HELSINKI UNIVERSITY OF TECHNOLOGY
Faculty of Information Technology
Department of Computer Science and Engineering
Telecommunications Software and Multimedia Laboratory

Dan Forsberg

Communication availability with Mobile IP in wireless LANs

Master's Thesis
March, 2000

Supervisor — Professor Hannu H. Kari
Instructor — M.Sc. Jari Malinen

HELSINKI UNIVERSITY OF TECHNOLOGY
Department for Computer Science and Engineering

ABSTRACT OF MASTER'S THESIS

Author: Dan Forsberg
Title: Communication availability with Mobile IP in wireless LANs
Date: March, 2000 **Pages:** 12+69

Professorship: Tik-109/110 Telecommunications Software
Major: Information Processing Science
Minor: Telecommunications Software

Supervisor: Prof. Hannu H. Kari
Instructor: M.Sc. Jari T. Malinen

Various portable computing devices such as laptops, handheld computers, and other personal digital assistants with networking capabilities increases the demand for seamless communication both in wired and wireless networks. This thesis focuses on the handoff decisions based on the wireless link quality information of the communication partners and how that information can be used to enhance communication availability.

My solution is to use soft, horizontal, policy-based, and mobile controlled two-phase handoffs in wireless local area networks with hierarchical Mobile IP. Together with advanced mobility agent prioritization and signal quality awareness the solution provides seamless handoffs and needed communication availability.

The presented performance analysis shows that a simple handoff policy provides an essential increase in communication availability. This improvement, as implemented in the HUT Dynamics, is sufficient for continuous communication availability under frequent location updates.

Traditionally Mobile IP is made for macro mobility but this thesis shows that hierarchical Mobile IP can be extended to support micro mobility without changes to the protocol. Signal quality awareness, two-phase handoff, and node prioritization are the key factors that make this possible. Additionally, configurable node selection policies can be used to maximize the communication availability.

The presented solution provides simple but essential improvement for mobile controlled handoffs when using simple link level technologies. It is used in the Wireless MediaPoli where it provides an essential support for multimedia reception into and from mobile computers.

Keywords hierarchical Mobile IP, soft handoff, glitchless handoff, two-phase handoff, seamless handoff, horizontal handoff, mobile controlled handoff, node prioritization, node selection policy, wireless LAN, ad hoc network, micro mobility, macro mobility

Acknowledgements

I would like to thank my instructor M.Sc. Jari T. Malinen who gave me invaluable advices and great support during my thesis work in the Mobile Ad hoc Routing Testbed (MART) project. He provided academic guidance and help in organizing the thesis and ideas both as a research advisor and a friend. As the project leader of the project he gave me excellent possibilities to test and measure the system done for this thesis. I thank also Heikki Arppe, Petteri Massetti and Janne Salmi for the pleasant and supportive atmosphere in the MART project.

I thank my supervisor prof. Hannu H. Kari who gave me the communication availability problem concept and excellent support in formulating the problem and criteria and the whole thesis. He is a very innovative person and inspired many of the ideas presented here.

I would also like to thank my friend and colleague Jouni K. Malinen who is a very talented researcher in the Mobile IP concept and in many other fields. He helped me a lot with the implementation and measurements by providing excellent advices and code fragments. He is also the key person as an implementor and developer in the Dynamics project. I thank all other friends and colleagues that made the Dynamics project successful including Björn Andersson, Jari Hautio, Kimmo Mustonen and Tom Weckström, the project leader.

Otaniemi, March 2000

Dan Forsberg

Table of contents

List of figures

List of tables

Glossary

access point (AP) An entity that has station functionality and provides access to the distribution services, via the WM.

ad hoc network (IBSS) Formed by nodes that can directly communicate with each other.

Address Resolution Protocol (ARP) Link layer protocol used to resolve MAC addresses.

backward handoff The MN initiates the handoff request to the communicating MA.

care-of address (CA) With the aid of the FAs, HA only needs to know under which FA the MN is in care of, i.e., the MN registers with the HA the location of the FA. The address for this FA is called CA. The data packets destined for the MN go through the HA and are tunneled up to the LFA or to the MN.

co-located care-of address (CCOA) When the data packets are tunneled up to the MN the MN has to have a CCOA in order for the LFA to be able to establish a tunnel to the MN.

communicating MA A communicating MA is an LFA or HA that the MN is connected to.

correspondent node (CN) Node in the network that communicates with the MN.

domain FA (DFA) A FA that assigns a multicast address unique within its domain to each MN.

FA decapsulation Tunnel endpoint is in the LFA and not in the MN.

foreign agent (FA) A router in the FN that provides routing services to the registered MNs in the FN.

foreign network (FN) Any another network than the HN of the MN.

forward handoff The MN initiates handoff request to the new MA (see also backward handoff).

forwarding Includes a rule, a route and a tunnel that together make up a tunneled data path for packets to and from the MN in the FA.

glitchless handoff Handoff delays and packet losses due to the handoff are eliminated from the data stream.

handoff management (HM) Includes the procedures and required information needed to make handoffs.

handoff policy Handoff policy is a set of rules that identify the best node that is selected if handoff is required.

handoff A process during which the routing responsibility of a node is handed over between designated MAs.

hard handoff The MN can not hear the old or the new MA simultaneously during the handoff (see also soft handoff).

highest FA (HFA) It is the root for the FA hierarchy tree. Upper level MA is a HA for this FA.

home agent (HA) A router in the home network of the MN. When MN is in the foreign network, it forwards tunneled packets to the FA or directly to the MN.

home network (HN) The IP address of the MN belongs to this network. Every MN has a home network.

horizontal handoff A handoff between APs with same link level technology. E.g., the MN does not need to change the network interface.

infrastructure network Network that includes an AP that is used to communicate with other nodes in the wireless network (see also ad hoc network).

interface information daemon (IID) Gives informative parameters about the underlying media used to deliver data packets to the network.

intermediate FA (IFA) A FA that is above the LFA and below the HFA in the FA hierarchy.

Internet Protocol (IP) The protocol that Internet is based on.

inter process communication (IPC) A method for processes in the operating system to communicate with each other.

lower tunnel An IP-in-IP encapsulated data path to the lower FA, or when in MN decapsulation mode, to the MN (see also upper tunnel).

lowest FA (LFA) A FA becomes lowest FA for the MN that is connected to it.

MA detection (MAD) The MN has to be aware of the available MAs. It includes procedures that enable the MN to find new MAs.

macro handoff A handoff in a wireless network with cell radius of several kilometers long (see also micro handoff and pico handoff).

MA selection (MAS) When MN has to decide which MA to use when connecting to the network it uses MA selection mechanism.

micro handoff A handoff in a wireless network with cell radius of tens or hundreds of meters (see also macro handoff and pico handoff).

MN decapsulation Tunnel endpoint is in the MN and the MN uses CCOA.

mobile assisted handoff (MAHO) The MN collects some information needed to do handoff decisions or otherwise else takes part to the handoff procedure initiated by the network.

mobile controlled handoff (MCHO) The MN makes the handoff decision and initiates it.

mobile node (MN) A host or router that can change the point of attachment from one network or subnetwork to another without changing the IP address.

mobility agent (MA) Provides connection for the MN to the home network with possible other MAs. FA and HA are MAs.

mobility management (MM) includes handoff management, MA detection and MA selection.

monitor A module that uses wireless extensions to improve the MA selection and handoff mechanisms.

network controlled handoff (NCHO) In NCHOs the network makes handoff decisions and collects data that is needed to make the decision (see also MCHO and MAHO).

next-hop routing Packets are routed in a host by host basis.

node information data This database contains information about the MAs the MN has detected.

node selector (NS) Chooses the best available MA according to the priority of the MAs.

pico handoff A handoff in wireless network with several meters long cell radius. The bandwidth in wireless networks with pico cell size is higher than in macro cell size wireless networks.

policy routing (PR) A set of rules for handling incoming and outgoing packets in the routing engine. PR can for example drop, modify and route packets.

priority balancing (PB) Occurs when the priority of the current MA is near enough to the MA with highest priority. The priority of the current MA is set to the same value as the compared MA.

priority decreasing (PDT) The priority of a certain MA is decreased even the SQ of this MA is not decreased (see also PIT).

priority increasing (PIT) The priority of a certain MA is increased even the SQ of this MA is not increased (see also PDT).

registration protocol (RP) The protocol that the MN uses when registering to the HA either directly or via the FA hierarchy.

reverse tunneling Both the MN and CN route packets through the HA when communicating with each other. Packets are tunneled between FAs and HA with FA decapsulation. With MN decapsulation packets are tunneled between HA and MN.

routing rule (RR) Used to decide which routing table to use for the handled packet in the routing engine.

seamless handoff If the user or a program that uses the network bandwidth does not notify the handoff only by examining the data stream over the network, the handoff is said to be seamless.

session key (SK) Unique shared secret in a session for security purposes.

signal quality (SQ) The difference between radio signal and noise levels. Signal quality is measured from received packets.

signal quality analyzer The purpose of the signal quality analyzer is to normalize and analyze the SQs. It can use averages of the signal qualities and decrease, increase or balance priorities.

signal quality collector Collects received SQs into a SQ history.

signal quality history Received SQ history. The history is kept in a FIFO buffer.

signal quality sensor Located in a network device and can measure the received and sent signal strengths.

signal quality tape (SQT) A file that contains signal quality and timestamp value pairs.

smooth handoff A seamless and soft handoff.

soft handoff In soft handoff the MN can hear both the old and the new MA when switching the communicating MA (see also hard handoff).

SQT set A set of SQT files that were recorded in the same session.

switching FA (SFA) A FA that replies to the MN in localized location updates.

Transmission Control Protocol (TCP) Belongs to the Internet protocol family. This protocol is on top of the IP and provides connection oriented services.

triangle tunneling Used when the MN does not route outgoing packets via HA to the CNs.

tunneling Used when packets are encapsulated into an another packet as the payload.

two-phase handoff During the handoff up and downstream routes are handled separately.

upper tunnel An IP-in-IP encapsulated data path to the upper MA (see also lower tunnel).

User Datagram Protocol (UDP) A protocol that belongs to the Internet protocol family. This protocol is on top of the IP protocol.

vertical handoff A handoff to wireless overlay network with different cell size and link level technology.

visiting network (VN) The network that MN is currently visiting. A VN is also a FN (see also FN).

wireless local area network (WLAN) Wireless networks that are suitable for example in the office environments. The IEEE 802.11 standard is an example WLAN.

wireless medium (WM) Uses air interfaces to deliver data between nodes in the wireless network.

Chapter 1

Introduction

Various portable computing devices such as laptops, handheld computers, and other personal digital assistants (PDAs) with networking capabilities increases the demand for seamless communication both in wired and wireless networks. Increased use of multimedia content with mobile computers makes seamless communication, an essential and required feature expected in mobile connections. Practical mobility management should provide a seamless handoff where the user does not observe communication disruptions.

Internet Protocol (IP) [1] based mobility management implementations have traditionally ignored link layer information with this respect. However, many link layer technologies provide signal based information that can be used by the network layer mobility control.

Traditionally, a user does not need to know the communication partner that provides the connection to the Internet. This is convenient but causes some restrictions to the system. When the user has several possibilities for connecting to the Internet she may want to choose or change the communication partner dynamically. For example, the cost, bandwidth, and available services may cause the user to change the communication partner, or more precisely the policy for choosing the gateway to the Internet.

1.1 Structure of the thesis

This thesis is constructed as follows. Chapter 2 explains the communication availability problem with Mobile IP in wireless local area networks (WLANs) and the criteria for an efficient solution. It also introduces the concept and general terms of the Mobile IP. Chapter 3 describes related work. It is divided into two sections as is the solution for the problem. The first section discusses network level handoffs and the second one handoff prioritization and handoff policies. Solution is presented in Chapter 4. Deeper aspects of the solution and detailed implementation is handled in Chapter 5. Evaluation of the system and different performance tests are described in Chapter 6. Finally, in the Chapter 7 I present conclusions.

Chapter 2

Communication availability

2.1 The Mobile IP concept

Mobile IP [2, 3] is a modification to the IP that allows nodes to continue processing datagrams no matter where they happen to be attached in Internet with the same reachable IP address. Control messages allow IP nodes involved to manage their IP routing tables reliably.

Mobile Node (MN) is a host that can change the point of attachment from one network or subnetwork to another without changing the IP address. It may continue to communicate with other Internet nodes called correspondent nodes (CN), at any location using the same IP address, assuming link-layer connectivity to a point of attachment is available. *Home agent* (HA) is a host in the home network (HN) of the MN. It tunnels datagrams for delivery to the MN when MN is away from home and maintains current location information for the MN. *Foreign network* (FN) is any other network than the HN of the MN. The FN, where the MN is currently visiting is called visiting network (VN). A *foreign Agent* (FA) is a host in a FN. It provides routing services to the registered MNs in the FN. FAs deliver decapsulated datagrams to the MNs tunneled by the HA and they may also act as default routers for registered MNs.

The packets that HAs and FAs route to the registered location of the MN must be kept intact. This is done by placing the original packet within another packet. The outer packet contains IP address of the next hop route to the registered location of the MN.

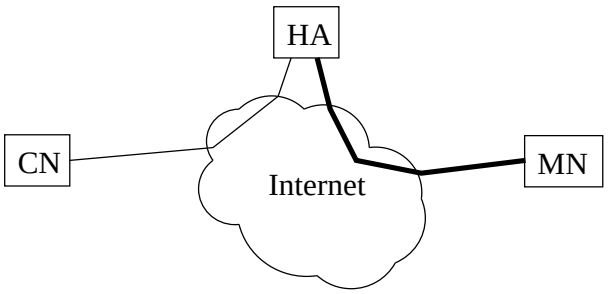

Figure 2.1: Basic Mobile IP

The receiver of the encapsulated packet can then unwrap the outer packet to retain the original inner packet intact. This method of encapsulating IP [4] packets within other IP packets is called *IP-within-IP*, and the routing of packets with the aid of the wrapper packets is called *tunneling* [5]. A FA is capable to tunnel packets in both directions, from the MN to the HA and from the HA to the MN.

The MN is the mobile computer where the user wants to have the Internet available. In addition, she wants to be reachable from other computers, even while visiting other networks. The IPv4 associates addresses with a certain point of attachment somewhere in the Internet. All packets destined to a particular computer will be routed through the fixed network. To let the MN move to other networks, there must be a way of routing the packets arriving at the HN to the location of the MN in the VN. For this purpose, each HN has a HA.

HA should always be aware of the current location of each registered MN. This is achieved by making the MN register to the HA each time it changes the point of attachment in the Internet. Each MN can register one of it's IP addresses only to one HA at the time. Figure 2.1 shows situation where there is no FA but the MN is registered to the HA directly and communicates with the CN via the HA.

When the packet latency between a HA and a MN is high the registration for the MN with the HA can take too much time. This can happen when the HA is far away and the MN is receiving real-time stream. If MN changes the point of attachment while

receiving the stream, it can take a while for the HA to receive the registration from the MN, and the mobile user would observe an annoying gap in the video stream.

With the aid of the FAs, HA only needs to know under which FA the MN is in care of, i.e. the MN registers with the HA the location of the FA. The address for this FA is called *care-of address* (CA). When the MN moves it registers the new FA with the HA.

2.2 Improving communication availability

In this thesis, I will consider the problem of communication availability under signal quality (SQ) based handoff management (HM) in wireless local area networks with Mobile IP.

The **communication availability** is readiness for usage [6]. Communication availability is defined at location l as the ability of the server to deliver the service that fulfills and maintains the requirements of service quality to the MN. I measure this both from the point of mobile users and the networks.

Signal quality gives information about the ability to transfer information in the wireless data path.

Handoff is an event that occurs between communicating mobility agents (MAs) and a MN. At least three nodes are involved in a handoff; one is the MN and the other two are the old and the new MA. Handoff starts when the decision for changing the MA is made and finishes when the MN has changed the MA. Thus a handoff is the process during which a node is "handed over" between two designated MAs [7]. During handoff the MA that is responsible of routing the packets to and from MN is changed.

In *soft handoff* [8, 9] the MN can communicate with both the new and old MA while performing handoff between them. This is not possible in *hard handoff* [9], because the MN can listen only one MA at a time.

Network controlled handoffs (NCHO) are handoffs where the network makes the handoff decision. In *mobile assisted handoff* (MAHO) the MN makes the handoff decision together with the network and in the *mobile controlled handoff* (MCHO) mobile node

decides by itself when to make handoffs. [10]

One method to separate MCHOs is to divide them into *forward* and *backward handoffs* [9]. In backward handoff the MN sends the handoff request to the current MA of the MN. In forward handoff MN initiates the handoff by sending request to the new MA.

Wireless network interfaces can have different cell sizes, for example in-room, in-building, campus, metropolitan, and regional. *Wireless overlay networks* are a combination of wireless networks that have different cell sizes. If the wireless network interface and the cell size is changed during the handoff process, this is called a *vertical handoff* [11], otherwise a *horizontal handoff* where old and new MA uses same radio technology. Depending on the cell size handoffs can be classified into *macro, micro*, and *pico handoffs*. Macro level cells have at least several kilometers long radius and lower bandwidth than micro cells or pico cells. The radius in micro level cells is in tens or hundreds of meters and with pico level cells the radius is in meters.

Handoff management includes the procedures and required information needed to make handoffs. In this thesis it is done in the network layer. The handoff management problem can be divided into two subproblems. In MCHO the MN has to be aware of available MAs and the services they offer. This can be expressed as *mobility agent detection* (MAD) problem. Secondly, the *mobility agent selection* (MAS) problem arises when MN detects several MAs and requires communication availability. One of the detected MAs has to be selected. Communication availability requires routing of the signaling messages that the handoff management must take care of.

2.3 Criteria

The criteria for the solution should be fulfilled at least with one MN in the system.

Minimal impact on data transmission

In the ideal situation the handoff is transparent and has no impact on the data transmission. This means that the throughput and latency of the routed packets are not affected

by the handoff management. Sessions are maintained and no packets are lost. The mobile user does not notice any difference whether the handoff management is used or not. Minimal impact on data transmission can be measured with throughput analysis, signaling latency and packet loss measurements.

Tolerance for congestion

The solution should tolerate congestion. Congestion occurs when the data path is fully used and there is demand for more capacity. The word "tolerate" means that the MN and the MA are able to communicate with each other. This can be measured with signaling latency under heavy load on the data path generated by the MAs and the MN.

Efficiency

The solution should be efficient. An efficient solution uses the radio bandwidth sparingly. Signaling messages are small and the signaling itself is lightweight. This can be measured with the number and size of the signaling messages required in the handoff management protocol.

MN should use the MA with best communication availability

To measure communication availability readiness of the service must be measured. Readiness of the service is related to the throughput and packet latency of the communication path. Smaller latency indicates better readiness. Higher throughput gives better readiness for services that require more bandwidth. Throughput and latency can be used to measure this criteria.

Independence of the underlying radio technology

The solution should not be dependent on the physical characteristics of the underlying radio technology. This can be measured by identifying the layer in which the solution is functional.

Modular node selection system

Flexibility to add new and modify existing node selection policies affects most the implementation. *Node selection policy* is a set of rules that affect the handoff decision. The implementation should be modular and easy to improve. This involves clear interfaces between modules and data flows.

Chapter 3

Related work

Handoffs have been studied widely in recent years. G. P. Pollini presents an overview of published work on handoff performance and control [12]. He also discusses current trends in handoff research. Furthermore, he also presents different handoff methods based on signal strengths. Challenges in seamless handoff design in mobile multimedia networks are handled by L. Taylor et al. [13].

Mobile IP is not specifically planned to support micro mobility [3] and it has not been considered to be a good solution for network-level micro mobility [14, 15, 16, 17]. Thus, several micro mobility proposals with and without Mobile IP have been introduced [18, 19, 20]. M. Stemm and R. H. Katz describes vertical and horizontal handoffs in [11]. C. Toh et al. presents different handoff protocol design issues in [21].

3.1 Network level seamless handoffs

If the user or a program that uses the network bandwidth does not notify the handoff by only examining the data stream over the network, the handoff is said to be *seamless*. In a *glitchless handoff* delays due to the handoff are eliminated from the data stream. Multicast and buffering are the most used methods to provide seamless and glitchless handoffs. R. Cáceres and V. N. Padmanabhan describe a buffer based solution with four-packet buffer in the access point (AP) [15]. In their handoff protocol MN initiates handoffs and the new AP sends a notify message to the old AP when the MN has moved. Their test nodes

in the system used IP addresses from the same subnet. Proxy and gratuitous ARP [22] messages were used in the wired side of the APs to route packets to the right AP. Beacon period varied between 10ms and 1s.

K. Brown and S. Singh researched User Datagram Protocol (UDP) [23] for mobile cellular networks and the results for the buffer based solution are in [14]. They use Mobile IP together with buffered UDP packets and achieve a 50% increase in throughput with M-UDP compared to UDP. Bakre's and Badrinath's similar work with Transmission Control Protocol (TCP) [24] can be found in [25]. They split the TCP connection into wireless and wired parts to get better throughput.

C. Perkins and K-Y. Wang present a scheme for optimized *smooth handoffs* in [26]. They use buffering with Mobile IP as a basis for the handoff. FAs buffer packets for MNs and when the MN switches FA the old FA is signaled to send the buffered packets to the new FA which then forwards the packets to the MN. Packet identifiers are used to eliminate duplicate packets sent to MN. Packet buffer is required for every MN and in multiple APs. This increases the requirements for resources and decreases the scalability of the system.

K. Keeton et al. present an incremental reestablishment scheme, which modifies an existing connection by establishing only the portion of the channel between the AP and the MN where the old and new channels would diverge [27]. They also present multicasting support for handoffs. Multicasting-based solution for handoffs in the Internet is described by Jayanth Mysore and Vaduvur Bharghavan [17]. Every MN has a unique multicast address and packets destined to MNs have this multicast destination address. Packets from the MN have unicast destination addresses. Neighbor multicasting routers join to the same multicast address as the MN. When MN initiates handoff with new AP it is already in the multicasting address of the MN and thus the handoff can be made seamless. C. L. Tan et al. describe a fast handoff scheme for wireless networks using a multicast based handoff in [16]. They describe a *domain FA* (DFA) which assigns a multicast address unique

within its domain to each MN. The domain FA has logically many APs in the lower level. Thus, the approach can be seen to have a two level hierarchy. The AP in which the MN is connected to has joined to the multicast group of the MN and actively forwards packets to the MN. Adjacent APs have also joined to the same multicast group but do not send packets to the wireless network. The handoff is similar to the handoff in [17] described above. The Ph.D. thesis of S. Seshan [28] and the paper of H. Balakrishnan, S. Seshan and R. H. Katz [29] favor multicast based handoff solutions also.

H. Balakrishnan et al. [29] and the thesis from Seshan [28] introduce also a *snoop module* that listens TCP traffic between the AP and MN. The idea of the snoop module is to resend packets that were lost between the MN and the AP by monitoring the acknowledgments to TCP packets generated by the receiver.

The drawback in multicast solutions is that multicasting has to be supported by the routers and the network bandwidth is wasted since the data stream is duplicated to several APs. The APs have to allocate resources for every MH that is directly connected to it or to the adjacent APs. Thus, resources are not used efficiently. The buffering in APs may also affect the packet routing latency between CN and MN. Additionally, if multicasting and buffering are used together the resource requirement becomes more demanding. In this kind of a scenario the wired network has to be capable of handling more bandwidth than the wireless network. This is not a problem if the bandwidth difference is considerably high. Today, especially in wireless local area networks the bandwidth is increasing. A problem is that the cell structure of the wireless networks and the many radio channels make it possible to overload the wired core network multiple times the wireless bandwidth.

3.2 Handoff prioritization and policies

S. Tekinay and B. Jabbari describe a measurement prioritization scheme for handoffs in mobile cellular networks [30]. The prioritization is made in the wired network side. S. Tekinay et al. illustrates how the SQs can be used to prioritize handoffs to get better

performance from the system. *Adaptation and Mobility in Wireless Information Systems* by R. H. Katz [8] presents generally the problem of improving communication through situation awareness. N. D. Tripathi et al. discuss about the handoffs in cellular systems [31]. They state that a handoff algorithm with fixed parameters cannot perform well in different system environments. Handoff prioritization schemes are described and the prioritization is based on the signal quality.

Different prioritization schemes are related to the handoff policies. H. J. Wang et al. introduce handoff policies that take into consideration many different aspects of the handoff [32]. For example, performance, power consumption, and cost could be measured and compared to calculate the best wireless system at any moment. Wang concludes that "Policies on what the 'best' reachable network is, and when to handoff to it, can be complex to specify" and that "A single, hard coded policy is suboptimal" [32]. Different network technologies differ in bandwidth, latency, power consumption, connection setup times, various hints, and possibly their charge model. H. J. Wang et al. introduce a policy calculation function that uses different cost parameters as input and produces the total cost of the network. The total cost can also be thought as the priority for the network. In addition APs may also have different priorities which H. J. Wang et al. do not handle in their paper. They separate the networks but do not separate different APs in the same network. More generally, the separation of different APs and the different services these may offer is not considered.

H. J. Wang et al. describe a handoff synchronization problem [32]. If several MNs are using the same policy in the same place, they may change the network simultaneously and affect the dynamic parameters of the network. These parameters affect the policy and thus may cause MNs to oscillate between different networks. This can also happen between APs in the same network. As the solution for this problem H. J. Wang et al. introduce a *stability period*. It is a time period that the MN waits before initiating a handoff. The reason for this time period is stabilization. Wang concludes that "Only if a

network is consistently better than the current one in use for the *stability period* does the mobile host perform handoff" [32]. This kind of approach adds latency to the handoff.

Chapter 4

Mobility agent switching

Mobile IP proposes a macro mobility solution for the mobility problem. Therefore, I am using it as a basis for the solution to the problem described in Chapter 2. More precisely, the HUT Dynamics Mobile IP is used for the mobility management to achieve communication availability. The hierarchical structure of HUT Dynamics provides an efficient platform for fast macro mobility [33].

The MN makes handoff decisions and the FA hierarchy assists the MN in the handoff management. This makes it possible for the MN to use different handoff policies independently of the MAs. Thus, the solution uses MCHO scheme. Handoff policies are described in Section 4.3. MN can compare different MAs and gather information about them from agent advertisements. MNs may use different policies and criteria to choose communicating MA. The disadvantage of the MCHO is that MN has to support handoff management. This includes ability to make handoff decisions and initiate handoffs. With NCHOs the access network is responsible for handoffs and the MN can be made simpler.

The whole idea for mobility lies on the *mobility management* system which includes also the handoff management. I divided the system into *MA detection* and *MA selection*. Figure 4.1 shows the data flow in the mobility management system. MA detection includes *signal quality sensor* and *signal quality collector*. *Node selector* and *signal quality analyzer* belong to the MA selection part of the handoff management. *Signal quality history* data is used by both of the MA selection and MA detection. *Node information data* contains

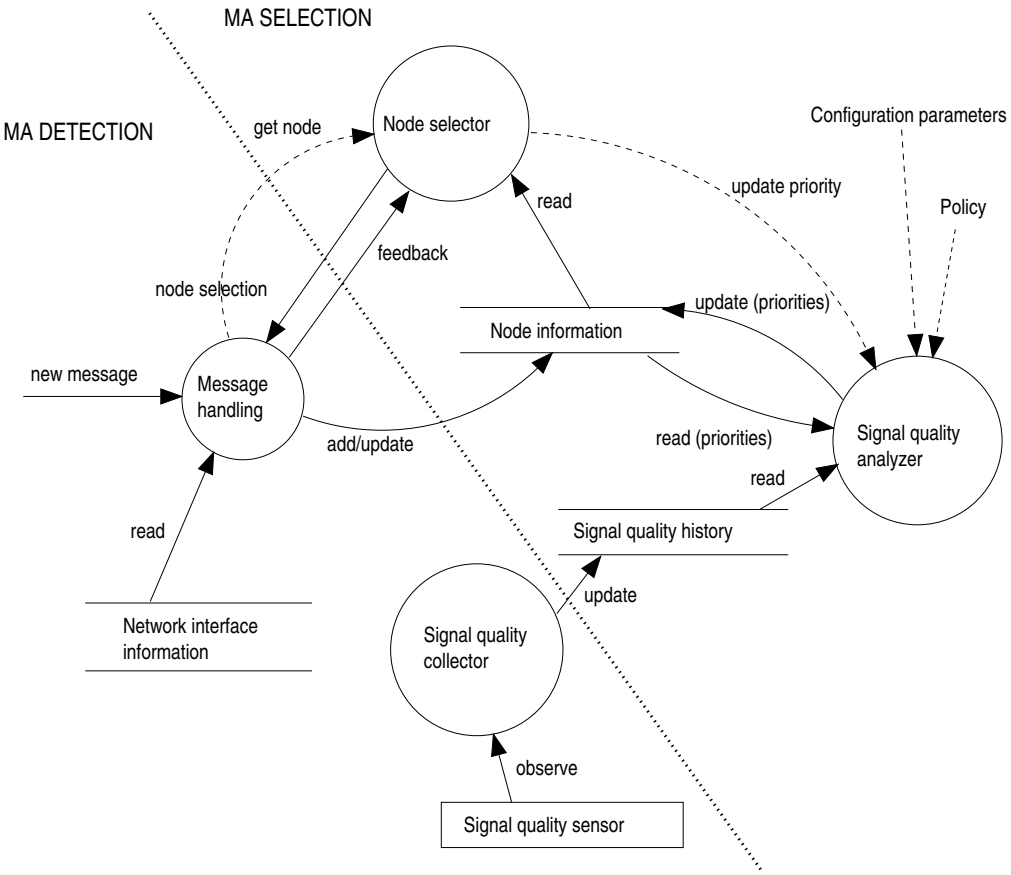

Figure 4.1: Data flow

information about FAs that the MN has received in the agent advertisements. This information storage is used and updated by the MA selection component. The concept is clarified below.

4.1 MA detection

The expiration of the agent advertisements in Mobile IP provides a method for MA detection. MN receives advertisements from MAs and knows which MAs are available. The agent advertisement lifetime is the maximum length of time that the advertisement is considered valid in the absence of further advertisements [3]. It is used to get some resolution for mobility in time. If MN moves it can detect movements from the advertisements. Either it does not receive agent advertisements from certain FAs anymore, or advertise-

ments itself contain some information from which MN can notice the movement. This is sufficient in wired static networks, where the MN is switched from one subnet to another, but in WLANs the movement and thus mobility is different. In a WLAN the mobile user is able to move inside the wireless cell of the MA without losing connection to it. The MA is acting also as an AP for the MN. If the MN is in range of several MAs it has to decide which one to use as a gateway for the communication with CNs. When the mobile user moves outside the current MA MN has to initiate a handoff with a new MA.

Signal quality collector

If a SQ sensor is available the solution has much more possibilities. SQ sensor monitors the link quality to other nodes in range. SQ is measured from received packets and is related to the data throughput between signal source and receiver.

SQ collector is a component that reads in the SQ values from the SQ sensor. Values are converted into a more general form and stored in a *signal quality history* data storage. Conversion is needed to support different kinds of SQ sensors. Collector makes the conversion so that the data storage contains comparable values from possibly different SQ sensors. This simplifies the SQ analyzer because it does not have to be aware of different SQ sensors.

Message handling

Connections, location updates and disconnects are handled in the *message handling* component. It receives all agent advertisement messages, parses them and saves the information into a *node information* data storage. Message handling communicates with the node selector which controls the location update decisions. When location update or connection is made, the message handling component sends the registration request to the selected MA and handles the received registration reply from the MA.

4.2 MA selection

Mobility agent selection is based on priority comparison. Priorities are modified and analyzed in the SQ analyzer. Node selector makes the final decision based on the priority and currently used MA. Different *mobility agent selection policies* are used to help the decision.

There is only one priority variable for each available MA. Priorities are based on the SQ values received via the interfaces with SQ sensor. With interfaces that do not have SQ sensor a specific interface priority is used as a basis for the MA priority. The whole monitoring system is built upon the idea that different MAs can be separated by some means related to the communication availability. Priorities have been chosen to separate the MAs in the monitoring system because they are flexible and abstract enough.

Priority balancing (PB) technique compares the best MA candidate that has the highest priority to the priority of the current MA. If PB occurs the priority of the current MA is set to the same value as the value of the best MA. When the best MA candidate has same priority than the current MA, the node selector does not make a decision to change the MA. PB occurs if the compared priorities are close enough.

Priority decreasing technique (PDT) decreases the MA priority with a certain percent value in the MA selection system. PDT uses triggers to modify the *degradation percent value* which is used to degrade the priority for the MA. The degradation percent value is increased every time the PDT is triggered, until the maximum of hundred percent is reached. The function for the degradation percent value that the PDT uses is exponential. *Priority increasing technique* (PIT) is used with companion of PDT. It has an opposite effect to the PDT. The value of the degradation percent variable is decreased with PIT and increased with PDT. PIT is also trigger based. It multiplies the degradation percent value with a constant fraction when triggered. *Configurable interface priorities*, SQ values, and SQ averages affect the priority of a MA, but also priority balancing, PIT, and PDT are used to enhance the MA selection system.

Node selector

Node selector is a simple component that compares the priority values. It communicates with the message handling module and decides if MN should change the MA. If change decision is made, the node selector gives information about the node it has selected to the message handling component.

Message handling module provides feedback for the node selector. If the registration process is unsuccessful the node selector has to decide what to do with the problem. The MA can not be used if the registration fails. After a time period the MA may become functional and the MN should be aware of it in order to take advantage of it. The possibility to separate functional and nonfunctional MAs increases the fault tolerance, efficiency, and functionality of the system.

If registration fails through the MA, the PDT is used to decrease the priority of the MA. Every time MA sends agent advertisement PIT is used for that MA. This enables MAs to become slowly available again. This technique makes the system more robust and small temporary failures in the MAs or the network are better recovered.

Signal quality analyzer

Signal quality analyzer is the main component for the MA selection process. It modifies and prepares the priorities for the node selector, which finds the MA with highest priority and handles the registration process feedback.

4.3 Selection policies

Node selector uses certain rules to make selection decisions. The set of rules that affects the node selection process is called a *policy*. Comparisons and selections are based on the node priority, which the policies modify to achieve the needed results. The node selector picks up the node that has the highest priority.

There are currently four different policies for the node selector: *eager-switching,*

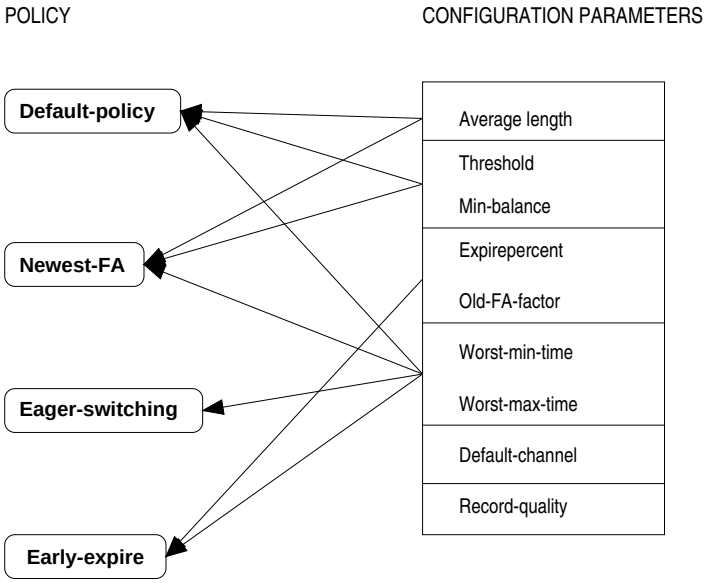

Figure 4.2: Policy and configuration parameter relations

newest-fa, *early-expire*, and the *default-policy*. SQ analyzer contains some configurable parameters that are related with policies. Figure 4.2 illustrates the relationships between configuration parameters and policies. The figure also shows how the parameters are conceptually related together, as explained in Section **??**.

Eager-switching

With *eager-switching* the node selector takes the MA with highest priority and does not calculate any averages from the link quality values. This means that the MA with immediate highest SQ is used. Depending on the SQ sensor characteristics the SQ values for the FAs can vary even the MN is not moving at all. Thus, frequent location updates is a characteristic for this policy.

Early-expire

Every agent advertisement has a lifetime that starts from zero when a new agent advertisement is received from the MA. The default policy is to use the agent advertisement lifetime expire entries from the node selector. *Early-expire* policy uses *expiration-time* configuration parameter to calculate the validity for MAs in the node selector. If the

entry becomes older than the expiration-time, the *old-FA-factor* percent value is used to degrade the priority.

Newest-FA

Newest-FA policy selects always the most recently detected MA and it acts like the default policy when no new FAs are detected.

Consider a situation where a MA is in an area where no other MAs are heard. The MA can be in a different radio channel than other surrounding MAs or there can be a wall between the MAs that does not pass the radio waves through. The mobile user can enter this area very quickly from an area where many MAs are heard. In our example this can happen when the mobile user changes radio channel or walks around the dense wall. Due to the nature of the expiration process for the MAs, the node selector will remember the old FAs when mobile user has entered the new area.

When MN detects the FA in the new area, newest-FA policy sets the priority for this FA at maximum and thus the node selector will select it. After the second agent advertisement from the new FA, the old FA entries may not have expired yet. The SQ value is used as a basis for the priority for the new FA and at this point it may become lower than the priorities of the old nonexpired FAs. This brings up a problem for the MN. After the second advertisement from the new FA the SQ analysis of the old FAs may still be valid, even they are not actually reachable. If the node selector selects one of the old FAs, it is clearly a mistake since the MN can not communicate with it. Solution for this problem is to set the priorities of the other FAs to the lowest possible, when new FA is detected. Old FA priorities are restored with new agent advertisements from each of the old FAs. This prevents MN from registering to the old FAs that may not be reachable when new FAs are detected. On the other hand, if some of the old FAs are still reachable and heard, the priorities will become normal with the next agent advertisements from these FAs.

Default-policy

The default policy uses averages of the last received SQ values to calculate priorities for different FAs. The number of SQ values used in the average calculation is configurable. The three other policies can be combined and with the default-policy, since they have slightly different effects. Different combinations make the system more flexible. Eight different combinations are listed in the Table 4.1.

Table 4.1: Policy combinations

eager-switching	newest-FA	early-expire
ON	OFF	OFF
ON	OFF	ON
ON	ON	OFF
ON	ON	ON
OFF	OFF	OFF
OFF	OFF	ON
OFF	ON	OFF
OFF	ON	ON

Chapter 5

Enhancing HUT Dynamics Mobile IP

In this chapter I explain the implementation that has been done for this thesis. It is built into the HUT Dynamics Mobile IP software, which is a hierarchical Mobile IP software. I have enhanced and further developed it.

The implementation is composed of MA selection and MA detection systems, interface handling in MN, device driver enhancements and several enhancements into the FA and MN components. Several tools were produced to help the testing process. I developed and implemented *record* and *replay testing methods* to support extensive testing. Some lower level modules and tools were also created and enhanced.

5.1 Base implementation

Mobile IP implementation

The HUT Dynamics Mobile IP [33] system has been developed in Helsinki University of Technology (HUT). It is a scalable and hierarchical Mobile IP implementation for the Linux operating system [34]. Development started in October 1998 in a student group of a software engineering course in HUT. After the course ended in spring 1999 HUT Dynamics has been further developed.

In a hierarchical Mobile IP several FAs are put into FNs, so that the FAs make up a hierarchical structure (Figure 5.1). The registrations need to travel only a minimal distance when the MN has already registered via the FA hierarchy. HUT Dynamics supports

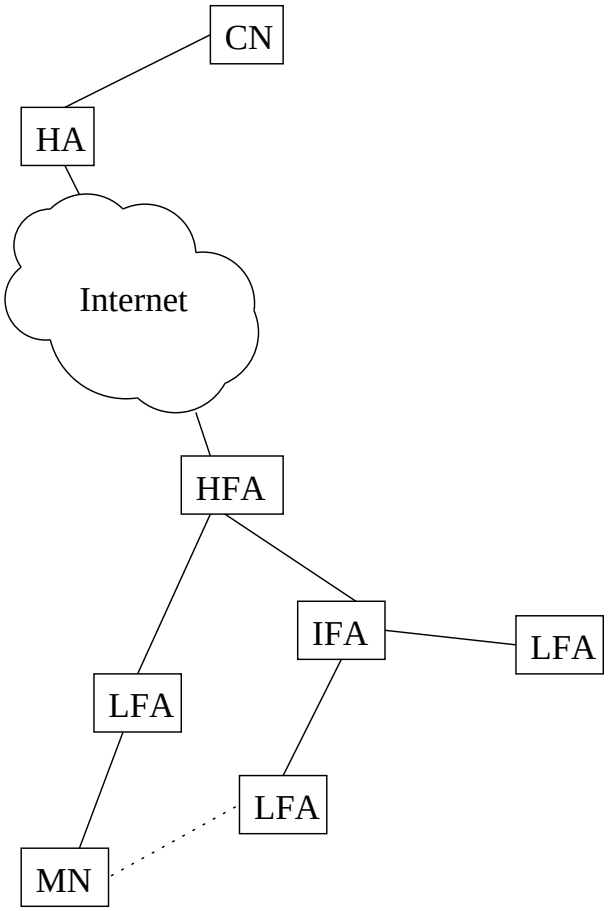

Figure 5.1: Hierarchical foreign agents

arbitrary number of FAs and hierarchy levels. It is also possible not to use FAs at all. In this scenario the MN registers directly with the HA [33].

Tunneling modes

The tunnel is established between the HA and the registered location of the MN. Tunnel is built with segments between the HA and the root FA, between each FA in the path down to the *lowest FA* (LFA). In *FA decapsulation* mode the LFA decapsulates the encapsulated packets and sends them directly to the MN. A tunnel may also continue down to the MN. This is called *MN decapsulation* mode, since MN decapsulates the tunneled packets.

Operation Overview

If MN decapsulation is used the MN needs a *co-located care-of-address* (CCOA) in the FN. In MN decapsulation mode the tunnel endpoints are between the MN and the HA. With FA decapsulation the home IP address of the MN is sufficient for registration and tunnel endpoints are between LFA and HA.

When MN moves to the FN the *registration protocol* (RP) is used for registration with HA. RP is implemented hierarchically and the mobility binding is created through the FA hierarchy step by step. This allows each FA on the path from MN to the HA to examine if they already have a binding for the specified MN. This allows them to perform local location updates. For a new registration, the protocol reaches the HA which then confirms the mobility binding creation [33].

Tunnels are created from root downwards after the MN has properly been authenticated by the HA. During registration, the MAs agree on the lifetime for the tunnel. A keep-alive protocol is used to keep the tunnel open and refreshed well before the lifetime of the tunnel expires [33].

During tunnel creation each MA stores information about the next MA in upstream and downstream directions for each tunnel. This information is called *mobility binding* [2] and it allows packet forwarding to the next MA.

An *access point* (AP) is an entity that has the station functionality and provides access to the distribution services, via the *wireless medium*. When MN moves to another place in the FN and connects to a new AP, the RP is executed again. This time a *localized location update* [33] is performed in the FA hierarchy. This includes creating new tunnels, if needed, to the new path location of the MN. *Switching FA* (SFA) is the FA that replies to the registration request of the MN. Localized location update is illustrated in Figure 5.2.

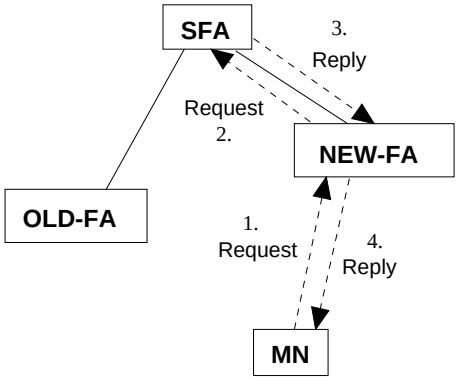

Figure 5.2: Local registration update

Security

HUT Dynamics Mobile IP supports secure signaling through authentication and replay protection mechanisms. A separate session key (SK) management protocol and Dynamics-specific vendor extensions [35] are used to support secure localized location updates in the FA hierarchy. MAs may have preconfigured security associations, which are used to authenticate the signaling packets.

HA acts as a SK distributor for the FA hierarchy and MN. SK management protocol is used to distribute the SK into registered path of the MN, which includes HA, FAs in the path and the MN [33]. If the security association exists the SK can be encrypted with the help of shared secret and thus man-in-the-middle style attacks can be prevented. If no security association is set for a certain FA-FA pair, public key encryption (RSA [36]) is used [33]. SK management protocol affects the performance of the system, because security calculations must be made and authentication extensions must be sent with the signaling packets.

Policy routing

In Internet IP usually routes packets in a host by host basis, that is, by *next-hop routing*. Routers are hosts that forward IP packets based on the IP addresses. HUT Dynamics is heavily based on *policy routing* (PR) in the Linux operating system. In the context of this

thesis PR refers to policy-based packet filtering and forwarding [37] in Linux.

The PR rules use the knowledge of incoming and outgoing network interfaces and the source and destination addresses of IP packets. Other information may also be used for routing decisions, e.g. the firewall chain in the Linux can mark packets. The routing code can use this mark to make routing decisions. Further, traffic control in the Linux enables the use of traffic flows that can be identified by flow identifiers. These flow identifiers can be used to decide the route for the packets.

PR implementation of Alexey Kuznetsov in the Linux kernel supports several routing tables. Routing table is a list of routing entries that are applied to the currently routed packet one by one until matching routing table entry is found.

Routing Rules (RR) are used to decide which routing table to use for the packet. The rule list is also scanned from the beginning to the end. The first matching rule is applied. By default, there are three rules in the Linux kernel. Rules are indexed with preference number and the rule with smallest preference is made the first entry in the list. New rules can be added with preference number. With one rule you can make routing decisions based on the source and destination addresses of the IP packet, incoming device, type of service field of the IP packet and firewall mark. The effects of the matching rules differ. Each rule has an *action* part. If matching rule is found the packet is further processed by the action part. Action usually contains pointer to one of the routing tables, which means that the packet is routed based on the routing entries in that particular table. Below is an example of the routing rules.

```
0:      from all lookup local
32764:  from 130.233.193.94 iif eth1 lookup 2
32765:  from all iif TUNL0 lookup 1
32766:  from all lookup main
32767:  from all lookup default
```

Tunnels, RRs, and routing tables makes possible to route packets from MN to CN through the FA hierarchy and HA without any modifications to the default routing engine. FA uses a separate routing table for MNs. This minimizes the effects of the FA in the host

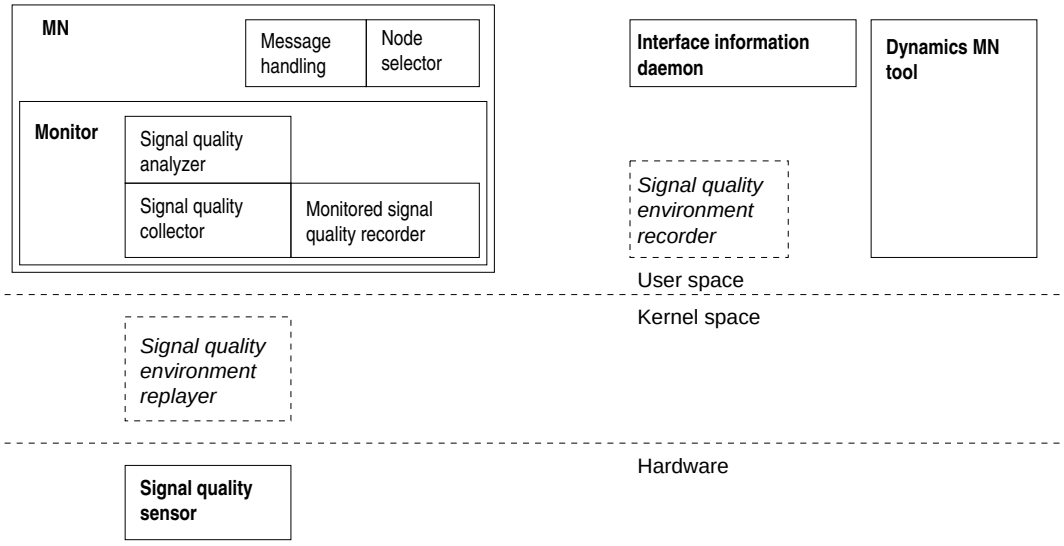

Figure 5.3: MN architecture

when it is used for any other routing purposes.

5.2 MN architecture

The MA detection and the MA selection systems are included in the *monitor component* that is attached into the MN. *Interface information daemon* (IID) is a separate entity that communicates with MN and it belongs to the interface handling category. *Signal quality environment recorder* and *signal quality environment replayer* are used for testing purposes and are explained in the Chapter 6. Figure 5.3 illustrates the architecture.

A special handler mechanism was developed to support hooks for different *events* in the MN. Received agent advertisements and inserted/removed interfaces generate such events. The developer can dynamically register or unregister *hooks* for the events at any time, but the calling order for the hooks is fixed to the same as the registration order. This could be easily enhanced to support hook prioritization. Figure 5.4 illustrates the events used in the Monitor module.

Events are categorized into two main categories: *interface-events* and *FA-events*. New categories can be added. Interface event category includes *interface up detection*

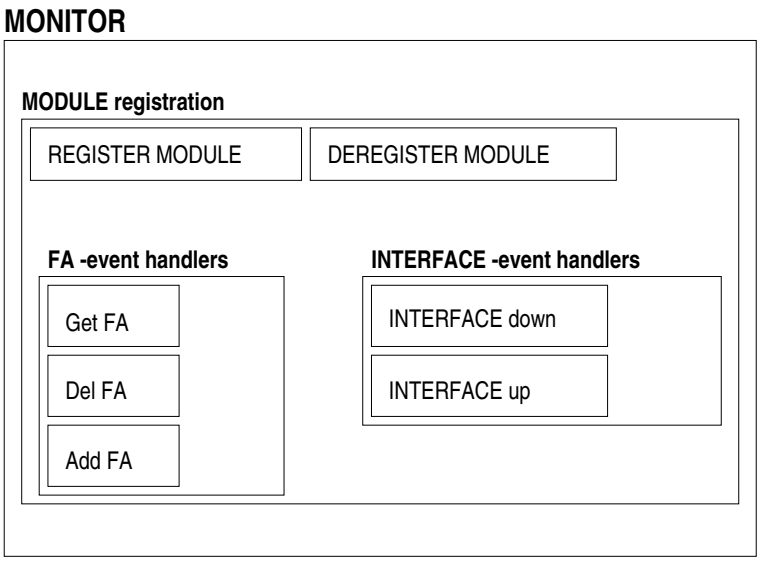

Figure 5.4: Event driven Monitor

(INTERFACE_INIT) and *interface down detection* (INTERFACE_DOWN) events. FA category includes *FA advertisement expiration* (FA_ADV_EXPIRE), *FA advertisement reception* (FA_ADV_RECEIVE), and *get best FA candidate* (FA_GET). Table 5.1 shows the events and assigned hooks.

INTERFACE_INIT event related handlers are called when new interfaces are detected. INTERFACE_DOWN event related handlers are called when the interface goes down. Communication through the device is not possible when it goes down. PCMCIA [38] based devices such as network interface cards are a good example of removable devices.

MA detection system keeps information about received agent advertisements in the memory and removes the entries when the lifetime has exceeded and no new advertisements from the FA are received. The handlers registered for the event FA_ADV_EXPIRE are called when the FA agent advertisement lifetime expires. When new FAs are detected through received agent advertisements, handlers registered for the FA_ADV_RECEIVE event are called. FA_GET handlers are called every time a FA agent advertisement is received. FA_GET handlers reinitialize, modify, and process the FA priorities for the node

selector. The node selector then compares the current MA priority with the MA with best priority.

Table 5.1: Event hooks

Event	Hooks
INTERFACE_INIT	mn_default_INTERFACE_INIT_handler monitor_interface_up
INTERFACE_DOWN	mn_default_INTERFACE_DOWN_handler monitor_interface_down
FA_ADV_EXPIRE	monitor_del_fa
FA_ADV_RECEIVE	monitor_add_fa
FA_GET	mn_default_FA_GET_handler monitor_get_fa

5.3 Policy-based MA selection and detection

Figure 5.5 describes the algorithm used in the MA selection. Policies, priorities, and configurability affect the MA selection and detection in the MN.

Priorities and normalization

The interface priority is combined with the priorities of each MA in the MA selection system. The Monitor component can be configured to give a weight for the interface priority. By default, the interface priority is used as the default priority for FAs that are reachable via interfaces without SQ sensor. With wireless interfaces that are capable of measuring SQ values no interface priority is used but the SQ value is used to produce the priority.

SQ values are queried with `iwspy(8)` ioctls [39]. An ioctl is a function that is used for exchanging information with the lower level driver in the kernel. The iwspy ioctls are implemented in the device driver [40] and are defined in the Linux kernel source (linux/wireless.h) [34]. Other ioctls are also used to get information about the wireless interface parameters.

Different wireless interface cards may produce different signal quality values. Ta-

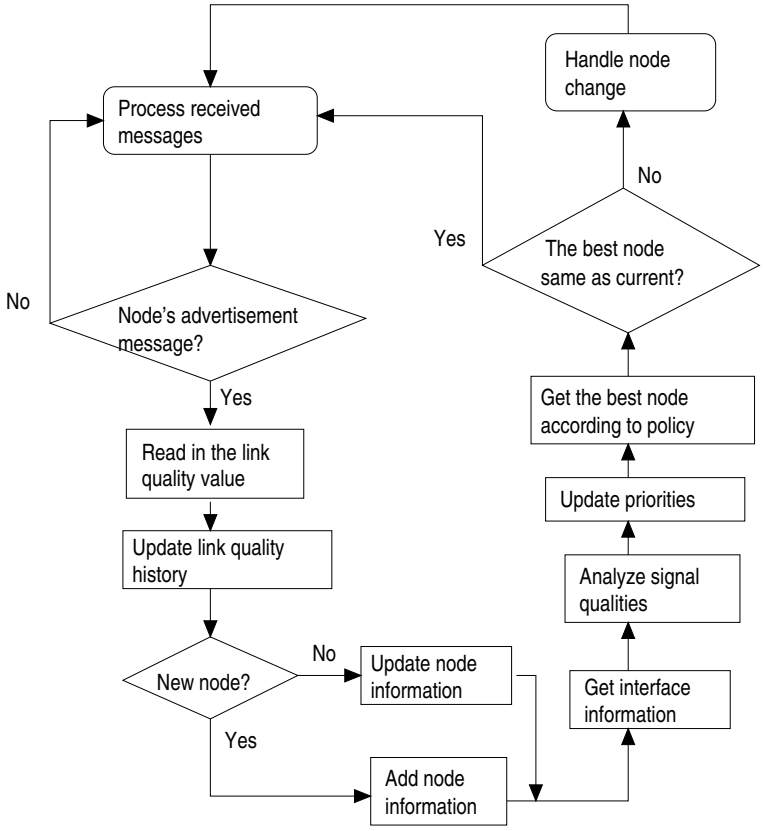

Figure 5.5: Flowchart

ble 5.2 shows an example difference between the IEEE (Institute of Electrical and Electronics Engineers, Inc.) 802.11 standard based wireless interface card implementations from Nokia and Lucent. The IEEE 802.11 standard [41] specifies that the SQ values may vary between zero and 255. Additionally, there are two kinds of values that are used to measure the wireless data path quality to the MA. The first is the SQ value and the other one is the *Received Signal Strength Indicator* (RSSI) [41]. Both values are optional and the exact meaning for these variables is not specified in the IEEE 802.11 standard. The only requirement is that they have to be comparable with each other in one implementation. For this purpose, the SQ analyzer normalizes the priorities to a range from 0 to 100. The maximum value is queried from the driver and is used to normalize the values.

After the normalization all MAs that are reachable via different devices can be compared roughly since the normalization does not take into account the different charac-

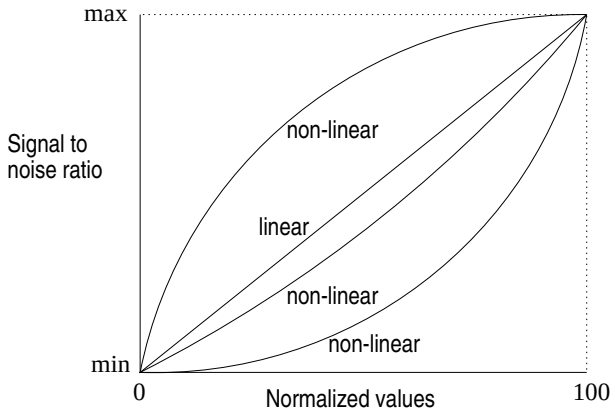

Figure 5.6: SQ normalization problem

teristics of the SQ values of different cards. The scale may not be linear compared to the real signal to noise ratio. The normalization is not distorted if the scale is linear. But if the scale is not linear the normalization distorts the SQ values. Figure 5.6 illustrates this. For accurate comparisons the compared normalized values should be equally distorted.

Additionally, in some cards the SQ value may change rapidly while in other cards it is quite steady.

Table 5.2: Signal qualities

Card type	Range	Explanation
*Nokia C*020	0 - 63	relative values
Lucent WaveLAN	0 - 92	dB values

Configurable Monitor

Monitor contains the FA selection mechanism. It is used with wireless interfaces that have SQ sensor. Monitor includes SQ collector that monitors the SQ values and keeps a SQ history in memory for further processing in SQ analyzer.

In addition to different policies the monitoring system in MN is configurable. Configurable parameters include: *threshold*, *min-balance*, *expire-percent*, *old-FA-factor*, *worst-min-time*, *worst-max-time*, and *average-length*. Each parameter has a default value that

can be changed with the `dynmn_tool(8)` configuration tool. Figure 4.2 shows the relations between the configuration variables. By tuning the values we can meet the requirements for different environments and needs.

Agent expiration

The MA selection system chooses the communicating MA for the MN from a list of MAs. Each entry in the list has a certain expiration time that is bound to the MA agent advertisement lifetime. This lifetime is three times the agent advertisement interval which is configurable in the MA. When the MA agent advertisement lifetime is expired the entry is removed from the SQ collector and all information about the MA is cleared. The MA is handled as newly detected next time an agent advertisement is heard from it. In addition to this the *node selector* can be configured to use its own expiration method with early-expire policy for the MA selection system. The expire-percent configuration parameter value that is used to calculate the expiration time for the MA from the agent advertisement lifetime. After the calculated expiration time is exceeded the *old-FA-factor* is used to decrease the MA priority. The idea behind this is that MAs may not be reachable when the MN moves relatively to the MAs. To speed up the MA detection this variable is used to decrease the priority of the FAs that are heard more seldom than the true FA agent advertisement interval is.

Threshold and minimum balance

MN changes MAs based on the priorities. The *MA switching threshold* is a percent value that is used in priority comparisons. If the current MA priority is at least *threshold* percents of the compared MA, MN will not change the current MA, but priority balancing is used.

When the compared priority is below *minimum balance* percents of the maximum, threshold is not used, but the MA with the highest priority is choosed. The best value for this variable depends on the underlying link layer technology.

The number of monitored MAs

MA is monitored when the SQ values are measured from the received packets of the MA. The Linux kernel limits the maximum number of MAs that the SQ collector can monitor at the same time to eight. When SQ collector monitors maximum number of MAs and new MA is detected some compromises have to be made. Either, one entry is dropped from the currently monitored entries in the SQ collector, or the new MA is discarded. When MA is discarded it is not available for comparison with other MAs and thus a potential candidate for the communication partner is lost. This affects the communication availability.

When space is needed for the newly detected MA and all slots are reserved, the following procedures are taken to drop one of the monitored MAs.

- If the MA is monitored, but no advertisements are heard from it in *worst maximum time* seconds it is marked old. Old entries are dropped if new slots are needed for newly detected MAs.
- If no old entries are found then the entry with the worst signal quality value is dropped. The *worst minimum time* is a configurable parameter that tells the minimum time in seconds that the worst entry has to be in the SQ collector before it can be dropped.

A SQ cache in the device driver is a way to get rid of the limit in the Linux kernel for the maximum number of monitored nodes. The purpose for the SQ cache is to use the Linux kernel limit in a different way. The SQ cache makes sure that the SQ is available for at least for eight last received packets. The problem in this approach is that the SQ values need to be queried fast enough before the old values get replaced by the new values.

Average length (N)

$$a_{i+N} = \frac{\sum_{j=i}^{N} a_j}{N} \tag{5.1}$$

SQ collector uses this configuration variable to decide how many last received SQ

values are used in calculation of the SQ average (a). If set to 1, the effect is the same as with the eager-switching policy.

The greater the value for this variable is the slower the system is in detecting changes in SQs, which is highly related to the communication availability. If the changes in priorities with surrounding MAs is not detected fast enough the node selector may not select the optimal MA for the mobile user. On the other hand if we use eager-switching or value 1 for the average length, the system may over-react depending on the SQ sensor characteristics. The node selector changes the MA more often than is really needed to maintain or increase the communication availability. The Equation 5.1 shows how the average is calculated using the average length.

5.4 Seamless handoff

The routing capabilities of the Linux operating system have been used for routing and tunneling of data packets. An *upper tunnel* in a FA is an IP-in-IP encapsulated data path to the upper MA. A *lower tunnel* in a FA is an IP-in-IP encapsulated data path to the lower FA or, when in MN decapsulation mode, to the MN. Figure 5.7 shows how two tunnels, upper and lower, are connected together so that the data packets will go from the lower tunnel to the upper tunnel and vice versa. This is the basic mechanism that the intermediate FAs (IFA) use to handle routing of data packets to and from MNs. An IFA is a FA that is at least one level up from the LFA and at least one level down from the *highest FA* (HFA). HFA is a FA that is the root for the FA tree hierarchy.

Delayed deletion and enhanced message processing

I made some enhancements similar to caching to speed up the signaling in FAs. *Delayed mobility binding deletion* in FAs is one of these optimizations. Another caching optimization is the *delayed forward deletion* in the SFA.

A *forwarding* includes a rule, a route and a tunnel that together make up a tunneled data path for packets to and from the MN in the FA. With this optimization the location

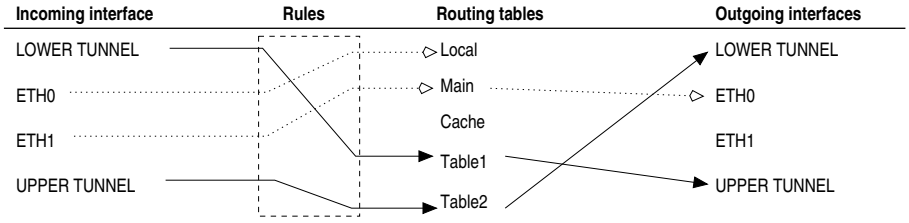

Figure 5.7: An example packet handling in a FA

update time for the MN that is switching FAs inside the same hierarchy is faster. Also some packets that may be under way in the data path will not get lost so easily.

Additionally, the registration reply processing in LFAs was enhanced. When MN receives the registration reply, it changes the default route to the new FA. In this stage the FA hierarchy must have data path ready for packets coming from the MN. Like in the registration request processing, the reply processing in IFA was enhanced. IFA creates forwardings downwards after forwarding the request to the child FA. LFA can not do this since the reply may reach the MN before the forwarding is ready. LFA is an exception to this enhancement and creates the tunnel with MN decapsulation to the MN before forwarding the reply to the MN.

Jouni K. Malinen enhanced the tunnel management in the FA hierarchy [42]. In FAs the lower level tunnel device creation overhead is avoided with tunnel devices that are statically created on FA startup.

The Impact on Data Transmission with Hierarchical Mobile IP

While MN changes the current MA, the route for packets to and from MN changes. This requires routing updates in MAs and MN. Packets destined to MNs are encapsulated in HA and decapsulated either in MN or in the lowest FA. Figure 5.8 shows tunnels between FAs and HA with *reverse tunneling* [43] and FA decapsulation. The HUT Dynamics FAs use explicit tunnels in both directions. This means that HFA and IFAs have to make one

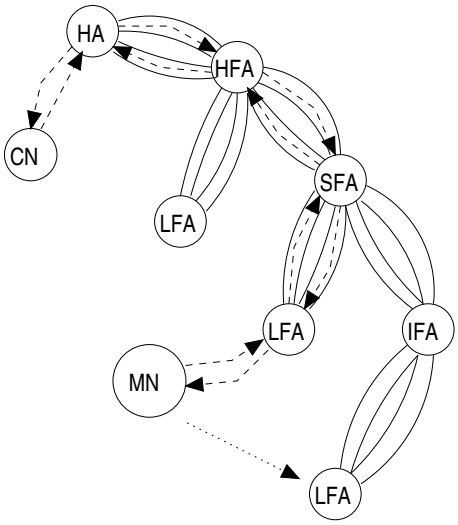

Figure 5.8: Reverse tunneling and FA decapsulation

tunnel upwards and another downwards. LFA also uses the tunnel upwards but does not make a tunnel downwards if FA decapsulation is used with the MN.

When MN changes the LFA in Figure 5.8 the new LFA forwards the request to the next upper FA. This IFA forwards it again up to the SFA which notices that the downward route for this MN has changed. SFA sends reply to the new location of MN via the IFAs and LFA. Then it connects the lower tunnel to the upper tunnel. Finally, it changes the route for packets destined for the MN and incoming from the upper tunnel.

When IFA receives the reply from SFA, it confirms the request and connects the lower tunnel to the upper tunnel as SFA did. IFA also adds a routing entry for the MN so that packets coming from the upper tunnel goes to the right lower tunnel. When MN receives the registration reply, it changes the default route to this new FA. This rises a race problem. When the SFA changes the route to the new location of MN, all packets destined for the MN will be routed to this new path. Depending on the efficiency of the SFA and FAs, network latency and network capacity, the forwarding in the LFA may not be ready when the data already comes from the next higher FA to the LFA. This means that some of the packets in certain time period do not have a route to the MN. Further this means that the MN will not receive some packets during the handoff and the deeper

the hierarchy is the bigger is the possibility for packet loss.

My solution for this problem is to enhance the functionality in the FA. It decreases the location update latency. Following steps are taken in the FA:

1. RECEIVE registration request
2. FORWARD request upwards
3. create tunnel downwards **if LFA and FA decapsulation mode in use**
4. add route for packets incoming from upper tunnel destined for MN to the lower tunnel

—

5. RECEIVE registration reply
6. **if IFA** FORWARD reply downwards
7. connect tunnel upwards
8. **if LFA** FORWARD reply downwards

In the LFA the lower tunnel is created after the request has been forwarded upwards. Inside the FA hierarchy tunnels exist between FAs already. With this approach the tunnel creation does not delay the message processing in LFA. Additionally, the data path is ready for downstream packets when the SFA switches the route. MN changes the default route as before when the registration reply is received.

Packet loss prevention without multicast or buffers

Soft handoff is a powerful method to change the communicating MA. With soft handoff MN can hear the old and the current FA simultaneously. This ability can be exploited in the system to prevent packet loss.

The FA hierarchy should not lose any packets from or to MN when MN is switching the FA. In the routing engine of the Linux kernel route changes can be made as an atomic operation so that no packets are lost between the route update process.

I enhanced the packet loss prevention in FA hierarchy so that the data path for downstream flow is generated in request handling stage. When SFA changes atomically the route to the new MN location, data path down to the MN is ready. This makes the downstream direction of the data stream to the MN more reliable. Unfortunately the request processing stage in FAs requires more resources compared to the solution where all

tunnels and routes are done in reply processing stage. This however does not compromise the security of the system since the data stream path is changed in the SFA or the HA and both can validate the request of the MN because of the security associations. Additionally, the data stream upwards from the MN is not opened during the registration request stage but in the reply processing stage.

Simultaneous Tunnels in MN

I enhanced the MN to use two tunnels with MN decapsulation mode, one to the old location and the another one to the new location. After the registration reply has been received MN starts sending packets to the new tunnel and deletes the old tunnel. The two tunnel decreases the packet loss since the MN has possibility to receive all packets. With only one tunnel at a time the synchronization with the FA hierarchy becomes a problem. These are the steps that MN takes with MN decapsulation mode during handoff:

1. Send registration update to new FA
2. Create a tunnel to the new FA
3. Change default route atomically to the new tunnel
4. Delete tunnel to the old FA

In the FA decapsulation mode MN does not use any tunnels. Thus, there is no tunneling handling problem in the MN. It can receive packets from several FAs at the same time.

5.5 Interface handling

Different link layer technologies have varying characteristics. The mobile user may want to use wired interfaces instead of wireless interfaces when she is not moving. While moving it is not convenient to use wired interfaces. Although, the system is not link layer technology dependent, the efficiency and solidness of the solution suffers if the resolution for node prioritization is not high enough. With wireless interfaces the signal parameters like power level and the noise level can be used to increase the resolution in the node selection process. Interface prioritization helps to solve the problem when many interfaces are simultaneously available for usage.

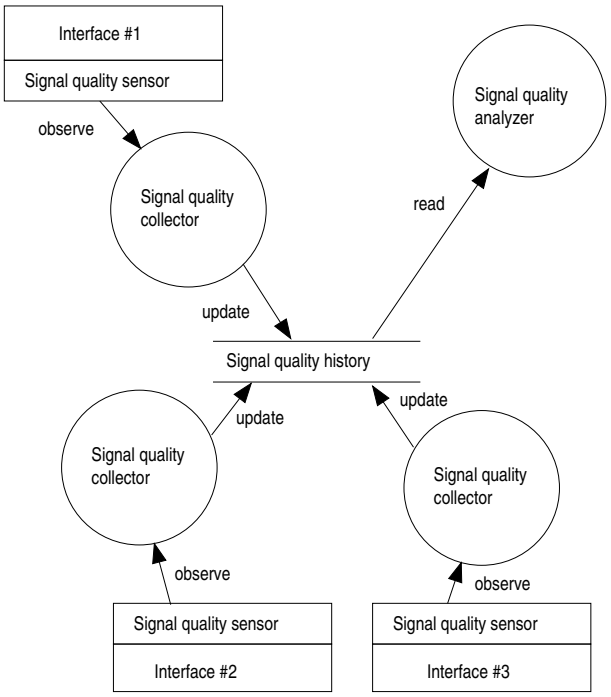

Figure 5.9: Multiple interfaces

Interface information daemon was developed for the interface priorization. The bandwidth, latency, quality of service (QoS), and cost of the used data path affects the priorization needs of the user. The IID contains a simple priorization mechanism for interfaces, but can be extended to support several parameters. IID uses an inter process communication (IPC) method with the MN and it uses a configuration file that contains the list of available interfaces for the MN.

Interface hot swapping and support for *multiple simultaneous interfaces* improves communication availability for users who are using different link layer technologies and several interfaces. Figure 5.9 illustrates the multiple interface support that was developed.

Lower level enhancements

I enhanced the Lucent IEEE 802.11 WaveLAN device driver to support iwspy ioctls (SIOCGIWSPY and SIOCSIWSPY). After a while this support became available also

in the GNU General Public License driver version that Andreas Neuhaus develops and maintains. We had also the possibility to test and use Nokia IEEE 802.11 wireless interface cards. The device driver for these cards did not contain iwspy support, which we needed for MA selection system. I made a patch for the Linux Wlan project development driver. The PRISM chipset that the Nokia card also uses does not support SQ values as specified in the IEEE 802.11 standard [41], but it supports RSSI values that are used for iwspy purposes.

I also implemented the SQ cache to the WaveLAN device driver from A. Neuhaus. Received packets are separated based on the source medium access control (MAC) address. Each MAC address has own cache entry that is updated every time a packet containing this source address is received. The cache entry that last received a packet is freshest and the other entries are older. The oldest cache entry is replaced if a packet is received with source address that can not be found from the cache. The SQ cache enables monitoring an arbitrary number of nodes.

Chapter 6

Performance analysis

In this chapter I explain my motivation for different tests. Different test setup environments are described before I show the test results and describe the tests more accurately. The focus was to test software not hardware. Tests for the handoff management and packet routing while nodes are moving proportionally to each other are included. Multiple MNs are not tested. In this chapter I also compare the results with related work.

IEEE 802.11 communication modes

The scope for IEEE 802.11 standard is to develop a MAC and physical layer (PHY) specification for wireless connectivity for fixed, portable and moving stations within a local area [41]. The standard describes two different operational modes for communication between nodes, an ad hoc (IBSS) and an infrastructure network.

In an infrastructure network an AP always exists and the communication is controlled by it. The AP usually acts as a gateway, or portal, to other parts of the network, to Internet for example. Infrastructure networks are formed around APs and moving nodes roam from one AP to another. The handoff is handled in the link layer and the network layer can not decide which AP to use or when to initiate handoffs.

In the ad hoc network mode every node in range participates to the communication control and can directly communicate with each other. There is no need for an AP and thus no link layer handoff management like in an infrastructure network.

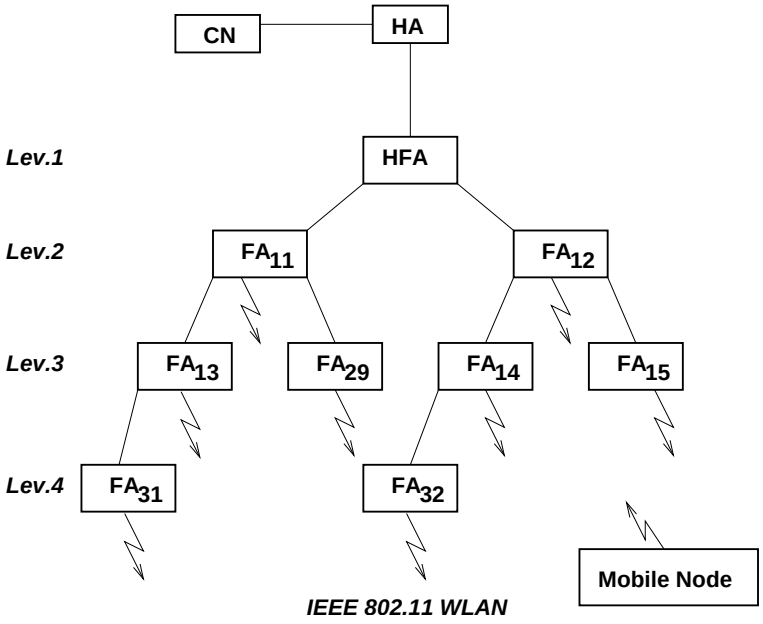

Figure 6.1: A four-level test bed setup

6.1 Test bed setup

Test bed includes hardware and software components. Some software components are made only for the tests to support emulation, measurements and result processing e.g. logging and log parsing.

Dynamics – HUT Mobile IP MN, FA and HA version 0.7pre3 were used in the test bed. All MAs resided in physically different hosts where the MN, HFA, HA, and the CN were Pentium class hosts whereas all the other FAs were less efficient, custom-built, 486-based embedded AP hosts, called Martnodes [44], with a wired and a wireless network interfaces. Martnodes and the CN used the Linux kernel version 2.2.9, MN the Linux kernel version 2.2.13 and the HA version 2.2.12. IEEE 802.11 [41] compatible 2 Mbps WLAN adapters from Lucent were used in the FAs and the MN in ad hoc mode. Device driver version 1.0.1 from Andreas Neuhaus was used in the MN and in the Martnode FAs.

The HA and the CN resided in a 100Mbps switched Ethernet laboratory network and the FA hierarchy in a dedicated switched 10Mbps network. The wired FA hierarchy was in private address-space subnet. Figure 6.1 shows the used FA hierarchy.

The MN used the wireless network for all its communication with the FAs, and all the other data between FAs and HA and between HA and CN were transferred in the wired network. The clear bottleneck on the network was the wireless part. The maximum obtainable throughput without location updates was 1.4 Mbps using TCP and 1.6 Mbps using UDP.

Real time link quality recording and emulation

The monitor module supports SQ recording into files. Produced data files can be post processed with `gnuplot(1)` to produce graphical representations of the SQs as a function of time. Monitor records the SQ values of agent advertisements from FAs. The agent advertisement interval defines the sampling frequency for each FA. The recorded trace can be used to measure the SQs for different environments and it helps to set up the wireless network. Monitor configuration variables such as the average-length affects the recorded trace. Thus, the monitor module SQ recorder is on top of the SQ analyzer.

To support easily configurable sampling frequencies, a more specific tool called `iwspy-gather` was made for SQ environment recording. The tool can be used to collect SQ information from different FAs in varying sampling frequencies. The collecting system is based on the MN agent solicitation [2] messages. Every time an agent solicitation is sent to the broadcast address, every FA that hears this message will reply automatically to it. This approach is not mandatory since ICMP echo messages could be used in environments that do not have FAs installed. The `iwspy-gather` tool gets messages from all heard FAs and collects the SQ values.

`Iwspy-gather` writes SQ values with timestamps into files. Each file contains information for only one FA. The number of produced data files depends on the number of heard FAs. One data file is called *signal quality tape* (SQT). One group of files produced in one recording session with `iwspy-gather` is called *SQT set*. Timestamps in the SQT files are synchronized within a SQT set. A SQT set can thus encode variations of the physical link and movements of all communicating parties within the time period.

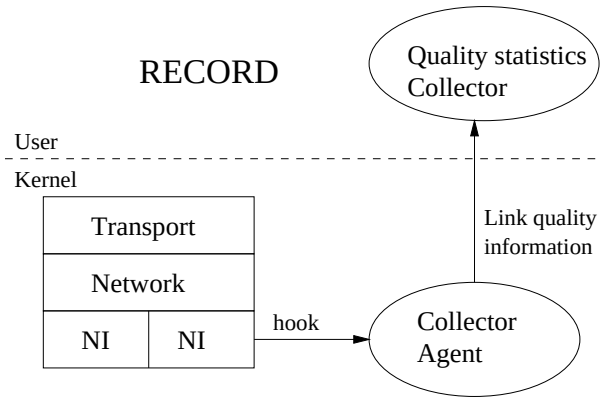

Figure 6.2: SQ environment recorder

`Iwspy-sim` is a program that uses SQTs. It was made for real life emulations and it communicates with the enhanced wireless interface network card device driver in the Linux kernel. It reads SQTs produced by the `iwspy-gather` and feeds SQ and timestamp value pairs to the driver which then replaces the actual SQ values sensored by SQ sensor. Each SQT is mapped to a MAC address so that SQTs do not get mixed. If the driver has a mapped SQT for a MAC address it is said to be in emulation mode for that MAC address. Thus, the driver can emulate received SQ values for some nodes simultaneously with nodes that do not have the SQT mapping. If the received packet is originated from an emulated node, the SQ value that was measured by SQ sensor will be replaced with a value from the mapped SQT. Emulation mode stops when the emulation starting time added with the last timestamp in the SQT is reached. The SQ values can be queried normally from the driver but the output is synchronized in time with the mapped SQT. Additionally, packet dropping can be emulated in the device driver level. The packet dropping percent is bound to the SQ value.

SQT sets can be divided and combined with other SQT sets. Imaginary SQTs can be created from scratch and combined with existing recorded SQT sets. This enables production of scenarios that would be otherwise hard to generate. SQT sets can be replayed with `iwspy-sim` multiple times which makes the emulation model convenient for testing different kinds of node selection policies. Also the characteristics for different

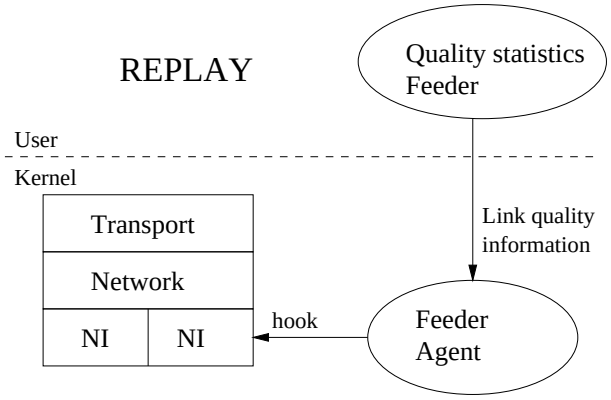

Figure 6.3: SQ environment emulator

configuration parameters can be examined.

6.2 System performance tests

I ran all the tests using FA decapsulation and reverse tunneling modes on the wireless environment. Security associations were configured between the HA and the MN, and separately in the FA hierarchy between FAs. The HFA and the HA did not have a preconfigured shared secret. Therefore, they used RSA public key encryption with 768-bit public keys for session key distribution. All the other key distribution operations used keyed MD5 [45] algorithm. This kind of configuration corresponds to the case where we do not have the complexity of managing shared secrets between the HN and each FN. However, it is feasible to use shared secrets between FAs in one administrative organization as is the usual case with FA hierarchies.

Handoff latency and packet traffic measurements

In these tests MN was forced to follow a predefined FA path and handoff frequency in the FA hierarchy illustrated in the Figure 6.1. Therefore, the MN did not use the agent discovery part of the Mobile IP. In practice, the MN received the agent advertisements, but it completed a location update only when requested by a test script.

Handoff latency

I measured the handoff latency by forcing the MN to initiate a handoff between different FAs once in a 100ms. Table 6.1 contains the resulted handoff latencies. The purpose for this test was to find the effect of the hierarchy level to the handoff time.

Table 6.1: Location update latencies for some transitions

Handoff type	Average (ms)	Standard deviation (ms)
$FA_{11} \rightarrow FA_{12}$	19.1	1.2
$FA_{12} \rightarrow FA_{11}$	19.2	1.4
$FA_{13} \rightarrow FA_{14}$	30.4	2.0
$FA_{14} \rightarrow FA_{13}$	30.3	1.0
$FA_{31} \rightarrow FA_{32}$	41.4	1.5
$FA_{32} \rightarrow FA_{31}$	41.1	1.3
$FA_{13} \rightarrow FA_{29}$	23.3	0.8
$FA_{29} \rightarrow FA_{13}$	23.5	0.9
$FA_{31} \rightarrow FA_{12}$	19.2	1.4
$FA_{12} \rightarrow FA_{31}$	41.5	1.7
$FA_{32} \rightarrow FA_{13}$	30.1	2.3
$FA_{13} \rightarrow FA_{32}$	41.3	1.6
$FA_{32} \rightarrow FA_{12}$	14.6	0.9
$FA_{12} \rightarrow FA_{32}$	37.4	1.4
$FA_{31} \rightarrow FA_{13}$	14.9	0.9

The results show that the handoff latency increases linearly with the hierarchy level at least up to fourth level. In this test bed the delay due to one hierarchy level is 11ms.

Packet loss, latency and order with location updates

This test describes the packet routing characteristics with handoffs. To better understand end-to-end effects requires a closer analysis of what happens to individual packets during a handoff. `Udpcat` and `udplisten` programs were made for packet *loss*, *latency*, *duplicates* and *order* testings. `Udpcat` sends UDP [23] packets across the IP network to the `udplisten` program in certain interval. Every UDP packet contains an increasing serial number. `Udplisten` saves the serial number and timestamp of the received packets into a log file. `Udplisten` can also initiate location updates in the MN via API calls if it is started in the same host as the MN is running in.

In the test the UDP packet size was 1024 bytes and the throughput 100 kB/s, 100 packets per second. Thus the average interval between packets was 10ms. Both the directions from CN to MN and from MN to CN were tested by sending 30000 packets several times. Generated log files were parsed to obtain needed information. I tested the system without location updates when the MN was registered to the FA_{31}. Without location updates no packet order changes or duplicates occurred in both directions. Maximum delay was between 20ms and 400ms in both directions. 270000 packets were sent from CN to MN and 2 packets were lost (0.000007%). From MN to CN direction 840000 packets were sent and 10 packets was lost (0.00001%). Both packet losses are negligible but shows that packets are lost without location updates also.

Table 6.2 contains the packet losses per location update with data stream from CN to MN. The test included almost 8000 location updates per transition. The location update interval was one second. If every packet during the handoff is lost the estimated packet loss depends on the transition. If all packets are lost during the handoff, in 20ms handoff two packets are lost. The minumum handoff latency in Table 6.1 is 14ms and maximum 41ms. Thus, with non-optimized handoffs the packet loss per location update would be around one to four packets.

Transitions were tested in group. First the MN forced location update to the FA_{11} and then to the FA_{31} then to the FA_{29} etc. The `udplisten` program received UDP packets and forced location updates in the MN, logged every received packet into a log file and marked location updates in to the log file. Location update started when the `udplisten` program used the MN API to change the forced FA. After that the MN was forced to update location to the forced FA IP address. When MN received the reply and the location update was successful it replied to the `udplisten` program through the API. `Udplisten` program marked the location update end to the log file. All lost packets between the location update starting mark and 100ms after the location update ending mark were included into the packet loss calculations. The average number of duplicated

Table 6.2: Data stream from CN to MN: packet loss

transition	lost packets/update
$FA_{11} \leftrightarrow FA_{31}$	0.00
$FA_{31} \leftrightarrow FA_{29}$	0.00
$FA_{29} \leftrightarrow FA_{32}$	0.00
$FA_{31} \leftrightarrow FA_{13}$	0.00
$FA_{12} \leftrightarrow FA_{15}$	0.00
$FA_{15} \leftrightarrow FA_{31}$	0.03
$FA_{32} \leftrightarrow FA_{11}$	0.07
$FA_{13} \leftrightarrow FA_{12}$	0.10

packets in the 30000 packet sending session was 0.98 packets (0.003% of all packets) and the average number of packets that changed order was 0.76 packets (0.003% of all packets). The maximum packet delay between received packets per 30000 packets session changed between 23ms and 700ms. Average delay was 10ms, as expected.

Table 6.3 contains the packet losses per location update with data stream from MN to CN. The test included 20000 location updates per transition. The location update interval was 300ms. Transitions were tested separately. The average number of duplicates in the 30000 packet sending session was 0.006 packets (0.000% of all packets) and the average number of packets that changed order was 0.44 packets (0.002% of all packets). The maximum packet delay between received packets per 30000 packets session changed between 20ms and 400ms. Average delay was 10ms, as expected. In this test it was expected that every lost packet was due to the location update. The number of lost packets varied depending on the transition.

Table 6.3: Data stream from MN to CN: packet loss

transition	lost packets/update
$FA_{11} \leftrightarrow FA_{31}$	0.27
$FA_{31} \leftrightarrow FA_{29}$	0.27
$FA_{29} \leftrightarrow FA_{32}$	0.00
$FA_{31} \leftrightarrow FA_{13}$	0.15
$FA_{12} \leftrightarrow FA_{15}$	0.14
$FA_{15} \leftrightarrow FA_{31}$	0.00
$FA_{32} \leftrightarrow FA_{11}$	0.00
$FA_{13} \leftrightarrow FA_{12}$	0.00

The results show that packets are lost during a handoff in some transitions. Addi-

tionally, the proportional packet loss distribution differs with data streams from MN to CN and from CN to MN.

End-to-end performance

End-to-end performance was measured as throughput. In the test the location update frequency was increased and UDP and TCP data stream throughput was measured. A throughput suitable for video streams of 1.4 Mbps, such as a near TV-quality MPEG-1 [46], was chosen as the speed for the data streams. Motivation for the latter measurements was to find out how frequently the location updates could be performed with the system when using a representative multimedia application. End-to-end performance depends on the packet routing characteristics that the previous test identified. Especially the TCP protocol is more sensitive for packet loss and delay than UDP which can be seen from the Figure 6.4.

Location updates were forced so that the MN started a new registration with given intervals. If the MN could not complete the previous registration before the new one began, a script added a firewall filter to drop the incoming packets from the CN until a registration succeeded. This corresponds to the situation in which the MN is too fast for the registration procedure to complete in time. Figure 6.4 contains the throughput graph and Table 6.4 shows the data points more accurately.

Throughput tests were made with `netperf`. It is a benchmark that can be used to measure various aspects of networking performance. The primary focus of the `netperf` is on bulk data transfer and request/response performance using either TCP or UDP and the Berkeley Sockets interface [47]. Socket send and receive sizes were 1024 bytes with UDP stream and 4096 bytes with TCP stream. 60 second throughput test was repeated several times with each location update interval.

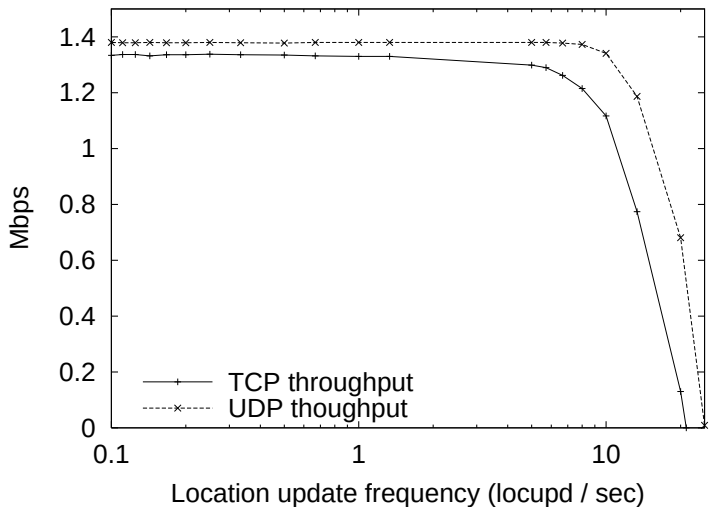

Figure 6.4: UDP and TCP throughput with location updates

Effects of the link quality to throughput

The 802.11 link layer has several power levels for sending data across the air [41]. The received SQ level depends on the power level of the sending partner and the surrounding environment e.g. walls, distance to the signal source, and the position of the antennas. The throughput was coarsely tested with different SQ levels to get a high level picture of the WLAN performance. Channel 6 was used with this test. Bulk UDP data stream throughput was measured with the `netperf` benchmark tool.

Test included two hosts, the AP and the MN. The SQ level was monitored and when a wanted range was achieved the `netperf` was used to test the bulk UDP data stream throughput from AP to the MN. Each test lasted 120 seconds and every received UDP packet in the MN was measured to get the SQ level. Totally 16 different ranges of SQ levels was measured. Figures from 6.5 to 6.12 shows the SQ histograms for the tests. Table 6.5 shows the corresponding UDP bulk transfer throughput obtained with each SQ level range. Test environment was an office floor with dense walls (see Figure 6.13).

The purpose for this test was to find out how the SQ affects the throughput in the network level. It can be seen that the throughput is relatively stable with SQs greater than

Table 6.4: UDP and TCP throughput with location updates

Location updates per second	UDP throughput (MB/s)	standard deviation (MB/s)	TCP throughput (MB/s)	stdandard deviation (MB/s)
50.00	0.00	0.00	0.00	0.00
20.00	0.68	0.16	0.13	0.03
13.33	1.19	0.17	1.12	0.10
10.00	1.34	0.07	1.12	0.10
8.00	1.37	0.02	1.22	0.05
6.67	1.38	0.01	1.26	0.04
5.71	1.38	0.00	1.29	0.03
5.00	1.38	0.00	1.30	0.02
1.33	1.38	0.00	1.33	0.02
1.00	1.38	0.00	1.33	0.02
0.67	1.38	0.00	1.33	0.02
0.50	1.38	0.02	1.34	0.01
0.33	1.38	0.01	1.34	0.01
0.25	1.38	0.00	1.34	0.01
0.20	1.38	0.00	1.34	0.01
0.17	1.38	0.00	1.34	0.02
0.14	1.38	0.00	1.33	0.05
0.13	1.38	0.01	1.34	0.01
0.11	1.38	0.01	1.34	0.01
0.10	1.38	0.00	1.33	0.04

10 dB. When the SQ is very low packets are dropped. Lower and thin histograms reveals this also. This test also shows that the SQ values with the Lucent WLAN cards changes about 7 dBs when the receiving and transferring cards are not moving. The estimate was that the SQ distribution generally sharpens when the SQ values increase. Unfortunately, this is not true all the time as can be seen from the figures. The changing power levels in the WLAN cards affect the SQ distribution.

The test results can be used to configure the monitor. For example the min-balance configuration variable would be good to set to 10 or above with IEEE 802.11 compliant WLAN cards from Lucent.

Different policies and configurations

Monitor supports four different policies: newest-FA, eager-switching, short-interval and the default policy. Additionally different configuration parameters are available for fine tuning and adjusting in different environments and with different link technologies.

In this test the SQ environment recording and re-playing system was used. Figure 6.13 shows the places for different APs in the office environment and route traveled

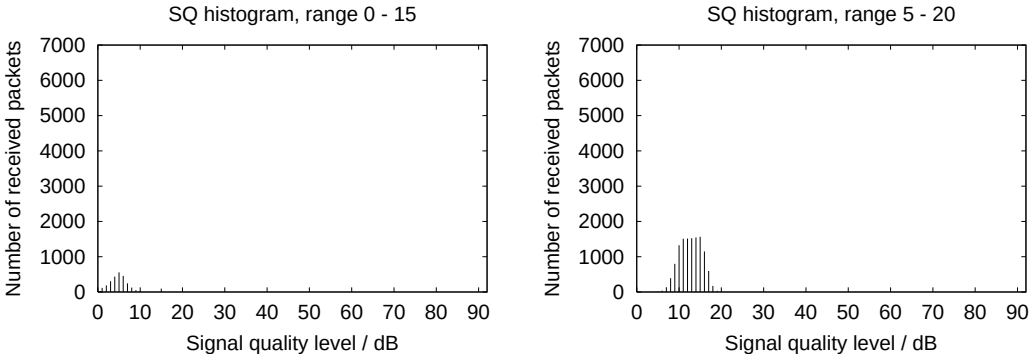

Figure 6.5: SQ 0 – 15 dB and SQ 5 – 20 dB

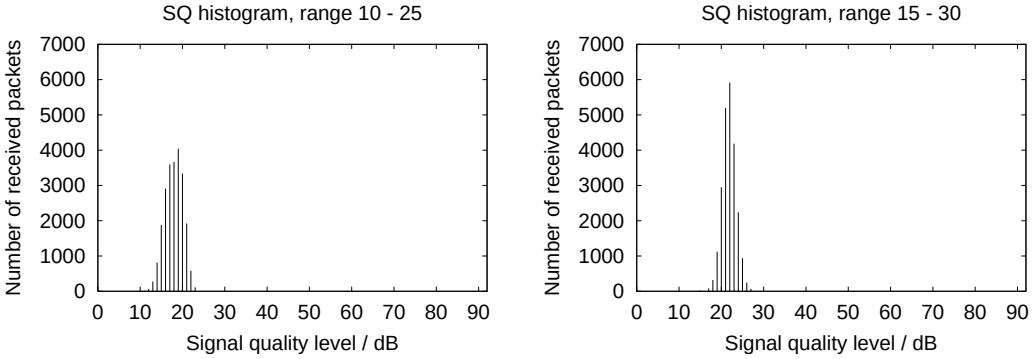

Figure 6.6: SQ 10 – 25 dB and SQ 15 – 30 dB

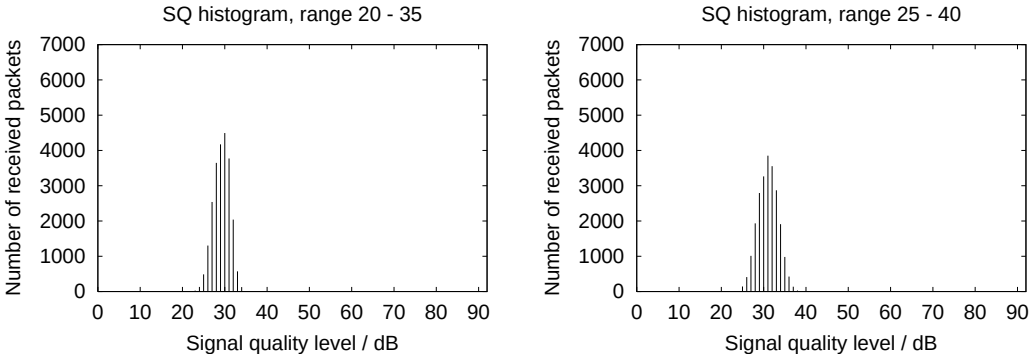

Figure 6.7: SQ 20 – 35 dB and SQ 25 – 40 dB

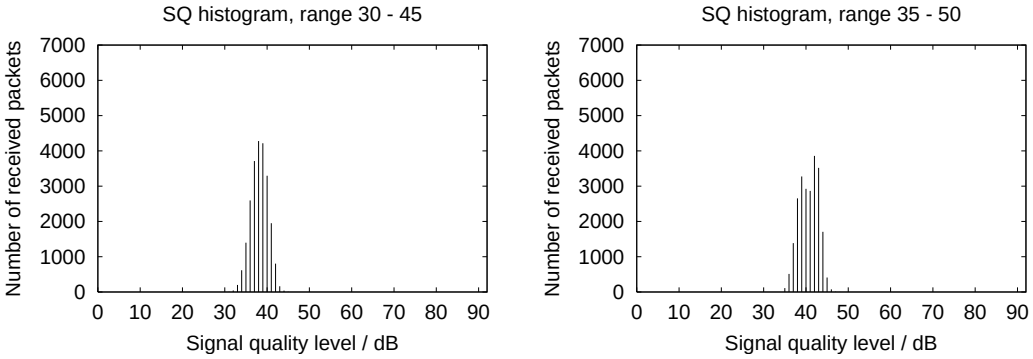

Figure 6.8: SQ 30 – 45 dB and SQ 35 – 50 dB

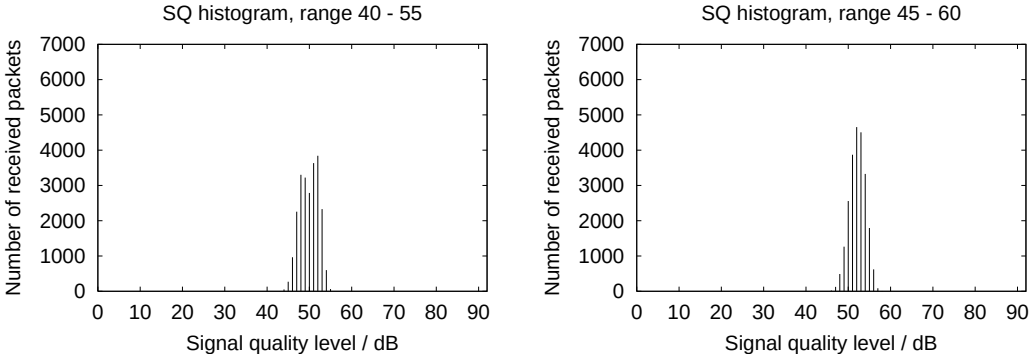

Figure 6.9: SQ 40 – 55 dB and SQ 45 – 60 dB

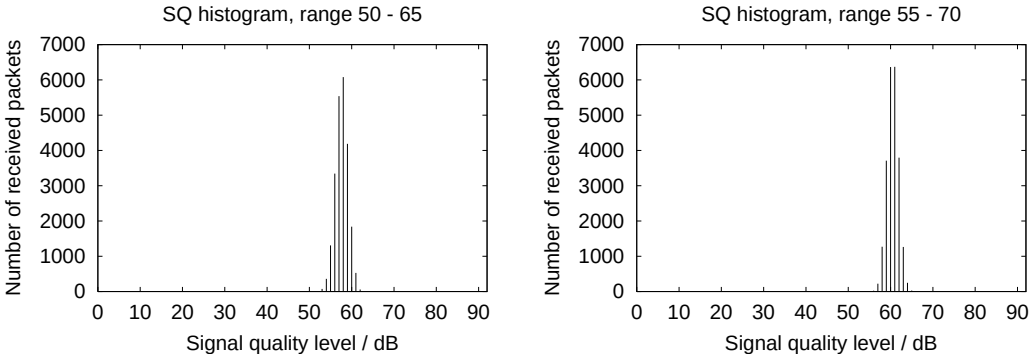

Figure 6.10: SQ 50 – 65 dB and SQ 55 – 70 dB

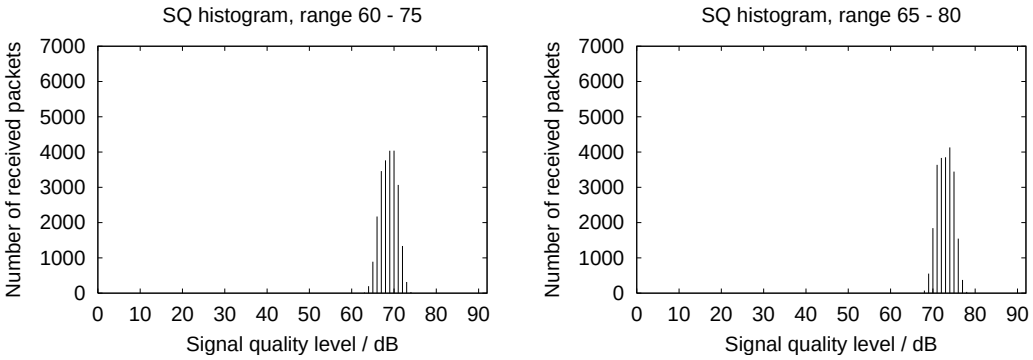

Figure 6.11: SQ 60 – 75 dB and SQ 65 – 80 dB

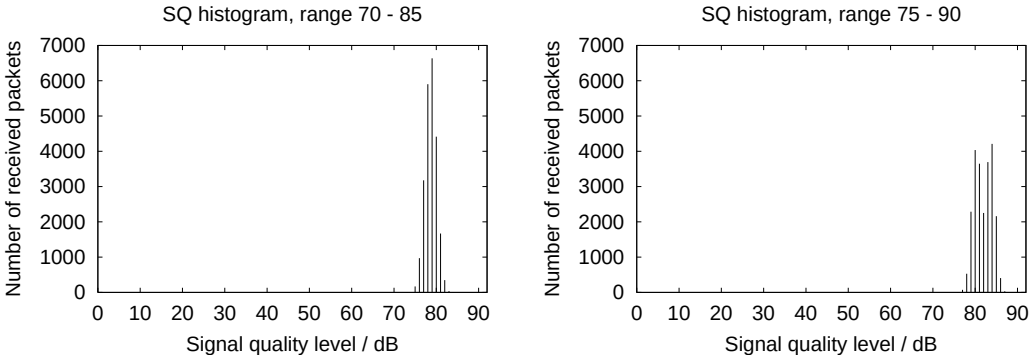

Figure 6.12: SQ 70 – 85 dB and SQ 75 – 90 dB

Table 6.5: Throughput in different signal quality ranges

SQ range (dB)	UDP throughput (MB/s)
0 – 15	0.17
5 – 19	0.83
10 – 25	1.58
15 – 30	1.58
20 – 35	1.58
25 – 40	1.58
30 – 45	1.58
35 – 50	1.58
40 – 55	1.58
45 – 60	1.58
50 – 65	1.58
55 – 70	1.58
60 – 75	1.58
65 – 80	1.58
70 – 85	1.58
75 – 90	1.58

by the mobile user. The route is marked with numbered bullets from 1 to 50. Figure 6.14 shows the recorded SQT set and the bullets in time line.

Monitor testings

I tested the monitor with two different settings with the recorded SQT set. Table 6.6 shows the two settings. Additionally a test was made without the help of the monitor e.g. the SQ sensor, SQ collector and SQ analyzer. The device driver was enhanced to drop packets when the SQ is low. Table 6.7 shows the corresponding SQ for each dropping percent. Packet dropping was used to simulate the wireless media and to find out the differences between the tests. The SQT set was re-played with the two different monitor

Table 6.6: Monitor settings

	Setting 1	Setting 2
Threshold	50	1
Min-balance	10	13
Expirepercent	50	50
Old-FA-factor	50	50
Worst-min-time	10	10
Worst-max-time	20	20
Average-length	1	3
Early-expire	OFF	OFF
Newest-FA	OFF	OFF
Eager-switching	ON	OFF

Table 6.7: Packet dropping percent bound to the SQ

SQ (dB)	Packet drop percent
<= 4	100%
5	90%
6	75%
7	33%
8	20%
9	10%

settings and without the monitor. Figure 6.15 contains the three different graphs that were recorded by the monitor. X axis describes time and y axis the SQ level and lost packet amount in last one second. The `Udplisten` and `udpcat` programs were used to flood 100 UDP packets per second from CN to the MN while re-playing the SQT set. Thus, the maximum packet loss amount is 100. Lost packets were calculated and marked into the graph with impulses. The FAs that the monitor used can be seen in the figures with different colors. A handoff has occurred when the color changes. Table 6.8 shows the number of location updates and lost packets with these three scenarios.

Table 6.8: Monitor testing results

	Plain Mobile IP	Monitor settings 1	Monitor settings 2
Lost packets	2179	66	117
Location updates	8	63	9

With plain Mobile IP settings SQ values are not used in FA selection. MN switches the FA if the agent advertisement lifetime expires. The agent advertisement interval is crucial since it determines the lifetime for the advertisement. By default it is three times the agent advertisement interval.

With plain Mobile IP setting the MN loses considerably more packets than with the monitor. MN with monitor settings 1 does location updates very eagerly compared to the other two, but the packet loss is lower. Eager-switching is not the best policy since it makes location updates much more frequently than the other two. When monitor settings 2 is used the packet loss is low and the number of location updates is almost as low as with the plain Mobile IP setting. The threshold with monitor setting 2 was set to 1 and

min-balance to 13 which makes the MN switch the MA when the current MA priority is below 13. Additionally the average-length was 3 with setting 2 which makes the priorities more stable than with the setting 1. All these changes in the setting 2 decreases the location updates compared to the setting 1.

6.3 Handoff protocol analysis

The handoff protocol that the system provides uses horizontal MCHOs. They are soft and classified as forward type handoffs. Additionally the handoff can be described as a *two-phase handoff*. In the first phase the route for downstream packets is changed and in the second phase the upstream route for the packets is changed. In the latter the system is in a state where the up- and downstream packets are routed via different APs. After the second phase the handoff has completed. This is possible only with soft handoff. Figure 6.16 illustrates the two-phase handoff.

Locality is exploited because of the hierarchical structure of the FAs. The localized location updates reuses partially the path between MN and HA, and depending on the FA hierarchy the re-used path may be relatively long. The lower in the hierarchy the SFA is the longer is the re-used path in the FA hierarchy during the location update. In a MA selection process radio hints are used to achieve seamless and glitchless handoffs. This is possible because of the finer granularity in FA comparison and direct knowledge of the wireless data path SQ characteristic. Coarsely, the better the SQ the better the packet delivery and thus better communication availability. No handoff request queuing is performed in the FAs.

Scalability issues are not tested. The soft handoff does not use neither specific buffers nor multicasting for packet loss prevention. Signaling load is shared in the hierarchical structure with localized location updates. These two things might improve the scalability with multiple MNs but need to be tested and analyzed more throughly. The HUT Dynamics Mobile IP supports signaling prioritization. Signling is prioritized over

the data packets, which makes it more tolerable for congestion in the network.

Service disruption time is comparable to the glitches that the MN or CN experiences during data stream transfers. Lost packets, packet order changing and relatively high latencies are sources for service disruptions. Service disruption affects communication availability. The more the service is disrupted the worse the communication availability becomes. With WLANs packet loss can not be eliminated completely if the AP coverage is sparse. Even in dense WLANs the reflection and interference may cause service disruption. The soft handoff with the localized location updates completes relatively fast and the packet loss rate is negligible. Data streams like UDP and TCP perform well with up to five location updates in one second. In an office environment such a high location update frequency is highly improbable and may indicate that the WLAN architecture should be re-planned.

The tree like structure of the FA hierarchy does not have any loops, but that alone does not prevent data looping. Data looping in the FA hierarchy is eliminated with routing rules and routing tables with black-hole routes.

6.4 Comparison with related work

Srinivasan Seshan in his Ph.D. thesis made handoff latency, packet loss and packet duplicate measurements [28]. In his implementation the handoff latency is measured between the registration request message and first data packet coming from the new AP. The implementation from Seshan does not use any registration reply messages. My test measured the latency between the registration request message and the registration reply message. In our two-phase handoff the MN may receive data packets from the old AP before the registration reply has arrived to the MN. Additionally, our handoff protocol uses replay protection and authentication which increases the handoff delay. The implementation from Seshan does not take care of the security issues. Thus, the handoff latency results are not directly comparable.

Seshan measured the packet loss during handoffs with a bit higher data rate as I have done, 1024 byte packets with 1.0 Mbit/s data rate. Without buffering or multicasting the implementation of Seshan lost several packets per handoff (2-5). Even with multicast-based handoffs the packet loss rate was several packets. When buffering was used with multicasting the packet loss rate was negligible. In our implementation the packet loss rate is negligible without multicasting or buffering.

Fikouras et al. measured the traffic disruption time with Mobile IP and with different handoff policies [7]. The traffic disruption time with Mobile IP handoffs and UDP traffic was up to six seconds and with TCP traffic more than ten seconds. They did not use hierarchical Mobile IP or SQ values to determine the best FA to register with. Their measurements showed that eager switching was the best choice when traffic disruption time is minimized. In my tests the eager switching policy behaved worse than the default-policy. Additionally, service disruption times were negligible in our environment.

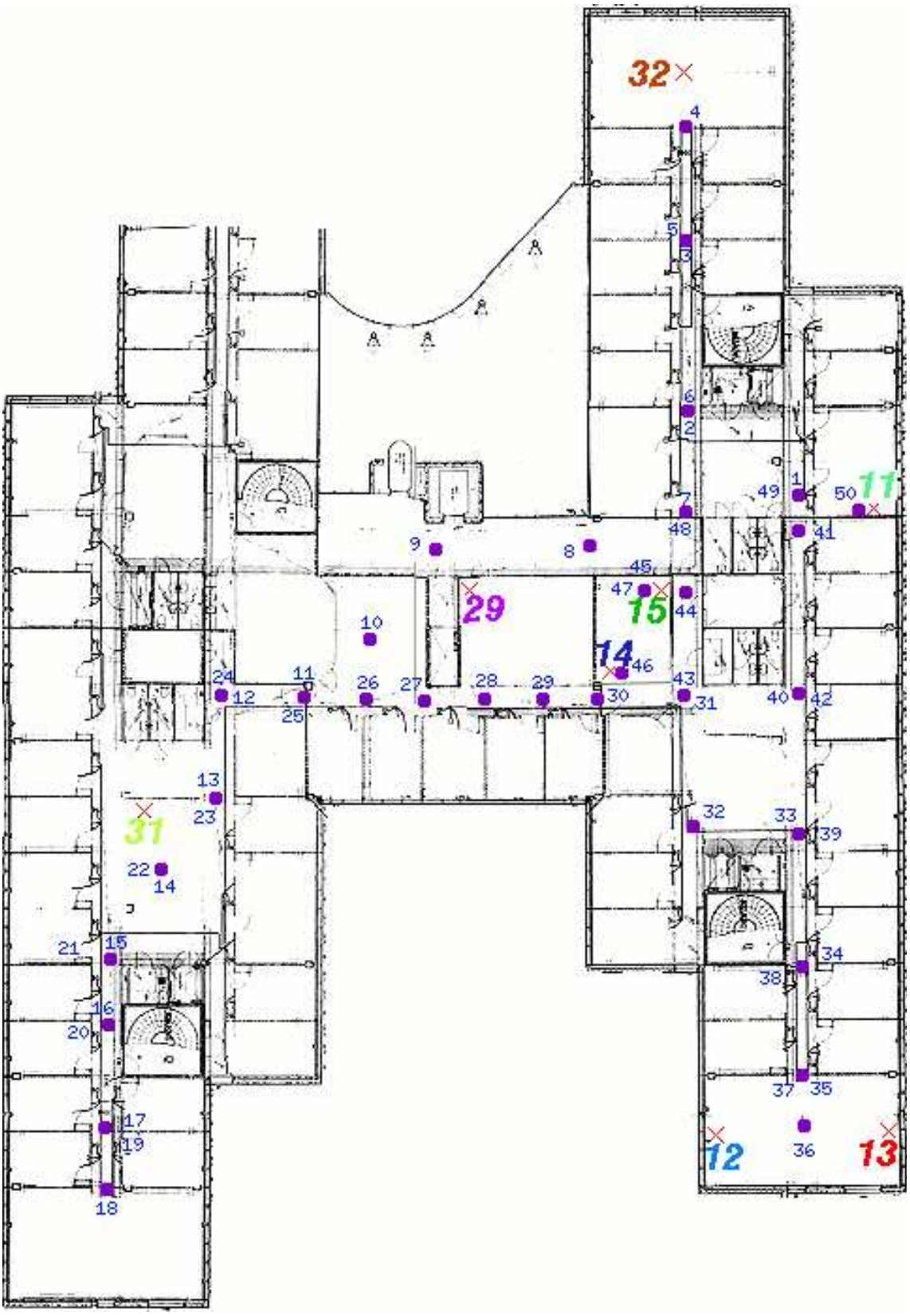

Figure 6.13: AP locations in a sample office environment

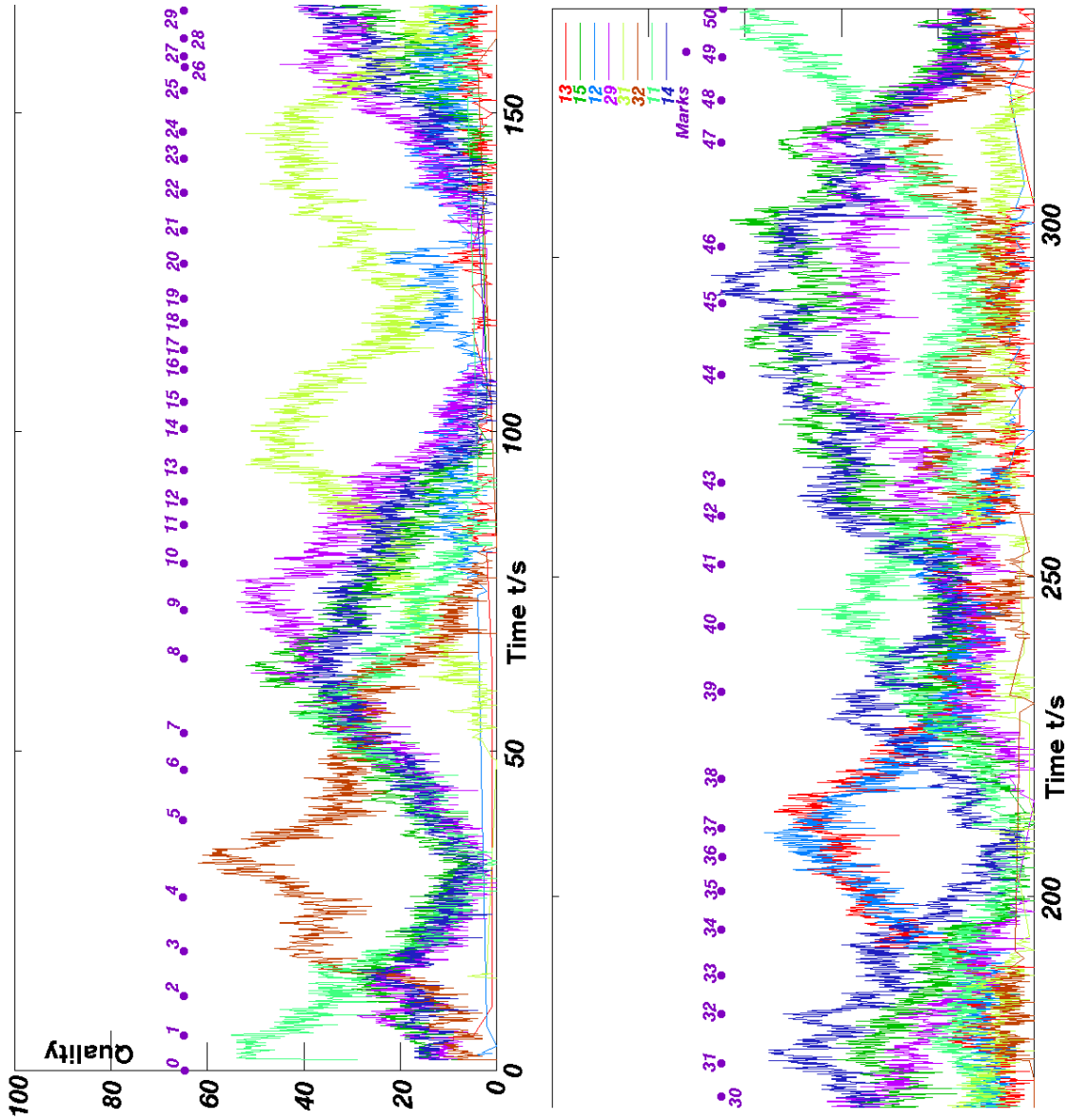

Figure 6.14: SQT set recorded in the office environment

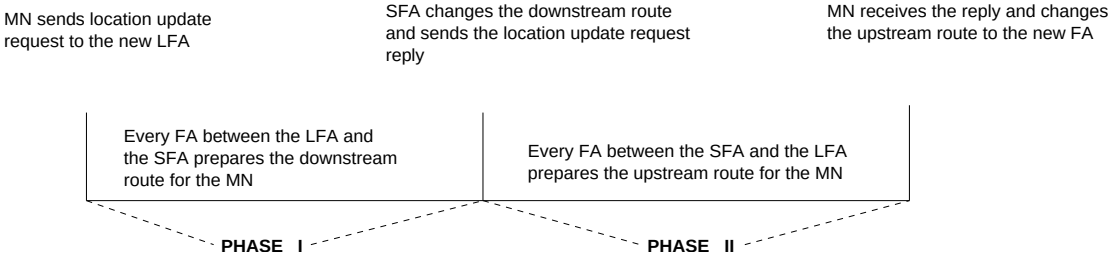

Figure 6.16: Two-phase handoff

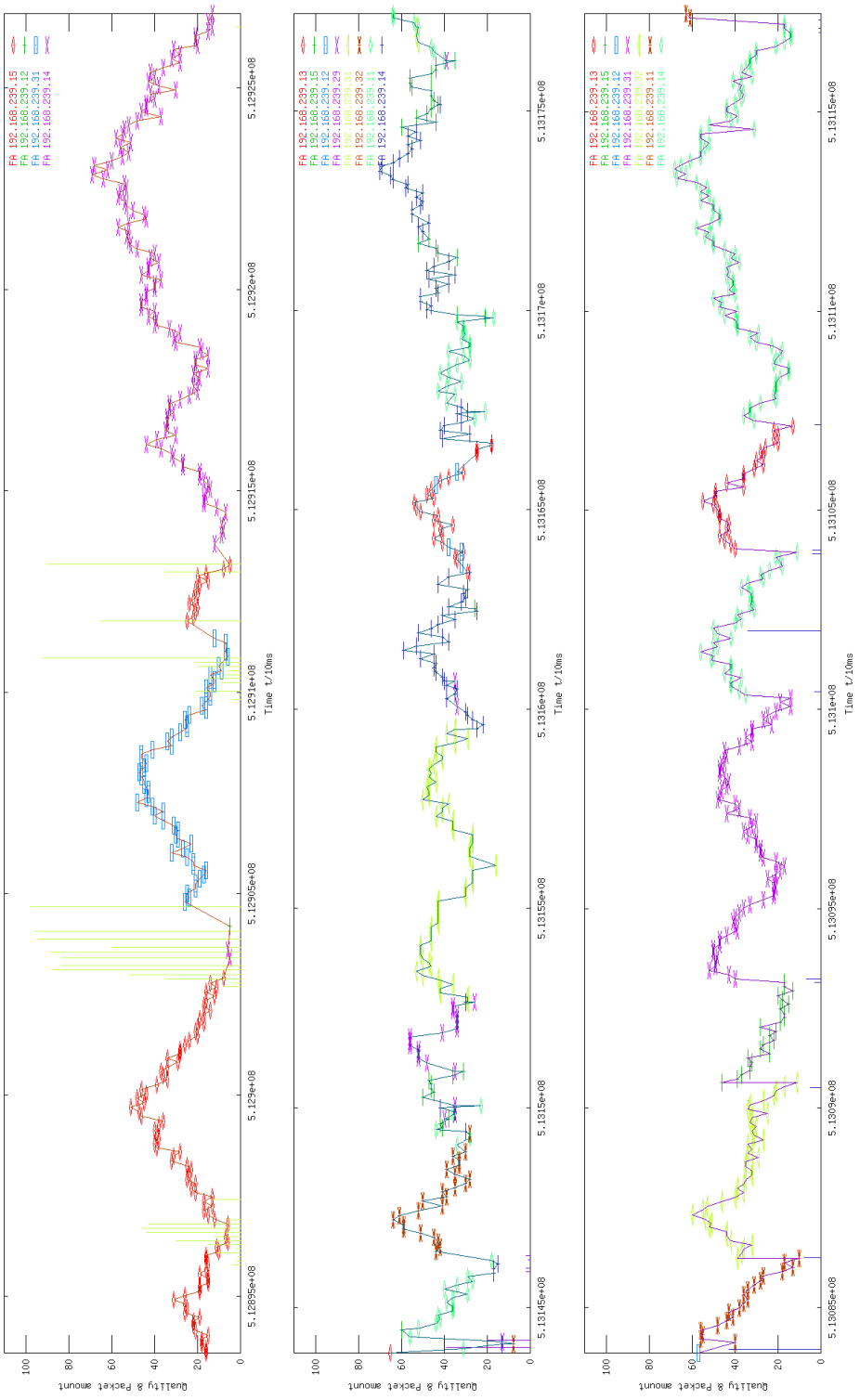

Figure 6.15: Plain Mobile IP and SQT replay with monitor settings 1 and SQT replay with monitor settings 2

Chapter 7

Conclusions

I developed a general event driven node selection mechanism based on the radio link signal qualities. I enhanced the Dynamics – HUT Mobile IP system to support glitchless and seamless handoffs in wireless local area networks. Configurable mobility agent selection system in the mobile node is based on prioritization and techniques that affect the priorities. I developed and made a signal quality environment emulator that can be used to test different mobility agent selection policies and configurations. Additionally, I added support for multiple interfaces and device hot swapping into the mobile node.

The tested and enhanced system improves packet delivery to and from the moving mobile node with the registered home IP address. Glitchless and seamless handoffs are automated in the mobile node. Handoff management does not affect the packet routing latency although it may change if the hierarchy level of the lowest foreign agent changes. Different policies and configurable handoff management with a modular implementation fulfills the modular node selection criteria. Additionally, the mobility agent detection and mobility agent selection systems make possible for the mobile node to choose the mobility agent that offers best communication availability.

- The enhancements help the user to switch from the wired office network to the WLAN. It also improves the communication availability since the user can attach to the network more easily and still maintain connections over different media. The mobile user can change the policy and handoff management parameters dynamically

while moving and without disturbing the communication sessions. Thus, the system is flexible and adaptively adjusts to the needs of the user.

- With soft handoffs neither buffering nor multicasting is required to get seamless handoffs. Soft handoff capable peer-to-peer link technologies enable simplicity both on the link and on the network layer. Thus, the implementation is simpler and more robust. On the other hand, this solution requires soft handoff capable point to multipoint link layer such as the Ethernet like IEEE 802.11 ad hoc mode network.

- The solution is not dependent on the handoff management below the network layer. Thus, it is also independent of the underlying physical characteristics of the link level radio technology. Although it uses the link level signal quality values when available to achieve the needed communication availability, there are different link level radio technologies that provide this information. In Mobile IP the mobile node controls the handoff management in the network layer. With respect to this the ad hoc mode suits well for this system. With newest-FA policy mobile node can be used in the wireless 802.11 infrastructure network also.

- The priority based foreign agent comparison is feasible because it is not bound to the signal quality values and thus only WLANs. With priorities different value functions can be combined to get the overall priority and the best choice over different possibilities. Priority degradation is one example of the value functions and can be used to improve communication availability in a more reliable way. Priority increasing supports recovering when priority decreasing is used. Both the priority decreasing and the priority increasing together make the system more tolerable on temporary and static failures of access points or network connections.

- Signal quality awareness is a simple but effective way to improve the communication availability without extending mobility protocols. It is scalable and independent, from intra-WLAN through micro mobility to macro mobility.

- The tests showed that hierarchical Mobile IP with signal quality awareness and two-phase handoff supports micro mobility. This is unique and has not been done before. Handoffs can be done more frequently than is on average needed in an office environment. Five handoffs per second with negligible packet loss and with session maintenance is sufficient for even higher needs.

The handoff protocol uses sparingly the radio channel since it does not send multiple signaling messages during a handoff. The registration request is used to initiate location updates and the registration reply is used for authentication purposes and to finish the two-phase handoff. The solution is architecturally natural with Internet mobility on WLANs where the smart mobile hosts can operate independently and the network is simple. Thus, I have demonstrated that a network-layer handoff support model in the mobile node is sufficient for continuous communication availability in mobile networks.

7.1 Future work

Future work includes scalability measurements with multiple mobile nodes under the same foreign agent hierarchy and a home agent. Hard handoff management is required when channel switching occurs in an ad hoc mode wireless local area network and is part of the future work. Future work includes also dynamic parameters in the foreign agent and agent advertisements to support mobility agent selection in the mobile node.

References

[1] J. Postel. RFC 760: DoD standard Internet Protocol. Jan. 1980. Obsoleted by RFC0791, RFC0777. Obsoletes IEN123. Status: UNKNOWN.

[2] C. E. Perkins. *Mobile IP Design Principles and Practice*. Addison-Wesley Publishing Company, One Jacob Way, Reading, Massachusetts 01867, 1. edition, Oct. 1997.

[3] C. Perkins. RFC 2002: IP Mobility Support. Oct. 1996. Updated by RFC2290. Status: PROPOSED STANDARD.

[4] J. Postel. RFC 791: Internet Protocol. Sept. 1981. Obsoletes RFC0760. See also STD0005. Status: STANDARD.

[5] C. Perkins. RFC 2003: IP Encapsulation within IP. Oct. 1996. Status: PROPOSED STANDARD.

[6] H. Laamanen. Serveability Issues, Series of Publications C, Report C-1998. Technical report, University of Helsinki, Department of Computer Science, 1998.

[7] N. Fikouras, K. E. Malki, S. Cvetkovic, and C. Smythe. Performance of TCP and UDP during Mobile IP handoffs in single-agent subnetworks. In *IEEE Wireless Communications and Networking Conference, WCNC*, pages 1258–1262, Sept. 1999.

[8] R. H. Katz. Adaptation and mobility in wireless information systems. *IEEE Personal Communications*, 1(1):6–17, 1994.

[9] J. Sevanto, M. Liljeberg, and K. Raatikainen. Introducing Quality-of-Service and Traffic Classes into Wireless Mobile Networks. In *WoWMoM'98. Proceedings of first ACM international workshop on Wireless mobile multimedia*, 1998.

[10] M. E. Kounavis, A. T. Campbell, G. Ito, and G. Bianchi. Supporting Programmable Handoff in Mobile Network. In *IEEE International Workshop on Mobile Multimedia Communications*, 1999.

[11] M. Stemm and R. H. Katz. Vertical handoffs in wireless overlay networks. 3(4):335–350, Winter 1998.

[12] G. P. Pollini. Trends in Handover Design. *IEEE Communications Magazine*, 34(3):82–90, Mar. 1996.

[13] L. Taylor, R. Titmuss, and C. Lebre. The challenges of seamless handover in future mobile multimedia networks. *IEEE Personal Communications*, 6(2):32–37, 1999.

[14] K. Brown and S. Singh. M-UDP: UDP for Mobile Networks. *ACM Computer Communication Review*, 26(5):60–78, 1996.

[15] R. Cáceres and V. N. Padmanabhan. Fast and Scalable Wireless Handoffs in Support of Mobile Internet Audio. *ACM Journal on Mobile Networks and Applications*, 3(4), Dec. 1998.

[16] C. Tan, S. Pink, and K. Lye. A Fast Handoff Scheme for Wireless Networks. In *2nd ACM International Workshop on Wireless Mobile Multimedia (WoWMoM'99), Seattle, Washington*, pages 83–90, Aug. 1999.

[17] J. Mysore and V. Bharghavan. A New Multicasting-based Architecture for Internet Host Mobility. In *MOBICOM 97 Budapest Hungary*, pages 161–172, Sept. 1997.

[18] A. Campbell, J. Gomez, C.-Y. Wan, Z. Turanyi, and A. Valko. Cellular IP. Internet Draft. October 1999. *(work in progress)*

[19] R. Ramjee, T. L. Porta, S. Thuel, K. Varadhan, and L. Salgarelli. IP micro-mobility support using HAWAII. Internet Draft. Jun 1999. *(work in progress)*

[20] P. McCann, T. H. J. Wang, A. Casati, C. Perkins, and P. Calhoun. Transparent Hierarchical Mobility Agents (THEMA). Internet Draft. Mar. 1999. *(work in progress)*

[21] C.-K. Toh. The design & implementation of a hybrid handover protocol for multimedia wireless LANs. In *The first annual international conference on Mobile computing and networking, MOBICOM*, pages 49–61, 1995.

[22] D. C. Plummer. RFC 826: Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware. Nov. 1982. Status: STANDARD.

[23] J. Postel. RFC 768: User Datagram Protocol. Aug. 1980. Status: STANDARD. See also STD0006.

[24] J. Postel. RFC 793: Transmission Control Protocol. Sept. 1981. See also STD0007. Status: STANDARD.

[25] A. Bakre and B. Badrinath. I-TCP: indirect TCP for mobile hosts, Distributed Computing Systems. In *Distributed Computing Systems, Proceedings of the 15th International Conference*, pages 136–143, June 1995.

[26] C. E. Perkins and K.-Y. Wang. Optimized Smooth Handoffs in Mobile IP. In *The Fourth IEEE Symposium on Computers and Communications*, pages 340–346, 1999.

[27] K. Keeton, B. A. Mah, S. Seshan, R. H. Katz, and D. Ferrari. Providing Connection-Oriented Network Services to Mobile Hosts. In *USENIX Symposium on Mobile and Location-Independent Computing*, Aug. 1993.

[28] S. Seshan. *Low-Latency Handoff of Cellular Data Networks.* PhD thesis, University of California at Berkeley, 1995.

[29] H. Balakrishnan, S. Sseshan, and R. Katz. Improving reliable transport and handoff performance in cellular wireless networks. *Wireless Networks*, 1(4):469–481, 1995.

[30] S. Tekinay and B. Jabbari. A measurement-based prioritization scheme for handovers in mobile cellular networks. *IEEE Journal on Selected Areas in Communications*, 10:1343–1350, 1992.

[31] N. D. Tripathi and J. H. R. H. VanLandinoham. Handoff in Cellular Systems. *IEEE Personal Communications*, 5(6):26–37, Dec. 1998.

[32] H. J. Wang, R. H. Katz, and J. Giese. Policy-Enabled Handoffs Across Heterogeneous Wireless Networks. In *2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 51–60, Feb. 1999.

[33] D. Forsberg, J. Malinen, J. Malinen, T. W. Tom, and M. Tiusanen. Distributing Mobility Agents Hierarchically under Frequent Location Updates. In *Sixth IEEE International Workshop on Mobile Multimedia Communications (MoMuC'99), San Diego.*, 1999.

[34] L. Torvalds. Linux: A Portable Operating System. Master's thesis, University of Helsinki, Department of Computer Science, Dec. 1997.

[35] G. Dommety. Mobile IP Vendor/Organization-Specific Extensions. Internet Draft. Nov. 1999. *(work in progress)*

[36] R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. *Communications of the ACM*, 21:120–126, 1978.

[37] H. W. Braun. RFC 1104: Models of policy based routing. June 1989. Status: UNKNOWN.

[38] Personal Computer Memory Card International Association. PC Card Standard, Mar. 1997. http://www.pc-card.com/pccardstandard.htm.

[39] W. R. Stevens. *UNIX Network Programming*, volume 1. Prentice Hall PTR, Upper Saddle River, NJ 07458, 2. edition, 1998.

[40] A. Rubini. *Linux Device Drivers.* O'Reilly & Associates, Inc., 101 Morris Street, Sebasstopol, CA 95472, 1. edition, Feb. 1998.

[41] IEEE. *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.* Institute of Electrical and Electronics Engineers, inc., New York, NY, Nov. 1997.

[42] J. K. Malinen. Using private addresses with hierarchical Mobile IPv4. Master's thesis, Helsinki University of Technology, Department of Computer Science, 2000.

[43] G. Montenegro. RFC 2344: Reverse Tunneling for Mobile IP. May 1998. Status: PROPOSED STANDARD.

[44] H. Arppe, D. Forsberg, J. Malinen, P. Massetti, and J. Salmi. Helsinki University of Technology research project: Mobile Ad-hoc Routing Testbed, MART, 1999. http://www.cs.hut.fi/Research/Mart/.

[45] R. Rivest. RFC 1321: The MD5 Message-Digest Algorithm. Apr. 1992. Status: INFORMATIONAL.

[46] ISO/IEC. ISO/IEC 11172:1993 Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s, 1993.

[47] Hewlett-Packard Company. Netperf: A Network Performance Benchmark, revision 2.1, Feb. 1996. http://www.netperf.org/netperf/NetperfPage.html.