will be implemented. It is assumed that a mobile node is having a reserved home IP address at a home network. Every communication session that the mobile node initiates with a correspondent node will be done with the home address of the mobile node. The mobile node will be able the switch physical connection to the Internet during a communication session to a corresponding node. The user will be able to switch the connection through a simple

Graphical User Interface (GUI). The GUI will display which interfaces are up and which is currently used for sending and receiving data to and from the Internet. All applications should run over the mobile node application, outgoing packet is just encapsulated and sent via the home agent. So the mobile node application will be application transparent.

10.2 Analysis

Since that the application prototype should be application and infrastructure transparent the handover in the prototype must occur at lower level than the transport layer. Neither modification at the network layer is possible since the prototype should work with the existing infrastructure. So no modification in the protocol stack is available according to the requirements. Even the mobile node must communicate with its correspondent nodes through the same IP-address, since both hosts' IP-addresses and port numbers identify a TCP-connection. The solution has to involve redirecting packets through a home network. The original packet sent from a correspondent node has to get encapsulated at a node in the home network. All packets sent to a correspondent node must have the correct source IP-address, the mobile nodes home address. This solution involves that the mobile node has to register at the home network with its current IP-address. A node at the home network has to handle the registrations and redirections of packets destined for the mobile node, a home agent. Since that the mobile node may have multiple network interfaces the home agent has to be able to handle that.

10.3 System overview

The mobile node (MN) will have one WLAN and one GPRS connection. The MN is always identified by its home IP-address, regardless of the interfaces IP-addresses. At the home network for the node a home agent (HA) will capture all incoming packets destined to the mobile node and tunnel them to one of the interfaces the mobile node is holding. So all packets sent to the MN from a correspondent node (CN) will be captured by the home agent at the home network and tunneled to the MN. The outgoing packets from the MN to a CN will also be tunneled to the HA. The HA forwards the packet to the CN with the MN:s home IP-address as source address in the packet. So the CN will automatically answer to the MN:s home address. OpenVPN [19] is used for managing the tunnel between the MN and the HA. Before any communication take place between the MN and a CN the MN has to register by the HA. The MN starts by sending a *registration request* over all interfaces to the HA. When the HA receives a *registration request* it answers with a *registration reply*. After the registration is completed the communication through the HA to a CN can take place. The MN continues to send *registration requests* periodically over all interfaces to the HA.

This system could only work if the HA can capture all incoming packets destined to the MN. The HA has to be within same local network as the MN:s home address. When the MN has registered from a care-of address (COA) outside the home network the HA has to start capturing the MN:s packets and tunnel or forward them. As soon as the HA receives a registration request from a MN the HA sends out a gratuitous ARP over the LAN. The HA also starts to answers ARP requests with ARP replies on the behalf of the MN:s home address. The gratuitous ARP is sent for updating the other nodes in the local network ARP caches. The gratuitous ARP is sent as an ARP

request. The MN:s home address will be linked to the HA:s physical address (MAC address). The technique when the HA answers to ARP requests on the behalf of the MN:s home address is called proxy ARP. The HA acts like a proxy and starts answering at ARP requests aimed for the MN:s home address. When the HA receives an ARP request aimed for the MN:s home address, the HA answers with its physical address in the ARP reply. The proxy ARP is simple permanent entry in the HA:s ARP cache.

For directing packets from an arbitrary application to the tunnel the routing table has to be configured in the mobile node. All packets should as default get routed to the tunnel. Only the *registration request* packets should get routed through one of the network interfaces directly. For a schematic overview of the system, see figure 12.

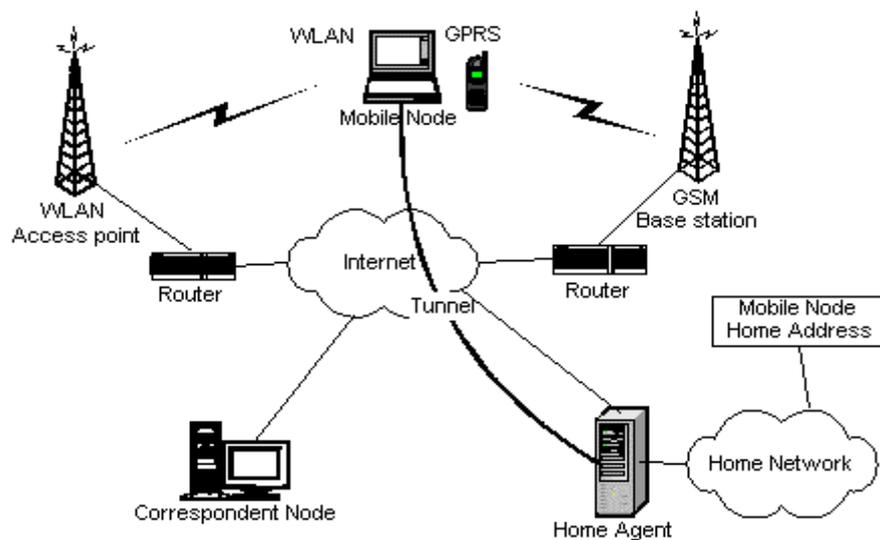


Figure 12: System overview

10.4 Environment

The mobile node was implemented on a laptop with a Linux kernel 2.4.20. The kernel was compiled with IP advanced routing and IP policy routing. The applications on both the mobile node and the home agent were developed with Java (J2SE 5.0). Although Java is platform independent the applications only runs under Linux. Some Linux specific modules like iproute2 [23] are used for manipulating routing rules and routing tables in the mobile node. A small routine in the home agent implementation was written in C, the part that sends out a gratuitous ARP. Since Java is platform independent and doesn't support such a low level socket that part had to be written in another language. The Java application at the home agent simply calls the gratuitous ARP routine when it wants to send out such a packet. The open source project openVPN is used for handling the tunnel. In both the mobile node and the home agent the tunnel is started from the Java applications. The input settings for the tunnel are read from a configuration file. So the tunnels on both sides are different processes and runs independently.

11 Mobile Node

Before any communication through a tunnel can take place the mobile node has to register by the home agent. The *registration request* messages have to be sent over all interfaces continuously. If an interface doesn't get any *registration reply* from the home agent within a certain time interval the interface will be considered down. That time has to be specified for every interface in a configuration file. The main reason for sending the *registration request* continuously is that the statuses of the different interfaces always have to be updated. So if you move with your mobile node from the radio coverage of your WLAN, the mobile node will discover that when the timer expires for that interface. If an interface that is active as tunnel endpoint timeout a handover to another interface will be performed. The mobile node is always co-located, that means that the tunnel endpoint is always located at the mobile node and no foreign agent is used.

11.1 Configuration

Before starting the mobile node application a configuration file (*mn.config*) has to be edited. See the User Guide in appendix B for more information about the configuration.

11.2 Script

Two scripts have been written for the mobile node. The script **getip** reads the IP address based on the argument to the script, which is the name of the device (e.g. *eth1*). The second script **getgw** reads the default gateway for that device. These scripts were written so that the user doesn't have to specify the IP address and gateways as input to the application. The scripts are also used for rereading the IP addresses for the devices during runtime if they have changed after the application started.

11.3 Architecture

Figure 13 shows the main classes that were implemented in the mobile node application. The class *Iface* represents a device and each device has its own registration and receiving thread, *RegThread* and *RegReplyRecv*. The *Iface* class has a socket object, which is bound to the IP address for that device. The class *Node* has a *Vector* with *Iface* objects. The class *Config* reads the configuration file and interacts with the class *Node*. The *Node* also has a *Gui* and a *Tunnel*.

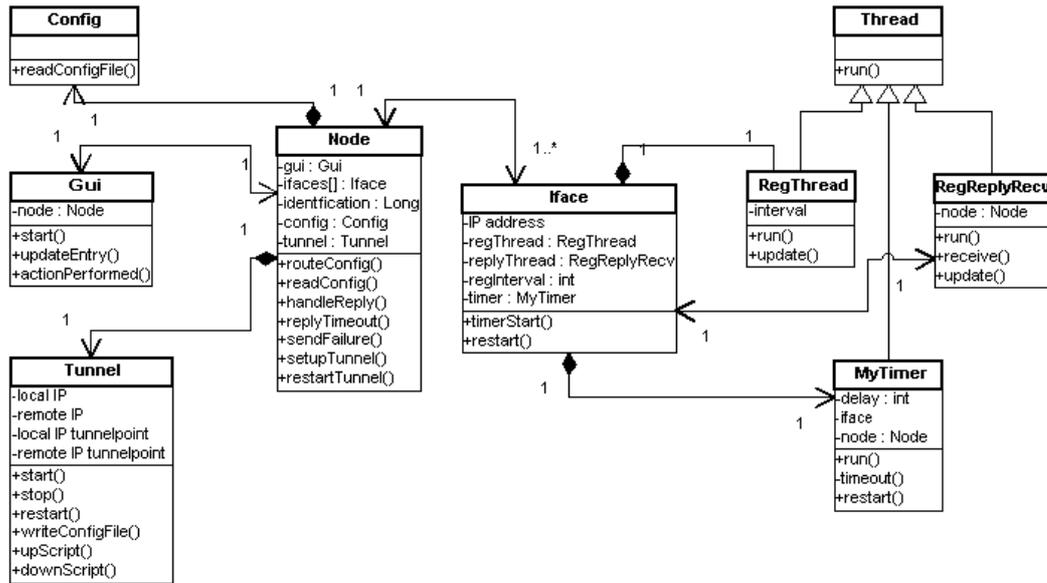


Figure 13: Class diagram for mobile node.

11.4 Routing

Since the *registration requests must* be sent parallel over all interfaces and over the correct physical connection, routing rules must be applied. Ip rule in the Linux-kernel is used for making this possible. A rule examines the source IP-address in an outgoing packet and is bound to a routing table. If the IP-address of an outgoing packet matches a rule then the bound routing table for the rule will be used for routing the packet. Every interface in the mobile node has a rule entry in the rule list and thus an own routing table. All the routing tables looks similar but packets destined to the home agent are handled differently. Packets destined to the home agent will be transferred through the device where the source IP-address of the packet match the IP address of the device. Figure 14 shows the routing rules for the mobile node with two devices, a WLAN and a GPRS device. If a packet has the source address 130.240.138.139 the table **wlan** will be used for routing the packet. Similar if the packet has a source address 10.177.23.14 the table **gprs** will be used for routing the packet. Figure 15 shows the routing table **wlan**. All packets will be sent to the tunnel device tun0 except those that has a destination address to the home agent. Those packets will be sent outside the tunnel via the gateway 130.240.138.1 through the device **ath0**. Figure 16 shows the table **gprs**, which looks the same, but without any gateway specified since the device is a point-to-point connection. GPRS is a Point-to-point-based network and WLANs are LANs. Packets from a LAN have to pass a gateway of the LAN. Since all outgoing packets from the LAN has to go through the gateway that is connected to the Internet the gateway has to specified in the routing table for the WLAN interface. The main routing table can be seen in figure 17.

Routing rules

From IP	Table
130.240.138.139	wlan
10.177.23.14	gprs

Figure 14: Routing rules

Routing table wlan

<u>Destination</u>	<u>Device</u>
130.240.136.214 via 130.240.138.1	ath0
default	tun0

Figure 15: Routing table wlan

Routing table gprs

<u>Destination</u>	<u>Device</u>
130.240.136.214	ppp0
default	tun0

Figure 16: Routing table gprs

Routing table main

<u>Destination</u>	<u>Mask</u>	<u>Device</u>
10.1.0.1	255.255.255.255	tun0
212.181.254.82	255.255.255.255	ppp0
130.240.138.0	255.255.255.0	ath0
default		tun0

Figure 17: Main routing table.

11.5 Algorithms

When the application is started it starts to read the configuration file, add routing rules and creates a routing table for each interface. For every interface a registration and a listening thread are started. Figure 18 and 19 shows the algorithm for sending *registration requests* and receiving *registration replies* to/from the home agent.

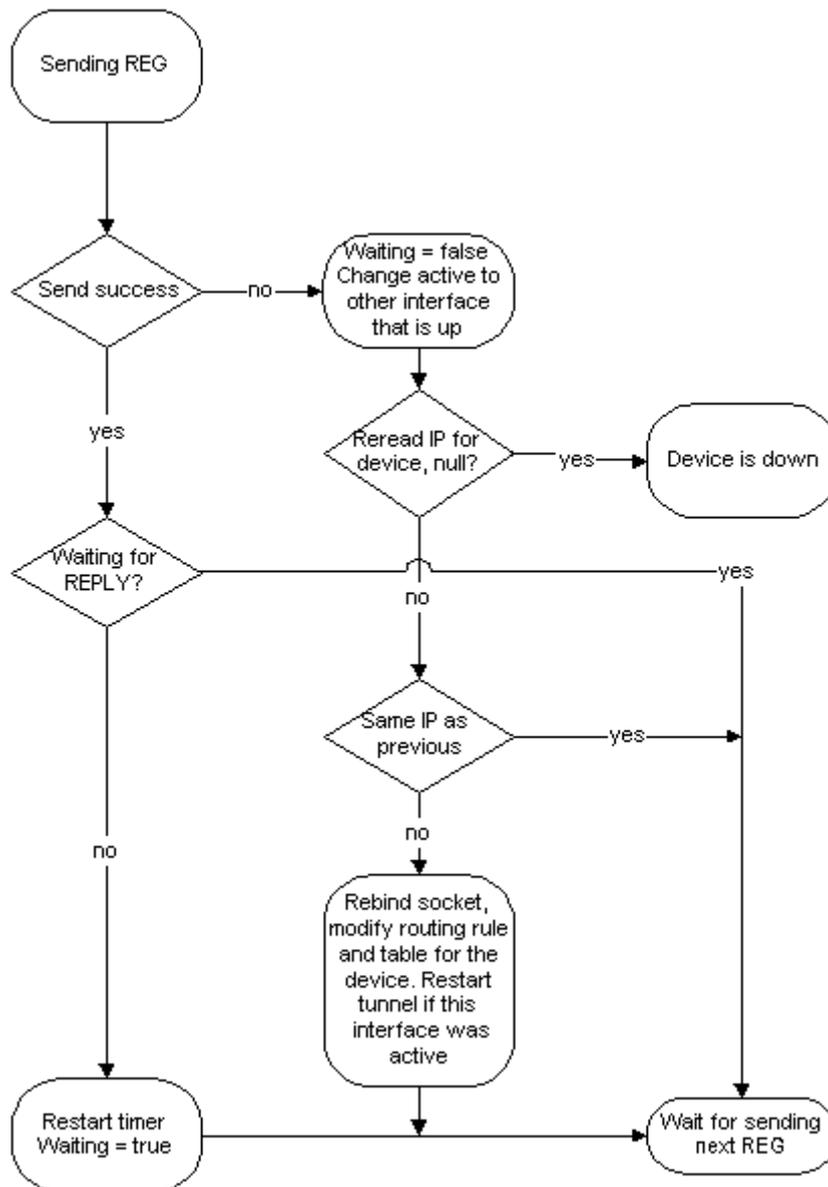


Figure 18: Sending Registration request algorithm

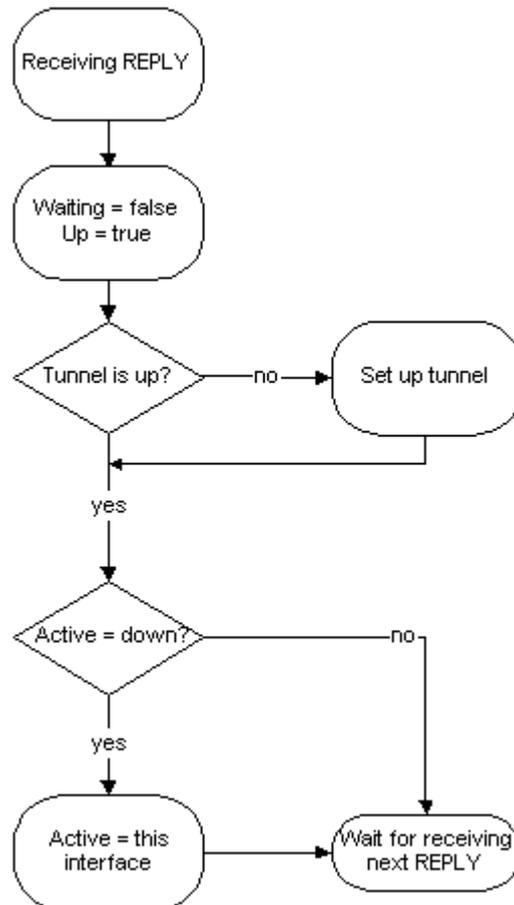


Figure 19: Receiving registration reply algorithm

If the timer for an interface expires before a *registration reply* is received, the interface is considered to be down. If the interface that became down was active a handover is performed to an interface that is up. Figure 20 shows the algorithm when a timer for an interface expires.

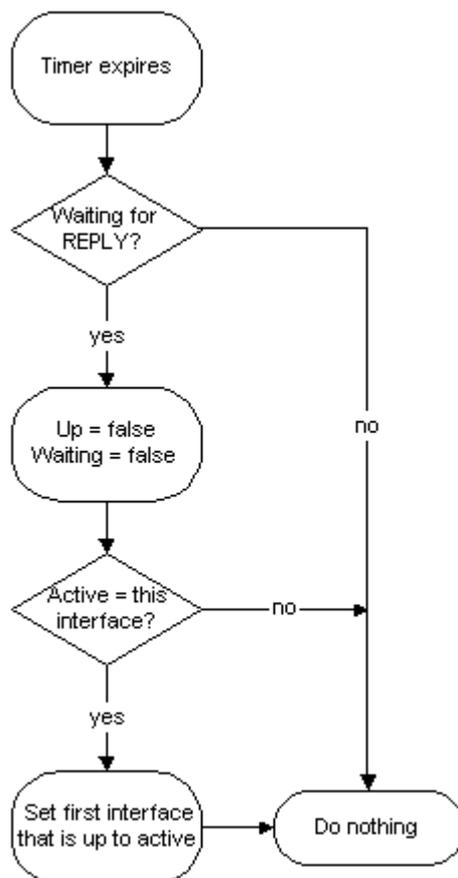


Figure 20: Algorithm when the registration reply timer expires.

11.6 NATP

Many LAN's are connected to external networks like Internet via Network Address and Port Translator (NAPT). A NAPT allows host to connect to the external network transparently. The internal host addresses has to be translated to one public IP-address. The NAPT does this; translate private IP-addresses and port numbers to public IP-addresses and port numbers. GPRS is behind a NAPT. Inbound traffic is only allowed for a limited period of time after an outbound session was initiated. So if corresponding node wants to connect a mobile node with a GPRS connection the mobile node has to initiate the communication session to allow incoming traffic.

11.7 Dynamic interfaces

The prototype can handle dynamic interfaces. For example: if you are located in a hotspot and the WLAN device has been assigned an IP address. You move with your mobile node and the application performs a handover to the GPRS interface. Later you enter a new hotspot and the WLAN device gets assigned a new IP address. The application will try to send registration as usual until a new IP address gets bound to the device. When a *registration* sending fails the application rereads the IP address for the device and tries to send *registrations* from the new IP address. This is implemented in the prototype but not tested so extensive.

12 Home Agent

The home agent has to be placed on a public network and at the same network as for the mobile nodes reserved home addresses. The home agent will listen for incoming

registration requests and respond with registration replies. When a mobile node register for the first time, the home agent sets up a tunnel to the mobile node and starts receiving packets destined to the mobile node home address.

12.1 Configuration

The home agent has to be configured before starting it. The name of the network device and the MAC-address of that device have to be specified in a configuration file. See the User Guide in appendix B for more information.

12.2 Architecture

Figure 21 shows the main classes for the home agent. The class *Agent* has a *Vector* with *MobileNode* as entries. Each mobile node that is registered has an entry in the vector *nodes* in the class *Agent*. Every *MobileNode* has a *MyTimer* object and a *Tunnel* object. The class *Config* reads the configuration file and interacts with the class *Agent*. Class *RegRecv* is a generalization of the class *Thread* and listens for incoming registration requests.

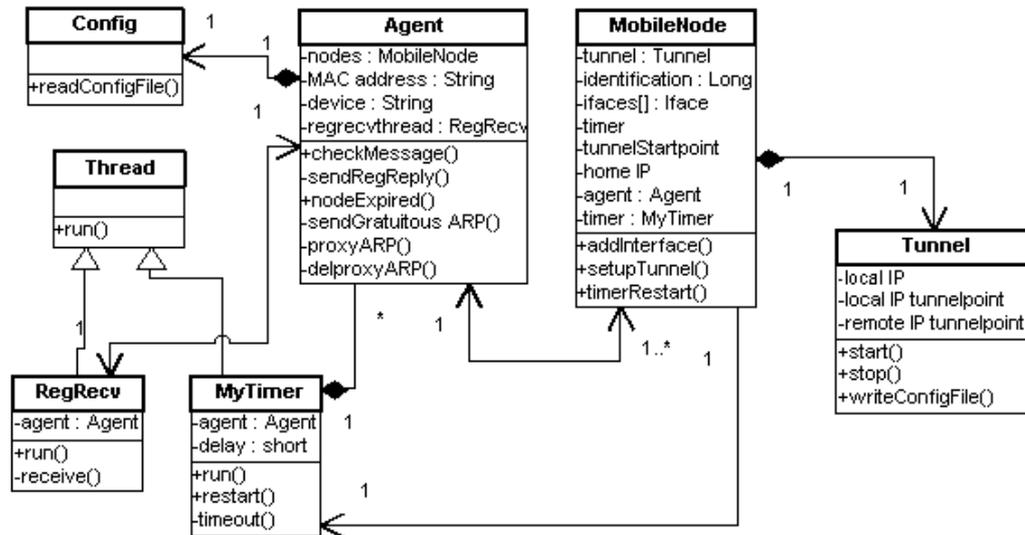


Figure 21: Class diagram for the home agent.

12.3 Algorithms

The home agent only has one main algorithm, to listen for registrations, handle them and respond with replies. Every mobile node that is registered will be stored in vector at the home agent. When a registration request is received a timer for the mobile node is restarted. As soon as the timer for a mobile node expires the mobile node entry in the vector will be deleted. Figure 22 shows the algorithm when the home agent receives a registration request from a mobile node.

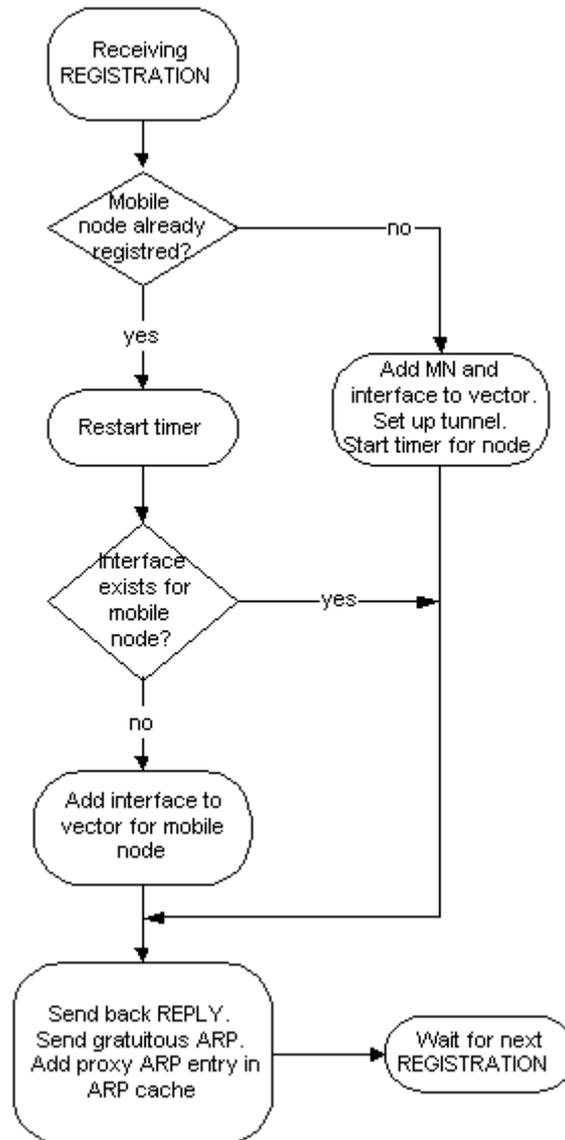


Figure 22: Receiving registration request algorithm

12.4 ARP terminology

ARP (Address Resolution Protocol) [22] is used for mapping IP addresses to physical devices. Every physical device (network interface) has an identifier (MAC-address) that is unique. Every node in a network has an ARP cache where there store the bindings between IP addresses and MAC-addresses for the nodes in the local network.

ARP request: An ARP request is sent by a node in a network asking for a MAC-address that is bound to a certain IP address.

ARP reply: An ARP reply is an answer to an ARP request. The ARP reply is sent to the sender and contains the MAC-address and IP address that is mapped together.

Gratuitous ARP: A node sends a gratuitous ARP over the local network to spontaneously cause other nodes to update their ARP cache. A gratuitous ARP may use either an ARP request or an ARP reply packet.

Proxy ARP: A proxy ARP is an ARP reply sent by a node on behalf of another node, which cannot answer to its own ARP requests.

12.5 Capturing the mobile nodes incoming packets

Since a mobile nodes home IP-address is different from the home agents IP-address, the home agent must capture incoming packets destined to the mobile node in some way. Since the home agent IP-address and mobile nodes home IP-address is at the same local network the problem is solved by manipulating the ARP cache at the home agent. The home agent starts to answers ARP requests for the mobile nodes home IP-address with ARP replies. Adding a proxy ARP entry in the local ARP cache does this. When a mobile node has registered at the home agent, the agent sends out a gratuitous ARP over the local network. A gratuitous ARP is sent for updating the other nodes in the network ARP caches, and is sent like an ARP reply. The home agent now will receive all packets destined to the mobile node.

13 Tunneling

The source package openVPN [19] is used for handling the tunnel between the mobile node and the home agent. OpenVPN is started with config files on both ends. The packets transferred to the tunnel are encapsulated with IP-in-IP encapsulation. The tunnel is using UDP as transport protocol, but the encapsulated packets may run any transport protocol. So if an application (e.g HTTP) is using TCP the transfer is still reliable, since it's just the encapsulated packets that are sent over UDP. The tunnel contains four IP-addresses, two that are the IP addresses of the physical devices and two that are virtual. The two physical addresses are used for sending the packets physically. When a packet is sent to one of the virtual IP-addresses, it will be sent to the tunnel with the virtual IP addresses as source and destination IP addresses. When a handover occurs the physical IP-address of the tunnel endpoint on the mobile node side is switched. The tunnel at the home agent side accepts incoming packets from different physical IP-addresses as long as the virtual IP-addresses match. The mobile node and the home agent may share keys for avoiding hijacking of tunnel connections. Both static keys and certificate-based public key encryption may be used in openVPN. No keys are used in this prototype. Figure 23 shows a tunnel where the mobile nodes home address is the virtual address at the mobile node side. The virtual IP address on the home agent side is just a private IP address. The home agent application uses the addresses 10.1.0.1 to 10.1.0.10 for virtual tunnel start points. The physical address at the home agent side is the public IP address for the home agent. The physical address on the mobile node side is the currently active device.

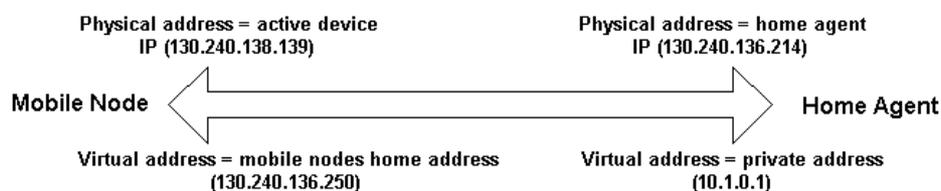


Figure 23: Tunnel configuration

Figure 24 shows how packets sent from a mobile node to a corresponding node via the home agent are encapsulated. Packets sent from a correspondent node to the mobile node via the home agent are also shown in the figure.

Mobile node home IP address: 130.240.136.250
 Home agent IP address: 130.240.136.214
 WLAN IP address: 130.240.138.139
 Correspondent Node IP address: 40.10.10.20

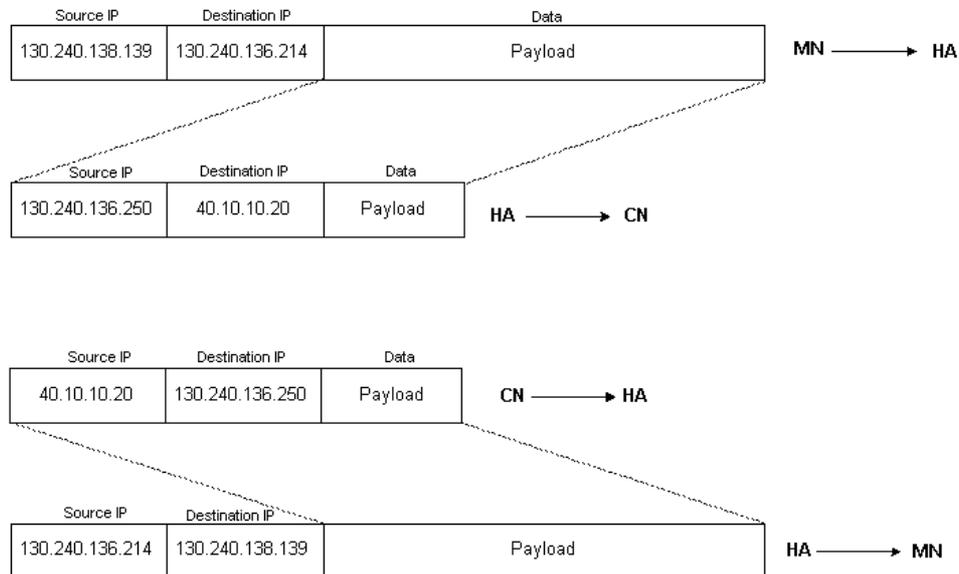


Figure 24: Packet encapsulation

14 Protocol

The protocol for this application is similar to Mobile IP [3]. All Mobile IP options and extensions are not implemented but the main idea is the same and the formats of the packets are similar. The registration request packet can be seen in figure 25. The field type is set to 1 in a registration request and 3 in a registration reply packet. The Code field isn't used, but in the original Mobile IP the field is used for options. Lifetime is number of seconds remaining before the registration is considered expired. The field home address is set to the reserved home IP address at the home network. The field home agent is the IP address of the home agent. The Care-of address is the address of the device that sends the registration. Identification is just a 64-bit random number that the mobile node sets in the start of the application. It's for matching registrations with replies, and for protecting against replay attacks of registration messages.

Type	Code	Lifetime
Home Address		
Home Agent		
Care-of Address		
Identification		
Local IP device		

Figure 25: Registration request

The registration reply is similar to the registration request as seen in the figure 26. Type is set to 3 in registration reply packet. The field lifetime indicates the time there is left before the registration for the mobile node at the home agent expires. The field receivers IP is the IP address that the reply packet was sent to. This is for the mobile node application, so it knows which device the reply packet was aimed for. The tunnel endpoint is the IP address of the virtual IP address on the home agent side.

Type	Code	Lifetime
Home Address		
Home Agent		
Identification		
Receivers IP		
Tunnel endpoint		

Figure 26: Registration reply

15 Handover cases

All handovers are performed and controlled by the mobile node. There are three different cases when a handover can occur. Figure 27 shows a handover performed manually. This handover is done by pressing the button for the desired device to be active in the GUI. When the mobile node performs a handover the local IP address for the tunnel is rebound to a new IP address. The configuration file for the openVPN process is rewritten with the new local IP address. A signal is sent to the openVPN process that restarts and rereads the configuration file. The tunnel is now locally bound to a new IP address. The openVPN process sends a ping message to the home

agent. The openVPN process on the home agent side is configured to accept packets from different IP addresses as long as all authentications are passed. The virtual IP addresses and keys must match. Since the openVPN process on the home agent side discovers that a packet from a new IP address has arrived it will rebound the active session to the new remote IP address. Figure 28 shows a case where the WLAN device has lost its connection to the home agent. The timer for the WLAN device will expire and a handover to the GPRS device will be performed. In figure 29 the handover is performed because of that the WLAN device couldn't send the registration request. An exception is caught and a handover is performed. If a registration request couldn't be sent the IP address for that device will be reread. If it is a new IP address the socket will rebound to the new IP address and the application will continue to try sending registration requests. This situation can occur when the mobile node has moved from one hotspot to another and the WLAN device has been assigned a new IP address.

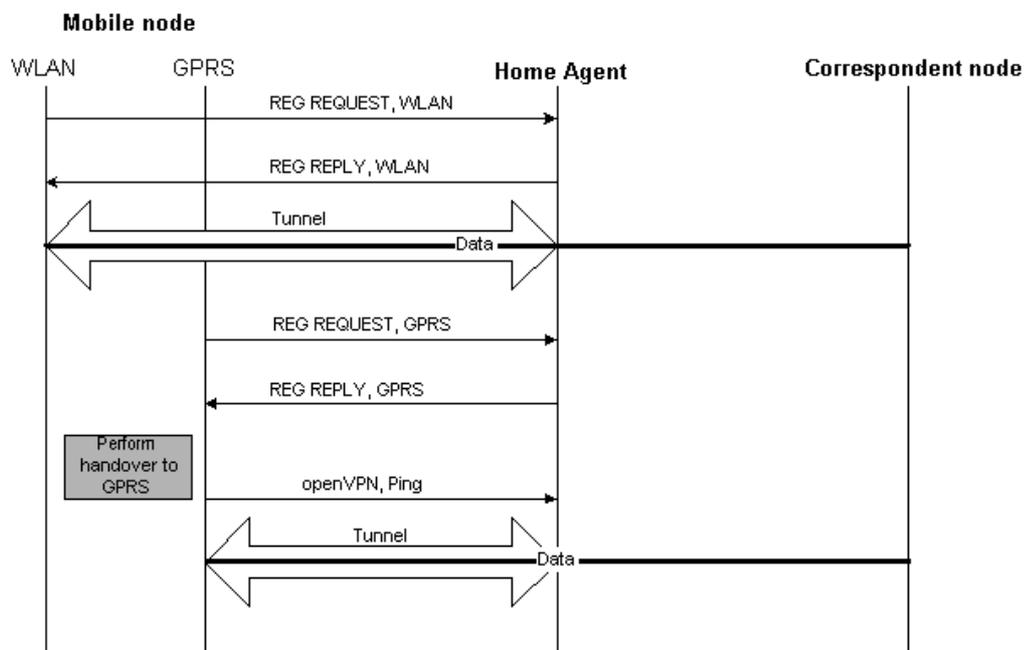


Figure 27: Packet exchange when performing a handover manually.

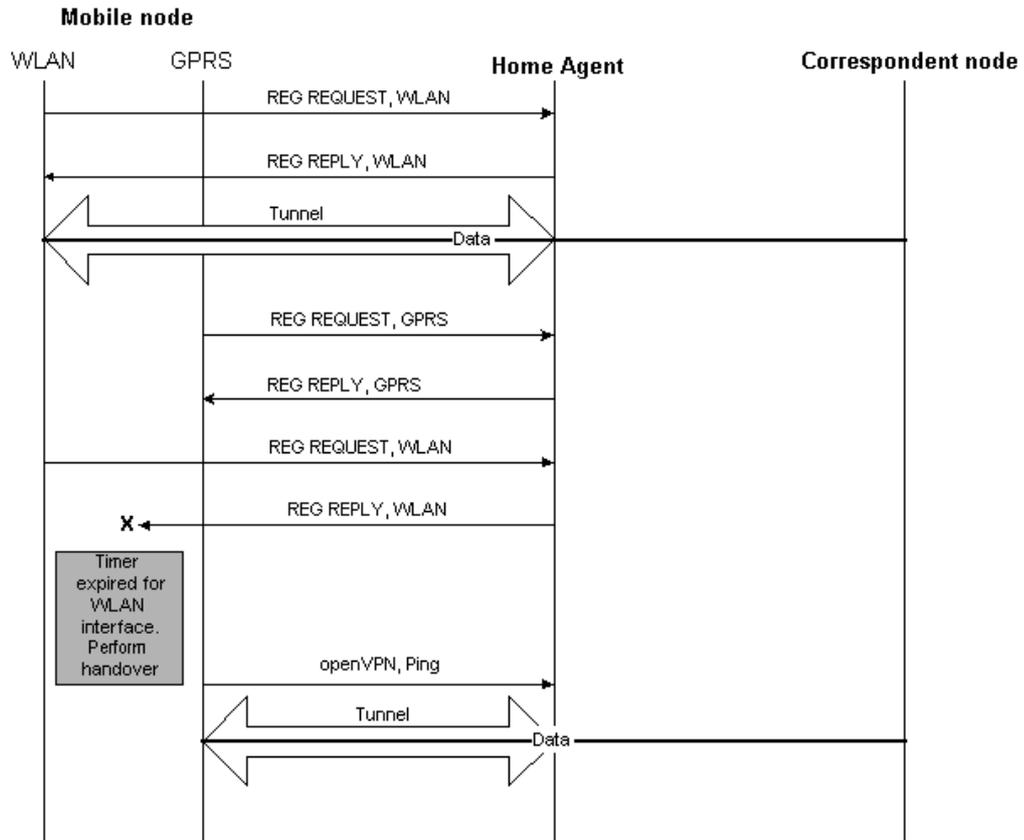


Figure 28: Packet exchange when WLAN interface loses its connection and a handover is performed.

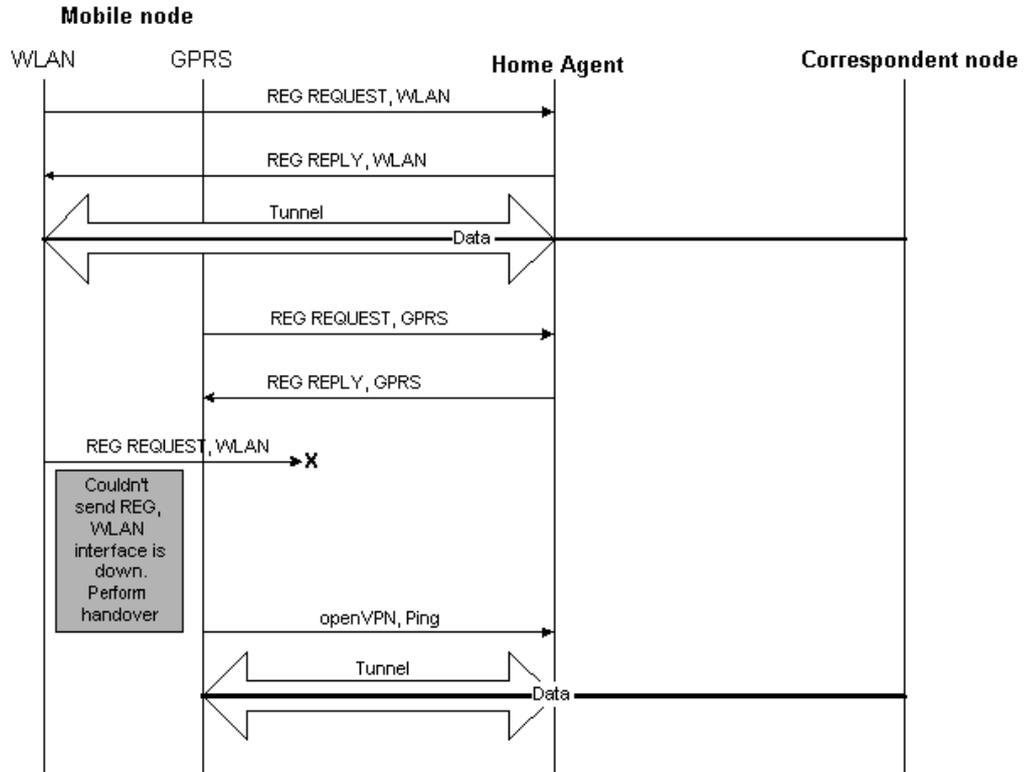


Figure 29: Packet exchange when WLAN interface is down and can't send a registration request. Handover to GPRS interface is performed.

16 Tests

Initially tests were done in a local environment. The first handover tests were done between a WLAN device and an Ethernet device in a private environment. Tests in a public environment were done at Campus in Skellefteå. The public WLAN Mobile City was used for the tests. Initially all handover tests were done with the command **ping**. The tool Ethereal [24] was used for reading the packets that were sent over the different devices including the tunnel. Ethereal were also good for confirming that a handover actually had taken place. Tests with the web browser Opera has been done for testing handovers with an application that uses TCP as transport protocol. Also an own written TCP application was used in the beginning for testing handovers with TCP. Satisfying tests where all nodes were on different LAN's were done. The main part of the testing has been done when the mobile node initiates the communication with the correspondent node. Although tests where the correspondent node initiates the communication also has been done. Most of those tests have only been done with **ping**. Although the tests only has involved handovers between WLAN, Ethernet, 3G and GPRS interfaces the system should work with other types of connection as long as a device has been assigned an IP address.

17 Conclusions

The result of this Master's Thesis is a working prototype. A mobile node can be reached through a home IP-address with my solution. A mobile node can perform seamless vertical handovers from one interface to another and keep the session connectivity. The prototype system is application transparent and works with both TCP and UDP applications. Both the two main goals with this work were fulfilled.

17.1 Limitations

Since the prototype application only starts and restart the openVPN tunnel the communication between the mobile node application and the openVPN application is very limited. Both applications are running on different processes, so there is no communication between the processes during runtime. It would be desirable to handle return messages from the openVPN process when the process is started or restarted from the mobile node application.

To check whether an interface is up or not is done by registration and reply messages as mentioned. But since packets sent via a GPRS-connection costs money, it would be desirable to send as minimal amount of packets as possible. Regular traffic through the interface could work as confirming the connection instead of the registration and reply messages. If no traffic currently is sent over the interface the registration and reply messages can be used for confirming the connection status.

It's very easy to hijack a connection and an IP address at the home network since no keys are used for authentication between the mobile node and the home agent. It is possible the send the data encrypted over the tunnel. If the mobile node and the home agent negotiates about a key the traffic over the tunnel can be sent encrypted. Even the registration and reply messages should be sent encrypted to make this system a little bit more secure.

18 Future work

This prototype is using a very easy handover policy, when a registration reply wasn't received within a certain timeout the interface is considered down and a handover is performed to the first interface that is up. It should be able for the user to specify more complex policies. Such policies could involve parameters like signal strength on the different connections, demands on connection speed or cost. The mobile node performs every handovers in this prototype. It would be considerable to allow the home agent to perform a handover for the mobile node. In some situations the home agent may know more about the traffic load for the different interfaces. The home agent may be in the best position for making a correct decision for the best connection available. Implementing some election algorithm in the home agent would be possible to do. No route optimizations are implemented in the prototype, all packets sent from a mobile node will go through its home agent. Packets from a mobile node should be able to be sent directly to the corresponding node for a more efficient system. As mentioned the security in this prototype is very limited. A secure key exchange solution would be desirable for avoiding hijacking of connections.

19 References

- [1] GPRS, Cisco, Web site, 20 Dec 2004.
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122limit/122y/122yy8/ggsn31/ggsn31_c/ggsnover.pdf
- [2] DHCPv4, RFC 3456, <http://www.faqs.org/rfcs/rfc3456.html>
- [3] Mobile IP Design Principles and Practices, Charles E Perkins. ISBN 0-201-63469-4
- [4] SCTP, RFC 2960, <http://www.faqs.org/rfcs/rfc2960.html>
- [5] L. Ma, F. Yu, V.C.M. Leung and T. Randhawa, "A new method to support UMTS/WLAN vertical handover using SCTP", IEEE Wireless Communications, vol. 11, no. 4, pp. 44-51, Aug. 2004.
- [6] J. H. Saltzer, D. P. Reed, and D. D. Clark, "End-to-end Arguments in System Design," *ACM Trans. Comp. Sys.* vol. 2, no. 4, Nov. 1984, pp. 278–88.
- [7] A. Matsumoto, M. Kozuka, K. Fujikawa, Y. Okabe, "TCP Multi-Home Options", draft-arifumi-tcp-mh-00.txt, IETF Internet draft, Oct. 2003.
- [8] Domain Name System (DNS), RFC 1034,
<http://www.faqs.org/rfcs/rfc1034.html>
- [9] Secure DNS Dynamic Update, RFC 2137,
<http://www.faqs.org/rfcs/rfc2137.html>
- [10] Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS), RFC 3278, <http://www.faqs.org/rfcs/rfc3278.html>
- [11] Universal Seamless Handoff Architecture in Wireless Overlay Networks, Web site, 20 Dec 2004,
http://www.cs.ucla.edu/~cclljj/publication/2004/UCLA_CSD_TR040012.pdf
- [12] IEEE 802.11 standard
- [13] D. Maltz, P. Bhagwat, "MSOCKS: An architecture for transport layer mobility," In Proc. of IEEE Infocom, p.p. 1037-1045, March 1998.
- [14] SOCKS Protocol, RFC 1928, <http://www.faqs.org/rfcs/rfc1928.html>
- [15] A.C Snoeren, & H. Balakrishnan, (2000). "An end-to-end approach to host mobility", *Proc. ACM/IEEE Mobicom* (pp. 155–166). Boston, MA.

- [16] A. C. Snoeren, H. Balakrishnan and M. Kaashoek, “Reconsidering Internet Mobility”, In *Proc. 8th Workshop on Hot Topics in Operating Systems (HotOS-VIII)*, May 9, 2000, available at <http://nms.lcs.mit.edu/papers/migratehotOS.pdf>
- [17] R. Hsieh, Zhe Guang Zhou, A. Seneviratne, “S-MIP: a seamless handoff architecture for mobile IP,” In *Proceedings of IEEE INFOCOM 2003*.
- [18] Mobility Support in IPv6, RFC 3775, <http://www.faqs.org/rfcs/rfc3775.html>
- [19] OpenVPN – An Open Source VPN Solution, Web Site, 20 Dec 2004. <http://openvpn.sourceforge.net>
- [20] Mobile Communications, Jochen Schiller, 2nd edition. ISBN 0 321 12381 6
- [21] Network Address Translation (NAT), RFC 2663, <http://www.faqs.org/rfcs/rfc2663.html>
- [22] Address Resolution Protocol (ARP), RFC 826, <http://www.faqs.org/rfcs/rfc826.html>
- [23] Linux Advanced Routing & Traffic Control HOWTO, Web Site, 1 Mar 2005 http://www.ibiblio.org/pub/Linux/docs/HOWTO/other-formats/html_single/Adv-Routing-HOWTO.html
- [24] Ethereal, Web site, 1 Mar 2005, <http://www.ethereal.com>

Appendix A

Abbreviations

ARP	Address Resolution Protocol
AP	Access Point
ASCONF	Address Configuration Message
BSS	Access points that are within the same radio coverage form a BSS (Basic Set Service).
Cell	Coverage area for an access point or a base station.
CN	Correspondent Node
COA	Care-of Address
DNS	Domain Name System
ESS	A logical WLAN with one or more BSS:es form a ESS (Extended Service Set).
ESSID	ESS identifier, WLAN network name
FA	Foreign Agent
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
GUI	Graphical User Interface
HA	Home Agent
Handover	When a mobile node switches the radio signal from one base station to another.
Handoff	See handover.
Hotspot	A specific geographic location in which an access point provides public wireless broadband network services to mobile visitors through a WLAN.
IEEE	Institute of Electrical and Electronics Engineers

IETF	Internet Engineering Task Force: IETF is the standardization organization for the Internet community.
IP	Internet Protocol
MAC	Medium Access Control
MAC-address	Unique hardware address that identifies a node in a network.
MN	Mobile Node
mSCTP	Mobile SCTP
MSOCKS	A transport layer solution which is built around the technique TCP Splice
NAT	Network Address Translation
NAPT	Network Address and Port Translation
PDN	Public Data Network
PPP	Point-to-Point Protocol
SCTP	Stream Control Transmission Protocol
SGSN	Service GPRS Support Node
S-MIP	A Seamless Handoff Architecture for Mobile IP
TCP	Transmission Control Protocol
TCP MH	Transmission Control Protocol Multi-Home
TCP Splice	A Technique that splits two independent TCP connections at one machine forming a single end-to-end TCP connection between the endpoints of the two original connections.
Tunnel	Path followed by a packet while it's encapsulated.
UDP	User Datagram Protocol
USHA	Universal Seamless Handoff Architecture
WLAN	Wireless Local Area Network

Appendix B

User Guide

This user guide describes how to configure and start both the mobile node and the home agent.

Mobile Node

Requirements

The mobile node only works with Linux and Java runtime environment (JRE) has to be installed. The Linux kernel has to be compiled with these network device supports:

- Universal TUN/TAP device driver
- PPP (point-to-point protocol)
- IP: advanced router
- IP: policy routing

The open source package openVPN has to be installed at the system. Install the openVPN rpm file and configure the TUN/TAP device node:

- `Mknod /dev/net/tun c 10 200`
- `Modprobe tun`
- `Echo 1 > /proc/sys/net/ipv4/ip_forward`

Look at the openVPN how-to [19] for more details.

The mobile node also needs a reserved public home IP address at the same local network as the home agent.

Configuration

When the devices for the mobile node are configured and have received an IP address the file `/etc/iproute2/rt_tables` has to be edited. All interfaces will have its own routing table when the application is running. For every interface a name has to be specified in the configuration file for the mobile node. Use the same names to create when editing the

`/etc/iproute2/rt_tables` file. Example: If the mobile node has two interfaces, a wlan and a gprs connection, add the tables.

- `Echo 200 wlan >> /etc/iproute2/rt_tables`
- `Echo 201 gprs >> /etc/iproute2/rt_tables`

This only has to be done once.

In the configuration file (`mn.config`) the parameters listed below has to be specified.

- Mobile nodes home IP address (has to be at the same local network as the home agent)
- Home agents IP address
- Number of interfaces
- Device name (for each device)
- Type of connection (for each device and must be the same name as the entries in the `/etc/iproute2/rt_tables` file)
- Registration request interval (how often the mobile node will send a registration request to the home agent, for each device in milliseconds)

- Registration reply timeout (for each device in seconds / 10)

Example of a configuration file with two devices:

```
mn-home 130.240.136.250
ha 130.240.136.214
interfaces 2
dev1 ath0
type1 wlan
reginterval1 3000
replytimeout1 20
dev2 ppp0
type2 gprs
reginterval2 10000
replytimeout2 200
```

The file `/etc/resolv.conf` has to be configured since the DNS-queries will be sent with the home address as source address. The DNS should be configured to a DNS-server appropriate for the home address. Edit the file `resolv.conf` in the same directory as where the application will be started. The file `resolv.conf` will be copied to `/etc/resolv.conf`. When you shuts down the application the original `resolv.conf` will be copied back to `/etc/resolv.conf` to restore the original DNS configuration.

Start

Enter **java MobileNode** at the prompt. When the application has started the routing rules can be viewed with the command **ip rule list**. **Ip route list table wlan** lists the wlan routing table.

GUI

When starting the mobile node application a graphical user interface (window) appears at the screen. All interfaces are listed in window. To change connection, simply press on the radio buttons below **Active** for the corresponding interface. When you close the application it's important that you do it by pressing the **Quit** button on the GUI. Else the added routing rules and the routing tables will not get deleted after the application has run.

Home Agent

Requirements

As for the mobile node the Linux kernel has to be compiled with universal TUN/TAP device driver. Install the openVPN package and configure the TUN/TAP device as described for the mobile node. It's very important that that IP-forwarding is enabled at the home agent, else will incoming packets to the home agent destined to the mobile node be discarded.

Configuration

In the configuration file (`ha.config`) the parameters listed below has to be specified.

- Network device for the home agent
- MAC address for the network device

Example of a configuration file:

```
Interface eth1  
Mac 00:C0:4F:A9:1E:EA
```

Start

Enter **java HomeAgent** at the prompt.