

Abstract

When moving between points of attachment in a Mobile IP enabled cellular network, which causes a change of the local IP-address, certain handoff mechanisms can be implemented to avoid disruptions in the ongoing sessions. We have implemented a Route Optimisation mechanism that will reduce the total latency. By tunneling the packets directly from correspondent node to the mobile node, instead of sending them via the home agent, as in Mobile IP triangular routing, we may reduce the delay between sender and receiver. This shorter route will also reduce the amount of packets being on the way to the old care-of address immediately after the mobile node gets a new address and the packet loss is reduced. The correspondent node receives binding updates from the Mobile Node (MN) and starts sending the packets to the new care-of address. An agent in the mobile node takes decisions on which correspondent node to send binding updates to based on information about e.g. the services running. To reduce the packet loss we have been testing when to send these updates in relation to when the new care-of address is activated. The agent in the mobile node should also take such decisions.

Our route optimisation mechanism is implemented on Flying Linux based portable computers and Foreign Agents. Its performance was measured on a Mobile IP network and compared with basic triangular routing.

Terminology

The following entities are defined in the RFC 2002:

- **Mobile Node (MN):** A host or a router capable of changing its attachment point from one network/subnetwork to another without changing its IP address or interrupting existing communications.
- **Home Agent (HA):** A router resident on MN's home network and forwarding traffic destined to the MN through encapsulation. HA also maintains current mobility bindings for the MN.
- **Foreign Agent (FA):** A router on the mobile node's visited network that provides routing services to the mobile node while it is registered.
- **Home Address:** The IP address by which a mobile node is known to its Correspondent Nodes (CN). It remains fixed as the mobile node moves through the Internet.
- **Care-of Address (COA):** An IP address either of a FA (Foreign Agent COA) or temporarily (Co-located COA) assigned to a MN. COA represents one end of a tunnel (the other end is the HA), allowing tunneling of traffic to the MN.

The following functions are defined in the RFC 2002:

- **Agent Discovery (AD):** The process by which a newly attached MN learns about the identity of HA or FA and obtains a foreign agent COA. AD is performed either through agent advertising their availability or through MN solicitation.
- **Registration (RG):** The authenticated process through which the MN informs its HA of its current COA. RG can be performed directly with the HA or through the FA.
- **Encapsulation (EP):** Also known as tunneling, the process allows forwarding of intercepted datagrams by the HA to the COA. This is accomplished through enveloping the intercepted datagrams into another IP datagram.
- **Decapsulation (DP):** The reverse process of EP. Performed at the COA, the original datagrams are extracted from the enveloped datagrams.

1 Introduction

1.1 Background

Wireless attachments to the Internet require mobility management for having the devices stay connected while moving between different points of attachment. Changing point of attachment increases the risk of packet loss, which can have a considerable impact on the ongoing session and cause disturbing disruptions. Different handover mechanisms are needed to maintain connectivity and

minimize the disruption of ongoing transfers. The Mobile IP standard [2] specifies a general handoff mechanism, which enables mobile nodes to change their point of attachment to the Internet without changing their IP-address.

Mobile IP can handle both local area and wide area movement in both wired and wireless networks. However, it requires that a mobile node's home network is notified of every change of location. When moving among cells within a visited network, the "care-of address" has to be registered with its home agent. If the distance between the visited network and the home network of the mobile node is large, the signaling delay for these registrations may be long.

In basic Mobile IP, packets destined for the mobile node are tunneled from the home network to the visited network. Correspondent nodes that want to send packets to the mobile node have to send them via the home network. This causes non optimal triangular routing (Figure 1.1). Even worse is that when a mobile node communicates with a correspondent node within the same visited network as it resides, packets still have to be routed via the home agent.

A route optimisation option [3] to Mobile IP can eliminate this triangle routing by allowing correspondent nodes to cache bindings of the mobile nodes current location. They can then tunnel the packets directly to the mobile node (to its care-of address) without going through the home network. Every new location has to be registered with hosts that are actively communicating with the mobile node.

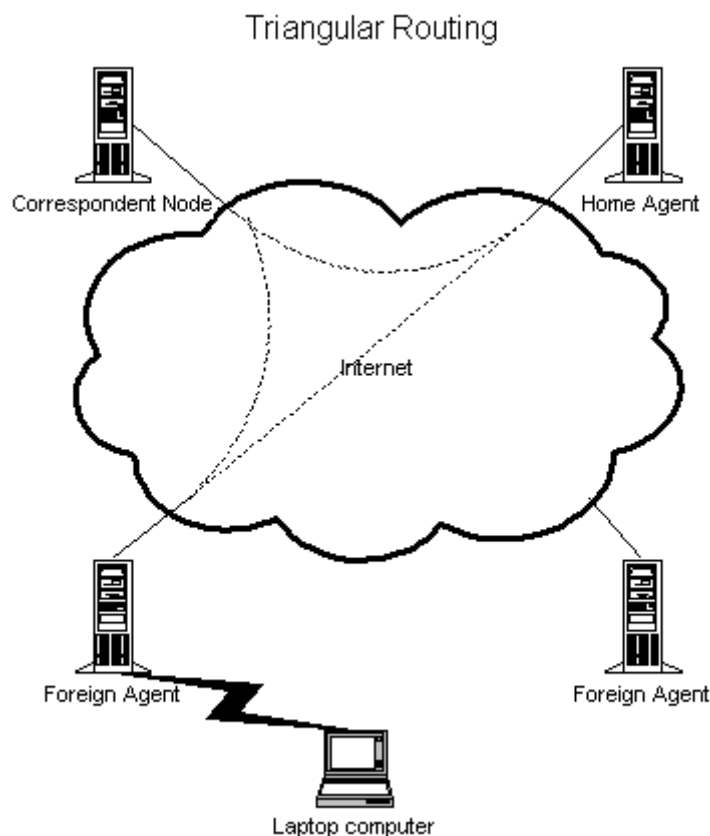


Figure 1.1: Non optimal triangular routing in basic Mobile IP

Due to ingress filtering in the border routers of the visited network, IP version 4 of Mobile IP (MIPv4), together with its route optimisation option, may be difficult to run in practice. RFC 2267 [6] (Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing) states: "All providers of Internet connectivity are urged to implement filtering described in this document to prohibit attackers from using forged source addresses which do not reside within a range

of legitimately advertised prefixes". Packets originated at a mobile node attached to a foreign network will not have a valid source address according to these filtering rules. Therefore, packets will then be dropped when ingress filtering is implemented.

In IPv6, the support for route optimisation is built in as a fundamental part of the protocol, rather than being added on as an optional set of extensions as in Mobile IPv4.

Support is also integrated into Mobile IPv6, and into IPv6 itself, for allowing mobile nodes and Mobile IP to coexist efficiently with routers that perform ingress filtering. A mobile node now uses its care-of address as the source address in the IP header of the packets it sends, allowing the packets to pass normally through ingress filtering routers. The home address of the mobile node is carried in the packet in a Home Address destination option, allowing the use of the care-of address in the packet to be transparent above the IP layer.

1.2 Scope of the Study

The objective of this study is to investigate route optimisation in Mobile IP. We want to study and quantify the effects it can have on the handoff performance.

We have implemented a route optimisation mechanism on top of the Mobile IP network. The implemented mechanism is similar to the route optimisation used in MIPv6. By sending a care-of address from the mobile node directly to the correspondent node (not via the home network as in MIPv4) to be cached there, a tunnel could be set up directly from the correspondent node to this care-of address. Similar to previously discussed route optimisation mechanisms, packets will bypass the mobile node's home agent and avoid triangular routing. There are delays and packet loss related to handovers, which we need to minimize to avoid disruptions in the ongoing sessions.

The route optimisation mechanism developed has the ability to run in three different modes with different features. When a handover is about to occur, the new care-of address can be sent to the correspondent host either before or after the actual handover is made. The care-of address is cached at the correspondent host and datagrams are now tunneled directly to the new Foreign Agent to which the mobile node is attached. Datagrams are detunneled and delivered to the mobile node. To show the benefits of this mechanism compared with basic Mobile IP, the mobile node needs to be far from its home network and close to the correspondent host.

2 Related Work

Department of Informatics at the University of Oslo, Norway, is active in the field of mobile networking. They have established a Wireless Experimental Metropolitan Area Network (WEMAN)[8], based on Mobile IPv6 for mobility and IEEE 802.11 technology for wireless links and access. They present a feasibility study of Metropolitan Area Network using Wireless LAN technology and explore four options of protocol support for mobile clients, ranging from simple DHCP solutions and IPv6 Route Advertisements through Dynamic IPv6 tunneling and finally Mobile IPv6 support, the latter implemented for Linux by University of Lancaster [10].

Aruna Seneviratne and Behcet Sarikaya [11] present several techniques and tools developed to provide wireless data services to applications with a focus on multimedia applications. They study performance degradation due to Mobile IP operations and conclude that the triangle routing hampers the performance most, suggesting the need for route optimisation. TCP which provides data integrity is shown to perform poorly in mobile links due to its congestion control mechanisms. The writers also discuss several proposals for handover optimization.

A handover mechanism for internetworks that include networks of small wireless cells populated by large numbers of portable devices is presented by Ramon Caceres and Venkata N. Padmanabhan [12]. According to the writers, such networks offer high aggregate bandwidth, support low powered mobile transceivers, and provide accurate location information. In these networks, users will often carry devices across cell boundaries in the midst of data transfers. They mean that Mobile IP does not meet the requirements of such networks including low latency and little or no data loss and therefore suggest a hierarchical mobility management scheme including Mobile IP for global mobility, that is, movement across administrative domains.

The 3rd Generation Partnership Project in its document about combined GSM and Mobile IP mobility

handling in UMTS [13] makes a feasibility study on using Mobile IP as a tunneling and mobility management protocol in combination with GSM/UMTS mobility management in the packet domain of UMTS.

3 Mobility in the Internet (Mobile IP)

The TCP/IP protocol suite is the most extensively used protocol for computer communications. Both TCP and IP were designed for stationary networks that seldom change their point of attachment to the Internet. The IP addresses for machines serve two purposes: they specify the identity of a machine and they determine its point of attachment. Thus IP address determine the routing of a datagram in the Internet.

For a mobile machine, changing its point of attachment to the Internet means that a route to the host cannot be determined from its IP address. Since IP addresses also represent the identity of a machine, they cannot be arbitrarily changed without the need to notify all the machines that may communicate with the mobile machine in question. Thus, clearly, a modification to the IP is required to support mobility in the IP networks.

The Internet Engineering Task Force (IETF) proposed a solution to the mobility in the Internet in its RFC 2002 document [1, 2].

3.1 Basic Mobile IP Architecture

Based on the traditional IP routing mechanism, datagrams destined to a mobile node (MN) will end up in the gateway of its home network. In so called triangular routing, when a mobile node is away from home, its datagrams are captured by a process called home agent (HA), which runs in the mobile's home network.

In MIPv4, the HA, which masquerades as the away-from-home mobile, encapsulates the captured datagram as a new datagram with destination address equal to the IP address of a process running on the visitor network and termed foreign agent (FA). The FA, upon receipt of the encapsulated datagrams, decapsulates them and delivers the datagram locally to the visitor mobile node.

In MIPv6 [4], each time the mobile node moves from one subnet to another, i.e. performs a Handover, it gets a new so called Co-located COA by an address autoconfiguration mechanism. Here again, the HA intercepts any packets addressed to the mobile node's home address but tunnels them directly to the mobile's COA by means of IPv6 tunneling mechanism.

Datagrams from any fixed host to a mobile node always go through the home network and are then redirected to the visitor network. However, datagrams from the mobile node to a fixed host, referred to as a corresponding host (CH), travel directly through the Internet without any redirection. Thus traffic between a MN and a CH create a "triangular" shape.

Through another process, termed the Registration process, the HA and the FA learn about the current point-of-attachment of the mobile to the Internet.

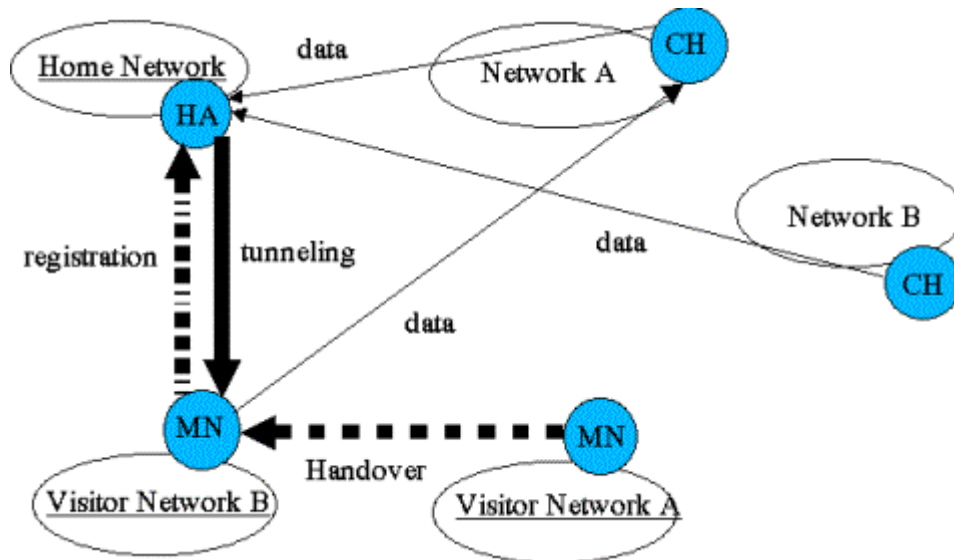


Figure 3.1 Triangular routing

3.2 Route Optimization

There are two sources of overhead associated with the triangular routing mechanism discussed in the preceding paragraph. It causes both an additional load in the home subnet and more latency time in transferring traffic to destination.

Route optimization techniques make extensions to the base Mobile IP protocol described earlier. Route optimization provides a means for nodes that implement them to cache the current location of a MN and then tunnel their own datagrams for the mobile node directly to that location, bypassing the possibly lengthy route for each datagram to and from the MN's home agent.

The *mobileip* working group within IETF has specified a Route Optimization protocol for both MIPv4 and MIPv6 [3].

3.2.1 Proposed scheme for Mobile IPv4

In the protocol specified for IPv4, the HA sends an appropriate Binding Updates message to the CH every time it receives a datagram intended for the MN, informing the CH about the mobile terminal's COA. The CH can subsequently send the datagrams directly to the MN's care-of address using IP-in-IP tunneling mechanism. A mechanism is also provided whereby the CH learn the mobile node's new location every time it gets a new COA during a session. As soon as the old FA receives data traffic from a CH, it sends the HA a Binding Warning message, asking that the CH be notified of the MN's new COA. The situation is illustrated below:

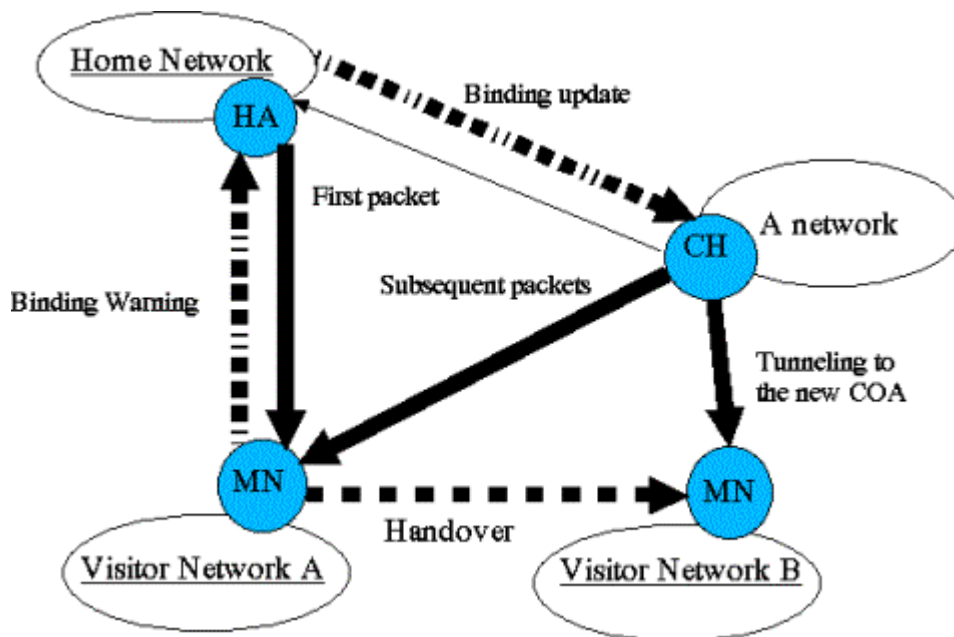


Figure 3.2 Route Optimization in MIPv4

3.2.2 Proposed scheme for Mobile IPv6

Mobile IPv6 introduces two new IPv6 Destination Options Header, namely a Binding Update and a Binding Acknowledgment. The destination options header is one of the so called IPv6 extension headers which is treated only by the final destination. The mobile node can send directly a Binding Update in the same packets carrying effective traffic to its correspondent nodes, which can then learn and cache the new mobile's care-of address. This minimizes signaling traffic. As a result of this mechanism, when sending a packet to any IPv6 destination, a host must first check if it has a binding for this destination. If a cache entry is found, the host sends the packets directly to the care-of address indicated in the binding, using an IPv6 Routing Header, i.e. a special extension header that forces the datagram to follow a predetermined route. This eliminates triangular routing. If no binding is found, the packet is sent to the mobile node's home address, which tunnels it to the care-of address as described previously. Mobile IPv6 also uses the IPv6 Neighbor Discovery protocol to perform a number of mobility related tasks. The situation is illustrated below:

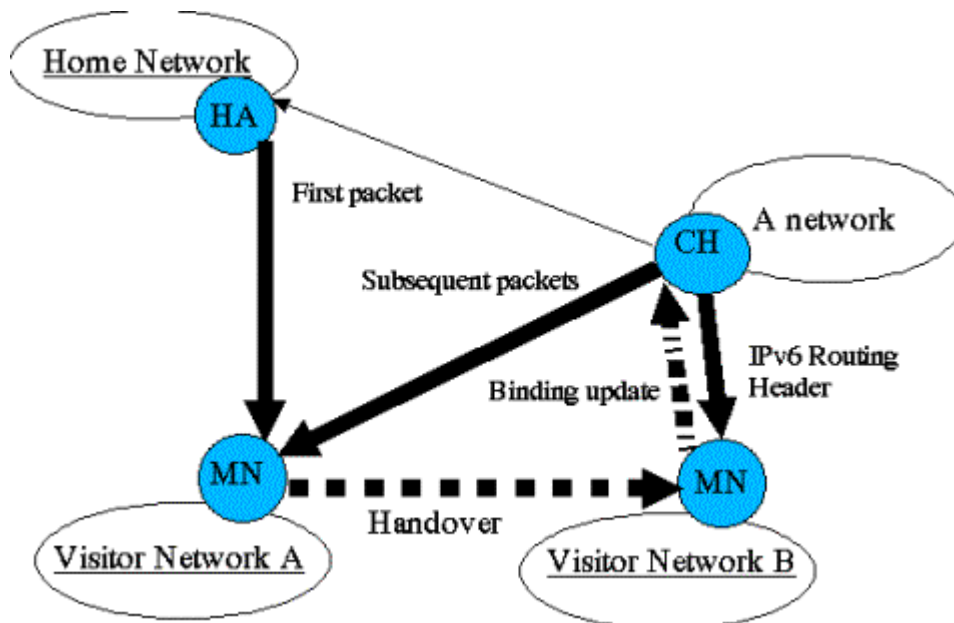


Figure 3.3 Route Optimization in MIPv6

4 The Test Network

The test network architecture consists of the FlyingLinux environment for the 2G1303 finger course, which took place at KTH/IT during the spring semester of year 2000 (figure 4.1). A picture describing the infrastructure can be viewed below.

4.1 Network Architecture

The public part of the network is separated into three subnets by the three routers/gateways. A masquerading server connects the private part of the network (not used for Mobile IP) to the public network. In addition to the infrastructure depicted below there are numerous HUBs of varying kind deployed.

The wireless network consists of Lucent Technologies ORiNOCO WavePoint II base stations working on different channels and Foreign Agents (FA) with Lucent WavaLan Bronze PCMCIA cards running on channel 2 in ad-hoc mode. The WavePoint II base stations are deployed at various locations and are connected to all the three subnets of the public network as well as to the private network. Most of the Foreign Agents (FA) are connected to the public subnet 130.237.14.64/26. The Home Agent (HA) as well as a FA is connected to the 130.237.15.192/26 subnet.

For the specific test scenario described in chapter 8 only the FA environment of the *wireless* network was used. To make measurement and analysis easier only FA 130.237.14.77/78 and FA 130.237.14.79/80 was used and monitored by connecting NetVCR to two separate HUBs placed in-between the FAs and the public subnet 130.237.14.64/26.

Firstly, an access point can filter traffic based on IP multicast groups and forward only the multicast traffic that has a receiver. Such filtering is especially important given the limited bandwidth of wireless links.

Secondly, a network layer access point can differentiate between packet types, based on the 'type of service' field in the IPv4 header or the 'flow id' field in the next generation IP-headers. This is especially important at the interface between a wired link and a slower wireless link.

Mobile IP provides a best effort service when it comes to handovers, it does not provide any explicit mechanisms for handing over active connections. Instead it assumes that packets that were routed to the old FA during a handover will be lost. Recovery from these errors is left to the higher level protocols. This is a potential weakness for Mobile IP because real-time applications can have certain Quality of Service (QoS) requirements, such as constraints on packet loss, delay and jitter.

5 Implementation of Route Optimization

The main purpose of our route optimization program is to investigate the behavior of a MN when visiting a foreign network. The program is a Java application that provides the user with an easily managed interface that lets the user actively take part in the decisions of when and how to make handovers between FAs.

The special implementation of Route Optimization tries to combat the weaknesses of Mobile IP, and in addition offers a new solution to the Quality of Service problem related to handovers. The migration from MIPv4 to a 'MIPv6 route optimization over MIPv4' simulation.

5.1 Basic Idea and Challenges

The purpose of our program is to extend the Dynamics HUT implementation of MIP with route optimization and to investigate the possibilities to make smoother handovers between different FAs. At first we examined the possibilities of changing the Dynamics code that was available to us. But in order to extend the Dynamics implementation with route optimization we would have to read hundreds of pages of C code and know where to make the necessary changes. Moreover it is also necessary to set up an automatic tunneling system in the Linux environment used in this project. To be able to perform tunneling in Linux, as well as to access most network specific information one has two choices, either to tap into the kernel directly to extract information (only for experienced users), or use available and ready software. The advantages of going directly to the kernel are obvious, since this gives full control and choice of the information. This kind of work is for an experienced Linux user and kernel programmer, and thus we had to find an alternate solution.

Since most of the members of the project team have had experience with Java it seemed to be an good choice. And indeed, the use of Java as our programming language proved to be excellent one, since the goal of our project was only to *investigate the behavior* of route optimization and *not* to implement a fully functional version in the Linux kernel.

Another challenge was about a choice between implementation of the route optimization extension to the MIPv4 and MIPv6. Due to just having MIPv4 implementation in our network, it seemed natural to only add this extension to the MIPv4. But there are some drawbacks in the route optimization in MIPv4 in comparison to MIPv6. Route optimization in MIPv4 adds to the complexity of the HA and requires security association between the HA and all CH's. Furthermore it still requires packets to be tunneled from the CH to the COA. In contrast, route optimization in MIPv6 removes the need to tunnel packets. The MN also has more control over deciding when to optimize routes since it creates the optimized route rather than the HA. In MIPv4 reversed tunneling is required to avoid ingress filtering [6] problems (where firewalls drop the mobile's outgoing packets) since packets are sent with the home address as the source. In MIPv6 packets may be sent with the COA as the source address, hence there should not be any problems with ingress filtering.

Because of the above reasons, a decision was made to simulate a fundamental MIPv6 route optimization behavior between the MN and CH.

5.2 Java Runtime Environment

Instead of going directly into kernel programming we developed our program as a Java application that performs a number of different runtime system calls and executes already existing software. The disadvantages of this are the direct opposite of writing your own, you rely on what the software can do, not what you want it to do. The result is a less optimal use of resources and requires the use of some ingenuity on behalf of the programmer.

Due to the well known fact that the Java Runtime environment has many bugs and has a tendency of not delivering what is promised, the construction of a program which utilizes a number of different executable external software is bound to be somewhat troublesome. Consequently more time and effort than desired was spent on solving and experimenting with different use of the Java Runtime environment

5.3 Program Description

To run the route optimization scheme, a client is run at the MN and a server at the CH (a server at the FAs is optional). The application components function as daemons that run in the background. In order for the user to manage the program in an easy way, we developed a graphical user interface, which is easily handled by a new user. With this GUI the user is able to start or stop the daemon (labeled 1 and 2 respectively). The user is also able to get information about the different FAs (3 and 4). Moreover the user can set the daemon to work in three different modes (5, 6, 7), which are described below. The source code of the first version is available at the project site [14].

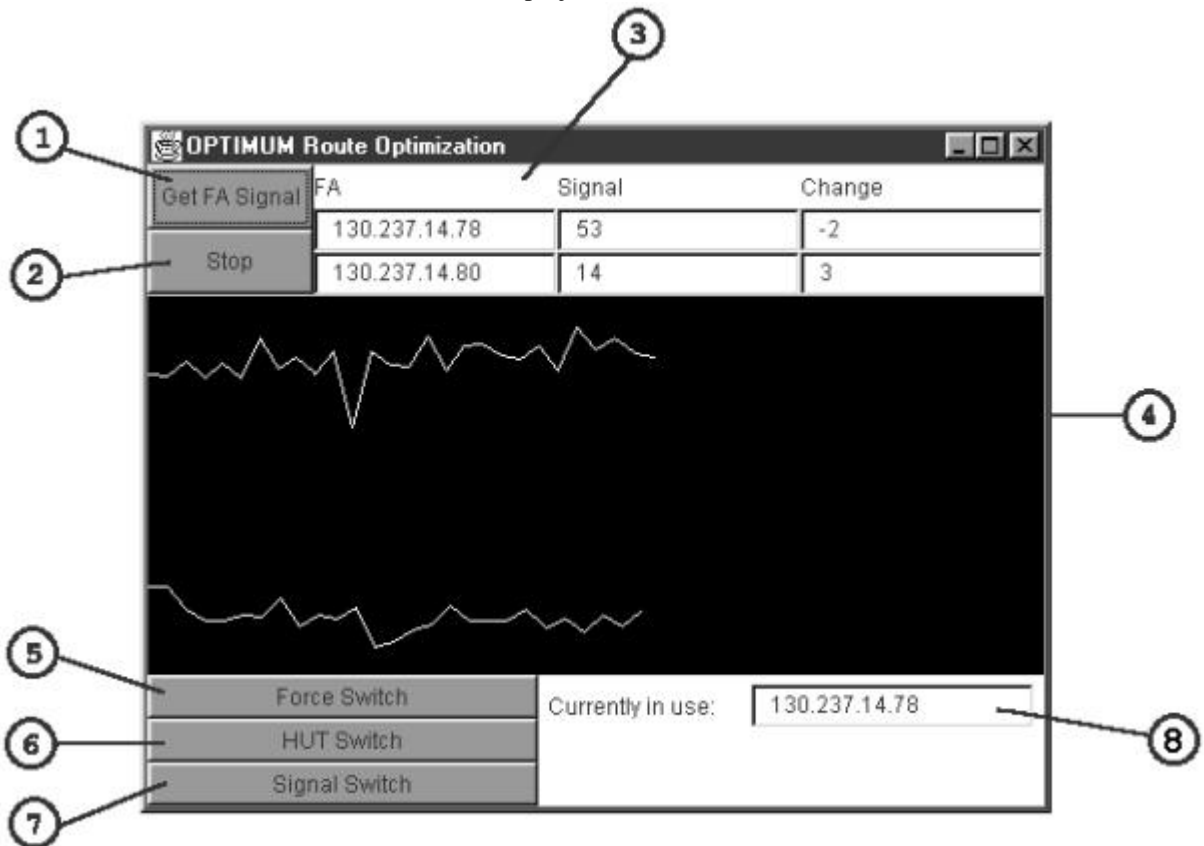


Figure 5.1: Graphical user interface

5.3.1 Mobile Node Components

The program at the MN performs various kinds of system calls and executes different software and makes different decisions based on the results of these calls. The calls to the Runtime environment are made periodically.

When executing the program the user gets running information about the FAs in range and it also shows which FA is currently in use. The program also keeps track of all the CNs it currently has a connection to, the state of the connection and which port numbers each connection uses.

As mentioned above there are three different modes (5, 6, 7) in which the program can run. These modes function in ways that seem only slightly different, but in fact they differ in ways that are of fundamental importance.

Force Switch

When clicking this button the program automatically changes FA no matter what the signal strengths of the FAs are. What makes this mode interesting is that the MN sends a BU to the CN *prior* to actually changing FA. This is an attempt to shorten the handover time. One has to note that when in this mode, the program is locked to one FA, and does not attempt to change FA even if the signal of the current FA is really low. The user has by himself click on this button to explicitly change FA again if he wants to.

HUT Switch

When using this mode the FA used is always the one that the Dynamics implementation thinks is the best. When the MN demon notices that a change of care-of address has occurred, it immediately sends a BU to the CN. The drawback of this mode is that the a BU is sent only *after* the actual handover is made.

Signal Switch

This mode is perhaps the most interesting one, and also the hardest one to implement in a good manner. This mode is supposed to look at the signal strength of the FAs and from this information calculate some kind of movement prediction. When detecting that there is time to change FA, the MN sends a BU to the CN, and after that it changes care-of address. This mode combines the best features of the previous two modes.

5.3.2 Correspondent Host Components

In order to make route optimization work at all, it is of course necessary to have a program running at the CH. This program also runs in the background as a daemon. Upon receiving a new BU from a MN, it sets up a tunnel to the MN's care-of address. This action prevents packets to go via the HA, i.e. route optimization is accomplished.

5.3.3 Foreign Agent Components

A nice but for the scope of this project non-wanted feature of the Dynamics Foreign Agent implementation is the effects of the ad-hoc mode in combination with the tunnel binding lifetime. It allows traffic destined to the old FA to be delivered to the MN unless it's out of range (a good enhancement for general use). Thus the experienced loss of packets (i.e. delay) will not be as easily noticeable. To avoid this good natured enhancement, an ability to shutdown this feature was implemented at the FAs. The only side effect of this is that the FA will decapsulate the packets and then route the original packets towards the home net, which is a non warranted behavior for Mobile IP.

5.4 Program Execution

The idea when writing the program was that it should work in mode *signal switch*, since this mode is the most interesting one. But with this mode some problems arose. When concerned with making correct decisions the amount of information available regarding the issue will greatly affect the process and result. All existing MIPv4 implementations use some kind of tool to measure the signal strengths of the different Foreign Agents to determine which FA to use (i.e. strong signal means source is close or that the reception of the signal is good).

The Dynamics implementation uses the Linux WaveLAN driver to get the signal strength information. An utility called *iw_spy* associated with this WaveLAN driver offers this functionality and is used in the MN client application. The only problem with this driver is that it is a passive recorder. This means that the driver itself cannot make sure that data is sent continuously to it (i.e. it doesn't waste

bandwidth) and thus can only do the measurements when data is being sent to it, when for example requested by higher levels (normally the IP protocol). Since the recommended shortest Agent Advertisement interval for MIPv4, as well as for the Dynamics implementation, is *one* second this means that all signal strength measurements of all *non sending* FAs will be less accurate. This fact makes it nearly impossible to make any kind of accurate movement detection, since the signal strength can vary quite a lot from one second to another. With a high concentration of many wireless devices communicating at close frequencies, the interference and noise will make any decision based on signal strength very difficult. As an example the interference with other devices caused ‘unexpected’ peaks (spikes) of the signal strengths, even for FAs very close to the MN.

6 Measurements

Real-time services have high requirements on low maximum end-to-end delay. This maximum is often considered to be around 150 ms, but may be stretched up to 400 ms [5]. Satisfying this requirement becomes even harder taking into account mobile users moving between different access points i.e. doing a handover.

Mobile IP will give users a great degree of mobility across private and public networks, but at the same time includes functions which are time consuming every time a handover is done, thus reducing performances of real-time services.

This project aims to study the differences in perceived end user Quality of Service on real-time services between the cases when basic mobile IP is used and when some route optimization mechanisms beside that are applied.

6.1 Test Specifications

The following functions in basic mobile IP are important for doing our measurements:

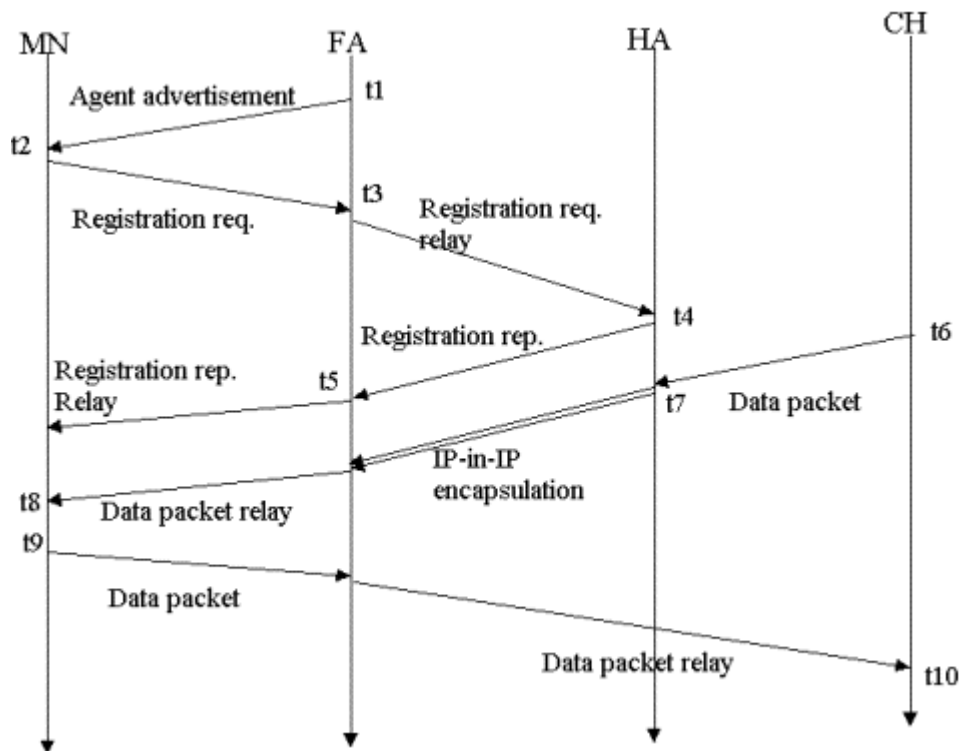


Figure 6.1 Time sequence of functions in basic MIP

Agent advertisement: At different instances t_1 and with different controllable time intervals, the existing foreign agents (FA) inform their availability to mobile node (MN). This is done through an ICMP message. $t_2 - t_1$ is negligible (a link layer communication between MN and FA).

Handover: At some time t_2 the MN decides to change the access net. This decision is implementation dependent and in our case is a function of agent advertisement signal strength sent by different FAs. At this time the MN has got a new care-of address (COA). It is also possible to force a foreign agent change, thereby forcing a handover.

Registration: The process through which the away-from-home MN registers its new COA with its home agent (HA). Registration is performed through the FA, which forwards the registration information to the HA. Both the registration request and registration reply are UDP datagrams.

Data packet between correspondent host (CH) and HA: This is an ordinary IP packet but we must somehow provide virtual delays in HA to be able to represent cases where HA and CH stand far from each other.

Tunneling: The process allows forwarding of intercepted datagrams by the HA to the COA. This is accomplished through enveloping the intercepted datagrams into another IP datagram.

In addition to the above functions, our proposed route optimization extension to MIP implements two user applications written in JAVA in order to reduce the end-to-end delays. The first one, which we can call Mobile Agent (MA), is responsible for sending a Binding Update directly to the CH, as soon as the MN has got a new COA (figure 6.2). In this way, the CH is provided with the MN's new COA earlier and can directly tunnel packets there. Direct tunneling (IP-in-IP encapsulation) is in fact the main responsibility of the second user process, called the Correspondent Agent (CA).

6.1.1 Measurements with Basic Mobile IP

1. Suppose that our MN is residing at a given place such as our room and has a stable COA (figure 6.1). Moreover we assume that both the CH and the HA are at fix places. Now we can, for example PING [7] from a CH to the MN and measure the time delays CH \rightarrow HA and HA \rightarrow MN. In order to compare base mobile IP with other enhancements presented later, the time delay HA \rightarrow MN should be rather long. This requires some type of virtual delay implementation in HA. If we define end-to-end delay (Tee) [5] as the time elapsed from when a CH sends a packet until it arrives at the MN, we have:

$$(Tee)_{old} = (t_8 - t_7)_{old} + (t_7 - t_6)$$

where old means the MN's previous point of attachment.

2. Measuring the time interval within which the transferring of a rather big file from a CH to the MN completes while the MN performs no handover is a good idea. This provides us with a measure to compare with the case where a handover actually occurs.

3. At some registered instance t_0 , when MN has a stable COA, the MN begins moving on a predetermined path until it sends its registration request corresponding to a new point of attachment. If we call this delay, change decision delay (Tcd), we have:

$$T_{cd} = t_2 - t_0$$

4. The time delay needed for registration request to reach the HA is also important since during that time the HA continues to send packets to the old COA. If we call this delay, registration request delay (Trr), we have:

$$T_{rr} = t_4 - t_2$$

5. The redirection delay (Tredir) [5], is defined as the amount of time when packets are lost after a MN has changed its COA. During this time interval, packets in-flight between the HA and the mobile user's previous COA plus the ones arriving at HA while

the registration request travels from the MN to the HA may be lost. We have:

$$T_{redir} = (t_8 - t_7)_{old} + T_{rr}$$

At some instance t we begin running the same file transferring as in (2) while MN is moving on the same path as in (3) and measure the following:

The instance t when file transferring begins

The instance t_2 when registration request is sent

The instance t_8 for the first packet arrived at the new COA

The number of end-to-end packet loss after t_2 and before the above t_8

The time interval within which the whole file transferring completes (the instance t8 for the last packet of the file)

6.1.2 Measurements with Proposed Route Optimization Extension

6. At some instance T, we begin running the same file transferring as in (2) while MN is moving on the same path as in (3) and measure the following (figure 6.2):

- The instance T when file transferring begins
- The instance T1 when Binding Update is sent by the HA process in the MN
- The instance T2 when Binding Update arrives at the CA process in the CH
- The instance T3 for the first packet tunneled by the CA after receiving Binding Update from HA
- HA
- The instance T4 for the first packet sent at T3
- The time interval within which the whole file transferring completes (the instance T4 for the last packet of the file)

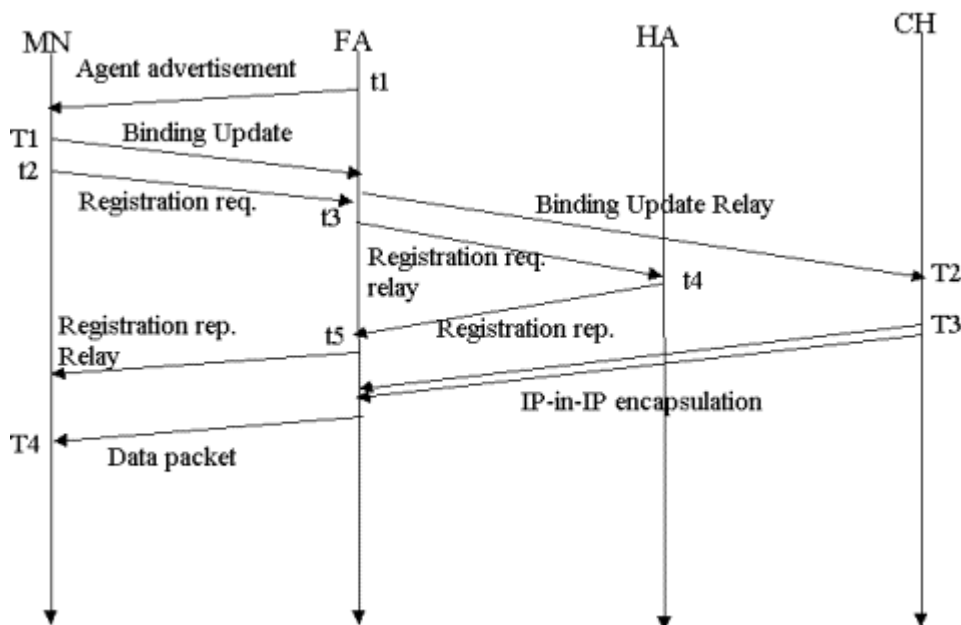


Figure 6.2 Time sequence of proposed MIP

6.2 Preliminary Test Results

In the Scope of this study we set out to investigate different ways of doing Binding Updates for a simple MIPv6 Route Optimization, mimicking the MIPv6 behavior of MN to CH Binding Updates. The first test results include two basic schemes for *when* to send BUs from the MN, namely

1. After a change to a known FA
2. Before actively changing to a new known FA

As a first preliminary test we investigated each of the above schemes five times while doing a handover (while downloading a large file from the CH). The reason we did not utilize *all* of the test specifications of chapter 6 was because that kind of in-depth measurements and analysis should be conducted under a more ideal environment. To this end these preliminary test results helped us confirm that. The basis of this can be read out of the discussions of previous chapters, the most important being :

- For a reliable and correct route optimization analysis the HA needs to be considerably further away.
- Foreign Agents which belong to more separated networks should be used.
- The Java Runtime system calls dominate the processing time of both the MN and the CH (a Linux hardware/driver programmer is needed).

6.2.1 Sending BU after changing COA

When sending BU after receiving a new COA it causes the packet loss from an already updated CH to increase, roughly proportional to the sum of the propagation time from the MN to the CH and the time it takes the CH to send and update it's BU cache. During that time, packets will be sent to the old COA and either be lost or retransmitted.

However our running Java program causes a much more complex 'black hole' time. Instead we made the preliminary measurements of the time between the registration reply was sent from the MN and the time the first 'new' packet arrived to the new COA.

Test #	Measured time
1	2387 ms
2	2853 ms
3	2173 ms
4	2412 ms
5	2340 ms

6.2.2 Sending BU before changing COA

The optimal result for this scheme is to have the time when the first packets from the CH arrive to the new FA, as close as possible to the time when the FA receives registration reply from the HA (See Figure 6.2 : T4 would be very near t5).

In this scheme the theoretical 'black hole' time is the time when the BU cache is updated in the CH, to the time when the new FA receives the registration reply from the HA. As in the case of sending BU after changing COA, the implemented Java program (i.e. the system calls) will influence the 'black hole' time. Here the preliminary tests measures the time between when the BU was received by the CH, and the time the new FA received the registration reply from the HA.

Test #	Measured time
1	2000 ms
2	1941 ms
3	1736 ms
4	2103 ms
5	1855 ms

6.2.3 Short comments on the preliminary measurements

The reason why the measured times for both the schemes are so long is simply the Java application. The system calls causes the extremely long measured times. Inbedded in the times are many costly operations such as tearing down a tunnel and building a new (a total of 6 system calls). When forcing a switch of FA the time until an agent advertisement arrives can be (if one was just missed) around one second.

With some extensive additional work from an experience Linux kernel and driver programmer the overhead caused by Java and the system/user calls can be greatly reduced. For example the tearing down and building of a tunnel in the Dynamics implementation is around 10-20 ms. Thus direct kernel access and driver support is essential in order to have route optimization schemes working satisfactory. Still the experience from the Java implementation and the measurements have provided us with key

insight and some new ideas, which we cannot reveal while the research is still in an early stage.

7 Future work

We can extend the work presented in this paper in a number of ways. First, we would like to experiment in a more realistic test network, where the benefits with applying Route Optimization mechanism are actually measurable; for example the MN and the CH may be residing on the subnets near each other and far from the HA. Second, it would be worthwhile to investigate and develop mechanisms to predict when a handover is likely to occur and consequently sending a binding update earlier, thus reducing packet loss. Here, it may also be beneficial to differentiate between traffic types and just sending binding updates to those CH that sends delay-sensitive traffic streams such as audio. Experimenting with Mobile IPv6 and applications using that as well as experimenting with resource reservations protocols such as RSVP are plausible future works.

8 References

- [1] Charles E. Perkins, *Mobile IP: Design Principles and Practices*, Addison-Wesley, Reading, MA, 1998.
- [2] C. Perkins, *IP mobility support*, IETF RFC 2002, October 1996.
- [3] C. Perkins, D.B. Johnson, *Route Optimization in Mobile IP*, IETF Mobile IP working Group Internet Draft, February 2000.
- [4] C. Perkins, D.B. Johnson, *Mobility Support in IPv6*, IETF Mobile IP Working Group Internet Draft, March 2000.
- [5] Jan-Olov Vatn, *End-to-End and redirection delay in IP Based Mobility*, Project report, Department of Teleinformatics, Royal Institute of Technology, Sweden, April 2000.
- [6] P. Ferguson, D. Senie, *Network Ingress Filtering*, IETF RFC2267, January 1998.
- [7] W.R. Stevens, *TCP/IP Illustrated Volume 1*, Addison-Wesley, Reading, MA, 1994
- [8] Lars G. Bjonnes et al, *Wireless Experimental Metropolitan Area Network Using IPv6 in Norway (WEMAN)*, Proceedings of the 32nd Hawaii International Conference on System Sciences, 1999.
- [9] *Dynamics-HUT Mobile IP*, Helsinki University of Technology, <http://www.cs.hut.fi/Research/Dynamics/>
- [10] University of Lancaster, *Mobile IPv6 implementation for Linux*, See <http://www.cs-ipv6.lancs.ac.uk/ipv6/MobileIP>.
- [11] A. Seneviratne, B. Sarikaya, *Cellular networks and mobile internet*, Computer Communications 21 (1998) 1244-1255.
- [12] R. Caceres, V. N. Padmanabhan, *Fast and Scalable Handoffs for Wireless Internetworks*, Proc. of ACM MobiCom '96, November 1996.
- [13] 3rd Generation Partnership Project, *Combined GSM and Mobile IP Mobility Handling in UMTS IP CN*, 3G TR 23.923, V1.2.0, November 1999.
- [14] *Optimum Project*, Royal Institute of Technology, <http://fl.ssvl.kth.se/~g5/>