AALBORG UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE
FREDRIK BAJERS VEJ 7E, 9220 AALBORG ØST, DENMARK

Seamless Handoff in Mobile IPv6

Development of the *Parametric Cell Switching* handoff initiation concept based on a theoretical and empirical study of common handoff initiation strategies

Master Thesis

by

Torben Wittrup Andersen
&
Anders Lildballe

Supervisor: Brian Nielsen

June 2001

Pages: 157
Number of copies: 10

Abstract

The Internet Protocol (IP) is expected to become the main carrier of traffic to mobile and wireless nodes. This includes ordinary data traffic like http, ftp and email as well as voice, video and other time sensitive data. To support mobile nodes, basic IP has been extended with protocols (Mobile IP) to support intercepting and forwarding of packets to a mobile and possibly roaming node. Seamless roaming requires that users and applications do not experience loss of connectivity or any noticeable hick-ups in traffic. This thesis proposes improvements to the Mobile IPv6 protocol.

In this thesis we focus on handoff initiation in Mobile IPv6. We have investigated two existing handoff initiation strategies, the *Eager Cell Switching* strategy and the *Lazy Cell Switching* strategy. To determine the handoff latency as a function of the configuration of essential protocol parameters, for each handoff initiation strategy we have proposed a mathematical model describing handoff performance.

To bridge the gap between theory and practice we have setup a testbed running FreeBSD 4.1 and installed it with KAME Mobile IPv6 software. Using this testbed we have studied the handoff latency experienced by an actual roaming mobile node using either of the two handoff initiation strategies. From an extensive empirical study using this testbed we conclude, that the proposed theoretical models are in excellent conformance with empirical results.

Using the mathematical models for the *Eager Cell Switching* strategy and the *Lazy Cell Switching* strategy we have proposed a new protocol configuration resulting in improved handoff performance for both strategies but without increasing network load. This theoretical performance improvement was also confirmed in experiments conducted in the testbed.

As both *Lazy Cell Switching* and *Eager Cell Switching* were found to have serious performance lacks we have proposed an advanced handoff initiation strategy (*Parametric Cell Switching*) which utilizes link layer information. A novelty of *Parametric Cell Switching* is that it can be configured to use many different criteria in a handoff decision. A particular instance of *Parametric Cell Switching*, implemented as a prototype and merged into the KAME software, showed much improved performance to both of the existing handoff initiation strategies.

We conclude that we have been successful in reducing the handoff latency for the existing handoff initiation strategies, but that in order to experience seamless roaming a mobile node must use link layer information in a handoff decision. Our prototype implementation seems promising, as in an informal building wide experiment using Wireless LAN as the link media, it was able to roam between three base stations serving three different networks without losing packets.

Dansk sammenfatning

Internet Protokollen (IP) forventes på længere sigt at blive den centrale protokol til transport af data til og fra mobile enheder. Dette inkluderer traditionelle data såsom http, ftp og email såvel som tale, video og andre tidsfølsomme data. For at understøtte mobile enheder er standard IP blevet udvidet med protokoller (Mobile IP) til at supportere opfangelse og videresendelse af data til en mobil og muligvis roamende enhed. Seamless roaming kræver at brugere og applikationer ikke oplever tab af forbindelse eller nævneværdige afbrydelser i trafikken.

Vi fokuserer i denne tese på handoff initiering i Mobile IPv6. Vi har undersøgt to eksisterende handoff initieringsstrategier, den såkaldte *Eager Cell Switching* og den såkaldte *Lazy Cell Switching*. For at bestemme handoff latency som funktion af konfigurationen af essentielle protokol parametre, har vi for hver af de to handoff initieringsstrategier foreslået en matematisk model der beskriver handoff performance.

For at relatere teori til praksis har vi opstillet en testopstilling installeret med FreeBSD 4.1 og KAME Mobile IPv6 software. Ved anvendelse af denne testopstilling har vi studeret den handoff latency som en roamende mobil enhed oplever, ved anvendelse af både *Eager Cell Switching* og *Lazy Cell Switching*. På baggrund af et indgående empirisk studie i denne testopstilling kan vi konkludere, at de foreslåede matematiske modeller er i glimrende overensstemmelse med de empiriske resultater.

Ved at anvende de matematiske modeller opstillet for henholdvis *Eager Cell Switching* og *Lazy Cell Switching* har vi foreslået en ny protokol konfiguration, der resulterer i signifikant lavere handoff latency uden at øge netværksbelastningen. Denne teoretiske forbedring af performance viste sig også ved aktuelle eksperimenter foretaget i testopstillingen.

Da både *Eager Cell Switching* og *Lazy Cell Switching* fandtes at have alvorlige mangler vedrørende performance, har vi foreslået en ny handoff initieringsstrategi kaldet *Parametric Cell Switching*, som anvender information fra linklaget. Et særkende ved *Parametric Cell Switching* er desuden at den kan anvende mange forskellige parametre til at træffe en handoff beslutning. Vi har implementeret en instans af *Eager Cell Switching* som en prototype i KAME softwaren. Denne instans viste væsentlig forbedret ydelse i forhold til de to eksisterende handoff initieringsstrategier.

Vi konkluderer, at vi har været succesfulde med hensyn til at sænke handoff latency for de eksisterende handoff algoritmer, men at disse ikke kan tilbyde seamless roaming. Det er derfor nødvendigt for en handoff initieringsalgoritme at anvende information fra linklaget. Vores prototype implementation er lovende, og i et uformelt forsøg, hvor vi installerede tre access routere i en af universitetets bygninger, var den modsat de eksisterende handoff algoritmer i stand til at roame mellem disse tre netværk uden at tabe data.

Preface

This master thesis presents the work carried out at our final semester at the Department of Computer Science, Aalborg University. The work is a continuance of initial work carried out at a previous semester, and this thesis comprises a stand alone presentation of essentials from both semesters.

In this thesis we assume the reader to be familiar with the Internet Protocol (IP), both in version 4 (IPv4) and in version 6 (IPv6). IPv4 is described in various literature including [Comer, 1997] (introductory). For a thorough description of IPv6 we refer to [Huitema, 1997]. We do not assume familiarity with the concept of Mobile IP, as this is introduced in chapter 1 and elaborated for IPv6 in appendix A.

The thesis is structured as follows. After introducing the concepts of mobility and handoff in chapter 1, chapter 2 motivates the content of this thesis, presents contributions and gives an overview of its structure. Chapter 3 contains a theoretical study of existing handoff initiation strategies and chapter 4 motivates and describes the results of an empirical study of these handoff initiation strategies. In chapter 5 we propose an advanced handoff initiation strategy which uses link layer information to decide when to initiate a handoff and in chapter 6 we present an instantiation of such an advanced handoff initiation strategy. Finally in chapter 7 we summarize our conclusions and present ideas for future work.

We recommend that all readers read chapter 1 for an introduction to Mobile IP and for a definition of generally used terms and chapter 2 for an overview of this thesis and an explicit reading guide.

In this thesis references are specified using the Harvard method. Thus a reference presents the name of the author and the year of publishing, as in [Johnson *et al.*, 2000]. A full description of references are given in the bibliography. Throughout the thesis, we have used the convention of **bold facing** a term, when defining it explicitly. Terms are generally defined in the context they are first used.

We would like to thank our supervisor Brian Nielsen, for encouragement throughout the work, for providing feedback through discussions and for reading and commenting on earlier versions of this thesis. Thanks to Karen E. Nielsen from Ericsson Telebit in Aarhus, for valuable assistance and feedback regarding mathematics.

Aalborg, 7th June 2001

Torben Wittrup Andersen

Anders Lildballe

Contents

Chapter 1

Introduction

In a press release from Ericsson on November 7th 2000, Jan Lindgren, Vice President and General Manager for Ericsson Mobile Internet Solutions, was quoted for saying [Ericsson Press]:

> "Ericsson believes that within three years, [the number of] mobile Internet users will exceed [the number of] fixed Internet users... Automobile travelers will play a major role in this development."

This is just one out of many examples of the hype that mobile Internet access is causing at the moment.

Today the fastest growing businesses in the telecommunication market are mobile telephony and the Internet. Mobile telephony operators foresee an increasing share of their revenues coming from new data services, while Internet Service Providers (ISPs) are attracted to wireless technology in order to reduce costs and to provide omnipresent access to the Internet [Guardini *et al.*].

Predictions and forecasts about future development of the Internet and mobile telephony are many and often disagree. However, the growth in both fields have been explosive during several years and this positive development is expected to continue.

One of the hottest areas of growth in the computer industry is within portable computing devices [Kotz *et al.*, 1999]. Examples are laptops, Personal Digital Assistants (PDAs), mobile phones etc. Connecting these portable computing devices to the Internet is a natural step towards **mobile computing**, which means that we will be able to access the same services from portable computing devices as from home computers. Deployment of mobile Internet is still at an initial level, but commercial products are available. For instance 3Com offers wireless Internet access for their PDAs (Palm Pilots) at selected locations. In the very near future we will be able to access the Internet from a variety of mobile devices [Kotz *et al.*, 1999]. This means that we will always be able to read our email, browse the world wide web, etc. wherever we are located. But as wireless access to the Internet matures, the possibilities become even more appealing.

An application with very high potential is Voice over IP (VoIP). If the real time demands of this application can be met, mobile telephony operators are looking at better bandwidth utilization and easier network management. Eventually todays circuit switched networks will give way to the future of networking – packet switching [Seymour, 2000] [Flanagan, 1999].

One might ask, whether it is feasible to expect the Internet of the future to be built around the same core protocols which has constituted it, since its emergence as a research network in the early 1970's. Even though protocols has of cause been extended as seen by the introduction of a new version of IP, IPv6 [Huitema, 1997], the basic principles of the core protocols are unchanged.

We believe that the fact that the Internet has now existed for 30 years with an ever increasing popularity resulting in more users and more traffic being generated every day proves the soundness of those principles. Thus the introduction of mobile computing into the Internet must be done by improving existing protocols.

1.1 Initial problem

The core protocol of the Internet is the Internet Protocol (IP). The IP protocol was designed to interconnect stationary hosts in static networks. Each host is identified by an IP address. If a host moves to another network, IP does not allow the host to keep its former IP address. A host moving from one network to another, will be seen as two independent actions by IP: A host disappearing from one network and a host appearing at another network. Why is this a problem? Applications and protocols used in the Internet assume that a host keeps its IP address for its lifetime. Consequently, other hosts will not be able to contact a host if its IP address is changed.

Mobile IP is the leading proposal to solve this problem. Mobile IP provides mechanisms that let a host keep one IP address, even when it moves to other networks. This means that other hosts can reach a host even after it has moved to another network. A handoff is, in the context of Mobile IP, a host which moves from one network to another.

When a host performs a handoff, it must inform other hosts about its new location. This process can be time consuming. Thus for a period of time other hosts may send packets to the host's old location, while it is located at the new location. Packets sent in this period may be lost.

The consequence of lost packets is always degraded performance. The severity of this performance degradation is, however, dependent on the kind of application that uses the transfered data. During connection-less communication, e.g. VoIP, the data are permanently lost and the user will experience it as disturbances in or drop outs of the other person's voice. During connection-oriented communication, e.g. transferring files or browsing the web, transport protocols will reduce flow to avoid congestion and retransmit lost data. To the users it will seem as if they are using a slower connection.

To summarize; it will not be transparent to users whether they are using Mobile IP via wireless devices or ordinary IP via wired devices. Today Mobile IP suffers from higher rates of packet loss than ordinary IP. The scope of the work presented in this thesis is thus to provide mechanisms to reduce the number of lost packets during a handoff in Mobile IP. Especially we focus on proactive handoff initiation as a method to reduce handoff latency.

1.2 Structure of the Internet

IP is a protocol at the network layer in the hybrid reference model defined in [Tanenbaum, 1996] and depicted in figure 1.1.

5	Application layer
4	Transport layer
3	Network layer
2	Data Link layer
1	Physical layer

Figure 1.1: The hybrid reference model [Tanenbaum, 1996, p. 44].

The hybrid reference model is in fact the OSI Reference Model [Tanenbaum, 1996] in which the Session Layer and the Presentation Layer has been removed. This modification is made, as these layers are seldom referred to in practice. In this thesis we refer to the hybrid reference model and not the also commonly used TCP/IP reference model [Tanenbaum, 1996], as it is more precise in its specification.

In the Internet the transport layer protocol is either the connection-less User Datagram Protocol (UDP) or the connection-oriented Transmission Control Protocol (TCP). At the application layer typical examples are HyperText Transfer Protocol (HTTP) and File Transfer Protocol (FTP).

A **node** is the term for a computing device which has access to the Internet. All nodes must support the IP protocol. An **IP address** uniquely identifies a network interface of a node on the Internet. An IP address is a number which can be viewed as consisting of two parts: A **network prefix**, which identifies the network at which the node is located, and a **host ID** which identifies the node at the network. A **host** is a node that does not forward packets between networks.

The Internet is composed of tens of thousands of networks linked together by nodes with routing capabilities (routers). A **router** is thus a node that forwards packets between networks. All networks must connect to the Internet through a router. An example showing two networks with network prefixes X and Y and the routers connecting them to the Internet, is shown in figure 1.2.

Packets are forwarded by routers based only on the network prefix. In figure 1.2 a node A is depicted to be at the network with prefix X and a node B is at the network with prefix Y. If a packet is sent from A to B, its destination address is composed of prefix Y and the host ID for node B. The packet is forwarded by routers in the Internet by means of the network prefix. The host ID is not used, until the packet has arrived at the network with prefix Y.

1.3 Mobile nodes

A node which can move around and still connect to the Internet is called a **mobile node**. Thus in principle every desktop computer connected to the Internet is also a mobile node if it is moved

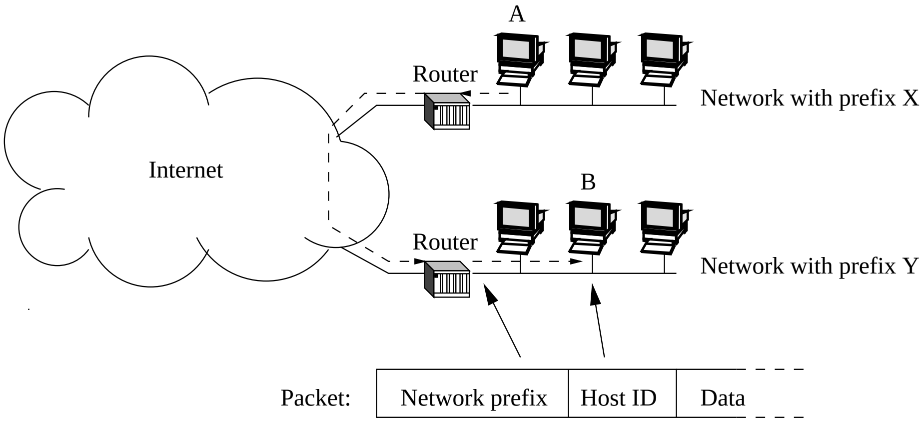


Figure 1.2: Connecting networks to the Internet.

and then reconnected to the Internet.

Usually, however, the term mobile node is used to refer to nodes, which are designed to move around and still preserve Internet access during the movement. Examples could be a laptop or a PDA. We adopt this last definition of a mobile node. Thus mobile nodes normally utilize wireless connections in order to access the Internet.

1.3.1 Attaching mobile nodes to the Internet

A node is connected to a network through an **access point**. An access point to which a mobile node can attach is often referred to as a **base station**. This is especially the case, whenever the access point provides a wireless connection. Examples of access points are hubs, switches and base stations. The access point at which a node is currently connected to the Internet is called its **point of attachment**. Accordingly, a router which connects a network to the Internet is called the **access router** of that network.

When a mobile node moves it changes its point of attachment accordingly, as it moves out of range of its current point of attachment and gets within range of new access points.

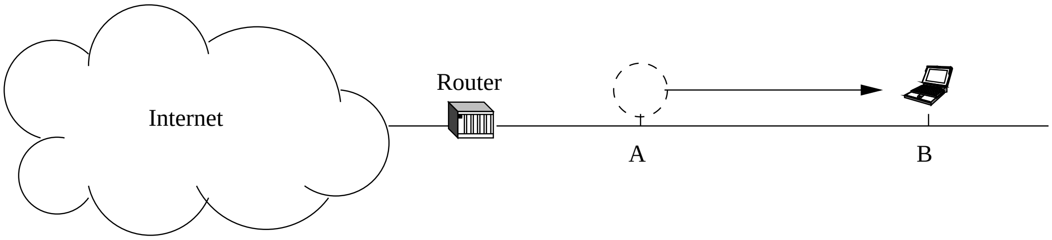


Figure 1.3: Changing point of attachment within a network.

The principle of changing the point of attachment within a network is illustrated in figure 1.3, where the point of attachment is changed from access point *A* to access point *B*. This operation can be performed without changing the mobile nodes IP address because the mobile node is still attached to the same network and does therefore not have to change either network prefix or host ID.

The movement could also be from a point of attachment at one network to a point of attachment at another network. This situation is illustrated in figure 1.4.

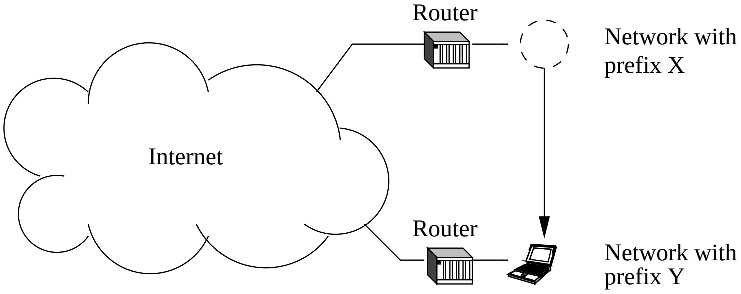


Figure 1.4: Changing point of attachment between networks.

In figure 1.4 a mobile node changes its point of attachment from the network identified by the prefix *X* to the network identified by the prefix *Y*.

A conflict regarding the IP address of the mobile node thus arises:

- If the IP address from the network with prefix *X* is kept, packets destined for the mobile node would not be routed to its new location at the network with prefix *Y*.
- If the IP address of the mobile node is changed to an IP address with prefix *Y*, then nodes wanting to communicate with the mobile node would not know the mobile node's IP address.

This conflict must be solved in the IP protocol, as for instance a TCP connection is established between two IP addresses and does not allow one of these IP addresses to be changed during operation. If the conflict was to be solved at the transport layer, all transport layer protocols would have to solve the conflict. By addressing the conflict at the network layer, it is made transparent to all upper layer protocols that a mobile node has changed its point of attachment from one network to another. This conflict is solved at the network layer by the introduction of Mobile IP.

1.3.2 Mobile IP

Mobile IP is an extension to the IP protocol. It provides a mechanism for allowing a mobile node to move between networks and still be able to both send and receive packets. This is done by allowing the mobile node to have several IP addresses.

The first IP address is the **home address**, which identifies the mobile node globally. It is static, meaning that it does not change when the mobile node moves. The network, at which this address is chosen, is referred to as the **home network** of the mobile node. All mobile nodes must be assigned a static home address.

When the mobile node is attached to a network other than its home network, referred to as a **foreign network**, it obtains an additional IP address known as a **care-of address**, which identifies the current location of the mobile node. The care-of address must always be an IP address valid at the current network. When the mobile node is attached to its home network, it is not assigned a care-of address.

At the home network, a router serving as a **home agent** must be present. The responsibility of the home agent is to forward data packets sent to the mobile node's home address to the mobile node's current care-of address.

Figure 1.5 depicts the basic principle of forwarding packets in Mobile IP, when a correspondent node sends packets to a mobile node. A **correspondent node** is the term used for a node which either sends packets to or receives packets from the mobile node.

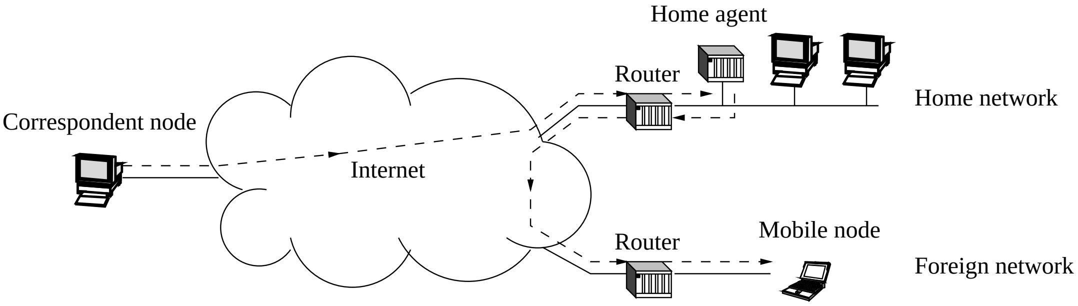


Figure 1.5: A mobile node receiving packets forwarded by its home agent.

In figure 1.5 a correspondent node sends a packet to the mobile node. The packet is destined to the mobile node's home address. Because the mobile node is attached to a foreign network, it has informed the home agent at its home network of its care-of address. The packet destined for the mobile node arrive at the mobile node's home network, where it is intercepted by the home agent, and forwarded to the mobile node's care-of address.

In order for the home agent to know the current point of attachment of the mobile node, the mobile node must notify the home agent, whenever it changes its point of attachment and thus changes the care-of address. This scenario is depicted in figure 1.6.

In figure 1.6 a mobile node has just changed its point of attachment to another network. It must then inform its home agent about its new care-of address.

1.3.3 Discovering new networks

For a mobile node to be able to attach to different foreign networks, means must be provided for the mobile node to discover those networks. In Mobile IP, networks are discovered through the reception of **router advertisements** at mobile nodes. Router advertisements are broadcasted from access routers at regular intervals.

Router advertisements contains the prefix of the advertised network and a lifetime denoting for how long this prefix can be considered valid by the mobile node. Upon each reception of a router advertisement from a certain network, the mobile node renews the lifetime for that particular network prefix.

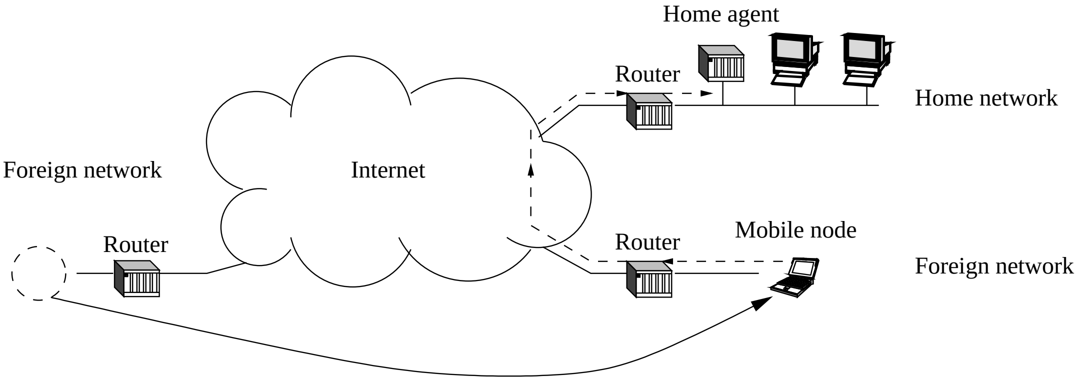


Figure 1.6: A mobile node updating its home agent with its new care-of address after moving to a new foreign network.

1.4 Handoff in Mobile IP

Besides providing several IP addresses for mobile nodes, an important aspect of Mobile IP is the support for changing the point of attachment from one network to another.

The process of changing access point from an access point at one network to an access point at another network for a mobile node is referred to as a **handoff**[1]. If a handoff is performed without any loss of data during the handoff, we refer to it as a **seamless handoff**[2].

It is the responsibility of a **handoff algorithm** to initiate and perform handoffs. We will view a handoff algorithm as consisting of three separate parts:

- Quality assessment of available networks.
- Handoff initiation.
- Handoff execution.

Quality assessment must be performed for every available network on a regularly basis, e.g. once every second. The purpose is to determine the value of the set of selected quality parameters for each network. A quality parameter for a network could be the signal strength with which data are received or the price of using the network.

When updated quality parameters exists for all networks, a handoff initiation entity must decide whether to perform a handoff to a new network. Handoff initiation and quality assessment are, although two separate parts, closely related in that the handoff initiation entity uses parameters of which the values have been determined by quality assessment. When a handoff is decided, it is the job of a handoff execution entity to perform the actual handoff. In this thesis our focus lies within quality assessment and handoff initiation. We will refer to an algorithm performing quality assessment and handoff initiation as a **handoff initiation algorithm**.

---

[1] Some literature uses the equivalent term **handover**.

[2] Some literature uses the equivalent terms **smooth handoff** or **smooth handover**.

1.4.1 Motivation for performing handoffs

That it is important to perform handoffs can be seen from the scenarios depicted in figure 1.7.

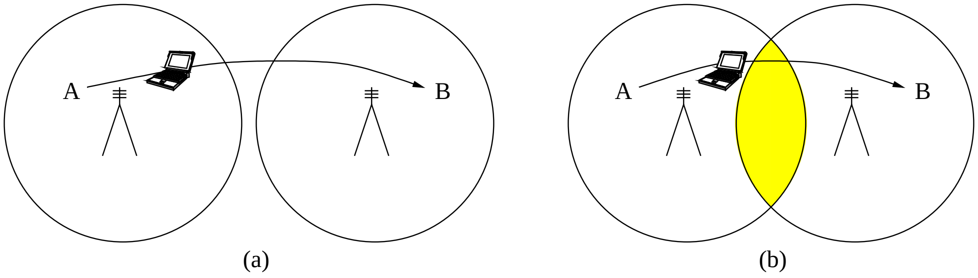


Figure 1.7: a) Seamless handoff not possible. b) Seamless handoff theoretically possible.

Figure 1.7a depicts two base stations at different networks and a mobile node. The circles illustrate the ranges of the base stations. The mobile node is moved from location *A* to location *B*. It is not possible to preserve Internet access during the entire move, because part of the path does not lie within the range of either of the base stations. In this scenario it is important to establish a new point of attachment as soon as the mobile node gets within the range of the base station covering location *B*.

In figure 1.7b a mobile node is again moved from location *A* to location *B*. In this scenario the entire path lies within the range of a base station. It is important to handoff between the networks attached to these base stations so that no interruption in the connection to the Internet is experienced by the mobile node.

It is the job of Mobile IP to provide mechanisms that perform handoffs. These handoffs must be characterized by **high performance**, meaning that there must be no data loss and that a minimal overlap between base station ranges is required. If no data are lost, applications are not disturbed. Minimal overlap between base stations results in potentially higher utilization of the range of base stations.

However, it is not always as obviously as in figure 1.7 whether it would be desirable to perform a handoff or not. Consider the scenario depicted in figure 1.8.

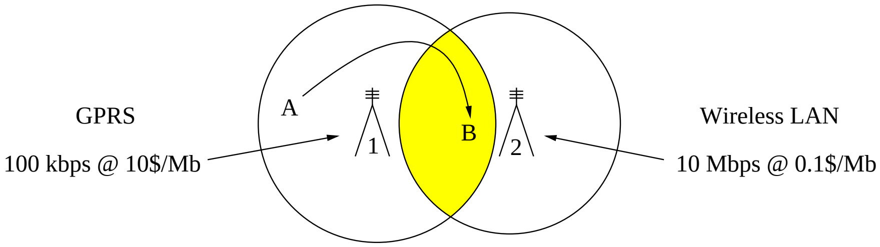


Figure 1.8: A situation where handoff could be performed.

In figure 1.8 a mobile node is moved from location *A* to location *B*. As both location *A* and location *B* is within the range of base station *1*, it is not necessary to perform a handoff in order to preserve attachment to the Internet.

However, assume that the attachment offered through base station *2* is both cheaper and offers better bandwidth than the attachment offered through base station *1*. Clearly it would then be desirable to perform a handoff to the network attached to base station *2*. This suggests that the reason for performing a handoff could be other than just preserving Internet access.

We have thus identified two basic motivations for performing a handoff:

- Handoffs must be performed in order to preserve attachment to the Internet.
- Handoffs must be performed to utilize the point of attachment providing the best performance and/or the lowest price.

Other parameters than just price and bandwidth can be considered. Different bandwidths and prices are often introduced, when base stations do not use the same link layer technology. If the mobile node is able to connect to both of these different link layer technologies a handoff must still be performed. An example is a mobile node performing a handoff from General Packet Radio Service (GPRS) to Wireless LAN.

1.4.2 Performance considerations for handoffs

In this section a handoff is examined in greater detail. A handoff consists of the following actions:

- Discover a new network.
- Obtain and validate a new care-of address.
- Obtain authorization to access the new network.
- Make a decision that a handoff should be initiated.
- Perform the handoff:
  - Notify the home agent of the new care-of address.
  - Receive acknowledgment from the home agent.

Thus the time it takes to perform a handoff is the sum of the time it takes to perform the above mentioned actions. The **handoff latency** is defined as the time from when a packet from a correspondent node or home agent was last received via the previous network until a packet from a correspondent node or home agent is first received via the new network. The handoff latency expresses a period of time in which packets may be lost or delayed.

A handoff initiation strategy must seek to minimize the handoff latency in order to improve performance. Two different basic handoff initiation strategies are initially identified. A strategy is **reactive** if it does not initiate a handoff until the current network becomes unavailable. A strategy is **proactive** if it initiates a handoff before the current network becomes unavailable.

1.4.3 Basic handoff initiation strategies

Recall that the decision whether or not to perform a handoff is decided by a **handoff initiation algorithm**. We examine how two simple handoff initiation algorithms will react to the scenario depicted in figure 1.9, where a mobile node is moved from the range of base station *1* into the range of base station *2*. Base station *1* and base station *2* offers access to two different networks, which we denote network *1* and network *2* respectively.

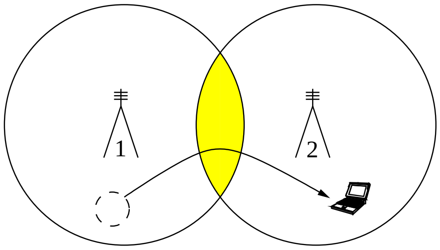


Figure 1.9: A simple handoff scenario.

In the specification of Mobile IP [Johnson *et al.*, 2000] [Perkins, 2000A], a simple reactive strategy seeking to minimize the number of handoffs is specified. This handoff initiation strategy is referred to as ***Lazy Cell Switching*** [Perkins, 1998]. A handoff is never initiated before the current point of attachment is declared unreachable. The reaction of this strategy to the scenario depicted in figure 1.9 is shown in figure 1.10.

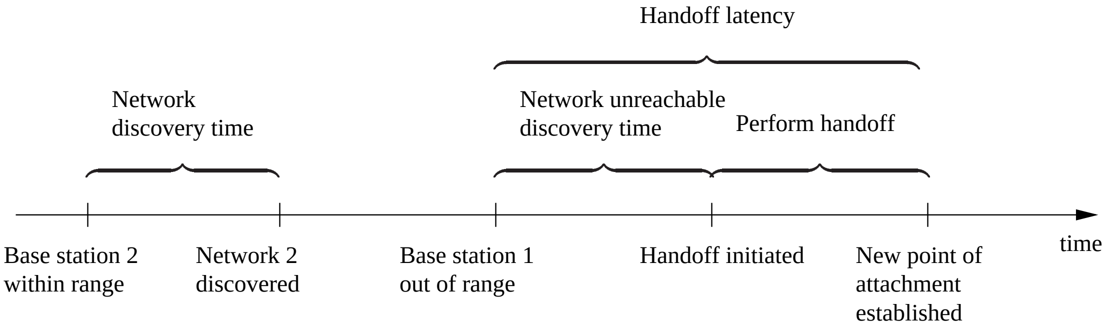


Figure 1.10: *Lazy Cell Switching* handoff initiation strategy.

As can be seen from figure 1.10, the handoff latency in the *Lazy Cell Switching* handoff initiation strategy depends on the time it takes to discover that base station *1* is unreachable plus the time it takes to perform the handoff. If base station *1* becomes unreachable before network *2* is discovered, this would further add to the handoff latency.

A simple proactive handoff initiation strategy, referred to as ***Eager Cell Switching***, is to change network as soon as a new network is discovered [Perkins, 1998]. The reaction of this strategy to the scenario depicted in figure 1.9 is shown in figure 1.11.

From figure 1.11 it can be seen, that the handoff latency for the *Eager Cell Switching* handoff initiation strategy only consists of the time from when base station *1* becomes unreachable until a new point of attachment is established. The mobile node is able to receive packets

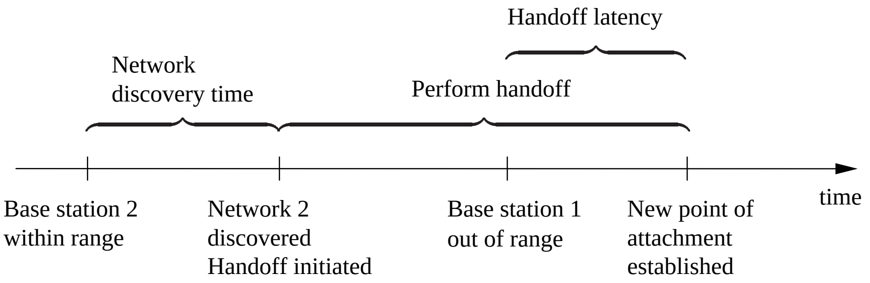


Figure 1.11: *Eager Cell Switching* handoff initiation strategy.

from both network *1* and network *2* until base station *1* becomes unreachable. If base station *1* becomes unreachable after the new point of attachment is established, the handoff latency is zero. Compared to the *Lazy Cell Switching* strategy the *Eager Cell Switching* strategy thus offers lower handoff latency.

This might lead to the conclusion, that the *Eager Cell Switching* strategy is always preferable to the *Lazy Cell Switching* strategy. However this is not the case, as can be seen from the scenario depicted in figure 1.12.

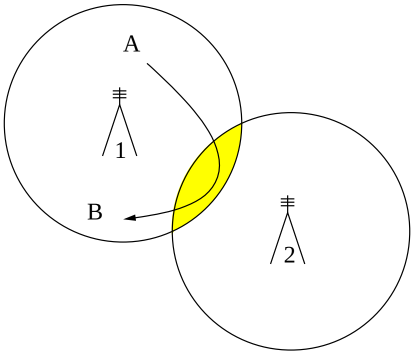


Figure 1.12: A scenario where a mobile node should not perform handoff.

In figure 1.12 a mobile node is moved from location *A* to location *B*. If *Lazy Cell Switching* is used, no handoff is performed. If *Eager Cell Switching* is used, first a handoff to network *2* is performed and later another handoff to network *1* must be performed when base station *2* becomes unreachable. In this particular scenario the *Lazy Cell Switching* handoff initiation strategy would be preferable.

Our observation is, that it is not at all obvious which of these handoff initiation strategies should be selected. The performance depends on the particular scenario in which the handoff initiation strategy is applied. In order to obtain better general performance a more advanced handoff initiation strategy is needed.

1.4.4 Loss of data during handoff

An important performance metric for a handoff strategy is the number of packets that are lost during a handoff. A **lost packet** is one that is sent from a correspondent node but never received at the mobile node, or vice versa.

Even though a mobile node has changed its point of attachment by obtaining a care-of address at a new network, the care-of address from the previous network can still be used for receiving packets. This of cause requires that the base station which provides access to the previous network is within range of the mobile node.

In principle a mobile node can have as many care-of addresses as the number of different networks within range. The care-of address currently advertised to the home agent is referred to as the **primary care-of address** [Johnson *et al.*, 2000]. Accordingly, the network to which the primary care-of address belongs is called the **primary network**.

Using *Lazy Cell Switching*, packets are not received for a period of time identified by the handoff latency. This is because, the primary network is no longer reachable.

Using *Eager Cell Switching* packets can still be received from the primary network during the handoff. If the mobile node stays within reach of the primary network until the handoff is completed, no packets are lost. A proactive handoff initiation strategy can thus dramatically reduce the number of lost packets.

A proactive handoff initiation strategy can cause packets to arrive out of order[3]. This can happen, if the route from a node to the new network is faster than the route from this node to the previous network. Packets sent from a node to the mobile node at its new primary care-of address would then arrive before the packets routed via the previous network. Some applications will react to packets arriving out of order by simply dropping the packets that arrived too late. An example is Voice over IP (VoIP). Another example is TCP, which is optimized for packets being delivered in the correct order. Packets arriving out of order will thus degrade the performance of TCP. The effect of packets arriving out of order depends on the application which the packets are addressed to.

1.5 Security considerations in Mobile IP

Even in ordinary non mobile IP networks the security threats are almost countless. In an uncontrolled global network like the Internet one must assume that packets can always be intercepted or modified by someone with bad intentions. Likewise a network can be flooded with packets only with the intention of disturbing traffic.

As defined in [Atkinson, 1995], security involves four related topics:

- Confidentiality. Data must be encoded so they can only be decoded by authorized parties.
- Authentication. It must be possible to prove or disprove someone's claimed identity.

---

[3] The possibility of packets arriving out of order is always present even in standard IP. This can happen if a route is changed somewhere in the Internet during transmission of packets. However this situation rarely occurs in practice in standard IP.

- Integrity. It must be ensured that data cannot be modified without such modification being detectable.
- Non-repudiation. The property of a receiver being able to prove that a sender of some data did in fact send the data that might later be denied being sent.

Various protocols and mechanisms have been designed to deal with the above mentioned topics. In the following we will only consider the security flaws introduced especially by Mobile IP.

Research within security regarding Mobile IP is coordinated by the IETF working group for Authentication, Authorization and Accounting (AAA). Their responsibility is to coordinate the development of protocols to be used by Mobile IP in the following areas:

- Authentication. A mobile node must be able to identify itself to a network to which it wants to attach. The network must be able to verify the mobile nodes claimed identity.
- Authorization. A network must be able to determine, whether an authenticated mobile node should be granted access to the network.
- Accounting. A mobile node having been authorized to use a network must be billed by the network service provider.

The above mentioned areas are often referred to as AAA. Our motivation for considering the AAA aspects of Mobile IP are primarily the time constraints which AAA will impose on handoff. Performing authentication and authorization carry the potential to be lengthy processes and can thus influence the handoff latency dramatically.

It is generally agreed, that networks allowing mobile nodes to attach must provide the AAA services. This should be the responsibility of AAA servers located at these network [Perkins, 2000B]. However, the aspects of AAA are still undergoing work and a standard is not yet agreed upon. Here we will consider AAA for Mobile IPv6 as proposed by Charles Perkins in [Perkins, 2000B]. As depicted in figure 1.13, there must be a AAA server at both the home network and the foreign network to which the mobile node is trying to attach.

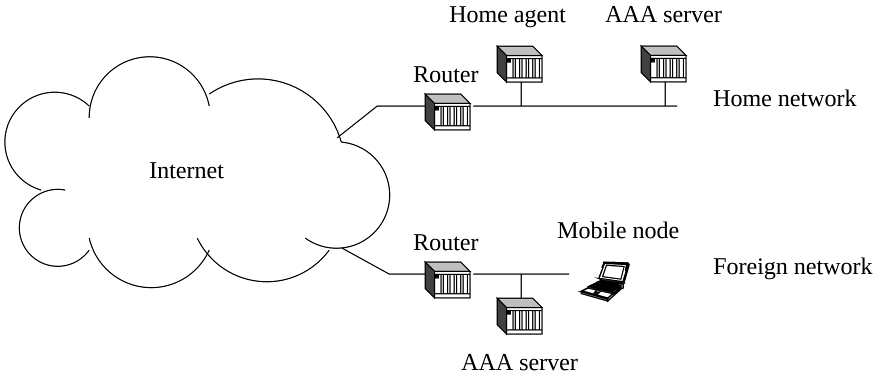


Figure 1.13: Networks with AAA servers.

The basic principle of AAA is that traffic from an unauthenticated host cannot travel beyond the access router of a network. A mobile node wanting to attach to a specific network should therefore initiate a series of events which if successful leads to authorization at that particular network. In the following it is assumed that all communication between nodes are secure.

- The mobile node identifies itself to the AAA server at the foreign network. This is done by the transmission of a so called **client identifier**. A client identifier should contain means of securely identifying a node and means for the AAA server at the foreign network to identify which AAA server it must contact to verify the credentials. This information is denoted the credentials of the mobile node.
- The AAA server at the foreign network must verify that the client identifier received from the mobile node is valid. This is done by transmitting the credentials to the AAA server at the mobile nodes home network. By use of an encryption key only known to this AAA server and the mobile node, the AAA server at the home network can determine whether the mobile node has the claimed identity. The answer to this question is returned to the AAA server at the foreign network.
- The AAA server at the foreign network can now decide whether to authorize the mobile node to attach to the network or not.

In order for this procedure to be secure, means must be provided for the AAA servers to verify each others identify. The need for AAA adds a potentially large overhead to the handoff latency in Mobile IP. There are numerous suggestions for how to deal with this overhead. If a mobile node could be authorized by a AAA server serving several networks the mobile node need not perform AAA actions when moving between these networks. Likewise, mobile nodes might be allowed to gain unauthorized access to a network for a limited amount of time [Tsirtsis, 2001], thus allowing the AAA procedures to be deferred to a later stage, e.g. until right after the handoff procedure has been completed. AAA servers might also consider caching the credentials of mobile nodes, thus allowing for faster verification of a mobile node's identity if it revisits a network.

We believe, that the potential overhead of AAA is just another incitement to explore proactive handoff initiation. The added overhead might be compensated for by initiating handoffs so that packets can still be received from the previous network while authorization is ongoing at the new network. Such a strategy can also avoid packets being lost, because the mobile node can receive packets from both networks during the handoff.

1.6 Standardization of Mobile IP

Mobile IP is a concept which is receiving heavy attention. Much effort is currently put into improving the performance of the handoff execution algorithm in Mobile IP.

Specifications exists for both IPv4 [Perkins, 1996] and IPv6 [Johnson *et al.*, 2000]. Both specifications are based on the same principles and maintained by the same people. They are currently at different maturity levels. The specifications are published according to the directions described in the RFC *The Internet Standards Process – Revision 3* [Bradner, 1996]. The standardization of protocols for the Internet is managed by the Internet Engineering Task Force (IETF) which is an organization under the Internet Society (ISOC).

The levels of maturity of a standard are the following [Bradner, 1996]:

Internet-Drafts: Used for making draft specifications public available. May be replaced at any time by a newer version. Internet-Drafts are removed if they are not updated for six months or if they are promoted to a RFC.

Request For Comments (RFCs): When a specification is accepted as an RFC it cannot be altered. RFCs are never deleted but can be obsoleted by other RFCs.

Proposed Standard: When a RFC is believed to be well specified and has attracted a certain amount of interest it might be advanced to a proposed standard. Neither implementation or operational experience is required.

Draft Standard: A specification from which at least two independent and inter-operable implementations from different code bases have been developed, and for which sufficient successful operational experience has been obtained, may be elevated to the Draft Standard level.

Internet Standard: A specification for which significant implementation and successful operational experience has been obtained may be elevated to the Internet Standard level.

Mobile IPv4 is currently a RFC at the Proposed Standard level [Perkins, 1996]. Because changes has been made to the Mobile IPv4 proposal since this RFC was released, the proposal is no longer up to date. The newest version of Mobile IPv4 is thus available as an Internet-Draft [Perkins, 2000A].

Mobile IPv6 is currently specified as an Internet-Draft [Johnson *et al.*, 2000] in its 13th version.

The standardization status of Mobile IP clearly suggests, that more work is needed before a standard of Mobile IP is agreed upon. However much active work is ongoing. The work is supervised by a working group under IETF. It is definitely expected, that higher levels of standardization will be reached in the future. This is a natural development given the huge interest in Mobile IP.

Commercial products supporting Mobile IP are starting to appear on the market. An example is that Cisco, the worlds leading manufacturer of routers, has started to support the RFCs specifying Mobile IPv4 in a wide range of their router products. Ericsson Telebit supports part of the Mobile IPv6 specification [Johnson *et al.*, 2000] in some of their router products.

1.7 Related work

A wide range of specifications aiming at optimizing the performance of basic Mobile IP are currently under development. Many of these specifications are published as Internet-Drafts.

A common denominator for the specifications, is that they are all based on the basic Mobile IP specification for either IPv4 [Perkins, 2000A] or IPv6 [Johnson *et al.*, 2000]. This means that they do not question the feasibility of the overall Mobile IP strategy, but addresses some subset of the many different aspects that must be handled by Mobile IP.

The aspects of Mobile IP that are the most commonly dealt with in the published Internet-Drafts, fall within three categories:

1. General improvements to reduce the handoff latency. The approach taken is mostly to improve the handoff *execution* procedure. Only little work concerns handoff *initiation* in Mobile IP.
2. Improvements to reduce the handoff latency using specific link layer technologies.
3. Suggestions regarding Authentication, Authorization and Accounting (AAA).

The scope of this report is mainly within 1). Therefore we briefly presents the basic ideas of some of these proposals in section 1.7.1.

The aspects of 2) are relevant seen from the view of deploying Mobile IP but does not provide general improvements as results often cannot be generalized.

The Mobile IP specifications are not aimed at any particular link layer technology. This is a great advantage when considering that mobile nodes might be able to attach to access points based on different link layer technologies. At the same time, the Mobile IP specification will be applicable with new link layer technologies.

Often using link layer specific information can provide better performance. This is true if for a wireless link one have access to the signal strength which can indicate that a base station soon will get out of range. This is recognized in both the Mobile IPv4 and the Mobile IPv6 specifications by allowing implementations to use link layer specific information if available. However an implementation must of cause still comply with the specification in the case, that no link layer information is available.

The aspects of 3) are highly relevant indeed, but lies outside the scope of this thesis.

1.7.1 Proposed improvements to handoff execution in Mobile IP

The current proposals for minimizing the handoff latency lies within two groups:

- Hierarchical mobility agent models. These proposals are aimed at reducing the amount of signaling between the mobile node and its home agent, by grouping several networks into administrative domains inside which handoff can be handled locally. Hierarchical proposals are developed by, among others, people from the IETF working group [Soliman *et al.*, 2000] and Lucent Bell Labs [La Porta *et al.*, 2000].
- Multiple care-of addresses. These proposals uses routing to multiple care-of addresses in order to reduce packet loss [Yegin *et al.*, 2000] [Rose *et al.*, 2000].

Examples of proposals that lies with the two groups are presented in the following two sections. All of these proposals are extensions to the basic Mobile IP specifications for either IPv4 or IPv6.

1.7.2 Hierarchical mobility agent models

The general principle of the hierarchical models is presented in figure 1.14.

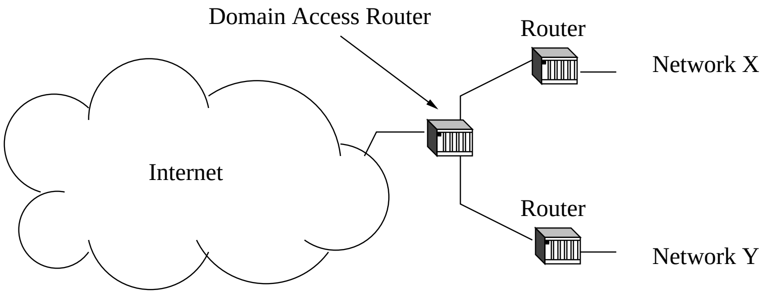


Figure 1.14: General principle of hierarchical mobility models.

A number of networks are grouped into an **administrative domain**. In figure 1.14 a simple example grouping only two networks is shown. An administrative domain is connected to the Internet through a **domain access router**, which is a gateway into the Internet for the administrative domain. If a mobile node changes network within the administrative domain, it must only notify the domain access router about its new care-of address. It is then the responsibility of the domain access router to route data packets destined for the mobile node to the right network.

The advantage of these models is, that they reduce the signaling load between mobile node and home agent and that the care-of addresses only needs to be registered with the domain access router (intra domain handoff).

The disadvantages are that special software has to be deployed at the routers within the administrative domain, and that movement between administrative domains does not benefit from hierarchical models.

In order to have any effect an administrative domain needs to span several networks possibly covering large geographical areas. Furthermore these networks should share a common gateway (the domain access router) to the Internet. It remains to be seen whether these demands can be met in realistic scenarios.

1.7.3 Multiple care-of addresses

A proposal within the group of multiple care-of addresses is *neighborhood routing* [Yegin *et al.*, 2000]. In this proposal the mobile node is allowed to update the home agent with multiple care-of addresses. The mobile node can then be reached, even though its exact location is not known. This is achieved by introducing a modified *routing header*, which contains the care-of addresses known to the home agent. If a mobile node is not reachable at the first care-of address, the packet is routed to the next care-of address etc.

Other proposals exploits what is known as **bicasting** or **IP diversity** [Rose *et al.*, 2000]. This means sending identical packet to different networks. This is often coupled with a hierarchical model so that the overhead generated by the extra packets only occurs within an administrative domain.

As for the hierarchical models no documentation of the performance to be expected exists. These approaches for reducing the handoff latency are still at an initial level.

Chapter 2

The thesis

The approach taken in Mobile IPv6 [Johnson *et al.*, 2000] is the leading proposal for adding support for mobility in the Internet Protocol (IP), and is likely to be the base on which IPv6 will support mobility in the future.

In this work we have decided to focus exclusively on Mobile IPv6 thus leaving Mobile IPv4 unattended. The reason for this choice is that Mobile IPv6 presents a more complete and elegant solution to mobility than Mobile IPv4, because IPv6 has built in support for many of the features needed to provide mobility. In IPv4 these features must be provided by a mobility extension. Furthermore IPv6 is expected to eventually substitute IPv4 in the future Internet. Our choice to focus on IPv6 will not affect the generality of obtained results. On the contrary we regard results obtained with Mobile IPv6 as generic enough to be applicable for Mobile IPv4 also, because of the similarities of the basic concepts of Mobile IPv4 and Mobile IPv6. For an introduction to the specifics of Mobile IPv6 we refer to appendix A

2.1 Means of improving Mobile IP performance

As stated in the introduction in chapter 1 our initial objective was to investigate how to reduce the handoff latency introduced by the Mobile IP protocol.

By studying the Mobile IPv6 specification [Johnson *et al.*, 2000] it has become apparent that mechanisms for performing seamless (and low latency) handoffs are very weakly specified. This is a recognized fact among the members of the IETF Mobile IP Working Group and numerous suggestions for improving handoff performance are under consideration.

By conducting a literature study we have revealed that much work is ongoing within the field of reducing the time consumed by *handoff execution*. However we have been surprised to learn, that in the field of *handoff initiation* we are able to find no ongoing work whatsoever, even though as illustrated in chapter 1, a successful handoff initiation strategy can result in zero packet loss. The work presented in this thesis regards handoff initiation strategies and their effect on the handoff latency.

In appendix A we have included a summary of the major protocols constituting Mobile IPv6. This summary shows that in order to deploy Mobile IPv6 in existing IPv6 networks, the main implementation effort needed is at the mobile node. This indicates that supporting Mobile IPv6 is likely to be fairly straightforward for router vendors and system administrators. To support that Mobile IPv6 should be as easy as possible to integrate into existing IPv6 networks, we pursue the idea that handoffs should be initiated by mobile nodes without requiring the use of services from special entities at networks.

2.2 Assessing existing handoff initiation strategies

The starting point of our work have been that of two existing handoff initiation strategies: The proactive *Eager Cell Switching* handoff initiation strategy and the reactive *Lazy Cell Switching* handoff initiation strategy, both described in section 1.4.3. These are the basic handoff initiation strategies against which more advanced handoff initiation strategies must be evaluated.

Although simple in concept, the performance of these two handoff initiation strategies have never been thoroughly investigated. The effect of various protocol parameters on handoff performance are not well understood and for instance the currently suggested default configuration of access routers [Johnson *et al.*, 2000] appears to be a mere ad hoc suggestion.

For each of the two simple handoff initiation strategies we have therefore conducted a theoretical study of the effect of these protocol parameters on handoff performance. As a result we derive two mathematical models (one for each of the basic handoff initiation strategies) reflecting handoff performance as a function of the configuration of protocol parameters.

To relate our theory to practice we have setup a testbed running Mobile IPv6. This is done as we believe that any proposal of protocol optimization for Mobile IP should be supported by empirical evidence. As for all network protocols it can be difficult to predict certain behavior, a fact which is emphasized by the many *best practice* configurations of protocols used in the Internet. Often a certain configuration is based only on the fact, that it has proven successful in actual operation. Using this testbed we conduct extensive empirical studies of the handoff performance of the two simple handoff initiation strategies. We then compare the theoretically predicted handoff performance with that experienced in the testbed to see whether the predictions of the mathematical models conform to the empirically obtained results.

2.3 Proposal of advanced proactive handoff initiation strategy

We believe that a need for more advanced handoff initiation strategies exists. Through our study of *Eager Cell Switching* and *Lazy Cell Switching* we conclude that they both have serious performance lacks. The *Eager Cell Switching* handoff initiation algorithm initiates far too many handoff and the *Lazy Cell Switching* handoff initiation algorithm results in an unacceptable number of lost packets.

However, this is not the only reason why more advanced handoff initiation strategies need to be considered. Another issue is that a handoff initiation strategy should also consider metrics such as e.g. the price of using a network and the available bandwidth.

To minimize the number of handoffs and to reduce the number of lost packets we investigate how information obtained from the link layer can assist. Furthermore we propose a set of simple metrics which could improve the performance experienced by the end user. These metrics include the possibility of statically defining the price of using a network and dynamically assessing indications of the available bandwidth. We propose the *Parametric Cell Switching* concept which allows a handoff initiation algorithm to base its handoff decision on several handoff metrics. To demonstrate the proposed concept we have implemented a prototype of *Parametric Cell Switching* using two selected handoff metrics. This prototype demonstrates improved performance to both *Eager Cell Switching* and *Lazy Cell Switching*.

2.4 Contributions

The following is a list of our main contributions:

- For both the *Eager Cell Switching* and the *Lazy Cell Switching* algorithms we propose a mathematical model describing handoff performance as a function of the configuration of essential protocol parameters.

  Using these mathematical models we propose a new default router configuration resulting in reduced handoff latency for both *Eager Cell Switching* and *Lazy Cell Switching* but without increasing network load.

- We bridge the gap between theory and practice by performing an extensive empirical study of handoff performance in a Mobile IPv6 testbed. Using this testbed we are able to conclude that the measured handoff latencies essentially conform with those predicted by the theoretical models and that our proposed router configuration does in fact yield the expected reduction in handoff latency.

- We propose an advanced handoff initiation strategy, which we denote *Parametric Cell Switching*, that uses link layer information and is able to consider metrics such as price and bandwidth. We expect *Parametric Cell Switching* to exhibit the responsiveness of *Eager Cell Switching* but to perform fewer and more informed handoffs.

  Through the development of a mathematical model we show that theoretically this advanced handoff initiation strategy is able to react almost as fast as *Eager Cell Switching* in scenarios where link layer information is not available.

- By means of a working prototype of the *Parametric Cell Switching* handoff initiation strategy applied in the Mobile IPv6 testbed we confirmed the theoretical model. By applying *Parametric Cell Switching* in a building wide scenario with three wireless networks we obtained results indicating the feasibility of using link layer information to assist in a handoff decision.

2.5 Structure of the master thesis

The remainder of this master thesis is structured as follows.

In chapter 3 we derive mathematical models for the *Eager Cell Switching* and the *Lazy Cell Switching* handoff initiation strategies. Based on these models we propose a new default configuration of protocol parameters resulting in lower handoff latency without increasing network load. In chapter 4 we present the Mobile IPv6 testbed and the results of the empirical studies of *Eager Cell Switching* and *Lazy Cell Switching*. We compare these results to the results predicted by the mathematical models.

In chapter 5 we propose the *Parametric Cell Switching* handoff initiation strategy using multiple metrics. For an instantiation of the *Parametric Cell Switching* handoff initiation strategy we theoretically determine its performance when link layer information is not available. Then, in chapter 6 we document the integration of *Parametric Cell Switching* into the KAME Mobile IPv6 software. We also present the results of two experiments conducted in the Mobile IPv6 testbed confirming that *Parametric Cell Switching* performs as predicted. We describe how we have deployed three wireless networks at the Department of Computer Science, Aalborg University, using Wireless LAN as the link media. We present informal results obtained with these networks indicating the feasibility of *Parametric Cell Switching* and that it has superior performance to both *Eager Cell Switching* and *Lazy Cell Switching*. Finally in chapter 7 we summarize our conclusions and presents ideas for future work.

We have included several appendices. Readers not familiar with Mobile IP and Mobile IPv6 in particular, are encouraged to read appendix A in which we present some of the basics of Mobile IPv6 operation. We suggest that this appendix is read **before** proceeding to chapter 3.

Readers interested in installing FreeBSD and the KAME Mobile IPv6 software should also read appendix B through E. Appendix B describes how we installed FreeBSD 4.1 on the computers constituting the testbed presented in chapter 4. Appendix C describes the configuration of the KAME Mobile IPv6 software used in the testbed. Appendix D summarizes a number of bugs found in the KAME software. Finally appendix E describes modifications that were made to FreeBSD tools supplied with KAME, in order to be able to run the Mobile IPv6 software.

Chapter 3

Theoretical study of existing handoff initiation strategies

In this chapter we first present two mathematical models able to predict handoff latency. The first model presented concerns the *Eager Cell Switching* handoff initiation strategy and the second model presented concerns the *Lazy Cell Switching* handoff initiation strategy.

Based on results obtained using the mathematical models we then propose a new default configuration for access routers resulting in a significant reduction of handoff latency without increasing network load. Finally we describe the implementation of two independent simulators originating from different code bases, which we use to stress the correctness of the mathematical models.

3.1 Motivation

As described in section 1.4.3 two simple handoff initiation strategies for Mobile IP exists. These are the *Lazy Cell Switching* handoff initiation strategy, which does not initiate a handoff until the current point of attachment is confirmed to be unreachable, and the *Eager Cell Switching* handoff initiation strategy which initiates a handoff immediately upon learning a new network prefix.

In this chapter we present mathematical models to predict the handoff latency as a function of the interval between broadcasting router advertisements from access routers and of the lifetime of the network prefixes specified in these broadcasted router advertisements. We present a model for both *Eager Cell Switching* and *Lazy Cell Switching*.

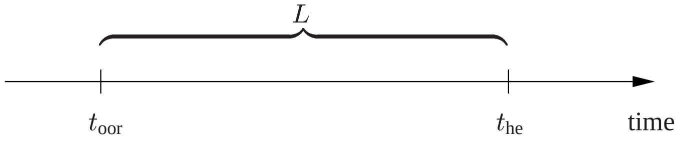
This task is undertaken in order to compare theoretically predicted handoff latency with results later to be obtained in experiments. If an accurate model can be devised and one is able to predict the handoff latency, this could be used for configuring the interval between sending router advertisements and the lifetime specified in the sent router advertisements at access routers. This configuration is important, as sending too many router advertisements increases network load and requires heavy processing at mobile nodes while sending too few router advertisements results in higher handoff latencies. A tradeoff between network load and handoff latency must therefore be found.

3.2 Assumptions and basic definitions

In the following we will assume that no packets are lost to simplify reasoning, i.e. that at any position all mobile nodes can receive all or none of the broadcasted router advertisements from any access router. When well within range of a base station it is a reasonable assumption that all broadcasted packets are received, as many wireless link layer technologies both retransmits lost packets and have very low bit error rates. When approaching the maximum range of a base station it might be that e.g. only some of the router advertisements sent via that base station are received by the mobile node. However, here we will assume that a mobile node is either inside or outside the range of any base station and that no such *grey zones* exist. We will further assume, that getting out of range of the primary network coincides with getting within range of a base station at a new network.

Recall that the **handoff latency** is defined as the time from when a packet from a correspondent node or home agent is last received via the primary network until a packet from a correspondent node or home agent is received via the new network. The handoff latency expresses a time in which packets may be lost.

By L we will denote the period of time from getting out of range of the base station serving the primary network at time t_{oor} (out of reach) until the occurrence of an event which triggers a handoff to a new network at time t_{he} (handoff event). What kind of event that can trigger handoff initiation depends on the handoff initiation strategy used. The principle is illustrated in figure 3.1.



t_{oor}: Out of range of primary network

t_{he}: Event that triggers handoff initiation

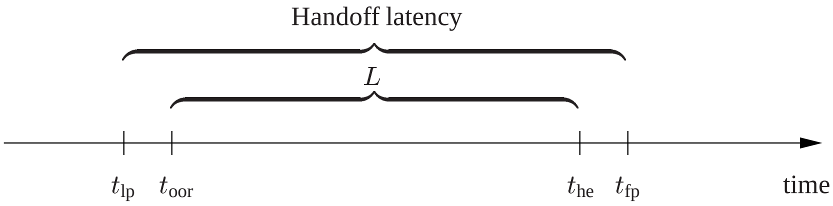
Figure 3.1: Definition of the period L.

The period L is affected by t_{he} which again depends on the handoff initiation strategy used. For the *Lazy Cell Switching* handoff initiation strategy, t_{he} will be the time when the primary network is concluded to be unreachable. For the *Eager Cell Switching* handoff initiation strategy, t_{he} will be the time at which a router advertisement is first received from the new network.

The period L relates to the handoff latency in the following way: L denotes a period of time in which packets are always lost. If a handoff is executed while both the primary network and the new network is reachable meaning that the primary network does not get out of range until after the handoff is completed, the value of L will be negative (to be interpreted as 0) yielding no loss of packets. In contrast, the handoff latency depends on the rate at which a mobile node is receiving packets.

As opposed to the handoff latency, L cannot be measured in real scenarios (how should one determine when the primary network is out of reach). On the other hand, deriving a mathematical model for calculating L is more appropriate as we cannot know at which rate packets will be send from correspondent nodes to a mobile node.

When the frequency of sending packets towards a mobile node is increased, the handoff latency converges towards the value of L. This is because the time of receiving the last packet from the primary network will be closer to the time when the mobile node gets out of range of that network and the time of receiving the first packet from a correspondent node via the new network will be shorter after the occurrence of the event that triggered the handoff to that new network. This relation between L and the handoff latency is illustrated in figure 3.2.



t_{lp}: Last packet received from primary network

t_{oor}: Out of range of primary network

t_{he}: Event that triggers handoff initiation

t_{fp}: First packet received from correspondent node via new network

Figure 3.2: Relation between L and the handoff latency.

In chapter 4 we use this property to compare the theoretical values for L with values for handoff latency measured in a real scenario, thus obtaining knowledge of the conformance between theory and practice. We will use L and theoretical handoff latency as interchangeable terms.

The time between broadcasting unsolicited router advertisements is configured by specifying a minimum time (denoted R_{min}) and a maximum time (denoted R_{max}) that a router must wait before broadcasting the next unsolicited router advertisement. After each broadcast of a router advertisement a router must pick a random delay between R_{min} and R_{max} before broadcasting the next router advertisement [Deering, 1991], in order to avoid routers synchronizing.

The lifetime of network prefixes specified in all broadcasted router advertisements on a link are identical and is denoted by T_{l}. For *Lazy Cell Switching* the lifetime of the network prefix broadcasted in the router advertisements affects the value of L, as it is used as an indicator of when the network should be considered unreachable. For *Eager Cell Switching* the lifetime of advertised network prefixes does not directly affect the handoff latency.

However, consider the situation, in which a mobile node at regular intervals is receiving router advertisement from two access routers advertising two different network prefixes. Clearly not even the *Eager Cell Switching* handoff initiation strategy should perform a handoff upon every reception of a router advertisement in this situation, as this would lead to the mobile node continuously performing unnecessary handoffs[1]. It must therefore be determined exactly when the reception of a network prefix should be taken as a sign of the entering of a new network. When using Mobile IPv6, a solution is to check whether the prefix is present in the *prefix list* maintained by IPv6 neighbor discovery. If not, the network prefix is considered to belong to a new network and a handoff is performed. A network prefix is present in the *prefix list* until

---

[1] Often referred to as the ping-pong effect.

its lifetime expires, why the lifetime of broadcasted network prefixes can indirectly affect the behavior of *Eager Cell Switching*. In our mathematical model for the *Eager Cell Switching* handoff initiation strategy we choose to ignore this possible effect of the lifetime of network prefixes. This is a valid choice, as it does not affect the handoff latency, when a mobile node enters a new network.

The time that a mobile node enters the range of a new network is denoted by C_{time}. After the entering of the range of a new network, a mobile node is able to receive all further packets broadcasted at that network until leaving the range of the network.

In the theoretical models we do not consider propagation delays of *binding updates*, *binding acknowledgments* etc., as their impact on the handoff latency depends entirely on the scenario in which Mobile IPv6 is operated. By excluding these propagation delays from the theoretical models, the theoretical models only reflects scenarios in which propagation delays are not significant. If one wishes to account for propagation delays in a specific scenario, this delay must be estimated and added to the theoretical predicted handoff latency.

3.3 Eager Cell Switching handoff initiation strategy

We start by deriving the mathematical model for *Eager Cell Switching* as this is the simpler model and will serve as a basis for the mathematical model for *Lazy Cell Switching* which we derive in section 3.4.

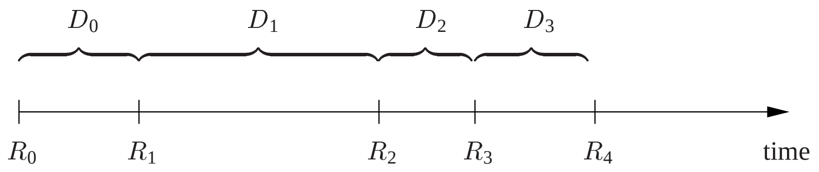
Recall that *Eager Cell Switching* is influenced by the interval between receiving unsolicited router advertisement and not by the lifetime of network prefixes. The mathematical model for this handoff initiation strategy should thus take as input the possible range of intervals between broadcasting router advertisements at access routers, $[R_{\text{min}}, R_{\text{max}}]$. As output we seek a function describing the probability distribution for the handoff latency. This function can then be used to determine theoretical values for the minimum, maximum and average handoff latency.

First we derive the probability distribution for L_{eager} which we denote $P_{L_{\text{eager}}}(l)$. In general, we use the letter P for probability distributions, $P_{L_{\text{eager}}}$ to reflect that it is a probability distribution for L using *Eager Cell Switching* and $P_{L_{\text{eager}}}(l)$ when the input variable is denoted by l. General properties of functions for probability distributions are, that they must not output any negative values and that the area under the probability curve must be exactly 1. Based on $P_{L_{\text{eager}}}(l)$ we can derive a function for the average value of L_{eager} denoted by $\overline{L}_{\text{eager}}$.

Determining L_{eager}

The abstraction considered is a router broadcasting unsolicited router advertisements, with the time between each broadcast uniformly distributed in the interval $[R_{\text{min}}, R_{\text{max}}]$. Mathematically, this is called a **random walk**. In figure 3.3 a possible distribution of router advertisements in a random walk is presented.

In the scenario depicted in figure 3.3, the size of D_{n} is uniformly distributed in the interval $[R_{\text{min}}, R_{\text{max}}]$. Recall from section 3.2, that the time a mobile node enters a link is denoted by C_{time}. Because the D's are independent and uniformly distributed, the distribution of C_{time} within an interval is independent of the interval in which it occurs. As a result it is adequate

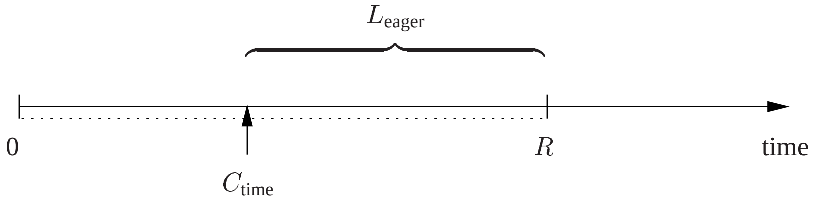


R_{n}: Unsolicited router advertisement number n

$D_{\text{n}} = R_{\text{n+1}} - R_{\text{n}}$: Distance between two consecutive router advertisements

Figure 3.3: Example of a random walk for a router broadcasting unsolicited router advertisements. The time between each broadcast is uniformly distributed in the interval $[R_{\text{min}}, R_{\text{max}}]$.

to consider only the interval in which C_{time} occurred. To simplify notation, the first router advertisement arriving after C_{time} is denoted R and the router advertisement sent prior to R is simply denoted by 0. The situation is depicted in figure 3.4.



0: Router advertisement sent prior to router advertisement R

R: First router advertisement received after C_{time}

$C_{\text{time}} \in [0, R]$

$L_{\text{eager}} = R - C_{\text{time}}$

Figure 3.4: Model for computing L_{eager} for the *Eager Cell Switching* handoff initiation algorithm.

The model in figure 3.4 reflects, that C_{time} must occur somewhere in the interval $[0, R]$ and is thus dependent of the value of R. From figure 3.4 we have that $L_{\text{eager}} = R - C_{\text{time}}$.

Determining $P_{L_{\text{eager}}}(l)$

To obtain the probability distribution $P_{L_{\text{eager}}}(l)$, we start by computing the joint probability distribution for C_{time} and R denoted by $P_{C_{\text{time}},R}(c_{\text{time}}, r)$. Because C_{time} is dependent of R, the joint probability distribution $P_{C_{\text{time}},R}(c_{\text{time}}, r)$ can be expressed as

$$P_{C_{\text{time}},R}(c_{\text{time}}, r) \quad = \quad P_{C_{\text{time}}|R}(c_{\text{time}}|r) \cdot P_R(r) \tag{3.1}$$

where $P_{C_{\text{time}}|R}(c_{\text{time}}|r)$ is the probability distribution for C_{time} given R and $P_R(r)$ is the probability distribution for the size of the interval in which C_{time} occurs.

As C_{time} is evenly distributed in the interval $[0, R]$, and knowing that for an evenly distributed random variable in an interval the probability distribution is always $\frac{1}{\text{interval size}}$, we can calculate

$P_{C_{\text{time}}|R}(c_{\text{time}}|r)$ as

$$P_{C_{\text{time}}|R}(c_{\text{time}}|r) \quad = \quad \frac{1}{r} \cdot 1_{c_{\text{time}} \in [0,r]} \tag{3.2}$$

where $1_{c_{\text{time}} \in [0,r]}$ is an indicator function with a value of 1 when $c_{\text{time}} \in [0, r]$ and 0 otherwise.

The probability distribution $P_R(r)$ expresses the probability that C_{time} should occur in an interval of size $R = r$. Intuitively, the probability of C_{time} occurring in an interval is proportional to the size of the interval. When the interval size is given as r, the function $f(r) = r \cdot 1_{r \in [R_{\text{min}}, R_{\text{max}}]}$ exhibits this property. We can obtain the probability distribution $P_R(r)$ (knowing that the area under the entire probability curve must always be 1) as $f(r)$ divided with the area of $f(r)$. This yields

$$\begin{aligned} P_R(r) \quad &= \quad \frac{r}{\int_{R_{\text{min}}}^{R_{\text{max}}} r \; dr} \cdot 1_{r \in [R_{\text{min}}, R_{\text{max}}]} \\ &= \quad \frac{2r}{{R_{\text{max}}}^2 - {R_{\text{min}}}^2} \cdot 1_{r \in [R_{\text{min}}, R_{\text{max}}]} \end{aligned} \tag{3.3}$$

The joint probability distribution $P_{C_{\text{time}},R}(c_{\text{time}}, r)$ is thus

$$\begin{aligned} P_{C_{\text{time}},R}(c_{\text{time}}, r) \quad &= \quad P_{C_{\text{time}}|R}(c_{\text{time}}|r) \cdot P_{C_{\text{time}} \in [0,R]}(r) \\ &= \quad \frac{1}{r} \cdot 1_{c_{\text{time}} \in [0,r]} \cdot \frac{2r}{{R_{\text{max}}}^2 - {R_{\text{min}}}^2} \cdot 1_{r \in [R_{\text{min}}, R_{\text{max}}]} \\ &= \quad \frac{2}{{R_{\text{max}}}^2 - {R_{\text{min}}}^2} \cdot 1_{c_{\text{time}} \in [0,r]} \cdot 1_{r \in [R_{\text{min}}, R_{\text{max}}]} \end{aligned} \tag{3.4}$$

The probability distribution $P_{L_{\text{eager}}}(l)$ for L_{eager} can be obtained by integrating over the joint probability distribution $P_{C_{\text{time}},R}(c_{\text{time}}, r)$ for all possible values of c_{time} and r. As r can be expressed as $r = c_{\text{time}} + l$ (which follows from $l = r - c_{\text{time}}$), we have

$$\begin{aligned} P_{L_{\text{eager}}}(l) \quad &= \quad \int_{-\infty}^{\infty} P_{C_{\text{time}},R}(c_{\text{time}}, c_{\text{time}} + l) dc_{\text{time}} \\ &= \quad \int_{-\infty}^{\infty} \frac{2}{{R_{\text{max}}}^2 - {R_{\text{min}}}^2} \cdot 1_{c_{\text{time}} \in [0, c_{\text{time}} + l]} \cdot 1_{c_{\text{time}} + l \in [R_{\text{min}}, R_{\text{max}}]} dc_{\text{time}} \\ &= \quad \frac{2}{{R_{\text{max}}}^2 - {R_{\text{min}}}^2} \cdot 1_{l \in [0,\infty[} \int_{-\infty}^{\infty} 1_{c_{\text{time}} \in [0,\infty[} \cdot 1_{c_{\text{time}} \in [R_{\text{min}} - l, R_{\text{max}} - l]} dc_{\text{time}} \\ &= \quad \frac{2}{{R_{\text{max}}}^2 - {R_{\text{min}}}^2} \cdot 1_{l \in [0,\infty[} \int_{0 \vee R_{\text{min}} - l}^{R_{\text{max}} - l} dc_{\text{time}} \\ &= \quad \frac{2}{{R_{\text{max}}}^2 - {R_{\text{min}}}^2} \cdot 1_{l \in [0, R_{\text{max}}]} \left[c_{\text{time}} \right]_{0 \vee R_{\text{min}} - l}^{R_{\text{max}} - l} \end{aligned} \tag{3.5}$$

Note, that when $x \vee y$ occurs in the limit of an integral is should be interpreted as $\max(x, y)$ and when $x \wedge y$ occurs in the limit of an integral is should be interpreted as $\min(x, y)$.

From formula 3.5 we then have

$$P_{L_{\text{eager}}}(l) \quad = \quad \begin{cases} \frac{2(R_{\text{max}} - R_{\text{min}})}{{R_{\text{max}}}^2 - {R_{\text{min}}}^2} = \frac{2}{R_{\text{max}} + R_{\text{min}}} & \text{if } 0 \leq l \leq R_{\text{min}} \\ \frac{2(R_{\text{max}} - l)}{{R_{\text{max}}}^2 - {R_{\text{min}}}^2} & \text{if } R_{\text{min}} < l \leq R_{\text{max}} \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

This probability distribution for $P_{L_{\text{eager}}}(l)$ is plotted in figure 3.5

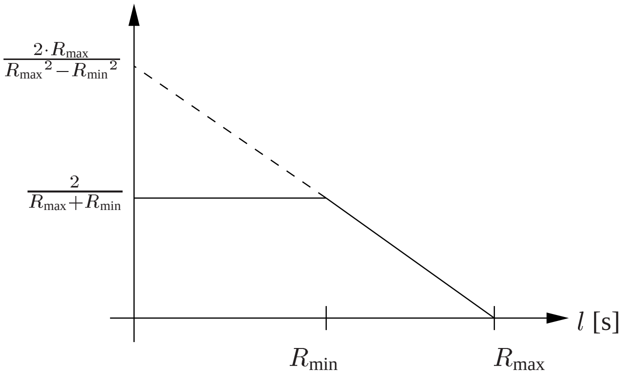


Figure 3.5: Probability distribution $P_{L_{\text{eager}}}(l)$ for the *Eager Cell Switching* handoff initiation strategy.

From figure 3.5 we observe, that there is the highest probability of obtaining handoff latencies in the range between 0 and R_{min}, that the maximum handoff latency is determined by R_{max} and that handoff latencies close to a value of R_{max} is less likely to occur.

In figure 3.6 we have plotted the probability distribution for *Eager Cell Switching* when $R_{\text{min}} = 0.5$ seconds and $R_{\text{max}} = 1.5$ seconds.

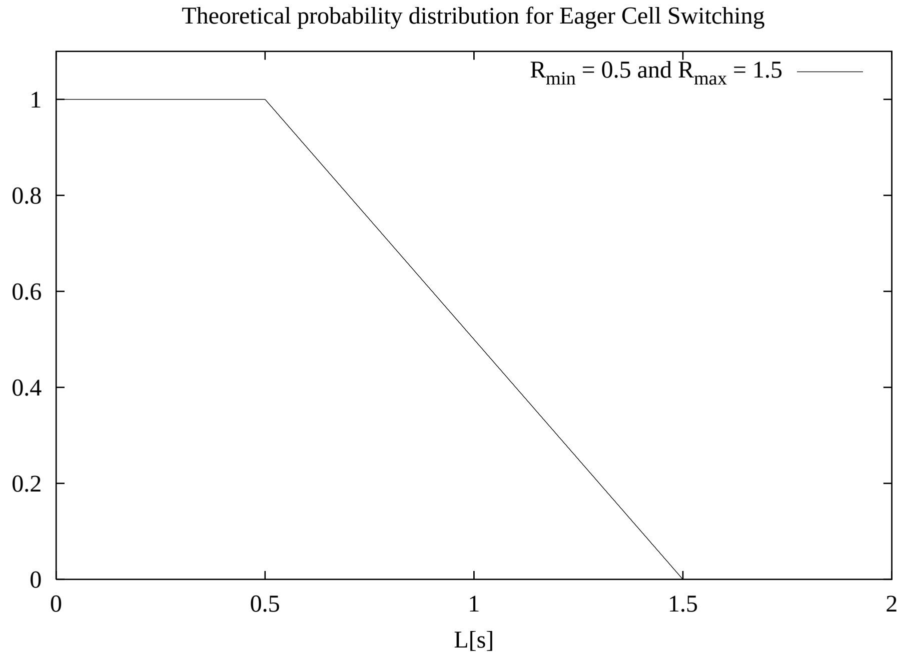


Figure 3.6: Probability distribution for *Eager Cell Switching* when $R_{\text{min}} = 0.5$ seconds and $R_{\text{max}} = 1.5$ seconds.

From figure 3.6 it is seen, that the probability distribution $P_{L_{\text{eager}}}(l)$ for $R_{\text{min}} = 0.5$ and $R_{\text{max}} = 1.5$ is constant in the interval $[0, 0.5]$ and then drops linearly in the interval $[0.5, 1.5]$. This probability distribution is also intuitive, as L_{eager} can assume a value in the interval $[0, 0.5]$ for all interval sizes $D \in [0.5, 1.5]$ resulting in a high probability for handoff latencies in the interval $[0, 0.5]$. For L_{eager} to assume a value in the interval $[0.5, 1.5]$, C_{time} must occur in a large interval. For example, for L_{eager} to assume the value of 1, C_{time} must occur in an interval of size $D \in [1, 1.5]$.

Determining $\overline{L}_{\text{eager}}$

Having calculated the probability distribution $P_{L_{\text{eager}}}(l)$, we can obtain $\overline{L}_{\text{eager}}$ by integrating over the product of $L_{\text{eager}} = l$ and $P_{L_{\text{eager}}}(l)$ for all possible values of L_{eager}. We can thus calculate $\overline{L}_{\text{eager}}$ by

$$\begin{aligned}
\overline{L}_{\text{eager}} &= \int_0^{R_{\text{max}}} l P_{L_{\text{eager}}}(l) dl \qquad (3.7)\\
&= \int_0^{R_{\text{min}}} l \frac{2}{R_{\text{max}} + R_{\text{min}}} dl + \int_{R_{\text{min}}}^{R_{\text{max}}} l \frac{2(R_{\text{max}} - l)}{{R_{\text{max}}}^2 - {R_{\text{min}}}^2} dl \\
&= \frac{1}{R_{\text{max}} + R_{\text{min}}} \left[l^2\right]_0^{R_{\text{min}}} + \frac{2}{{R_{\text{max}}}^2 - {R_{\text{min}}}^2} \int_{R_{\text{min}}}^{R_{\text{max}}} l(R_{\text{max}} - l) dl \\
&= \frac{{R_{\text{min}}}^2}{R_{\text{max}} + R_{\text{min}}} + \frac{2}{{R_{\text{max}}}^2 - {R_{\text{min}}}^2} \left[\frac{1}{2} l^2 R_{\text{max}} - \frac{1}{3} l^3\right]_{R_{\text{min}}}^{R_{\text{max}}} \\
&= \frac{{R_{\text{min}}}^2}{R_{\text{max}} + R_{\text{min}}} + \frac{\frac{1}{3}{R_{\text{max}}}^3 - {R_{\text{min}}}^2 R_{\text{max}} + \frac{2}{3}{R_{\text{min}}}^3}{{R_{\text{max}}}^2 - {R_{\text{min}}}^2} \\
&= \frac{{R_{\text{min}}}^2 (R_{\text{max}} - R_{\text{min}}) + \frac{1}{3}{R_{\text{max}}}^3 - {R_{\text{min}}}^2 R_{\text{max}} + \frac{2}{3}{R_{\text{min}}}^3}{{R_{\text{max}}}^2 - {R_{\text{min}}}^2} \\
&= \frac{{R_{\text{max}}}^3 - {R_{\text{min}}}^3}{3({R_{\text{max}}}^2 - {R_{\text{min}}}^2)}
\end{aligned}$$

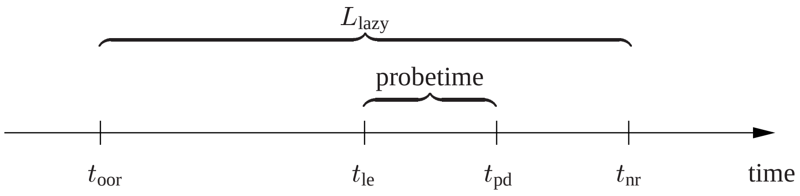
If we insert values of $R_{\text{min}} = 0.5$ and $R_{\text{max}} = 1.5$ (as plotted in figure 3.6) in formula 3.7 we obtain $\overline{L}_{\text{eager}} = \frac{13}{24} \approx 0.54$ seconds.

3.4 Lazy Cell Switching handoff initiation strategy

Recall that the *Lazy Cell Switching* handoff initiation strategy is influenced by the interval between receiving unsolicited router advertisement and the lifetime of network prefixes. The mathematical model for this handoff initiation strategy should thus take as input the possible range of intervals between broadcasting router advertisements at access routers, $[R_{\text{min}}, R_{\text{max}}]$, and the lifetime of the broadcasted network prefixes, T_{l}. As output we seek a function describing the probability distribution for the handoff latency. This function can then be used to determine theoretical values for the minimum, maximum and average handoff latency.

Determining L_{lazy}

Recall that *Lazy Cell Switching* does not initiate a handoff before the currently selected network has been confirmed to be unreachable. This corresponds to the situation depicted in figure 3.7.



t_{oor}: Out of range of primary network

t_{le}: Lifetime of primary network expires

t_{pd}: Mobile node finished probing primary network

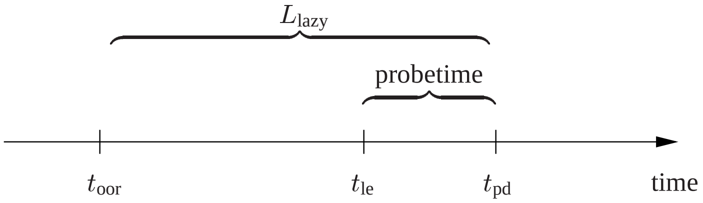
t_{nr}: Router advertisement which triggers handoff initiation received from new network

Figure 3.7: Possible definition of L_{lazy} for the *Lazy Cell Switching* handoff initiation strategy.

When the lifetime of a network prefix expires, the mobile node might[2] **probe** the default router at the primary network to learn if it is still reachable. After the probing is completed and the primary network is declared unreachable, a handoff is initiated upon reception of a router advertisement from a new network.

In practice however, if after probing the primary network is concluded to be unreachable, the mobile node should not wait until the reception of a router advertisement from a new network before initiating a handoff. It should try to immediately attach to the network which is most likely to be reachable. Hints for a probable reachable network could be a router advertisement received from a new network before the primary network was concluded to be unreachable or a network which has previously been known to be reachable.

If a handoff is initiated immediately upon concluding that the primary network is unreachable this corresponds to the situation depicted in figure 3.8.



t_{oor}: Out of range of primary network

t_{le}: Lifetime of primary network expires

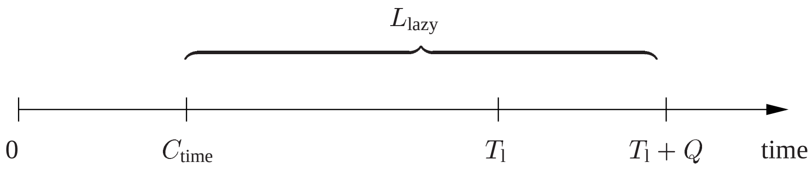
t_{pd}: Mobile node finished probing primary network

Figure 3.8: Chosen definition of L_{lazy} for the *Lazy Cell Switching* handoff initiation strategy.

---

[2] This depends on the actual mobile node implementation.

We choose to develop a mathematical model restricted to the definition depicted in 3.8. We do not consider the behavior depicted in figure 3.7 to be feasible, as it adds extra (and unnecessary) overhead to the handoff latency. We start from the model depicted in figure 3.9.



0: Time of receiving the last router advertisement from primary network

C_{time}: Time of leaving the range of the primary network

T_{l}: Lifetime of network prefix advertised in router advertisement

Q: Probing time

$L_{\text{lazy}} = T_{\text{l}} - C_{\text{time}} + Q$

Figure 3.9: Model for computing L_{lazy} for the *Lazy Cell Switching* handoff initiation algorithm.

If 0 denotes the time of receiving the last router advertisement from the primary network, T_{l} denotes the lifetime, C_{time} denotes the time when the mobile node leaves the range of the primary network and Q denotes the probing time, L_{lazy} can be expressed as

$$\begin{aligned} L_{\text{lazy}} &= \text{Lifetime remaining of previous network} + Q \\ &= T_{\text{l}} - C_{\text{time}} + Q \end{aligned} \tag{3.8}$$

Determining $P_{L_{\text{lazy}}}(l)$

As for the *Eager Cell Switching* handoff initiation algorithm we start by computing the probability distribution, which we for the *Lazy Cell Switching* handoff initiation strategy denotes $P_{L_{\text{lazy}}}(l)$.

We assume Q to be uniformly distributed within the interval $[Q_{\text{min}}, Q_{\text{max}}]$. The probability distribution for Q is thus

$$P_Q(q) = \frac{1}{Q_{\text{max}} - Q_{\text{min}}} \cdot 1_{q \in [Q_{\text{min}}, Q_{\text{max}}]} \tag{3.9}$$

The probability distribution for C_{time} using *Lazy Cell Switching* corresponds to $P_{L_{\text{eager}}}(l)$ which was calculated in formula 3.6 for *Eager Cell Switching*. This can be seen from the following argument. Recall that L_{eager} is given by $L_{\text{eager}} = R - C_{\text{time}}$. As C_{time} is evenly distributed in the interval between 0 and R, the probability distribution of $R - C_{\text{time}}$ is identical to the probability distribution of $C_{\text{time}} - 0$. We thus have the probability distribution for C_{time} as

$$P_{C_{\text{time}}}(c_{\text{time}}) = \begin{cases} \frac{2}{R_{\text{max}} + R_{\text{min}}} & \text{if } 0 \le c_{\text{time}} \le R_{\text{min}} \\ \frac{2(R_{\text{max}} - c_{\text{time}})}{{R_{\text{max}}}^2 - {R_{\text{min}}}^2} & \text{if } R_{\text{min}} < c_{\text{time}} \le R_{\text{max}} \\ 0 & \text{otherwise} \end{cases} \tag{3.10}$$

Given the probability distribution $P_{C_{\text{time}}}(c_{\text{time}})$ we can find the probability distribution for L_{lazy} when no probing is used, $P_{L_{\text{np}}}(l_{\text{np}})$. From formula 3.8 we deduce that $L_{\text{np}} = T_{\text{l}} - C_{\text{time}}$ which implies $C_{\text{time}} = T_{\text{l}} - L_{\text{np}}$. The probability distribution for L_{np} can thus be expressed as

$$P_{L_{\text{np}}}(l_{\text{np}}) \quad = \quad P_{C_{\text{time}}}(T_{\text{l}} - l_{\text{np}}) \tag{3.11}$$

To find the probability distribution $P_{L_{\text{lazy}}}(l)$ we integrate over the joint probability distribution of Q and L_{np} for all possible values of q and l_{np}. As q can be expressed as $q = l - l_{\text{np}}$ we have

$$\begin{aligned}
P_{L_{\text{lazy}}}(l) &= \int_{-\infty}^{\infty} P_{\text{Q}}(q) \cdot P_{L_{\text{np}}}(l_{\text{np}}) dl_{\text{np}} \qquad (3.12) \\
&= \int_{-\infty}^{\infty} P_{\text{Q}}(l - l_{\text{np}}) \cdot P_{C_{\text{time}}}(T_{\text{l}} - l_{\text{np}}) dl_{\text{np}} \\
&= \int_{-\infty}^{\infty} \frac{1}{Q_{\text{max}} - Q_{\text{min}}} \cdot 1_{(l - l_{\text{np}}) \in [Q_{\text{min}}, Q_{\text{max}}]} \cdot \Big(\frac{2}{R_{\text{min}} + R_{\text{max}}} \cdot 1_{(T_{\text{l}} - l_{\text{np}}) \in [0, R_{\text{min}}]} \\
&\qquad + \frac{2(R_{\text{max}} - T_{\text{l}} + l_{\text{np}})}{{R_{\text{max}}}^2 - {R_{\text{min}}}^2} \cdot 1_{(T_{\text{l}} - l_{\text{np}}) \in [R_{\text{min}}, R_{\text{max}}]} \Big) dl_{\text{np}} \\
&= \frac{2}{Q_{\text{max}} - Q_{\text{min}}} \int_{l - Q_{\text{max}}}^{l - Q_{\text{min}}} \frac{1}{R_{\text{min}} + R_{\text{max}}} \cdot 1_{l_{\text{np}} \in [T_{\text{l}} - R_{\text{min}}, T_{\text{l}}]} \\
&\qquad\qquad + \frac{R_{\text{max}} - T_{\text{l}} + l_{\text{np}}}{{R_{\text{max}}}^2 - {R_{\text{min}}}^2} \cdot 1_{l_{\text{np}} \in [T_{\text{l}} - R_{\text{max}}, T_{\text{l}} - R_{\text{min}}]} dl_{\text{np}} \\
&= \frac{2}{Q_{\text{max}} - Q_{\text{min}}} \Big(\frac{1}{R_{\text{min}} + R_{\text{max}}} \int_{l - Q_{\text{max}} \vee T_{\text{l}} - R_{\text{min}}}^{l - Q_{\text{min}} \wedge T_{\text{l}}} dl_{\text{np}} \\
&\qquad + \frac{1}{{R_{\text{max}}}^2 - {R_{\text{min}}}^2} \int_{l - Q_{\text{max}} \vee T_{\text{l}} - R_{\text{max}}}^{l - Q_{\text{min}} \wedge T_{\text{l}} - R_{\text{min}}} (R_{\text{max}} - T_{\text{l}} + l_{\text{np}}) dl_{\text{np}} \Big) \\
&= \frac{2}{Q_{\text{max}} - Q_{\text{min}}} \Big(\frac{1}{R_{\text{min}} + R_{\text{max}}} \Big[l_{\text{np}} \Big]_{l - Q_{\text{max}} \vee T_{\text{l}} - R_{\text{min}}}^{l - Q_{\text{min}} \wedge T_{\text{l}}} \\
&\qquad + \frac{1}{{R_{\text{max}}}^2 - {R_{\text{min}}}^2} \Big[\frac{1}{2} {l_{\text{np}}}^2 + l_{\text{np}}(R_{\text{max}} - T_{\text{l}}) \Big]_{l - Q_{\text{max}} \vee T_{\text{l}} - R_{\text{max}}}^{l - Q_{\text{min}} \wedge T_{\text{l}} - R_{\text{min}}} \Big) \\
&= \frac{2}{(Q_{\text{max}} - Q_{\text{min}})(R_{\text{min}} + R_{\text{max}})} \Big[l_{\text{np}} \Big]_{l - Q_{\text{max}} \vee T_{\text{l}} - R_{\text{min}}}^{l - Q_{\text{min}} \wedge T_{\text{l}}} \\
&\quad + \frac{2}{(Q_{\text{max}} - Q_{\text{min}})({R_{\text{max}}}^2 - {R_{\text{min}}}^2)} \Big[\frac{1}{2} {l_{\text{np}}}^2 + l_{\text{np}}(R_{\text{max}} - T_{\text{l}}) \Big]_{l - Q_{\text{max}} \vee T_{\text{l}} - R_{\text{max}}}^{l - Q_{\text{min}} \wedge T_{\text{l}} - R_{\text{min}}}
\end{aligned}$$

If the left half of the result of formula 3.12 is denoted by $P_L(l)_{\text{l}}$ and the right half of the result of formula 3.12 is denoted by $P_L(l)_{\text{r}}$ we have

$$P_{L_{\text{lazy}}}(l) \quad = \quad P_L(l)_{\text{l}} + P_L(l)_{\text{r}} \tag{3.13}$$

Further reduction of $P_L(l)_\text{l}$ yields

$$
\begin{aligned}
k_l &= \frac{2}{(Q_\text{max} - Q_\text{min})(R_\text{min} + R_\text{max})} \\
P_L(l)_\text{l} &= k_l \Big[l_\text{np} \Big]_{l - Q_\text{max} \vee T_\text{l} - R_\text{min}}^{l - Q_\text{min} \wedge T_\text{l}} \\
&= k_l \cdot \begin{cases}
l - Q_\text{min} - T_\text{l} + R_\text{min} & \text{if } T_\text{l}\text{-}R_\text{min}\text{+}Q_\text{min} \leq l \leq \text{min}(T_\text{l} + Q_\text{min}, \\
 & \qquad\qquad T_\text{l} - R_\text{min} + Q_\text{max}) \\
-Q_\text{min} + Q_\text{max} & \text{if } T_\text{l} - R_\text{min} + Q_\text{max} < l \leq T_\text{l} + Q_\text{min} \\
R_\text{min} & \text{if } T_\text{l} + Q_\text{min} < l \leq T_\text{l} - R_\text{min} + Q_\text{max} \\
T_\text{l} - l + Q_\text{max} & \text{if } \text{max}(T_\text{l} + Q_\text{min}, \\
 & \qquad T_\text{l} - R_\text{min} + Q_\text{max}) < l \leq T_\text{l} + Q_\text{max} \\
0 & \text{otherwise}
\end{cases}
\end{aligned}
\tag{3.14}
$$

Further reduction of $P_L(l)_\text{r}$ yields

$$
\begin{aligned}
k_r &= \frac{2}{(Q_\text{max} - Q_\text{min})({R_\text{max}}^2 - {R_\text{min}}^2)} \\
P_L(l)_\text{r} &= k_r \left[\frac{1}{2} {l_\text{np}}^2 + l_\text{np}(R_\text{max} - T_\text{l}) \right]_{l - Q_\text{max} \vee T_\text{l} - R_\text{max}}^{l - Q_\text{min} \wedge T_\text{l} - R_\text{min}} \\
&= k_r \cdot \begin{cases}
\left[\tfrac{1}{2} {l_\text{np}}^2 + l_\text{np}(R_\text{max} - T_\text{l}) \right]_{T_\text{l} - R_\text{max}}^{l - Q_\text{min}} & \text{if } T_\text{l} - R_\text{max} + Q_\text{min} \leq l \leq \\
 & \quad \text{min}(T_\text{l} - R_\text{min} + Q_\text{min}, T_\text{l} - R_\text{max} + Q_\text{max}) \\
\left[\tfrac{1}{2} {l_\text{np}}^2 + l_\text{np}(R_\text{max} - T_\text{l}) \right]_{l - Q_\text{max}}^{l - Q_\text{min}} & \text{if } T_\text{l} - R_\text{max} + Q_\text{max} < l \leq T_\text{l} - R_\text{min} + Q_\text{min} \\
\left[\tfrac{1}{2} {l_\text{np}}^2 + l_\text{np}(R_\text{max} - T_\text{l}) \right]_{l - Q_\text{max}}^{T_\text{l} - R_\text{min}} & \text{if } \text{max}(T_\text{l} - R_\text{min} + Q_\text{min}, T_\text{l} - R_\text{max} + Q_\text{max}) \\
 & \quad < l \leq T_\text{l} - R_\text{min} + Q_\text{max} \\
\left[\tfrac{1}{2} {l_\text{np}}^2 + l_\text{np}(R_\text{max} - T_\text{l}) \right]_{T_\text{l} - R_\text{max}}^{T_\text{l} - R_\text{min}} & \text{if } T_\text{l} - R_\text{min} + Q_\text{min} < l \leq T_\text{l} - R_\text{max} + Q_\text{max} \\
0 & \text{otherwise}
\end{cases}
\end{aligned}
\tag{3.15}
$$

As can be seen from formula 3.14 and formula 3.15, the probability distribution for $P_{L_\text{lazy}}(l)$ is composed of numerous functions each restricted to certain values of l. In figure 3.10 we have plotted examples of probability distributions for two different configurations of R_min, R_max, Q_min, Q_max and T_l.

From figure 3.10 we observe, that the highest probability is for obtaining a handoff latency around the value of the network prefix lifetime. We also note that the minimum handoff latency is determined by $T_\text{l} - R_\text{max} + Q_\text{min}$. This is intuitive, as $T_\text{l} - R_\text{max}$ is the minimum amount of time of the lifetime which can remain when leaving the range of the primary network and entering the range of a new network and Q_min is the minimum time that probing of the primary network will last. Similarly, the maximum handoff latency is determined by $T_\text{l} + Q_\text{max}$.

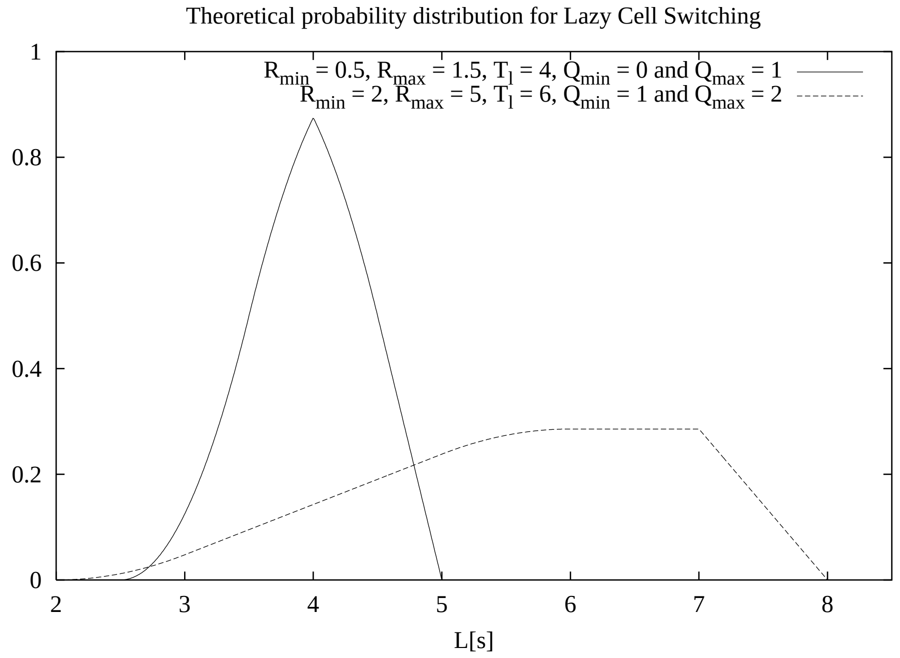


Figure 3.10: Probability distribution for *Lazy Cell Switching* for two different configurations.

Determining $\overline{L}_{\text{lazy}}$

To obtain $\overline{L}_{\text{lazy}}$ it is intuitive to integrate over the probability distribution for all possible values of L_{lazy}. However, due to the complex nature of the probability distribution for the *Lazy Cell Switching* handoff initiation algorithm, we calculate $\overline{L}_{\text{lazy}}$ in a simpler way.

Recall from formula 3.8, that $L_{\text{lazy}} = T_{\text{l}} - C_{\text{time}} + Q$. We have already argued, that the probability distribution for C_{time}, $P_{C_{\text{time}}}(c_{\text{time}})$, is equal to the probability distribution for L_{eager}, $P_{L_{\text{eager}}}(l)$. This implies that the average value of C_{time}, denoted $\overline{C}_{\text{time}}$, is equal to $\overline{L}_{\text{eager}}$, which we calculated in formula 3.7. As for the probing time Q, it is independent of the other variables, and its average $\overline{Q}$ is $\frac{1}{2}(Q_{\text{max}} - Q_{\text{min}})$. We can therefore calculate $\overline{L}_{\text{lazy}}$ by

$$\begin{aligned} \overline{L}_{\text{lazy}} &= T_{\text{l}} - \overline{C}_{\text{time}} + \overline{Q} \\ &= T_{\text{l}} - \frac{{R_{\text{max}}}^3 - {R_{\text{min}}}^3}{3({R_{\text{max}}}^2 - {R_{\text{min}}}^2)} + \frac{1}{2}(Q_{\text{max}} - Q_{\text{min}}) \end{aligned} \tag{3.16}$$

If we insert values of $R_{\text{min}} = 0.5$, $R_{\text{min}} = 1.5$, $Q_{\text{min}} = 0$, $Q_{\text{max}} = 1$ and $T_{\text{l}} = 4$ (as plotted in figure 3.10) in formula 3.16 we obtain $\overline{L}_{\text{lazy}} = \frac{95}{24} \approx 3.96$ seconds.

3.5 Optimizing router advertisement settings

In the Mobile IPv6 specification values of R_{min} = 0.5 and R_{max} = 1.5 seconds are recommended [Johnson *et al.*, 2000]. On average, this configuration will cause a network load of one router advertisement every second.

Using the theoretical models for *Lazy Cell Switching* and *Eager Cell Switching* we can calculate the minimum, maximum and average handoff latency when using the recommended settings, as shown in figure 3.11. Here we have chosen $Q_{\text{min}} = 0$ and $Q_{\text{max}} = 1$ as these settings reflect the KAME Mobile IPv6 configuration as we later describe in section 4.3.2.

Handoff	**Configuration**					L **[s] (theory)**		
strategy	R_{min}	R_{max}	T_{l}	Q_{min}	Q_{max}	**Avg**	**Min**	**Max**
Eager	0.5	1.5	N/A	N/A	N/A	0.54	0	1.5
Lazy	0.5	1.5	4	0	1	3.96	2.5	5.0

Figure 3.11: Theoretical values for L using the recommended configuration.

The average load at a network due to router advertisements is equal to half the sum of R_{min} and R_{max}. This means that if R_{min} is decreased by e.g. 0.2 seconds, the network load can be kept unchanged by adding 0.2 seconds to R_{max}. In the following we investigate the effect of varying R_{min} and R_{max} on the handoff latency.

3.5.1 Eager Cell Switching handoff initiation strategy

In figure 3.12 we have plotted $\overline{L}_{\text{eager}}$ for the *Eager Cell Switching* handoff initiation algorithm, when R_{min} and R_{max} are chosen so that their sum is always 2. A sum of 2 is chosen, as this is the sum of $R_{\text{min}} = 0.5$ and $R_{\text{max}} = 1.5$ which is the default configuration as proposed in [Johnson *et al.*, 2000].

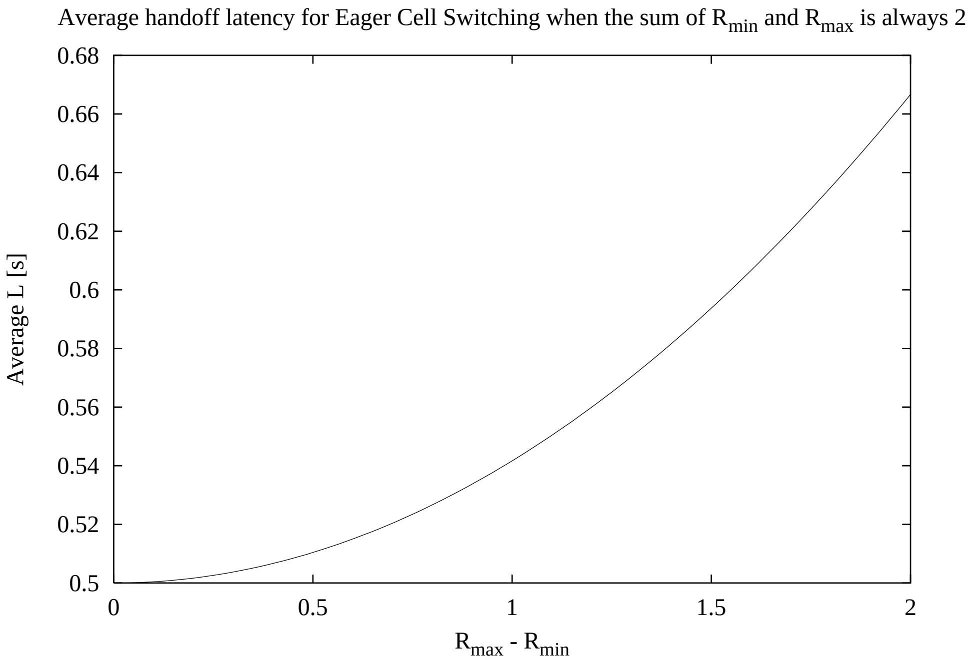


Figure 3.12: Development of $\overline{L}_{\text{eager}}$ for *Eager Cell Switching* when the sum of R_{min} and R_{max} is always 2.

As can be seen from figure 3.12, a lower value for $\overline{L}_{\text{eager}}$ is achieved by letting R_{min} and R_{max} have values close to each other. This corresponds to that mobile nodes are able to discover new networks faster if access routers are configured with R_{min} and R_{max} close to each other.

3.5.2 Lazy Cell Switching handoff initiation strategy

In figure 3.13 we have plotted $\overline{L}_{\text{lazy}}$ for the *Lazy Cell Switching* handoff algorithm, when R_{min} and R_{max} are chosen so that their sum is always 2 (yielding identical average network load).

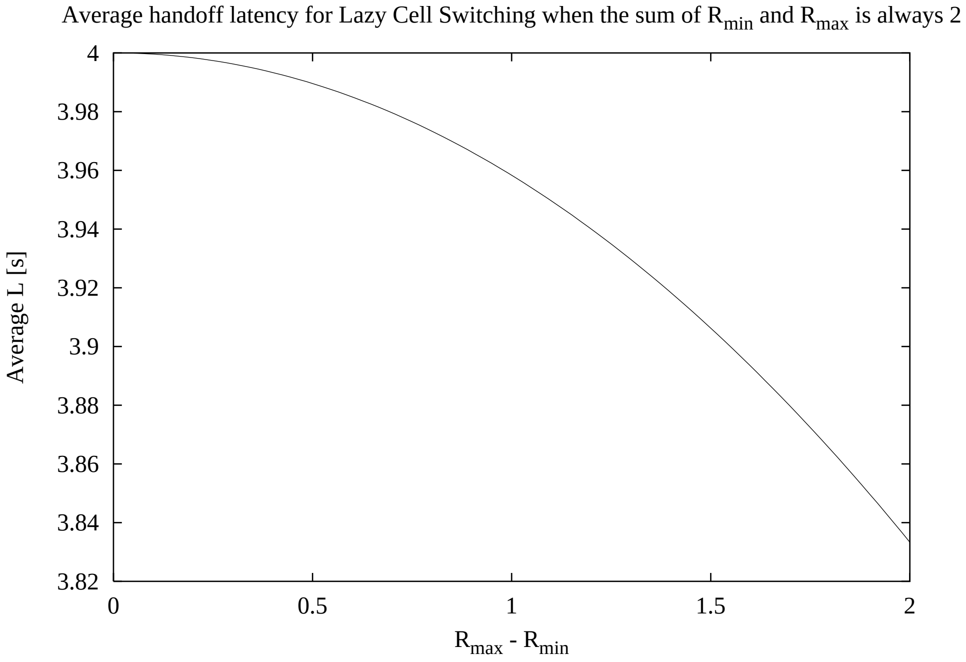


Figure 3.13: Development of $\overline{L}_{\text{lazy}}$ for *Lazy Cell Switching* when the sum of R_{min} and R_{max} is always 2.

As can be seen from figure 3.13, a lower value for $\overline{L}_{\text{lazy}}$ is achieved by letting R_{min} and R_{max} have values far from each other. This is in direct conflict with what is the optimal configuration for the *Eager Cell Switching* handoff initiation strategy.

Figure 3.14 depicts the development in average handoff latency using default settings, but for different configurations of lifetime.

As can be seen from figure 3.14 the configuration of the lifetime has a major impact on the handoff latency produced by the *Lazy Cell Switching* handoff initiation algorithm. The average handoff latency is proportional to the lifetime, which suggests that the lifetime should be configured close to R_{max} in order to obtain a lower average handoff latency.

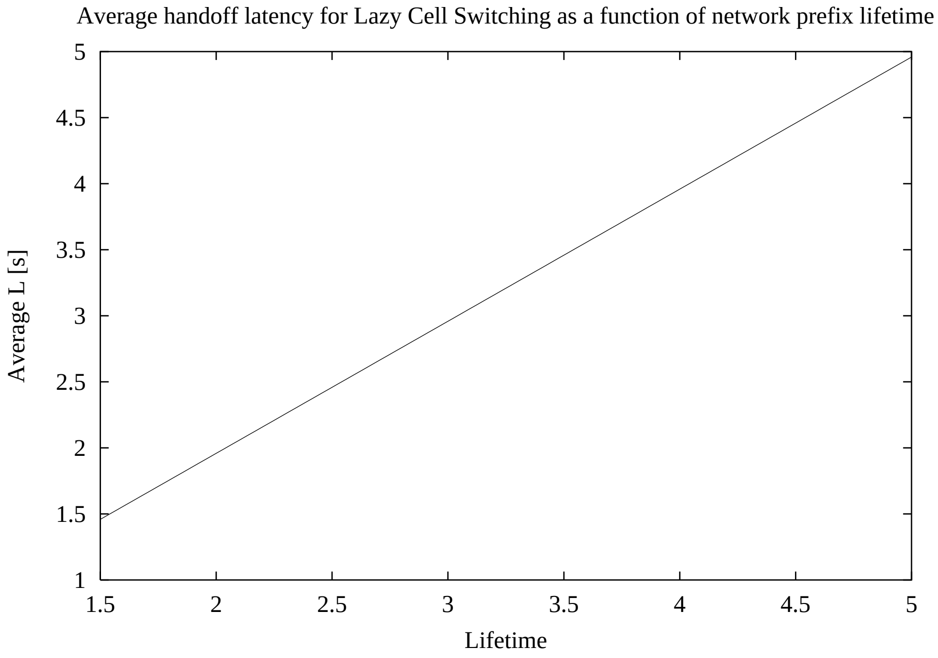


Figure 3.14: Development of $\overline{L}_{\text{lazy}}$ for the *Lazy Cell Switching* handoff initiation algorithm as a function of lifetime when $R_{\text{min}} = 0.5$, $R_{\text{max}} = 1.5$, $Q_{\text{min}} = 0$ and $Q_{\text{max}} = 1$.

3.5.3 Proposal of new default router configuration

As shown in the plots in figure 3.12 and figure 3.13, *Eager Cell Switching* requires R_{min} and R_{max} to be very close and *Lazy Cell Switching* requires R_{min} and R_{max} to be far apart, in order to obtain the lowest value of $\overline{L}$. In addition, from figure 3.14 it can be seen that the *Lazy Cell Switching* handoff initiation strategy yields lower average handoff latency by setting the lifetime close to R_{max}.

The results for *Eager Cell Switching* reflects how fast a new network is discovered after entering a link. In order to discover networks faster, R_{min} and R_{max} should be configured with almost identical values. It is possible to configure R_{min} and R_{max} with identical values, but this increases the risk that two or more routers should synchronize.

On the other hand, this will cause degraded performance for a mobile node using *Lazy Cell Switching*, if there is still some of the lifetime of the previous network left upon entering the new link. In order to reduce this effect the lifetime should be configured at a lower value.

As a network administrator cannot always know which handoff initiation strategy mobile nodes will use, we must consider both the *Eager Cell Switching* and the *Lazy Cell Switching* strategies. To allow for faster detection of new networks we propose that R_{min} and R_{max} is configured with very similar values. To reduce the effect on the average value of L for the *Lazy Cell Switching* handoff initiation strategy, we further suggest that the lifetime is configured to a value slightly above that of R_{max}. An alternative to lowering the lifetime is to include an *Advertisement Interval Option* in router advertisements as defined in [Johnson *et al.*, 2000]. This carries the value of R_{max} at the broadcasting router, which can be used by a handoff initiation algorithm to reason about when to expect the next router advertisement.

In figure 3.15 we propose a new set of default settings imposing an amount of network load equal to the network load of the current default settings, and list the corresponding values for L for both the *Eager Cell Switching* and the Lazy Cell Switching handoff initiation algorithms. The proposed value of T_l reflects that this constant must be configured with a granularity of seconds [Johnson *et al.*, 2000].

Handoff	**New configuration**					**L [s] (theory)**		
strategy	R_min	R_max	T_l	Q_min	Q_max	**Avg**	**Min**	**Max**
Eager	0.9	1.1	N/A	N/A	N/A	0.50	0	1.1
Lazy	0.9	1.1	2	0	1	2.00	0.9	3.0

Figure 3.15: Theoretical values for L using our proposed configuration.

As can be seen by comparing the average and maximum values of L for the default settings in figure 3.11 with the average and maximum values for L for our proposed settings in figure 3.15, our proposed settings have better average and worst case performance for both handoff initiation strategies. This improvement in performance is obtained without increasing network load due to router advertisements.

3.6 Simulating handoff initiation strategies

To stress the correctness of the mathematical models for *Eager Cell Switching* and *Lazy Cell Switching* we implemented two simulators; one running in real time and one able to simulate many handoffs in a very short time. The fast simulator was implemented to verify the correctness of the mathematical models. The real time simulator was implemented as it more closely emulates the behavior of the testbed and thus we were confident that it would be implemented correctly.

It should be noted that both the mathematical models and the simulators implement our interpretation of the Mobile IPv6 specification [Johnson *et al.*, 2000]. This means, that there might occur inconsistencies with actual implementations of Mobile IPv6. In chapter 4 we present the results of investigating the conformance between the KAME Mobile IPv6 implementation [KAME] and our theoretical models.

The simulator running in real time has the following characteristics:

- It can be made to simulate either the *Lazy Cell Switching* handoff initiation strategy or the *Eager Cell Switching* handoff initiation strategy.
- It consists of three independently running threads:
  - A mobile node moving between two links. The mobile node is always listening at exactly one of the links.
  - An access router broadcasting router advertisements onto the first link at a frequency configured by R_min and R_max and containing a lifetime of T_l.

  - A second access router broadcasting router advertisements onto the second link and configured with the same values for R_{min}, R_{max} and T_{l}.

- A histogram (frequency distribution) is output to a file upon request.

The fast simulator has the following characteristics:

- It can be made to simulate either the *Lazy Cell Switching* handoff initiation strategy or the *Eager Cell Switching* handoff initiation strategy.

- It can simulate millions of handoffs in only a few minutes.

- It is based on two variables holding virtual time stamps. A variable t_{oor} representing the time when the primary network goes out of range and a new network is within range. A variable t_{r} representing the time when a router advertisement was last received from a new network at the mobile node. A simulation is then performed in the following way:

  1. Randomly select a time t_{oor} for the primary network to become out of reach.
  2. Chose a time for the arrival of the next router advertisement randomly within the interval $[R_{\text{min}}, R_{\text{max}}]$ and add the time to t_{r} (perform a random walk).
  3. If $t_{\text{r}} \leq t_{\text{oor}}$ then goto 2, else goto 4
  4. Determine the handoff latency from t_{oor} and t_{r} according to the selected handoff initiation strategy.
  5. Goto 1

- A histogram (frequency distribution) is output to a file upon request.

The two simulators were found to output similar results. This was determined by comparing histograms for several different configurations. The fast simulator was used to verify the correctness of the mathematical models for the two simple handoff initiation strategies. By simulating one billion handoffs with the fast simulator and plotting a histogram over the obtained results, we increased our confidence in the proposed mathematical models. The probability distributions for both mathematical models for a series of different configurations were found to be in complete conformance with the results generated by the simulator.

3.7 Summary

In this section we summarize the derived mathematical models for the *Eager Cell Switching* and the *Lazy Cell Switching* handoff initiation strategies. We do not show the derived probability distributions for theoretical handoff latency, but these can be found in formula 3.6 and formula 3.12 respectively.

Recall that L is the theoretical handoff latency, C_{time} the time of leaving the range of the primary network and entering the range of a new network, R the time of the arrival of the first router advertisement from the new network, T_{l} the lifetime of network prefixes specified in router advertisements, Q_{min} the minimum time it can take before probing of the default router at the primary network is completed and Q_{max} the maximum time it can take before probing of the default router at the primary network is completed.

Figure 3.16 depicts selected results from the theory for the *Eager Cell Switching* handoff initiation algorithm.

$$\begin{aligned}
L_{\text{eager}} &= R - C_{\text{time}} \\
\overline{L}_{\text{eager}} &= \frac{{R_{\text{max}}}^3 - {R_{\text{min}}}^3}{3({R_{\text{max}}}^2 - {R_{\text{min}}}^2)} \\
L_{\text{eager}_{\text{min}}} &= 0 \\
L_{\text{eager}_{\text{max}}} &= R_{\text{max}}
\end{aligned}$$

Figure 3.16: Summary of theoretical results for *Eager Cell Switching*.

From figure 3.16 it can be seen, that the minimum handoff latency for the *Eager Cell Switching* handoff initiation algorithm is always 0 while the maximum handoff latency is determined by the value of R_{max}.

Figure 3.17 depicts selected results from the theory for the *Lazy Cell Switching* handoff initiation algorithm.

$$\begin{aligned}
L_{\text{lazy}} &= T_{\text{l}} - C_{\text{time}} + Q \\
\overline{L}_{\text{lazy}} &= T_{\text{l}} - \frac{{R_{\text{max}}}^3 - {R_{\text{min}}}^3}{3({R_{\text{max}}}^2 - {R_{\text{min}}}^2)} + \frac{1}{2}(Q_{\text{max}} - Q_{\text{min}}) \\
L_{\text{lazy}_{\text{min}}} &= T_{\text{l}} - R_{\text{max}} + Q_{\text{min}} \\
L_{\text{lazy}_{\text{max}}} &= T_{\text{l}} + Q_{\text{max}}
\end{aligned}$$

Figure 3.17: Summary of theoretical results for *Lazy Cell Switching*.

From figure 3.17 it can be seen, that the minimum handoff latency for the *Lazy Cell Switching* handoff initiation strategy is decided by the values of T_{l}, R_{max} and Q_{min}. Recall from section 3.2, that we assume that getting out of range of the primary network coincides with getting within range of a new network. The minimum time remaining of the prefix of the primary network upon leaving the range of the primary network is therefore given by $T_{\text{l}} - R_{\text{max}}$. A handoff cannot be initiated before probing of the primary network is completed which takes a minimal period of time specified by Q_{min}.

Chapter 4

Empirical study of existing handoff initiation strategies

In this chapter we present a testbed running Mobile IPv6, in which we empirically study the handoff performance of the *Eager Cell Switching* handoff initiation strategy and the *Lazy Cell Switching* handoff initiation strategy.

First we motivate the need for an experimental approach followed by the design of the Mobile IPv6 testbed. Then a series of experiments are presented and the results evaluated against the theoretical models presented in chapter 3. Finally the empirically obtained results are summarized and an overall conclusion regarding the conformance between theory and practice is made.

4.1 Motivation

In chapter 3 we proposed mathematical models to describe the expected handoff latency of *Eager Cell Switching* and *Lazy Cell Switching*. In order to validate whether these mathematical models are indeed applicable in a real network we decided to study the handoff performance of these two handoff initiation algorithms in a realistic scenario.

To determine the relation between theory and reality we consider a four step approach which allows us to compare theoretically predicted handoff latency with experimentally obtained handoff latency for a wide range of different configurations:

Step 1: By configuring access routers with default settings as suggested in [Johnson *et al.*, 2000] it should be determined how obtained handoff latency conforms to theoretically predicted handoff latency.

Step 2: Recall from chapter 3 that the theoretical models are concerned with the interval between broadcasting unsolicited router advertisements from access routers and the lifetime of network prefixes. To validate the mathematical models the one input variable should be kept fixed and different values for the other input value should be tried. We start by keeping the interval with which router advertisements are broadcasted from access routers fixed and try different settings for lifetime of network prefixes.

Step 3: Then we keep the lifetime of network prefixes fixed and try different intervals between broadcasting unsolicited router advertisements from access routers.

Step 4: Finally, by configuring access routers with our proposed settings it should be determined if the theoretically predicted reduction in handoff latency is also present in practice.

If these four experiment series should produce results that are in conformance with the mathematical models it would increase our confidence in these models.

What is needed is thus a platform on which this experimental study can be performed. We have decided to provide this platform by setting up a testbed in the laboratory. There are two main reasons for this choice:

- Performance of Mobile IPv6 is highly dependent of network specifics such as topology and load. A testbed allows experiments to be conducted in a controlled environment making it easier to reproduce test results and thus feasible to compare results for different handoff initiation strategies.
- Reports of performance measurements on actual implementations of Mobile IPv6 are almost non-existing. Thus providing an environment for measuring this performance is interesting in itself.

4.2 Design of testbed

In this section a description of the kind of scenarios, that the testbed should be able to emulate, is first given. Then the composition of the testbed is motivated and described, and finally the installed software is presented.

4.2.1 Scenario to be modeled

The testbed should be able to emulate a scenario as depicted in figure 4.1.

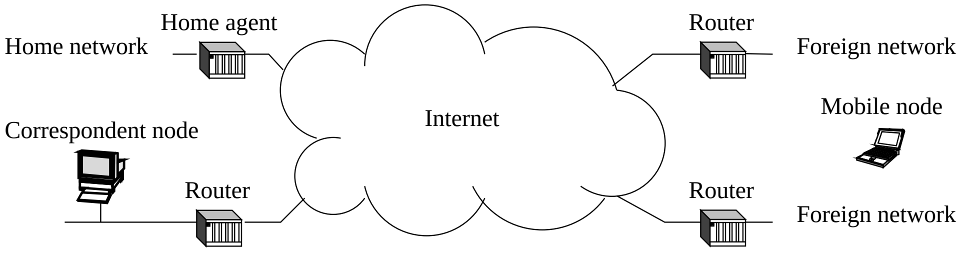


Figure 4.1: Scenario to be emulated by testbed.

In figure 4.1 four networks connected to the Internet are present: Two foreign networks, to which a mobile node can attach, a network for hosting the mobile node’s home agent and a network on which a correspondent node is present.

In this scenario the only situation which cannot be represented is the situation in which the mobile node has three or more foreign networks to attach to. However, if needed, more networks can easily be added.

4.2.2 Composition of testbed

A testbed which can emulate the scenario in figure 4.1 is shown in figure 4.2.

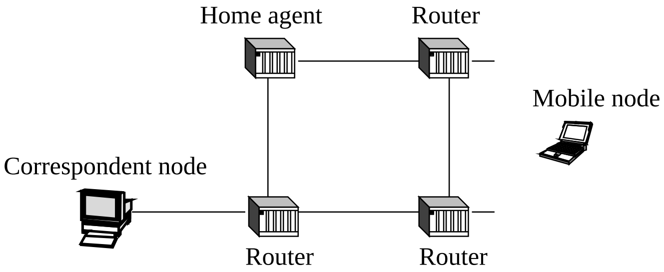


Figure 4.2: A testbed requiring six computers to realize.

The physical connection between the entities shown in figure 4.2 could be Ethernet. Different delays on the links can be emulated by using special software[1]. The testbed shown in figure 4.2 can be established using six computers. In order to reduce the need for hardware we propose a somewhat smaller testbed. This testbed is depicted in figure 4.3.

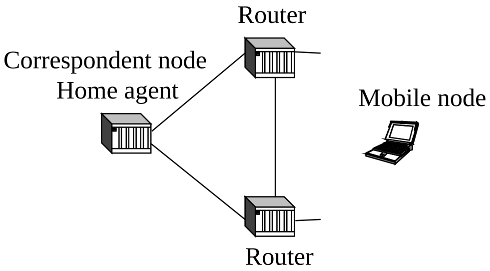


Figure 4.3: A testbed requiring only four computers to realize.

The reduced testbed shown in figure 4.3 differs from the testbed shown in figure 4.2 in that it contains two less networks. This is achieved by moving the home agent to the network at which the correspondent node is located. Furthermore, the software of the correspondent node has been moved to the access router of its network, the router that now also hosts the home agent. Reducing the testbed to contain only four entities of course reduces the degree of compliance with the scenario depicted in figure 4.1. One constraint is introduced by the testbed depicted in figure 4.3. It is not possible to emulate different delays on the routes from the mobile node to the home agent and the correspondent node or between the home agent and the correspondent node. For example this could be needed in an experiment where the effect of *binding updates* arriving later at the home agent than at a correspondent node should be investigated.

---

[1] With FreeBSD for instance, one could use the `dummynet` tool [Dummynet], which can introduce delays at routers, randomly drop packets etc.

This constraint does not affect the handoff latency when the mobile node moves between networks and thus does not affect the experiments to be presented in this chapter. The testbed that has been set up is therefore the reduced testbed. Of course this testbed can later be expanded to contain two additional computers and thus obtaining full compliance with the scenario depicted in figure 4.1.

In figure 4.4 the chosen testbed is shown with configured IPv6 addresses and prefixes.

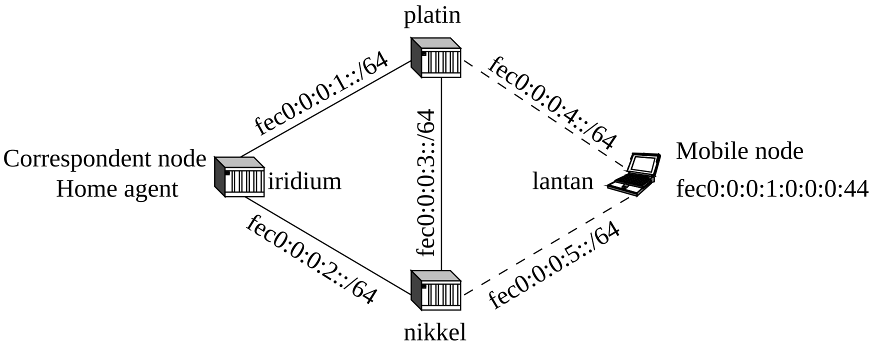


Figure 4.4: Testbed with IPv6 addresses and prefixes.

The testbed consist of four nodes; three routers and one host. The three routers, *iridium*, *platin* and *nikkel*, are assigned an IPv6 prefix for each network device. The mobile node, *lantan*, is manually assigned an IPv6 address at the fec0:0:0:1::/64 network, its home network. When *lantan* is not at its home network, it uses *stateless auto-configuration* (see section A.2) to assume an IPv6 address as its care-of address. The responsibilities of the four nodes are the following:

Iridium, besides being a router, acts as a correspondent node. It runs an application, that corresponds with an application at the mobile node *lantan*. *Iridium* also hosts the home agent.

Lantan is a mobile node, which hosts an application that corresponds with an application hosted at the correspondent node *iridium*.

Platin is a router providing a network to which the mobile node *lantan* can attach.

Nikkel is another router providing a network to which the mobile node *lantan* can attach.

4.2.3 Link media used in testbed

To interconnect the three routers *platin*, *nikkel* and *iridium* we use standard 802.3 10 Mbit/s Ethernet devices. From *platin* to *lantan* and from *nikkel* to *lantan* we have installed both 802.3 10 Mbit/s Ethernet and 802.11b 11 Mbit/s Wireless LAN connections. The Wireless LAN devices used are Silver Cards from Orinoco (Lucent Technologies) [ORiNOCO].

When the testbed is operated, it is either configured to use the Ethernet devices or the Wireless LAN devices to connect from the mobile node to the access routers. This means, that either the Ethernet devices from *platin* to *lantan* and from *nikkel* to *lantan* or the Wireless LAN devices from *platin* to *lantan* and from *nikkel* to *lantan* should be disabled.

The reason for providing both a wired and a wireless link media solution is that the Wireless LAN devices was not available until shortly before deadline. It was therefore decided to initially use Ethernet in order to get the testbed up and running. Having both possibilities also provides the opportunity to compare experimental results obtained using the wired solution with experimental results obtained using the wireless solution. Upon presenting experimental results, we will clearly state whether an experiment has been performed using the Ethernet or the Wireless LAN interfaces.

4.2.4 Software installed in the testbed

All nodes run FreeBSD version 4.1 [FreeBSD]. FreeBSD is a BSD UNIX operating system, that supports several types of computer architectures (Intel, DEC Alpha and PC-98). FreeBSD is chosen for five main reasons:

1. BSD UNIX is the platform where network development happens.
2. FreeBSD is free.
3. FreeBSD comes with a dual networking protocol stack supporting both IPv4 and IPv6.
4. FreeBSD is open source licensed.
5. There exist an Mobile IPv6 support package (KAME) for FreeBSD.

On top of FreeBSD the KAME package [KAME] is installed. The KAME package includes Mobile IPv6 support and IPsec support. The KAME package installed is the weekly snap-release of 25/9-2000. A snap-release is the newest version of the package and may include functionality that is still under development and is not fully tested. The Mobile IPv6 code supplied with KAME is an example of such functionality.

The Mobile IPv6 implementation included in KAME can be configured to use three different handoff strategies. In KAME, a handoff strategy is represented as an **eager mode** setting at the mobile node. The available eager modes are:

Eager mode 0: This mode corresponds to *Lazy Cell Switching* (see section 1.4.3). When eager mode 0 is used, a handoff is not performed until a mobile node has detected that its primary network is unavailable.

Eager mode 1: In this mode a mobile node reacts whenever a new prefix is discovered. After having received a router advertisement with a new prefix, the mobile node probes the default router at its primary network with a router solicitation in order to see if it is still reachable. If reachable, it retains the attachment to the primary network. If not a handoff to the newly discovered network is performed.

Eager mode 2: This mode corresponds to *Eager Cell Switching* (see section 1.4.3). When a new prefix is discovered a handoff to that network is immediately performed.

For a description of how we installed FreeBSD and KAME we refer to appendix B. The details of the configuration of IPv6 and Mobile IPv6 at the four nodes are described in appendix C. Encountered bugs in FreeBSD and KAME are described in appendix D.

4.3 Overview of experiments

As Mobile IPv6 uses the reception of router advertisements to discover new network prefixes, the interval between sending router advertisements can affect the time it takes to discover new networks. In non mobile networks a router advertisement is typically sent from routers once every 7 to 10 minutes [Deering, 1991]. However, in mobile networks it is recommended, that router advertisement should be sent with an interval between 0.5 and 1.5 seconds [Johnson *et al.*, 2000]. As sending many router advertisements increases the load on a network, the performance regarding loss of packets of Mobile IPv6 versus the interval between sending router advertisements is investigated. Similarly, the lifetime of network prefixes specified in the router advertisements affects the time that it takes a mobile node to discover that a router has become unreachable. The conducted experiments aim to reveal whether these parameter settings affect handoff performance as suggested by the mathematical models derived in chapter 3. The following experiments are conducted:

Default settings using wired connections: In this experiment the router advertisement interval and network prefix lifetime is set as recommended in [Johnson *et al.*, 2000]. This means an interval randomly chosen between 0.5 and 1.5 seconds and a lifetime of 4 seconds. The purpose of this experiment is to reveal handoff performance using the recommended settings.

Default settings using wireless connections: This experiment is the same as the **default settings using wired connections** experiment, but using wireless connections between access routers and the mobile node instead. The purpose is to reveal whether the introduction of wireless links affects handoff performance.

Handoff latency as a function of network prefix lifetime using wired connections: In this experiment the handoff latency is measured for different network prefix lifetimes, but with a fixed range for the intervals between sending router advertisements. The purpose of this experiment is to investigate how the lifetime of sent router advertisements affects the handoff performance.

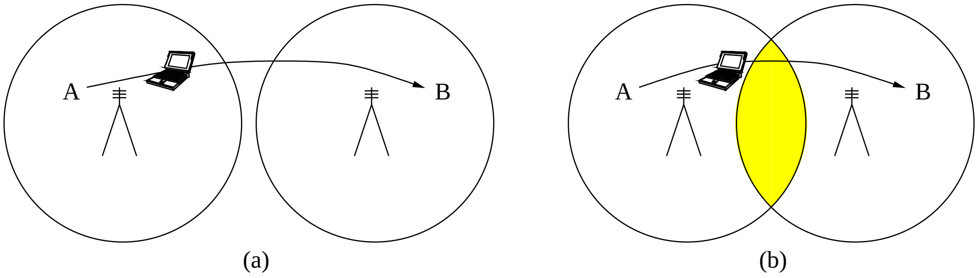
Handoff latency as a function of router advertisement interval using wired connections: In this experiment the handoff performance is measured for different router advertisements intervals, but with an identical network prefix lifetime. The purpose of this experiment is to investigate how the interval between sending router advertisements affects the handoff performance.

New proposed default settings using wired connections: In chapter 3 we proposed a new set of default settings which should improve handoff performance without increasing network load. We proposed that the interval between broadcasting router advertisements should be in the range between 0.9 and 1.1 seconds and that the lifetime of network prefixes should be 2 seconds. The objective of this experiment is to verify whether this theoretical performance gain can be realized in an actual setting.

All experiments have been conducted using both *Eager Cell Switching* (eager mode 2) and *Lazy Cell Switching* (eager mode 0). The overall purpose of performing these five experiments, is to establish the relationship between the mathematical models and practical experimental results. Furthermore, the results for eager mode 0 and eager mode 2 provides a baseline that more advanced handoff initiation strategies can be evaluated against. No experiments have been conducted for eager mode 1, as this eager mode can be seen as a combination of eager mode 0 and eager mode 2 and its behavior is predictable from the results for these two eager modes.

4.3.1 Scenarios modeled in experiments

The experiments described in section 4.3 are conducted using both *Lazy Cell Switching* and *Eager Cell Switching*. An experiment is performed using one of two different setups emulating the two scenarios depicted in figure 4.5.



(a) (b)

Figure 4.5: a) No overlap scenario. b) Overlap scenario.

The first setup, the **no overlap setup**, models the scenario in which a mobile node gets out of range of a base station serving the primary network before it gets within range of a base station serving a new network. The no overlap setup corresponds to the scenario depicted in figure 4.5a. The scenario is characterized by the fact, that for a certain period of time no point of attachment can exist as no access point is reachable. When this period approaches zero (as when exactly one base station is always reachable) this scenario corresponds to the one for which mathematical models was derived in chapter 3.

The second setup, the **overlap setup**, models the scenario in which a mobile node is moved from the range of one base station serving the current primary network into the range of a new base station serving a new network. In the overlap setup, an overlap zone where the mobile node can reach both networks exists. The overlap setup corresponds to the scenario depicted in figure 4.5b. The scenario is characterized by the fact, that at least one base station is always reachable and thus in theory packet loss could be avoided when moving out of the range of the first base station.

For all the conducted experiments the result wanted is the handoff latency. Recall that the handoff latency is defined as the time between receiving the last packet from the previous primary network until receiving the first packet from the new network.

The handoff latency has been determined using the following method:

- UDP packets are send from the correspondent node to the mobile node. Each packet contains a timestamp and a sequence number. The UDP packets are send with a random interval between 95 ms and 105 ms. The interval is randomized to make sure the network does not adjust itself to any particular sending frequency.

- The UDP packets are received at the mobile node. A timestamp is stored upon reception of a packet. An entry in a log file is generated, containing the following parameters:
  - The value of the sequence number in the received packet.
  - The time (denoted transmission delay) in μs between the time of transmitting the previous packet and this packet from the corresponding node. This value is computed by subtracting the timestamp in the previously received packet from the timestamp in this packet.
  - The time (denoted reception delay) in μs between the receiving of the previous packet and this packet. This value is computed by subtracting the timestamp stored when receiving the previous packet from the timestamp stored when receiving this packet.
- A handoff is registered as packets missing in the log file. If a handoff is performed without losing packets it will therefore not be registered.

A sample from such a log file is presented below:

```
Seq. no.    Transmission delay.  Reception delay.
235         99994us              100006us
236         102978us             102969us
237         96980us              97005us
275         3876413us            3876418us
276         96988us              96882us
277         101977us             101922us
278         105985us             105990us
```

This sample represents the situation where a handoff has been performed in which 37 (275-237-1) packets are lost. The handoff latency in this case was approximately 3.9 seconds.

By investigating log files, it has been revealed that the average time between sending packets is 101.5 ms. The correspondent node sleeps between 95 ms and 105 ms between sending packets, yielding an average sleep of 100 ms. The kernel at the correspondent node has been configured to reschedule once every 1 ms. After a sleep has timed out, thus an average 0.5 ms elapses before the thread is allowed to continue. The remaining (101.5-100-0.5) 1.0 ms is expected to be used at the correspondent node for processing. This processing includes sending the packet. In the following sections the two setups are described.

The interval of 95 ms to 105 ms was chosen to avoid too many UDP packets being sent. Due to a memory leakage in the KAME Mobile IPv6 software only a limited number of packets can be sent from a correspondent node before it crashes. In experiments presented in this chapter we were able to perform 300 to 400 handoffs in sequence before the correspondent node crashed.

The no overlap setup: At most one available point of attachment

The no overlap scenario from figure 4.5a is realized by first disallowing forwarding of IP packets from the access router through which the mobile node is currently attached and then immediately allowing forwarding of IP packets through the other access router. This situation is depicted in figure 4.6.

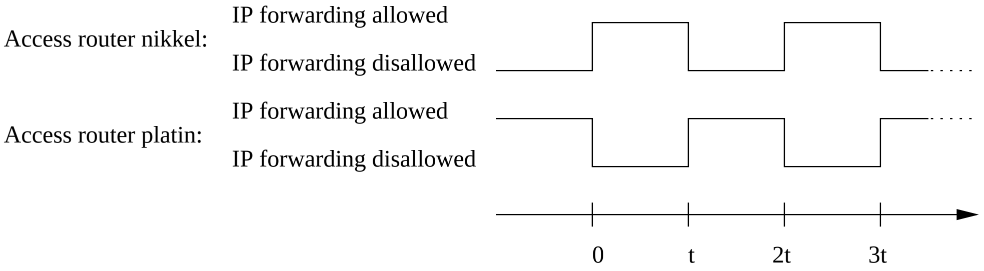


Figure 4.6: Realizing the no overlap scenario by allowing and disallowing forwarding of IP packets at the access routers towards the mobile node.

The allowing and disallowing of IP forwarding is done using a firewall which for IPv6 in FreeBSD is controllable by the `ip6fw` command. At *nikkel* we run a small program which controls the IPv6 firewall at both *nikkel* and *platin*. With regular intervals, the program sends an order to the IPv6 firewall at *platin* to either allow or disallow IP forwarding and immediately upon sending this order instructs the IPv6 firewall at *nikkel* to do the opposite.

The time when no access router is reachable is the time in which IP forwarding is disallowed at both access routers. To determine the overhead of allowing and disallowing IP forwarding a simple experiment has been conducted: UDP packets are sent every 2 ms from a correspondent node to the mobile node. At an access router IP forwarding was disabled and then immediately enabled. The period in which no packets were received at the mobile node was an average 12 ms. From this experiment we conclude, that it takes around 12 ms to allow IP forwarding Another overhead lies in the propagation time of orders sent between the two access routers. This time is in the range 1 ms to 2 ms depending on network load. As the handoff latencies predicted by the mathematical models often are in the range of several seconds, these overheads are considered insignificant.

Using the no overlap setup, the interval from when IP forwarding is allowed to when it is disallowed at one access router, is always chosen to be greater than the time it takes to perform the handoff. In figure 4.6 this corresponds to that a handoff always take below t time units to perform[2]. This choice ensures, that the obtained results reflects the handoff latency and not the interval between which IP forwarding is allowed and disallowed.

The overlap setup: At least one available point of attachment

The overlap scenario is realized by allowing and disallowing IP forwarding in the following way:

1. Allow IP forwarding at *nikkel*. Then wait t time units.
2. Disallow IP forwarding at *platin*. Then wait t time units.
3. Allow IP forwarding at *platin*. Then wait t time units.
4. Disallow IP forwarding at *nikkel*. Then wait t time units.

---

[2] We use $t = 10$ seconds in all experiments.

5. Goto 1.

These steps are repeated during an entire experiment. This algorithm yields the situation depicted in figure 4.7.

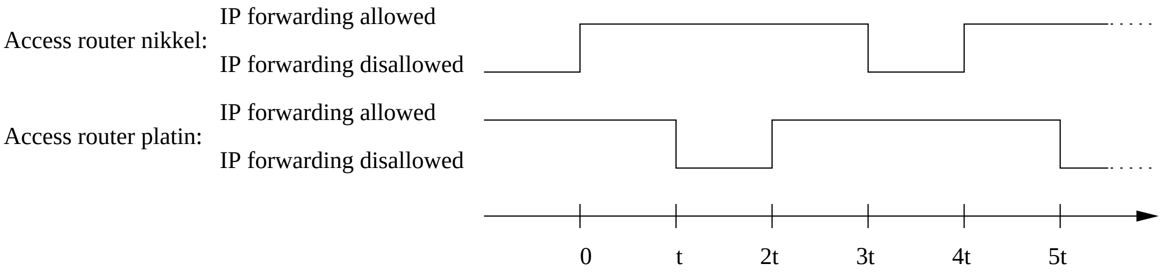


Figure 4.7: Realizing the overlap scenario by allowing and disallowing forwarding of IP packets at the access routers towards the mobile node.

The way of allowing and disallowing forwarding of IP packets depicted in figure 4.7, ensures that at least one point of attachment is always available to the mobile node.

Using the overlap setup, the interval between allowing and disallowing IP forwarding is always chosen to be more than three times the time it takes to perform the handoff. In figure 4.7 this corresponds to that a handoff always take below t time units to perform[3]. This choice ensures that if the allowing or disallowing of IP forwarding at an access routers triggers a handoff, that handoff is completed before the other access router is ordered to allow or disallow IP forwarding.

4.3.2 Evaluating and presenting results

For all experiments, we plot the experimentally obtained handoff latency together with the handoff latency predicted by the appropriate mathematical model. This allows for an easy comparison between theory and reality. The handoff latency in experiments are calculated as the number of lost packets times 0.1015 seconds, which is the average time between sending packets from the correspondent node.

When an experiment concerns a particular router configuration we plot the measured handoff latencies as a histogram. Such a histogram is also denoted a frequency distribution. An example of such a plot is the plot in figure 4.8.

In figure 4.8 the handoff latency is along the x-axis. We use two different y-axes, one for the continuous probability distribution of the theoretical handoff latency and one for the discrete frequency distribution of the measured handoff latency. The y-axis at the left is for the theoretical probability distribution and reflects, that the area under the probability curve must be 1. The y-axis at the right is for the empirical frequency distribution and reflects that the sum of all columns must be 1.

In some experiments we try different values for one variable (the independent variable) while keeping another variable fixed. An example is investigating the effect of varying the interval

---

[3] We use $t = 10$ in all experiments.

between broadcasting unsolicited router advertisements and keeping the lifetime of network prefixes fixed. In these experiments we have the average handoff latency along the y-axis and the value of the independent variable along the x-axis. We also plot a function for the theoretical handoff latency (these functions are directly obtainable from the mathematical models). To see how well the experimental results conform to theory, we fit a curve to the experimentally obtained average values. An example of such a plot can be found in figure 4.16.

4.3.3 Calculating predicted handoff latency

Recall from chapter 3 that the theoretical models need various protocol settings as inputs. In this section we motivate how we have configured these settings when using the mathematical models to produce theoretical results to compare with empirical results.

The settings for interval between broadcasting router advertisements have of course always been configured as they were on the access routers in each experiment. Similarly, the lifetime of network prefixes has always been configured to an identical value of that used in an experiment.

In section 3.4 we suggested, that *Lazy Cell Switching* might probe the default router at the primary network if the lifetime of that network expires. By investigating the source code for the KAME Mobile IPv6 implementation and the FreeBSD 4.1 IPv6 neighbor discovery implementation it was discovered, that when the lifetime of a network prefix expires in the default router list the default router at that network is **not** probed. Instead, the network is instantly declared unavailable. This implies that we should configure the probing time to zero in the mathematical models.

In the mathematical model for *Lazy Cell Switching* we assume, that a handoff to a new network (when the probetime is zero) is initiated immediately when the lifetime of the primary network expires. In practice the expire time of a network prefix is calculated in the following way.

Upon receiving a router advertisement at the mobile node, the lifetime of the advertised network prefix is extracted. This lifetime is specified in seconds. The expiration time for the received network prefix is calculated as the current system time (in seconds) plus the lifetime received in the router advertisement. A prefix is then expired when the current system time (in seconds) becomes larger than the expiration time for that particular prefix. This will happen when router advertisements advertising a particular network prefix are not received for a period of time corresponding to the lifetime of the network prefix.

The following is an example of calculating the expiration time of a network prefix in the FreeBSD 4.1 implementation of IPv6 neighbor discovery: At time 10h:20m:30.25s a router advertisement is received advertising network prefix X with a lifetime of 4 seconds. The expiration time of prefix X is then set to 10h:20m:34s. If no further router advertisement advertising network prefix X is received, the prefix will be declared unreachable when the system time (in a granularity of seconds) surpasses the expiration time. In this example that would happen at system time 10h:20m:35.00s. In this case, a period of 4.75 seconds will have elapsed before the prefix is expired and not 4 seconds as specified in the received router advertisement. This specific implementation causes the actual expiration time of a network prefix to be 0 to 1 second higher than that specified in router advertisements.

In order to compare empirically collected data for handoff latency with that expected by the theoretical model, we must compensate in the theoretical model for this extra expiration time.

By assuming, that the extra added expiration time is uniformly distributed within the interval [0,1] seconds, we can compensate the theoretical model for *Lazy Cell Switching* by adding a random delay in the interval [0,1]. All theoretical plots for handoff latency presented in this chapter thus uses $Q_{\text{min}} = 0$ and $Q_{\text{max}} = 1$.

4.4 Handoff latency using default settings

Using default settings for Mobile IPv6 a router sends router advertisements at a random interval between 0.5 and 1.5 seconds. The lifetime of the network prefixes in broadcasted router advertisements should be set to three times the maximum interval [Deering, 1991]. As only integer values can be specified, in this experiment it has been set to 4 seconds.

4.4.1 No overlap setup: Eager Cell Switching

The experiments performed using the no overlap setup are particularly interesting, as the mathematical models derived in chapter 3 predicts the expected results from these experiment. By comparing the expected and the actually obtained results we can get a bearing as to what extend the mathematical models comply with reality.

In figure 4.8 both the theoretical probability distribution and the actual values for handoff latency measured in an experiment with *Eager Cell Switching* using Ethernet links are plotted.

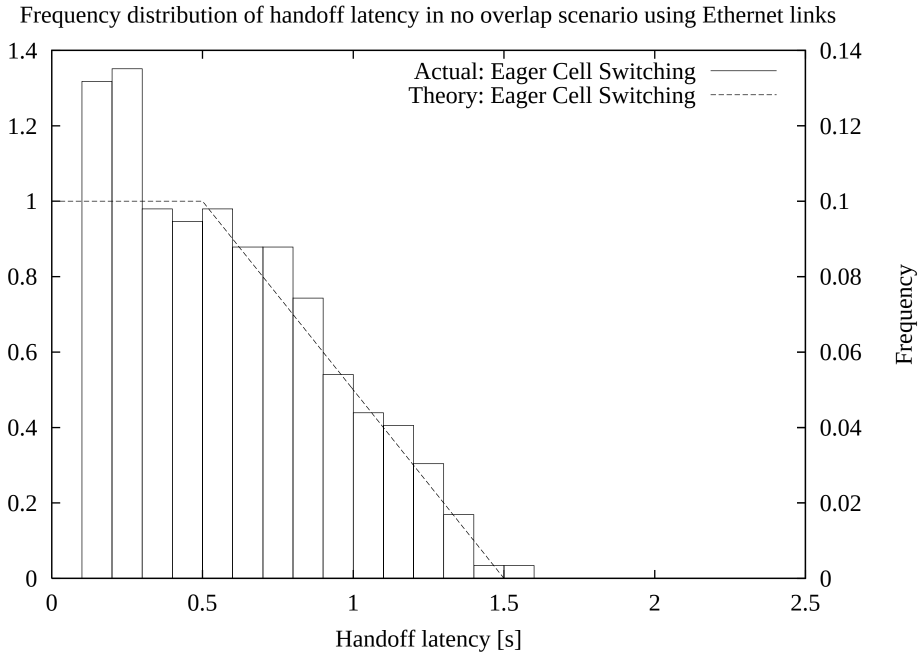


Figure 4.8: Frequency distribution for handoff latency using *Eager Cell Switching*, Ethernet links and default router configuration in the no overlap scenario.

As can be seen from figure 4.8 the frequency distribution obtained in the experiment by using

Eager Cell Switching apparently follows the probability distribution suggested by the theoretical model. The reason that there are no handoff latencies in the 0 to 0.1 seconds interval is that we cannot measure handoff latencies lower than 0.1 seconds due to the distance between sent UDP packets.

In figure 4.9 the actual values for handoff latency measured in an experiment with wireless links using *Eager Cell Switching* is plotted.

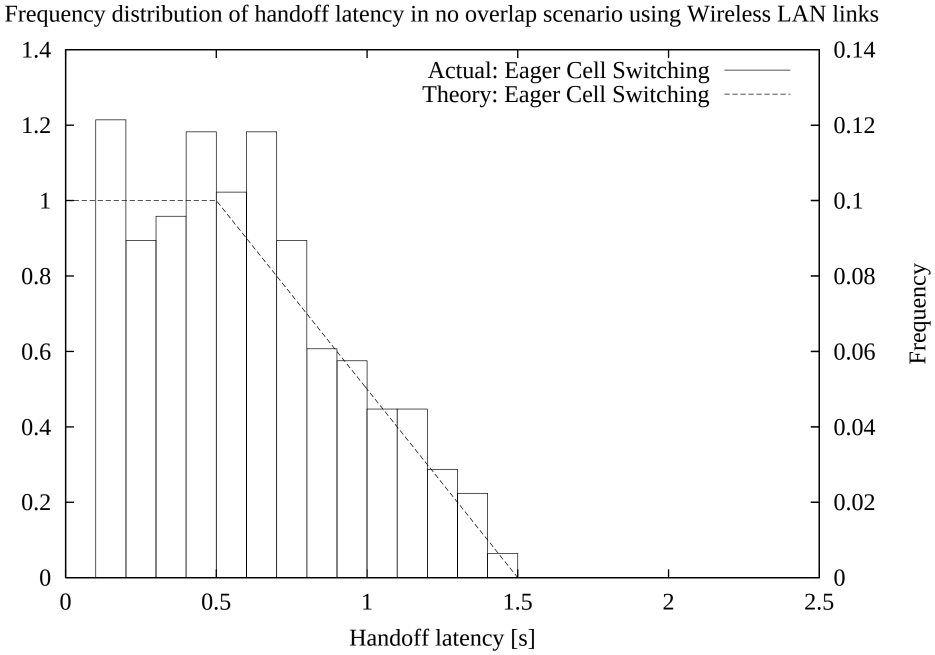


Figure 4.9: Frequency distribution for handoff latency using *Eager Cell Switching*, Wireless LAN links and default router configuration in the no overlap scenario.

As can be seen from figure 4.9 the measured handoff latencies using Wireless LAN apparently follow the theoretical probability distribution. The difference between the measured handoff latencies plotted for Ethernet (see figure 4.8) and the measured handoff latencies for Wireless LAN (see figure 4.9) are expected to be due to the relative low number of samples. By performing simulations with the fast simulator introduced in section 3.6, we experienced that at least 5000 samples were necessary in order to generate almost identical plots in two consecutive experiments. This suggest, that a much larger data set is needed in order to obtain more precise results.

From figure 4.8 and from figure 4.9 it can be seen that the results obtained using wired and wireless link media are almost identical. This implies that no significant overhead is introduced by the use of Wireless LAN.

4.4.2 No overlap setup: Lazy Cell Switching

In figure 4.10 both the theoretical probability distribution and the actual values for handoff latency measured in an experiment with the *Lazy Cell Switching* handoff initiation algorithm using Ethernet links are plotted.

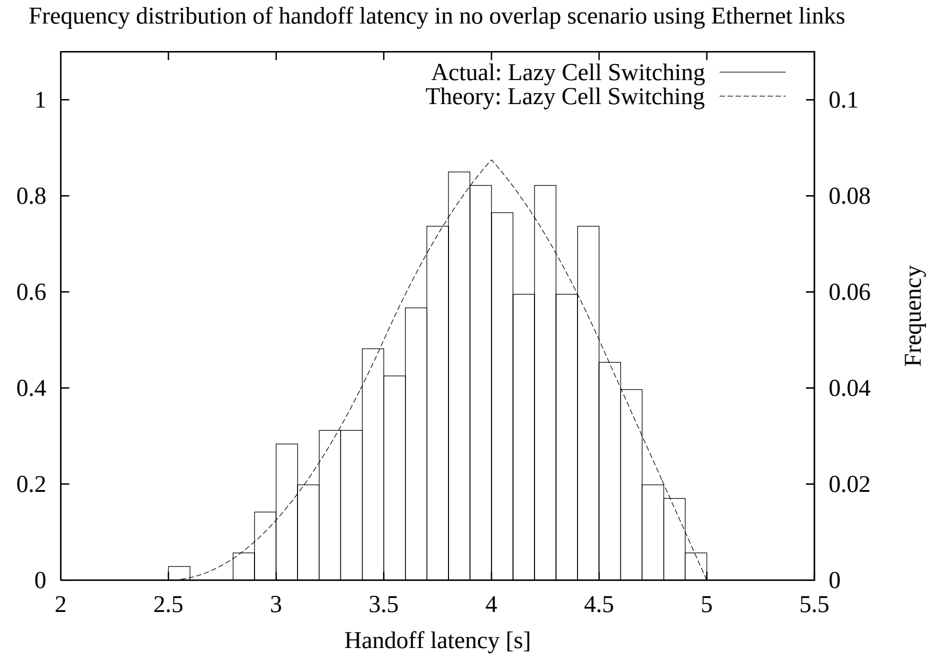


Figure 4.10: Frequency distribution for handoff latency using *Lazy Cell Switching*, Ethernet links and default router configuration in the no overlap scenario.

From figure 4.10 it can be seen, that the frequency distribution of the handoff latencies measured in the experiment apparently follows the theoretical probability distribution.

In figure 4.11 the same experiment has been performed, but using Wireless LAN as the link media. As for the experiment using Ethernet as link media, the frequency distribution of the measured handoff latencies apparently follows the probability distribution.

From figure 4.10 and from figure 4.11 it can be observed, that the experiment resulted in almost identical results using the wired and the wireless link media. This implies, that there is no significant overhead using the wireless link media. This was quite as expected, as the experiment was performed with the mobile node well within range of the both base stations.

4.4.3 Summary of results for the no overlap setup

In figure 4.12 a summary of the expected and obtained results for both the *Lazy Cell Switching* and the *Eager Cell Switching* handoff initiation strategy is presented.

The values in figure 4.12 reflects very well that predicted by the theoretical models. This is true for both the *Eager Cell Switching* and the *Lazy Cell Switching* handoff initiation strategies. Inconsistencies are mainly due to the method of measuring. A handoff latency of 1.5 seconds will be measured as ($\lfloor \frac{1.5s}{0.1015s} \rfloor \cdot 0.1015s$) 1.42 seconds. It is seen, that no significant propagation or processing delay are present in the Mobile IPv6 testbed. We therefore conclude, that the empirically obtained results were as predicted by the mathematical models.

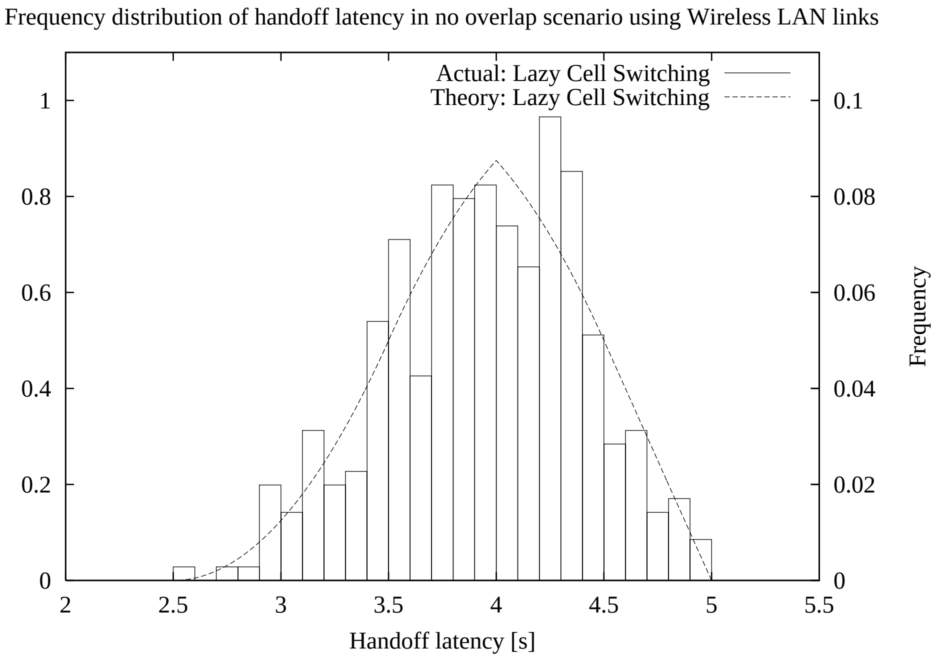


Figure 4.11: Frequency distribution for handoff latency using *Lazy Cell Switching*, Wireless LAN links and default router configuration in the no overlap scenario.

Handoff strategy	Router settings	L [s] (Theory)			Latency [s] (Ethernet)			Latency [s] (Wireless)		
		Avg	**Min**	**Max**	**Avg**	**Min**	**Max**	**Avg**	**Min**	**Max**
Eager	Default	0.54	0	1.5	0.54	0.10	1.52	0.57	0.10	1.42
Lazy	Default	3.95	2.5	5.0	3.97	2.54	4.97	4.06	2.64	4.97

Figure 4.12: Summary of expected and actual results for the no overlap setup.

4.4.4 Overlap setup: Eager Cell Switching

This experiment results in no packets being lost. The *Eager Cell Switching* handoff initiation strategy initiates a handoff as soon as a router advertisement from a new network is received. But because in the overlap setup the primary network is reachable during the handoff no packets are lost. This was confirmed both by using wired and wireless connections.

4.4.5 Overlap setup: Lazy Cell Switching

In this experiment we use the *Lazy Cell Switching* in the overlap setup. The experiment is expected to produce a similar result as the experiment performed in the no overlap setup. This is because the *Lazy Cell Switching* strategy does not initiate a handoff before the prefix lifetime of the primary network expires. Therefore *Lazy Cell Switching* is not able to take advantage of the period in which two networks are reachable as in the overlap setup.

In figure 4.13 the frequency distribution for handoff latency measured in the experiment using the *Lazy Cell Switching* handoff initiation algorithm with wired links is plotted.

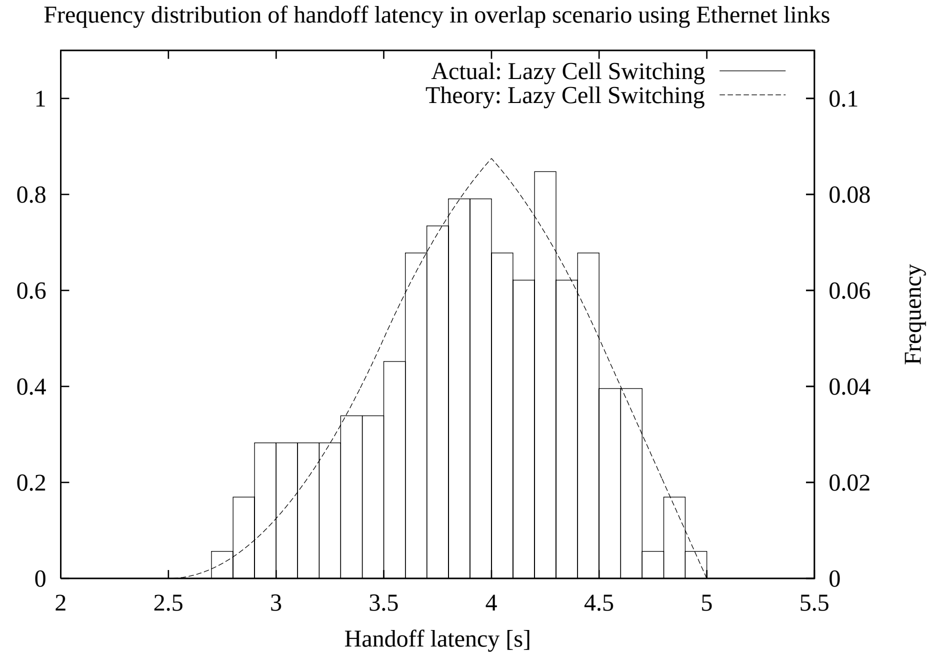


Figure 4.13: Frequency distribution for handoff latency using *Lazy Cell Switching*, Ethernet links and default router configuration in the overlap setup.

From figure 4.13 we observe, that the frequency distribution apparently conform with the probability distribution of the mathematical model.

In figure 4.14 the frequency distribution for handoff latency measured in a similar experiment, but using Wireless LAN, is plotted.

From figure 4.14 we can see, that the results obtained using the wireless link media are very

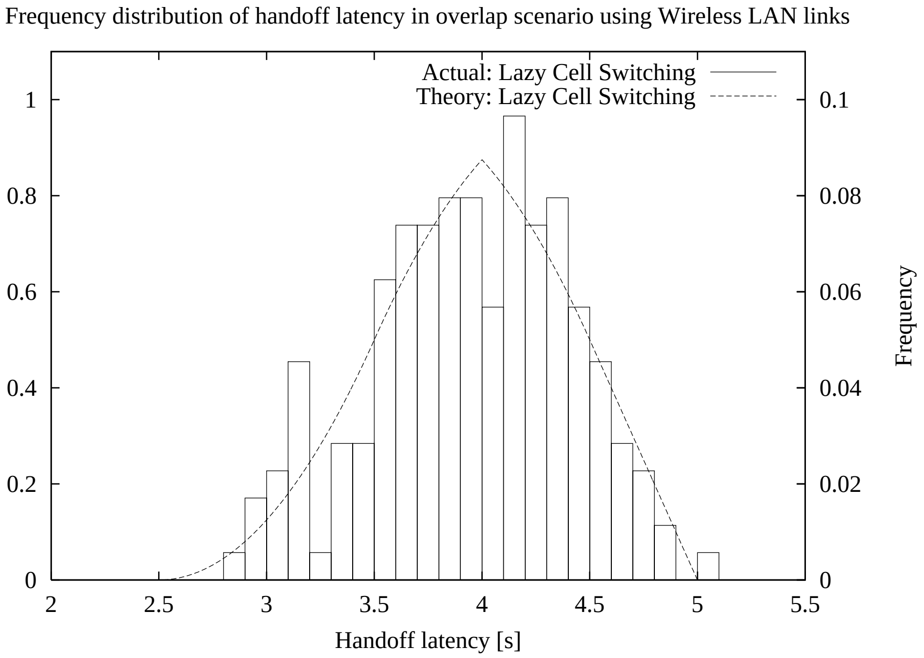


Figure 4.14: Frequency distribution for handoff latency using *Lazy Cell Switching* in the overlap setup using Wireless LAN.

similar to the results obtained when using a wired link media. Accordingly, the frequency distribution apparently conform with the probability distribution of the mathematical model.

From figure 4.13 and from figure 4.14 we observe, that the results obtained using the wired and the wireless link media are very similar.

4.4.6 Summary of results for the overlap setup

In figure 4.15 a summary of results obtained in the overlap setup is presented.

Handoff	**Router**	L **[s] (Theory)**			**Latency [s] (Ethernet)**			**Latency [s] (Wireless)**		
strategy	**settings**	**Avg**	**Min**	**Max**	**Avg**	**Min**	**Max**	**Avg**	**Min**	**Max**
Eager	Default	0	0	0	0	0	0	0	0	0
Lazy	Default	3.95	2.5	5.0	3.92	2.74	4.97	3.96	2.84	5.08

Figure 4.15: Summary of expected and actual results for the overlap setup.

The values for handoff latency for *Lazy Cell Switching* in figure 4.15 are very similar to the values obtained using the no overlap setup as presented in figure 4.12. As the *Lazy Cell Switching* handoff initiation strategy does not take advantage of a mobile node being able to reach more networks at the same time, these results were expected to be similar. For *Eager Cell Switching* we experienced no packet loss which was as expected.

4.5 Handoff latency as a function of router advertisement interval

In these experiments we measure the handoff latency for different ranges of intervals between sending router advertisements with a fixed network prefix lifetime. The purpose is to confirm the correctness of the theoretical models by comparing actual results to expected results, why the experiments are only conducted for the no overlap setup.

The lifetime is always kept at 5.0 seconds and the experiments are conducted with ranges for intervals between router advertisements configured between [0.5,1.5], [1.5,2.5], [2.5,3.5] and [3.5,4.5] seconds respectively.

4.5.1 Eager Cell Switching

Recall from formula 3.7, that the average handoff latency using the *Eager Cell Switching* handoff initiation strategy is given by

$$\overline{L}_{\text{eager}} \quad = \quad \frac{{R_{\text{max}}}^3 - {R_{\text{min}}}^3}{3({R_{\text{max}}}^2 - {R_{\text{min}}}^2)} \tag{4.1}$$

In figure 4.16 we have plotted the average handoff latency as a function of the router advertisement interval for both the theoretical model and for the actual experiments.

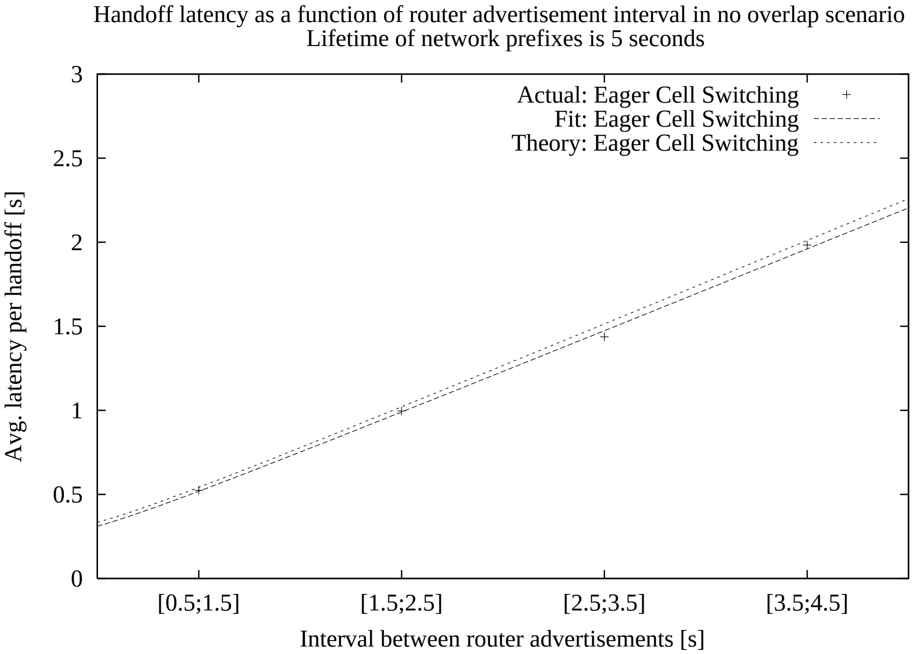


Figure 4.16: Handoff latency as a function of router advertisement interval for *Eager Cell Switching* in the no overlap setup.

As can be observed from figure 4.16 the obtained average handoff latencies in these experiments are in excellent conformance with those predicted by the theoretical model for the *Eager Cell Switching* handoff initiation algorithm.

4.5.2 Lazy Cell Switching

Recall from formula 3.16, that the average handoff latency using the *Lazy Cell Switching* handoff initiation strategy is given by

$$\overline{L}_{\text{lazy}} = \frac{1}{2}(Q_{\max} - Q_{\min}) + T_{\text{l}} - \frac{{R_{\max}}^3 - {R_{\min}}^3}{3({R_{\max}}^2 - {R_{\min}}^2)} \tag{4.2}$$

In figure 4.17 we have plotted the average handoff latency as a function of the router advertisement interval for both the theoretical model and for the actual experiments.

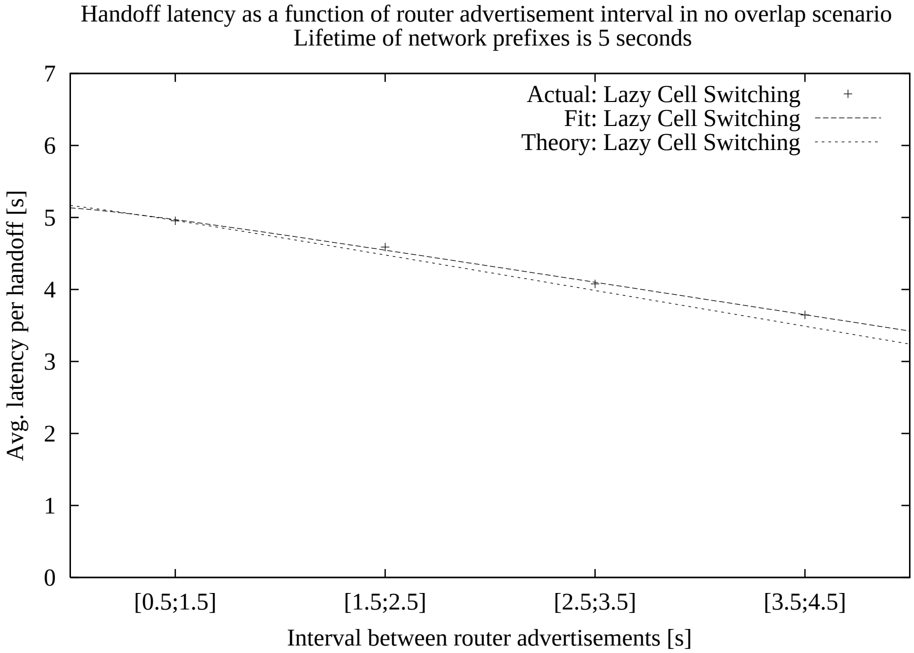


Figure 4.17: Handoff latency as a function of router advertisement interval for *Lazy Cell Switching* in the no overlap setup.

As can be observed from figure 4.17 the obtained average handoff latencies in these experiments are in excellent conformance with those predicted by the theoretical model for the *Lazy Cell Switching* handoff initiation algorithm.

4.5.3 Summary of results

For both the *Lazy Cell Switching* handoff initiation strategy and the *Eager Cell Switching* handoff initiation strategy we observed almost exactly the results which was predicted by the mathematical models. This leads us to conclude, that regarding the average handoff latency both theoretical models reflects very well handoff latency experienced when using the KAME Mobile IPv6 implementation.

4.6 Handoff latency as a function of network prefix lifetime

In these experiments we measure the handoff latency for different lifetimes of network prefixes sent with identical router advertisement intervals. The purpose is to confirm the correctness of the theoretical model by comparing actual results to expected results, why the experiments are only conducted for the no overlap setup.

The intervals between broadcasting router advertisements are always kept between [0.5,1.5] seconds. This interval is chosen as it is the default configuration. The experiment is conducted with values for lifetime of network prefixes configured at 2, 3, 4 and 5 seconds.

4.6.1 Eager Cell Switching

In figure 4.18 we have plotted the average handoff latency as a function of the router advertisement interval for both the theoretical model and for the actual experiments.

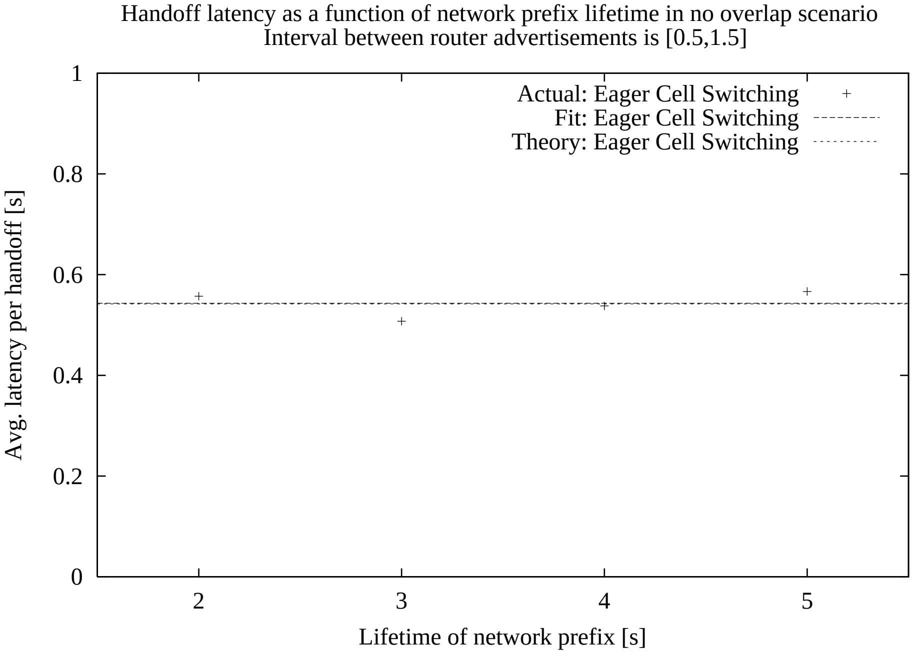


Figure 4.18: Handoff latency as a function of router advertisement lifetime for *Eager Cell Switching* in the no overlap setup.

From figure 4.18 we observe that the obtained results are in excellent conformance to those predicted by the mathematical model for the *Eager Cell Switching* handoff initiation strategy. We note, that in these experiments *Eager Cell Switching* is not influenced by the lifetime of network prefixes.

4.6.2 Lazy Cell Switching

In figure 4.19 we have plotted the average handoff latency as a function of the router advertisement interval for both the theoretical model and for the actual experiments.

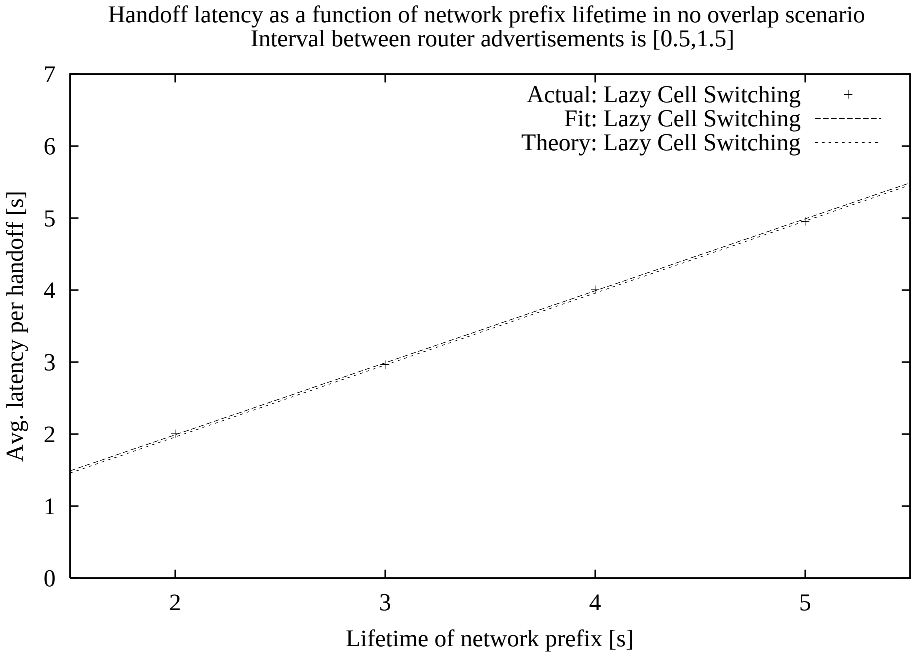


Figure 4.19: Handoff latency as a function of router advertisement lifetime for *Lazy Cell Switching* in the no overlap setup.

From figure 4.19 we observe that the obtained results conform completely to those predicted by the mathematical model for the *Lazy Cell Switching* handoff initiation strategy.

4.6.3 Summary of results

For both the *Lazy Cell Switching* handoff initiation strategy and the *Eager Cell Switching* handoff initiation strategy we observed almost exactly the results which was predicted by the mathematical models. This leads us to conclude, that regarding the average handoff latency both theoretical models reflects very well handoff latency experienced when using the KAME Mobile IPv6 implementation.

4.7 Handoff latency using new proposed settings

In chapter 3 we recommended a set of optimized settings for access routers supposedly yielding better handoff performance than the default settings without increasing network load. In this section we present experimental results using these optimized settings.

Recall that we proposed a new set of default settings at $R_{\min}$ = 0.9, $R_{\max}$ = 1.1 and lifetime

T_1 = 2 seconds. According to the theoretical model this would result in an improvement in the average handoff latency from 0.54 seconds to 0.50 seconds (a 7% reduction) using *Eager Cell Switching*. Furthermore, the worst case handoff latency will be reduced from 1.5 seconds to 1.1 seconds (a 27% reduction). Similarly, using *Lazy Cell Switching* we expect the average handoff latency to drop from 5.04 seconds to 2.00 seconds (a 60% reduction) when the expiration delay of prefixes in the neighbor discovery lists is between 0 and 1 second.

As determined in section 4.4, the *Lazy Cell Switching* handoff initiation algorithm yields the same handoff latency whether applied in the overlap or the no overlap scenario. The *Eager Cell Switching* handoff algorithm yields zero packet loss in the overlap scenario. For these reasons we chose to perform the experiment using the no overlap setup.

4.7.1 Eager Cell Switching

In figure 4.20 the frequency distribution for handoff latency measured in an experiment with our new proposed settings using *Eager Cell Switching* is plotted.

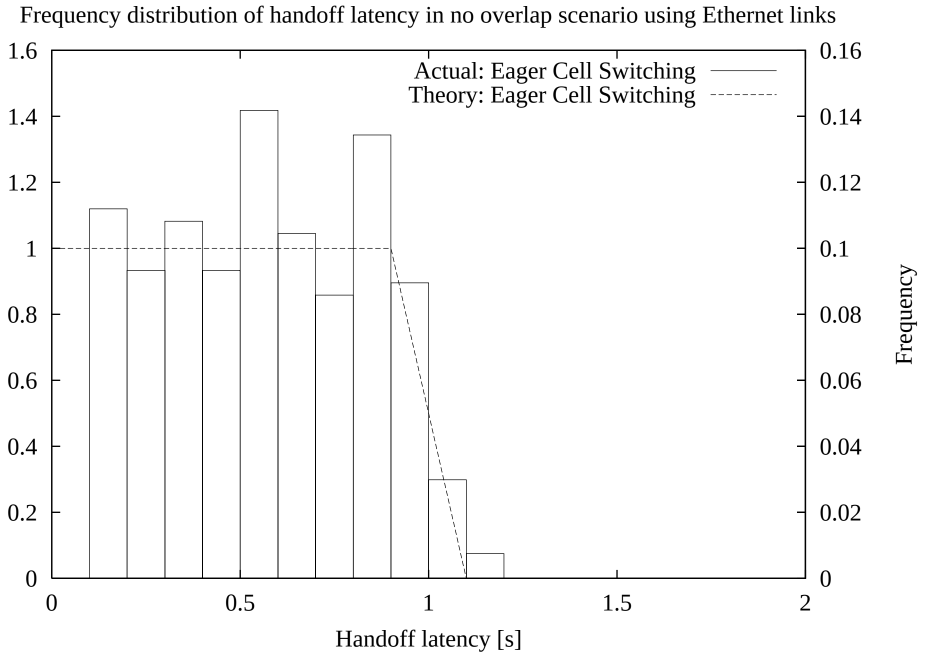


Figure 4.20: Frequency distribution for handoff latency using *Eager Cell Switching*, Ethernet links and proposed router configuration in the no overlap scenario.

From figure 4.20 we observe, that the frequency distribution for handoff latency using our proposed settings with the *Eager Cell Switching* handoff initiation algorithm apparently conforms to that predicted by the mathematical model.

4.7.2 Lazy Cell Switching

In figure 4.21 the frequency distribution for handoff latency measured in an experiment with our new proposed settings using *Lazy Cell Switching* is plotted.

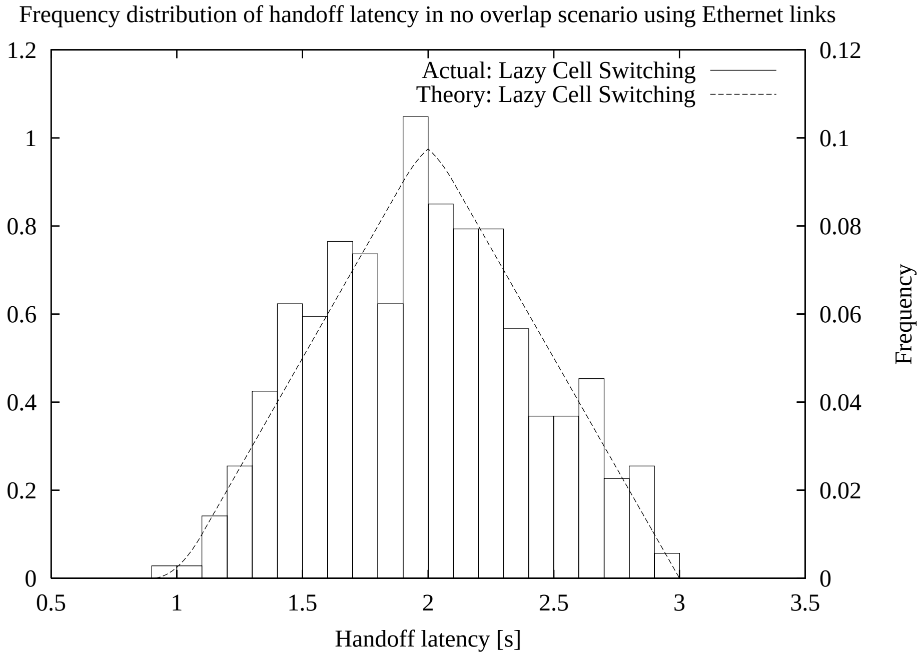


Figure 4.21: Frequency distribution for handoff latency using *Lazy Cell Switching*, Ethernet links and proposed router configuration in the no overlap scenario.

From figure 4.21 we observe, that the frequency distribution for handoff latency using our proposed settings with the *Lazy Cell Switching* handoff initiation algorithm apparently conforms to that predicted by the mathematical model.

4.7.3 Summary of results

The frequency distributions obtained for both the *Lazy Cell Switching* handoff initiation algorithm and the *Eager Cell Switching* handoff initiation algorithm appeared to be in conformance with that predicted by the two mathematical models. This implies that we indeed have been able to transfer the theoretically predicted improvement in handoff latency to an actual setting in the testbed. However, as for the frequency distributions we are unable to make firm conclusions whether they conform in every way to that predicted by the theoretical models. Apparently that is the case, as we have not been able to find any configuration for which the predicted probability distribution did not closely match the obtained frequency distribution. However, the number of performed handoffs in each experiment is only between 300 and 400 which is not enough samples to state a more firm conclusion. In figure 4.22 we have shown a summary of the results obtained using our proposed settings.

As can be seen from figure 4.22, as the average handoff latency is concerned, the results obtained from experiments are in conformance with that predicted by the mathematical models.

Handoff	**Router settings**			L **[s] (Theory)**			**Latency [s] (Ethernet)**		
strategy	R_{min}	R_{max}	T_{l}	**Avg**	**Min**	**Max**	**Avg**	**Min**	**Max**
Eager	0.9	1.1	2	0.5	0	1.1	0.53	0.10	1.12
Lazy	0.9	1.1	2	2.0	0.9	3.0	1.96	0.91	2.94

Figure 4.22: Summary of expected and actual results using our new proposed settings.

4.8 Conclusions drawn from empirical studies

In this section we present the main results obtained from our empirical study of the KAME Mobile IPv6 implementation. Recall that the main purpose of the empirical study was to investigate whether the handoff latency predicted by our proposed mathematical models would reflect the handoff latency observed in an actual network.

The main results obtained from the empirical studies are the following:

- The probability distributions predicted by the model for the *Eager Cell Switching* handoff initiation strategy were found to be in good conformance with the frequency distributions obtained in experiments. No results indicated that the probability distributions predicted were incorrect. Due to a memory leakage in the KAME implementation it was only possible to obtain the handoff latency for 300 to 400 handoffs before the correspondent node crashes. In order to make more firm conclusions regarding the predicted probability distributions a larger number of handoffs must be performed.

- The probability distributions predicted by the model for the *Lazy Cell Switching* handoff initiation strategy were found to be in good conformance with the frequency distributions obtained in experiments. No results indicated that the probability distributions predicted were incorrect. The results are subject to the same lack of samples as for the *Eager Cell Switching* handoff initiation strategy.

- The average handoff latency predicted by the mathematical model for *Eager Cell Switching* and by the mathematical model for *lazy Cell Switching* was in all experiments very close to the empirically obtained results. We conclude, that the predicted results for average handoff latency for both mathematical models are in excellent conformance with the empirically obtained results.

In general, the obtained frequency distributions all seemed to conform to those predicted by the theoretical models. However, due to a memory leakage in KAME the number of samples are too small to make more firm conclusions regarding the frequency distributions. We suggest that a dataset of around 5000 handoffs per experiment should be obtained, as the simulator described in section 3.6 were able to produce consistent results when 5000 handoff were performed.

The results obtained for the average handoff latency in the experiments all pointed towards that both theoretical models reflects extremely well the average handoff latency experimented by mobile nodes using the KAME Mobile IPv6 implementation. A set of between 300 and 400 samples (depending on the actual experiment) is enough for the average handoff latency to converge.

Due to the design of the Mobile IPv6 testbed and the way experiments were performed, some specific issues regarding the obtained results must be observed:

Destination caches: The IPv6 Neighbor Discovery Protocol (NDP) maintains at each node a *destination cache* mapping IPv6 addresses to link layer addresses. We moved between the same two foreign networks every 10 seconds which resulted in the two access routers always knowing the link layer address of the mobile node. Had our movement been less frequent, the entries in the *destination cache* would have timed out, as an entry typically lasts for 20 to 30 seconds when not updated. Had these entries not been present, the access routers would not have known the link layer address of the mobile node. The overhead introduced when NDP have to discover a link layer address should be insignificant, but we have sometimes observed an overhead in the 1 to 2 seconds range. A node can only transmit a *neighbor solicitation* message once every second [Narten *et al.*, 1998] so this unexpected overhead might be the effect of a lost *neighbor solicitation*.

A way to ensure that the *destination cache* at access routers get probably updated is by forcing the mobile node to send a *neighbor solicitation* to newly discovered routers upon receiving a router advertisement from a router for the first time. We have applied this strategy in an informal experiment and it showed to reduce the handoff latency when moving less frequent between networks. We therefore suggest that a handoff initiation algorithm should exhibit this behavior.

Duplicate address detection: Upon receiving a router advertisement from a new network the mobile node uses *stateless auto-configuration* to generate a valid IPv6 care-of address. To validate that no other node at the same network is using an identical IPv6 address the mobile node must perform *duplicate address detection* (DAD) which takes between zero and three seconds to complete [Huitema, 1997]. However, DAD is only performed when the care-of address is first generated and it then have a default lifetime of 30 days. Therefore our experimental results contains no overhead from DAD.

Propagation delay: As the Mobile IPv6 testbed is local area, we have been able to control the load at the networks. The result has been that the experimental results show no sign of propagation delay of *binding updates* and *binding acknowledgments*. When applied in the Internet such propagation delays must be accounted for.

From the empirical studies we conclude, that both mathematical models proposed in chapter 3 reflects very well the handoff latency to be expected using the KAME Mobile IPv6 implementation.

From the theoretical models and the empirical study we have observed, that *Eager Cell Switching* is capable of avoiding packet loss in an overlap scenario. However, the theoretical models do not reflect, that this strategy initiates far too many handoffs as a handoff is initiated each time a new network is discovered. This can lead to handoffs to unstable networks. We conclude that *Eager Cell Switching* has a fast but potential risky behavior.

Regarding *Lazy Cell Switching* we have observed, that it minimizes the number of performed handoffs but is not able to avoid packet loss. *Lazy Cell Switching* is based on the principle that packets (router advertisements) must be lost before it even considers initiating a handoff. We thus conclude, that this handoff initiation strategy has inadequate performance.

Chapter 5

Proposal of Parametric Cell Switching handoff initiation strategy

In this chapter we first present a framework for a handoff initiation strategy. Based on this framework we propose the concept of *Parametric Cell Switching*, which allows a handoff initiation strategy to base a handoff decision on possibly many handoff criteria.

We then present a set of criteria to be used in an instantiation of the *Parametric Cell Switching* concept. Finally we develop a mathematical model able to predict the handoff latency of this particular instantiation of the *Parametric Cell Switching* concept for a restricted class of scenarios.

5.1 Motivation

In chapter 3 and chapter 4 we showed first theoretically and then empirically, that the *Eager Cell Switching* and *Lazy Cell Switching* handoff initiation algorithms both have inadequate performance. Although *Eager Cell Switching* is able to avoid packet loss, it initiates far too many handoffs. Conversely, *Lazy Cell Switching* reduces the number of handoffs to a minimum, but suffers from high handoff latencies.

Through our study of Mobile IP and handoff latency in particular, we have come to the belief, that basing a handoff decision only on information available at the network layer or higher layers, does not suffice. Even though we were able to obtain respectable handoff latencies in our local area testbed by tuning protocol parameters, much overhead will be added when Mobile IP is deployed in the Internet. This overhead includes both the propagation delay of *binding updates* and *binding acknowledgments* introduced in large network, but also the effect of various AAA actions which carry the potential to be lengthy processes.

Link layer information such as signal strength or bit error rate may be continuously available and can thus be measured at any frequency, providing valuable information about the present quality of a link. Such information will allow a mobile node to quickly detect a decaying connection and respond by initiating a handoff before the primary network becomes unreachable. We therefore believe that link layer information should play a key role in a proactive handoff initiation strategy. This view is also supported by Charles E. Perkins, one of the leading authorities within the Mobile IP society [Perkins, 1998, p. 244].

In this chapter we therefore pursue a more advanced handoff initiation strategy, based on both link layer and network layer information, which we refer to as *Parametric Cell Switching.* We often refer to Wireless LAN as this is the only wireless link media which has been available for experiments.

5.2 Concept of Parametric Cell Switching

This section defines a handoff framework, which form the basis for deciding when to perform a handoff. The purpose of defining a general framework is to make it possible to initially abstract over link layer technologies and specific user scenarios.

By presenting a general handoff framework we set the context and basic structure of the *Parametric Cell Switching* concept. A general handoff framework has possibility for later fine tuning for special handoff scenarios and supporting many different link layer technologies.

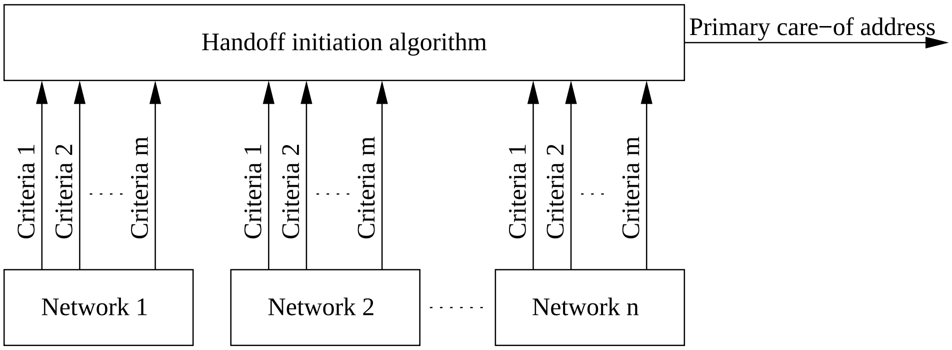


Figure 5.1: Abstract handoff framework.

The handoff framework is presented in figure 5.1. It consists of a handoff initiation algorithm, a set of available networks and a set of handoff criteria transfered for each network to the handoff algorithm.

The overall task of the **handoff initiation algorithm** is to select the care-of address at the "best"network as the primary care-of address. For each network the handoff algorithm periodically evaluates a set of handoff criteria and determines which network provides the best performance from the users point of view. If the "best"network is not the network at which the primary care-of address belongs, a handoff to that network should be initiated. The objective of this proactive behavior is to initiate handoffs before the primary network becomes unreachable and thereby avoid packet loss, but still keep the number of handoffs at a reasonable level.

A **handoff criteria** is a piece of information, that describes a characteristic of the network. It is suggested that the same set of criteria for every available network is provided to the handoff algorithm. If a certain criteria cannot be provided for a network, e.g. because the used link layer technology does not support it, two options arise:

- The value of the criteria should be chosen from a set of default values.

- The criteria should have no influence.

The first option is simple to implement as only one handoff initiation algorithm is needed. For the second option a handoff initiation algorithm is needed for each combination of missing criteria. However, the first option requires insight into the consequences of different default values. We choose to use the first option to avoid multiple handoff initiation algorithms.

The handoff framework provides a general model, where different handoff initiation algorithms based on different sets of handoff criteria can be developed.

5.3 Choice of handoff criteria

In this section we select a set of handoff criteria, that we later want to use in an instantiation of the *Parametric Cell Switching* concept. We will refer to this instance as the *Parametric Cell Switching* handoff initiation algorithm. First the set of criteria is listed, then each criteria is described and the choice of each criteria is defended.

In [Andersen *et al.*, 2000] we presented 18 potential handoff criteria, which was the result of a brainstorm. Each handoff criteria was evaluated with respect to usability/relevance, implementation/deployment effort, and run-time cost on a three-point scale. In [Andersen *et al.*, 2000] we recommended that handoff criteria, which measure the quality of the link from the mobile node to the access router (one-hop), should be given higher priority than handoff criteria, which measure the quality of one or more routes from the mobile node to correspondent nodes (the entire route), as the process of determining values for entire route criteria is often long in duration. The consequence is that we have chosen only to focus on one-hop handoff criteria. We have selected four of the 18 potential handoff criteria to be part of the *Parametric Cell Switching* handoff initiation algorithm. The four handoff criteria are:

- Signal-to-noise ratio (Link layer).
- Round trip time to the default router at a network (Network layer).
- Duration of network availability (Network layer).
- Price of using a network (Network layer).

The four chosen handoff criteria are spread across two layers in the network protocol stack. Handoff initiation algorithms based purely on network layer criteria can be used without utilizing special properties of the lower layers in the network protocol stack; the link and physical layer. This ensures that the handoff initiation algorithm is independent of the network device and its associated driver.

On the contrary handoff initiation algorithms based on link layer criteria may require modifications to each device driver to provide the necessary information. Link layer criteria however enables more precise assessments of the quality of a link to a network.

If link layer and network layer criteria are combined, the link layer criteria can ensure high performance when they are present, while the network layer criteria can constitute a fall-back mechanism, when link layer criteria are not present.

5.3.1 Signal-to-noise ratio

The signal-to-noise ratio is a measure of how clear a signal is compared to background noise. The background noise can be caused by devices communicating at other frequencies or electrical appliances in general. As the name implies the signal-to-noise ratio consist of two measures: A measure of the received signal strength and a measure of the background noise level.

When a mobile node communicates over a wireless link it must be within the range of a base station. The range of a base station depends on its location and the technology it uses. GSM ranges up to a few tens of kilometers in open land while its range in urban areas may be limited by buildings [Mouly *et al.*, 1992]. Wireless LAN ranges up to 160 m in open areas, while inside a building, internal structures will limit the range considerably [ORiNOCO].

A base station transmits with a certain effect. As the strength of a radio signal degrades as the distance between a base station and a mobile node increases, a mobile node far from a base station will receive the base station's signal at a lower level than a mobile node closer to the base station. The received signal strength of several base stations can to some extend indicate which base station offers the most stable link.

The received signal strength from a base station may change rapidly. Consider the following example. A terminal is receiving a high level signal from one base station. Then the terminal turns a corner and the radio signal to and from the base station is blocked. A handoff to a network with a reachable base station must be performed before communication can continue. A sudden drop in signal strength like described above is denoted a corner-effect [Pahlavan *et al.*].

If a measure of the background noise level is also obtainable from a mobile node's link layer, the signal-to-noise ratio is a better choice as a handoff criteria than the received signal strength, because the signal may be lost in the noise. Compared to the received signal strength, the signal-to-noise ratio gives a more precise measure of the quality of a link.

A mobile node can only measure the signal-to-noise ratio for traffic traveling from a base station to the mobile node. The signal-to-noise ratio in the other direction can be measured by the base station and delivered to the mobile node. We do not consider measurements at the base station as this may require modification of base stations which may be a serious deployment hurdle.

5.3.2 Round trip time

The round trip time to a router can be determined by a mobile node by performing a ping (echo request) of the router. A successful ping indicates that there exists a two-directional link between the router and the mobile node. In other words the router can receive packets from the mobile node, which implies that some base station at the network can hear the mobile node's signal. Similarly, the mobile node can receive packets from the router, which implies that the mobile node can hear some base station's signal.

The measured round trip time is affected by several factors:

The bandwidth of the network: Our informal measurements has shown that the round trip time to a router at a 10 Mbit/s network is approximately 25% higher than the round trip time to a router at a 100 Mbit/s network. The measurements were performed with no other traffic at the network using two identical network devices, which was forced to communicate at 10 Mbit/s and 100 Mbit/s respectively.

The amount of traffic at the network: Especially the amount of data queued at the router affects the round trip time, as ICMP packets does not have higher priority than other IP packets. If a lot of data is queued in the in-bound buffer for the wireless network at the router it will take longer before the ICMP packet is processed at the router. Also if the out-bound buffer is filled with data the ICMP packet will reside in the buffer for a period before it is transmitted over the network. Loaded buffers are also an issue at the mobile node.

It is our observation that heavy traffic through a router, as from point *A* to point *B* in figure 5.2, which does not travel at the network at which the ping is performed does not affect the round trip time. Whether this is the result of clever resource-sharing implemented in FreeBSD 4.1 or a general property of the buffer mechanisms commonly implemented in routers, must be determined by trying the experiment with other operating systems.

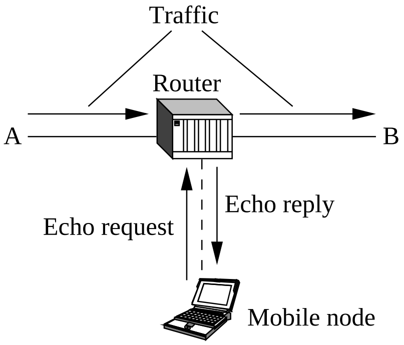


Figure 5.2: Heavy traffic at non-local networks does not affect the round trip time to the default router.

Wireless LAN, 802.11b, can operate at different transmission speeds: 11 Mbit/s, 5.5 Mbit/s, 2 Mbit/s and 1 Mbit/s. The actual speed used depends on external conditions like distance, radio noise and obstacles between the communicating devices. As mentioned earlier the available bandwidth affects the measured round trip time and the transmission speed affects the available bandwidth. Therefore the round trip time will indicate the actual network load at the actual transmission speed. The round trip time could be a simple way to reveal which link offers the best bandwidth.

We choose to use the round trip time because it is a pure network layer handoff criteria, which is very simple to implement.

It should be noted that routers may be configured to not reply to pings to protect themselves against denial-of-service attacks. However, if a network provider wants to provide mobile access, allowing a certain amount of ICMP traffic may be necessary.

5.3.3 Duration of network availability

The duration of network availability is the period of time in which a network on average is reachable from a particular mobile node. This time can be measured at several levels. One level is the period of time that a radio signal from a network is received at the link layer. Another level is the period of time in which router advertisements are delivered regularly from a network. In other words: The period of time a mobile node considers a router at a particular network reachable.

This criteria only applies for revisited networks. By remembering the duration that a network has been available in the past, the mobile node can assess which network in the past has provided connectivity for the longest duration. This is especially useful if the mobile node often travels the same physical path. In this case the mobile node has measured the reachability duration of the available networks several times, and therefore has information to choose the network, which is likely to remain reachable for the longest and thereby reduce the number of handoffs. Furthermore, the network availability duration is a handoff criteria which enables a mobile node to *learn* from past experiences.

5.3.4 Price of using a network

The price of using a network is the amount of dollars that a user must pay an Internet Service Provider (ISP) for each Mb the user receives/sends through the network or for each minute the user is connected to the network. By including this criteria, the handoff initiation algorithm can take the costs of using a network into consideration. For the user it is preferable if a reasonable tradeoff between price and performance is achieved.

5.4 Computing scores for handoff criteria

In this section we present how each handoff criteria is processed for use in the *Parametric Cell Switching* handoff initiation algorithm.

The general concept of *Parametric Cell Switching* is that the measures for each handoff criteria for a network are converted, weighted and summed to an overall score for the network. The scores for all available networks are compared and the care-of-address at the network with the highest score is used as primary care-of-address. If two networks fluctuate around the same score it may cause a ping-pong effect, where a new network gets the highest score every time the scores are calculated. This may cause the handoff initiation algorithm to initiate a handoff back and forth between the same two networks every time the scores are calculated. To reduce ping-pong effects we decide on an ad hoc basis that a network must have a score at least 10% of the score range better than the primary network to become the new primary network. If the score range is [-10,10] a network must thus have a score at least 2 points better than the primary network to become the new primary network. The general expression for the network evaluation is

$$S \triangleq \sum_{i=1}^{m} W_i \cdot P_i \tag{5.1}$$

where S is the evaluation value (score) measured for the network, while P_i is the score contribution from the ith of m handoff criteria and W_i is the customizable weight of the ith handoff criteria. In the case where the four handoff criteria described in section 5.3 are used, the network evaluation expression is

$$S = W_{\text{SNR}}P_{\text{SNR}} + W_{\text{RTT}}P_{\text{RTT}} + W_{\text{Avail}}P_{\text{Avail}} + W_{\text{Price}}P_{\text{Price}} \tag{5.2}$$

where *SNR* denotes signal-to-noise ratio, *RTT* denotes round trip time, *Avail* denotes duration of network availability and *Price* denotes the price of using the network.

The functions for P_i for handoff criteria where a high measured value counts in favor of a network, have the output range [0,10] for all valid input. An example is the signal-to-noise ratio. Handoff criteria where a low value counts in favor of a network have an output range of [-10,0]. An example is the round trip time. This is an important feature as it makes it easier to set the weights W_i as intended.

The weighted sum of the handoff criteria is used because it is fairly easy to control the relative importance of the used handoff criteria by adjusting the weights. It is also a robust construction, which is not subject to division by zero and other unintended behaviors.

In the following sections we present our suggestion for how the P_is should be calculated. We pursue a function for converting measured values to a score contribution for each handoff criteria P_i. The function should have the following characteristics:

- It has a limited output range. This ensures that the score contribution of each handoff criteria is limited.
- It is defined for all input values, which places no restrictions on the measured values for handoff criteria.
- It is continuous, which ensures no magic jumps in the score contribution.
- Below a certain input value x_1, the output is very close to minimum.
- Above a certain input value x_2, the output is very close to maximum.
- Between these input values the development is close to linear.

Figure 5.3 summarizes the wanted characteristics of a function for converting measured values to score contributions.

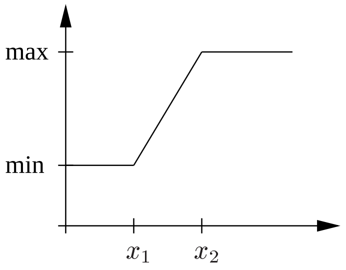


Figure 5.3: Pursued characteristics of function for converting measured values to score contributions.

After a scan of a mathematical handbook, we found that the hyperbolic tangent function ($\tanh$) exhibits these characteristics. The general function we choose for converting measured values to score contributions is

$$P(x) = a \cdot \tanh\left(\frac{x-b}{c}\right) + d \tag{5.3}$$

where *x* is the measured values, *a* determines the size of the output range, *d* determines vertical displacement of the function, *b* determines horizontal displacement of the function and finally *c* determines the slope of the middle part of the function. The characteristics of this function are visualized in figure 5.4.

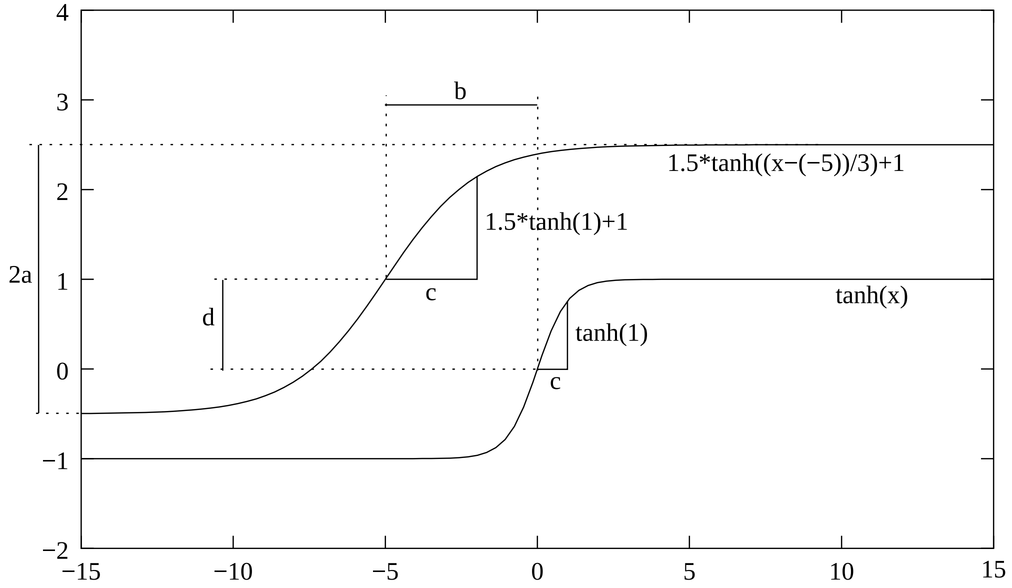


Figure 5.4: The characteristics of $a \cdot \tanh(\frac{x-b}{c}) + d$.

In the following sections we present settings of *a*, *b*, *c* and *d* for each handoff criteria. These settings are chosen as a suggestion of what apparently seems reasonable. In order to determine more appropriate settings clearly empirical data are needed. If one possesses explicit knowledge of the scenario in which the handoff initiation algorithm is operated further optimizing would be possible.

5.4.1 Computing score for signal-to-noise ratio

A link specific handoff criteria, in this case signal-to-noise ratio, must be handled individually for each link layer technology used. Consider GSM and Wireless LAN. The typical radio output power capability for a GSM mobile phone is 33 dBm[1] [Scourias], while for Wireless LAN it is 15 dBm [ORiNOCO]. On the other hand a GSM mobile phone can maintain a connection with signal strengths as low as -110 dBm, while Wireless LAN only can maintain connections with signal strengths above -83 dBm. As a result of this variance in signal strength behavior, the constants in equation 5.3 must be determined individually for the signal-to-noise ratio for each used link layer technology.

---

[1] Decibels of a milliwatt: $[\text{dBm}] = 10\log(1000[\text{watt}])$.

As a high signal-to-noise ratio counts in favor of a network the output range is selected to be [0,10]. A challenge when computing the score for the signal-to-noise ratio for a certain link layer technology is to determine the score contribution for different signal-to-noise ratio levels. In a simple experiment with Wireless LAN we have measured the round trip time and packet loss against the received signal-to-noise ratio. The result is plotted in figure 5.5.

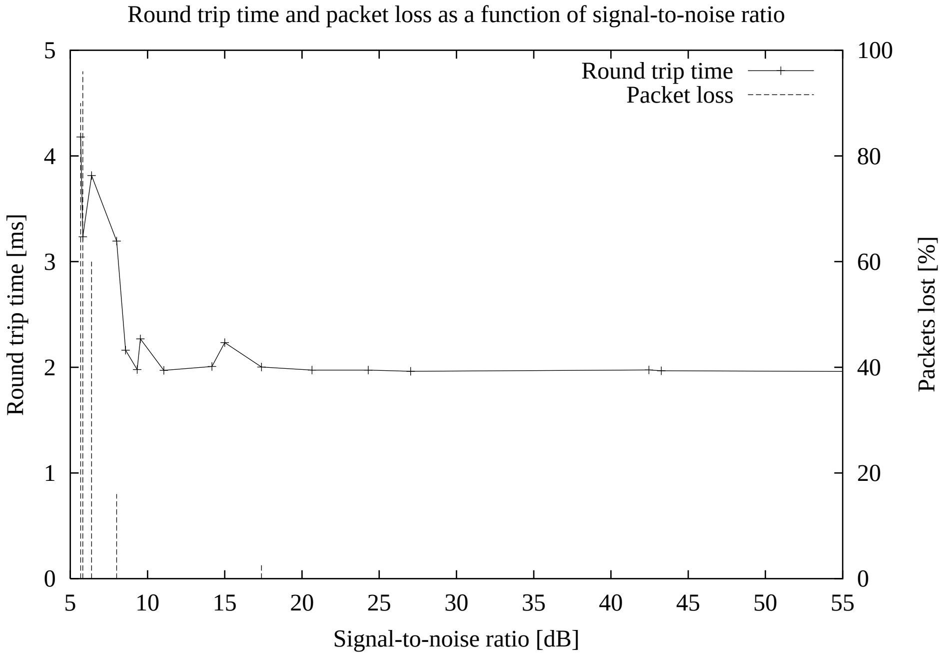


Figure 5.5: Round trip time and packet loss as a function of signal-to-noise ratio in a Wireless LAN experiment.

The experiment is performed by sending 30 pings from a mobile node to an access router with one second intervals, measuring the average signal-to-noise ratio for potential replies and the average round trip time. Unreplied pings are considered lost. During the experiment the mobile node is moved around to different locations each resulting in a different average signal-to-noise ratio. A limitation to the experiment is that the signal-to-noise ratio can only be measured when a packet has been received. This means that we cannot measure the signal-to-noise ratio if the ratio is so low that a packet is not successfully received.

Wireless LAN retransmits a packet, which is lost during transmission, three times before the packet is dropped. If one of these retransmissions are successful it shows as an increased round trip time, otherwise it shows as a lost packet. Figure 5.5 shows that packets are lost and the round trip time increases when the signal-to-noise ratio drops below 10 dB.

Based on this observation we conclude that the signal-to-noise ratio for Wireless LAN shall only contribute to the score when the signal-to-ratio is better than 10 dB. If the signal-to-noise ratio is below 10 dB for a connection to a particular base station, the connection is not considered stable. From 10 dB and up we want a steady increase in score contribution. Above some point between 20 dB and 80 dB the score contribution should level. We want this behavior as the performance of the Wireless LAN network does not improve significantly when the signal-to-noise ratio is above 20 dB, but we still want a better signal-to-noise ratio to result in an increased score, as we consider a high signal-to-noise ratio as an indication of a stable connection. The following definition fulfills these demands

$$P_{\text{SNR}} \triangleq 5 \cdot \tanh\left(\frac{\text{snr} - 40}{c}\right) + 5 \tag{5.4}$$

where *snr* is the signal-to-noise ratio for the network. The constant *c* determines how fast the score shall rise. Figure 5.6 illustrates the result of different settings for *c*.

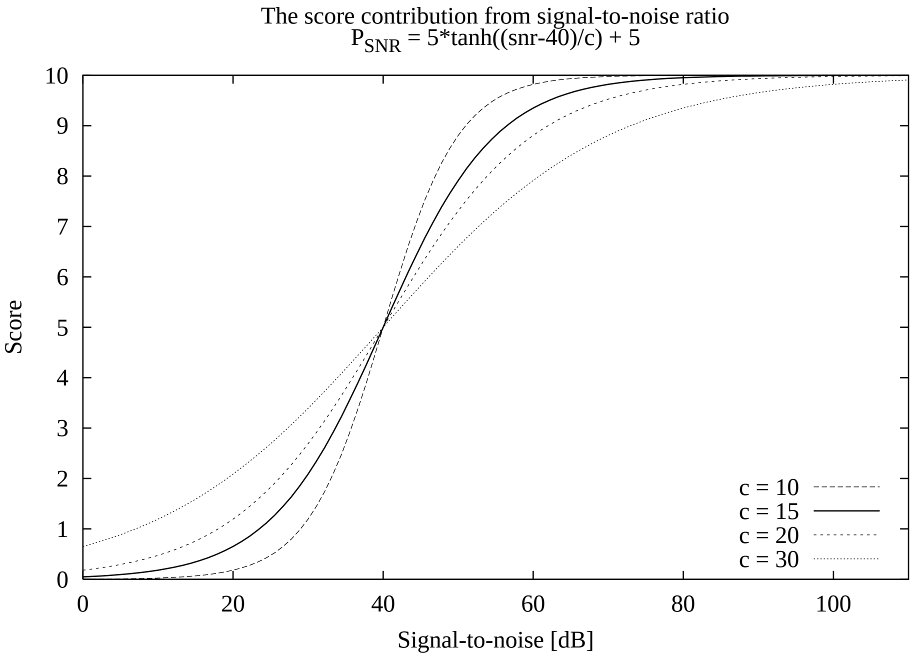


Figure 5.6: The score contribution for signal-to-noise ratio for different settings of c.

We choose to use $c = 15$, as it gives hardly any score when the ratio is below 10 dB, but still has a slope not too close to vertical. Note that the selection of $b = 40$ and $c = 15$ can only be expected to work for Wireless LAN. For other link layer technologies an experiment like the 30 pings-experiment must be performed to determine for which signal-to-noise ratio a connection is unstable. Then b and c should be adjusted to reflect this behavior.

5.4.2 Computing score for round trip time

The round trip time to the default router at a network fluctuates even when conditions are unchanged. The scheduling of processes may have changed or a collision may have occurred at the link media. To damp irrelevant fluctuations the round trip time is averaged across the last five samples. The round trip time is a network criteria and is directly comparable for all types of network interfaces. The score contribution for the round trip time is set to be in the range [-10,0] so that it reduces the score most when the round trip time is high. The term for the round trip time in the score evaluation is

$$P_{\text{RTT}} \triangleq -10 \cdot \tanh\left(\frac{\text{Avg}_n(\text{rtt})}{c}\right) \tag{5.5}$$

where *Avg*$_n$(*rtt*) denotes the average round trip time for the last n samples. The constant *c* here determines how fast the score contribution should approach -10. Figure 5.7 shows the score contribution for different values of *c*.

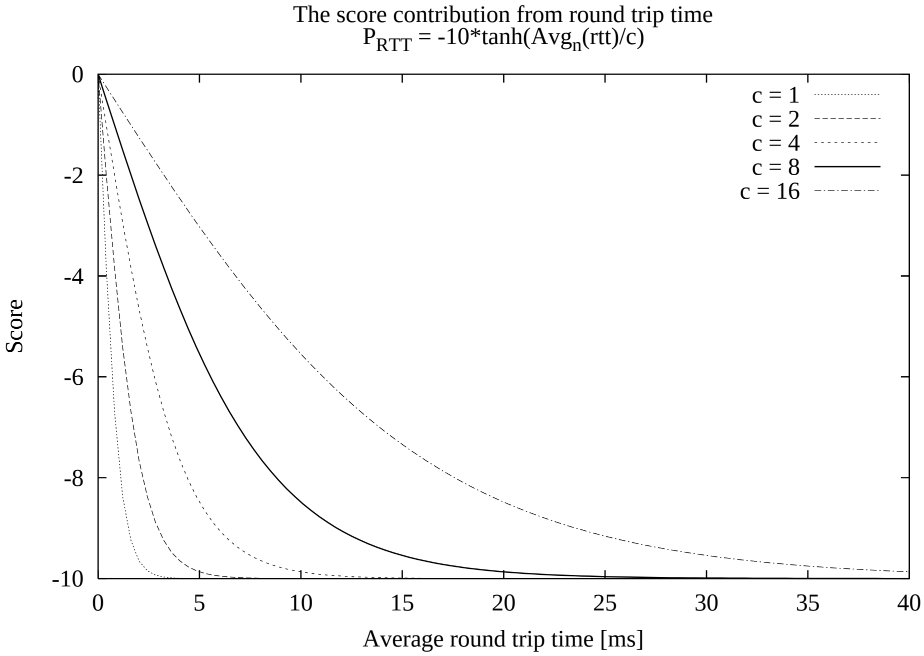


Figure 5.7: The score contribution for round trip time for different settings of *c*.

We choose to use $n = 5$ and $c = 8$ as we do not expect round trip times above 10 ms why round trip times above this value should give the lowest possible score. Note that $c = 8$ should be a general applicable value, as we want the score for round trip time to reflect differences in link layer technologies.

5.4.3 Computing score for duration of network availability

The duration of network availability is averaged across the last 10 samples, as we wish the handoff initiation algorithm to have memory of previous movement patterns and still be able to adapt relatively fast to new patterns. A high value of duration of network availability counts in favor of a network and the output range is therefore selected to be [0,10]. The term for the duration of network availability in the score evaluation is

$$P_{\text{Avail}} \triangleq 10 \cdot \tanh\left(\frac{\text{Avg}_n(\text{avail})}{c}\right) \tag{5.6}$$

where *Avg*$_n$(*avail*) is the average duration of continuous reachability of the network for the last n connection periods. The constant *c* determines how fast the score contribution should approach the maximum value 10. The score contribution for different settings of *c* are illustrated in figure 5.8.

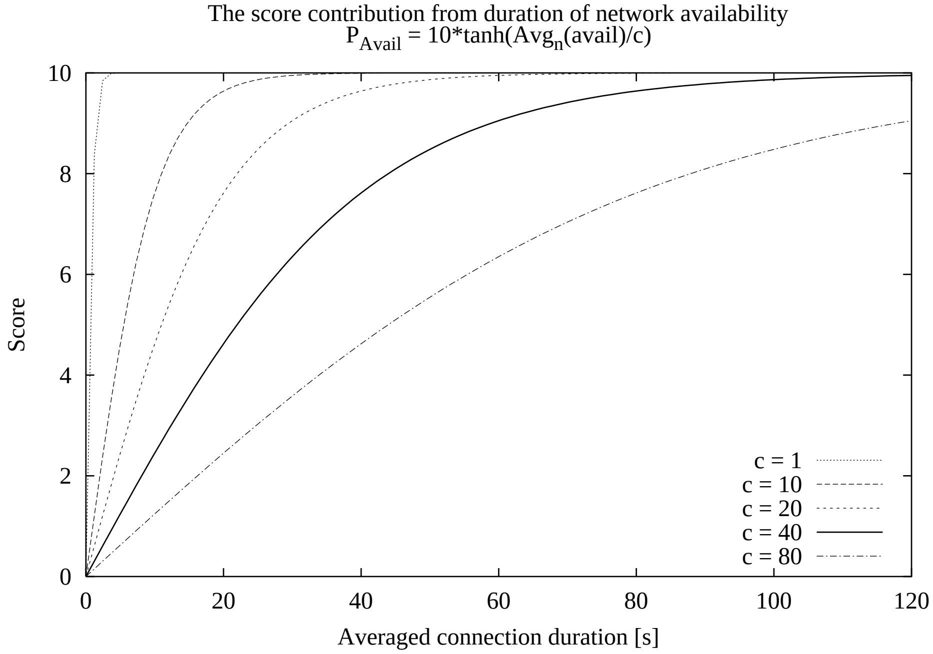


Figure 5.8: The score contribution for duration of network availability for different settings of *c*.

We choose a value of $n = 10$ and $c = 40$ as this gives all networks, that offers a connection duration greater than two minutes, the top score.

5.4.4 Computing score for price of using a network

We want changes in the price to have immediate effect why no average is used. The price should have close to proportional effect on the score, and a lower price should give a higher overall score. As a high price counts against a network the output range is selected to be [-10,0]. In this model the price is in \$/Mb. If the real price unit is e.g. \$/min some kind of mapping must be constructed. The term for the price criteria is

$$P_{\text{Price}} \triangleq -10 \cdot \tanh\left(\frac{\text{price}}{c}\right) \tag{5.7}$$

where *c* determines how fast the score contribution approaches the value -10. Figure 5.9 shows the development of the score contribution for price for different settings of *c*.

If not many networks are expected to cost over 6 \$/Mb, $c = 3$ seems like a good choice as the score contribution for prices above 6 \$/Mb for $c = 3$ is almost -10.

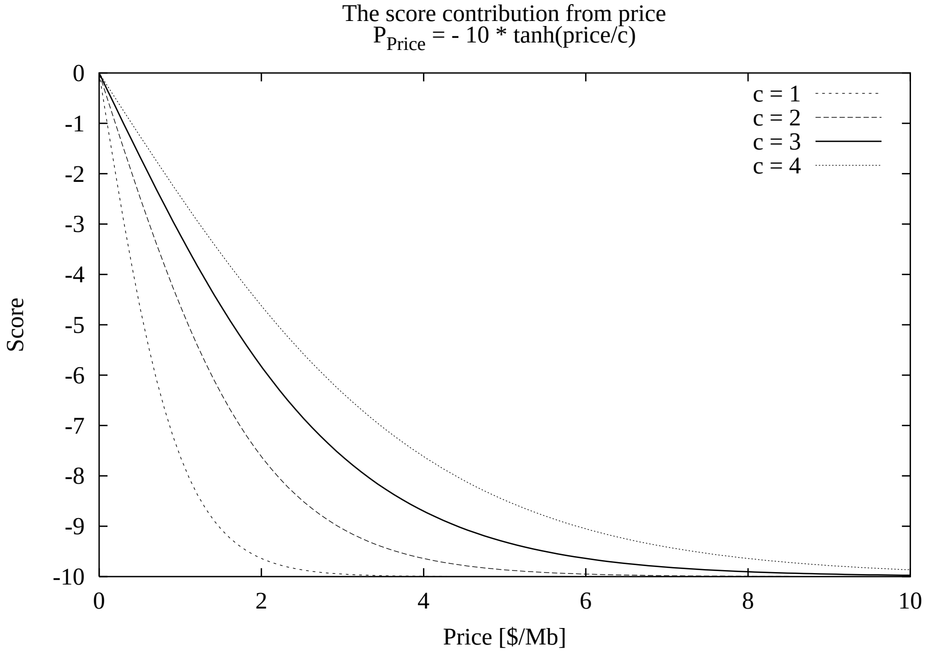


Figure 5.9: The score contribution for price of using a network for different settings of *c*.

5.5 Weighting of handoff criteria

In section 5.4 we presented how a score for each handoff criteria can be computed based on measured values. We did not consider how each handoff criteria should be weighted in an overall score calculation.

In this section we first discuss the effect of using weights. Then we present a set of weights which we use in the *Parametric Cell Switching* handoff initiation algorithm.

5.5.1 Effect of using weights

A feature of the *Parametric Cell Switching* concept is the possibility of setting different weights for each criteria used in the handoff algorithm.

Weights for the different criteria used in an instance of *Parametric Cell Switching* reflects the relative importance of each criteria in the handoff decision. The advantage of this strategy is that it is relatively easy for users to configure the behavior of the handoff initiation algorithm. This is important, as the performance of a handoff initiation algorithm is dependent on the particular scenario in which it is operated and it is therefore impossible to determine a setting of weights which is optimal in all scenarios.

An example of a criteria which different users might want to give different weight is the price of using a network. Some users might chose to set the weight for the price of using a network to a very low value, if their priority is to maintain the best connection at all times no matter the cost. Others might seek a more reasonable relation between price and performance and thus choose to configure the weight for price at a higher value.

Another example is the duration of network availability. This criteria might increase the performance of a handoff initiation algorithm in scenarios where a mobile node often travels the same physical path. However, it can slow down reaction of the handoff initiation algorithm when leaving the range of the networks to which the mobile node has often been connected. The weight for duration of network availability should thus reflect the preferred behavior of the handoff initiation algorithm in these scenarios.

A designer of a handoff initiation algorithm could provide numerous different sets of weights, each optimized for some specific scenario.

5.5.2 Weight setting in Parametric Cell Switching

What might be appropriate weight settings is dependent on the scenarios in which the *Parametric Cell Switching* handoff initiation algorithm is deployed. We consider setting the weights as $W_{\text{SNR}} = 1$, $W_{\text{RTT}} = 1$, $W_{\text{Avail}} = 0$ and $W_{\text{Price}} = 0$. This yield the formula for overall score as

$$S = P_{\text{SNR}} + P_{\text{RTT}} \tag{5.8}$$

We consider these simple settings as we do not want to engage in a study of accounting models, and we do not have the resources to perform experiments regarding duration of network availability. Furthermore we believe the signal-to-noise ratio and the round trip time to be the most important criteria in our proposal of a proactive handoff initiation algorithm.

The signal-to-noise ratio and the round trip time are given equal weights, as it is of equal importance to know whether a link is present and to know which link provides the best available bandwidth. Recall that the score contribution for P_{SNR} is in the range [0,10] and that the score contribution for P_{RTT} is in the range [-10,0]. When both P_{SNR} and P_{RTT} are given a weight of 1 the output of the proposed *Parametric Cell Switching* handoff initiation algorithm will be in the range [-10,10]. Figure 5.10 shows a surface plot of the overall score for the selected weights.

The round trip time and the signal-to-noise ratio is measured periodically. If there is no response to a ping within a short period, the ping times out and the round trip time is set equal to the timeout period or an infinitive value. In this fashion unreplied pings will be reflected in a poor score for the round trip time.

5.6 Theoretical study

In this section we present a mathematical model able to predict the performance of the proposed *Parametric Cell Switching* handoff initiation algorithm using the weights proposed in section 5.5.

We perform this theoretical study for the following reasons:

- To determine the influence of various settings on the handoff latency.

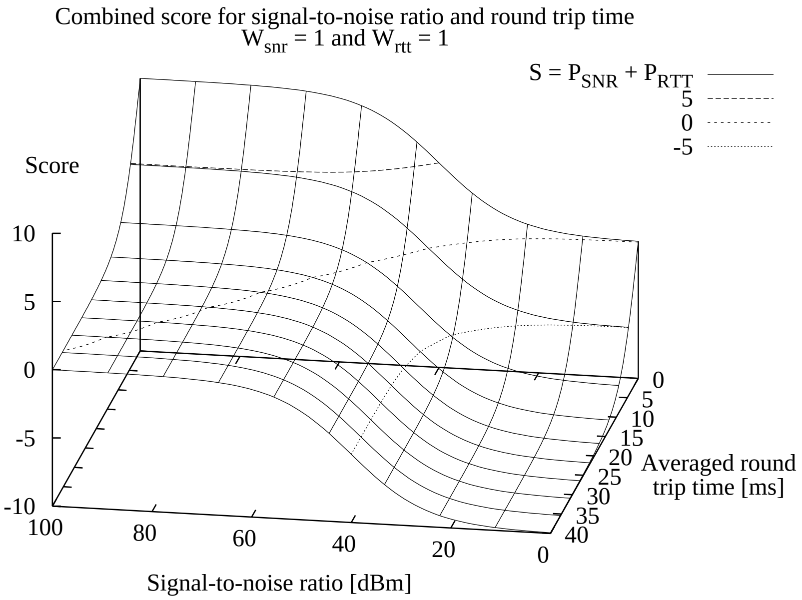


Figure 5.10: A surface plot of the score $S = P_{\text{SNR}} + P_{\text{RTT}}$.

- To learn what handoff latency can be expected. We are particularly interested in learning how fast (compared to *Eager Cell Switching*) *Parametric Cell Switching* is able to react to the discovery of new networks.

We consider the no overlap scenario from section 4.3.1 which means that there is no overlap zone between the two networks to which the mobile node can connect. There is exactly one network available to the mobile node at all times. The proposed mathematical model is restricted to the scenario of the Mobile IPv6 testbed as presented in chapter 4. We assume the use of wired connections and therefore the mathematical model does not consider signal-to-noise ratio. Therefore the mathematical model does not account for behavior triggered by signal-to-noise ratio measurements. We make these apparently very strict assumptions for two reasons:

- The main purpose of the mathematical model is to reveal how soon *Parametric Cell Switching* is able to react in a real scenario. This restricted scenario provokes the *Parametric Cell Switching* handoff initiation algorithm to perform a handoff as fast as possible.
- In the Mobile IPv6 testbed the nodes are stationary therefore the signal-to-noise ratio does not change significantly even for wireless links.

Recall that in the proposed handoff initiation algorithm, three events can trigger the mobile node to initiate a handoff:

- A poor signal-to-noise ratio is measured for the primary network.
- A poor round trip time is measured for the primary network.

- A poor combination of signal-to-noise ratio and round trip time is measured for the primary network.

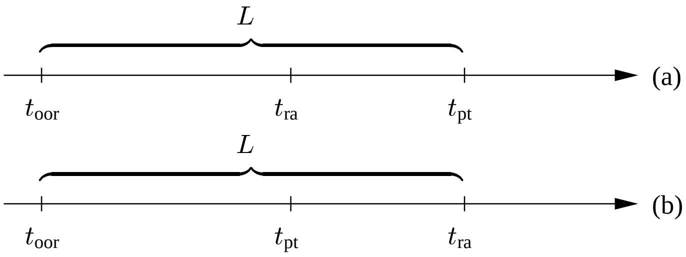
In an experiment in the Mobile IPv6 testbed using wired connections, the signal-to-noise ratio does not change during an experiment. Therefore a handoff is never triggered by a poor signal-to-noise ratio. Handoffs are only triggered by poor round trip times, and when pings are not replied.

When the *Parametric Cell Switching* handoff initiation algorithm is applied in the Mobile IPv6 testbed two situations affecting the way a handoff is triggered can occur:

- The mobile node becomes aware of a new available network before it realizes that the primary network is unavailable. More precisely a router advertisement arrives from the new network before a ping of the primary network times out.

- The mobile node determines that the primary network is unavailable before it becomes aware of a new available network. More precisely a ping of the primary network times out before a router advertisement arrives from the new network.

In figure 5.11 these two situations are shown.



L: Theoretical handoff lantency
t_{oor}: Out of range of primary network
t_{pt}: Timeout of ping
t_{ra}: First router advertisement from new network

Figure 5.11: Different triggers of handoff initiation in the *Parametric Cell Switching* handoff initiation algorithm.

From figure 5.11 it is seen, that if a ping to the default router at the primary network times out before the arrival of a router advertisement from a new network, the handoff latency is determined by the arrival of this router advertisement. Conversely, if a router advertisement from a new network arrives before a ping to the default router at the primary network times out, the handoff latency is determined by the timeout of this ping.

In order to derive an overall expression for the handoff latency for the *Parametric Cell Switching* handoff initiation algorithm we need to determine how often each situation occurs and how the handoff latency can be expressed in each situation.

Determining L_{ping}

First we consider the situation where the ping timeout occurs after the arrival of a router advertisement from a new network as depicted in figure 5.11a. The mobile node pings all available networks periodically as illustrated in figure 5.12.

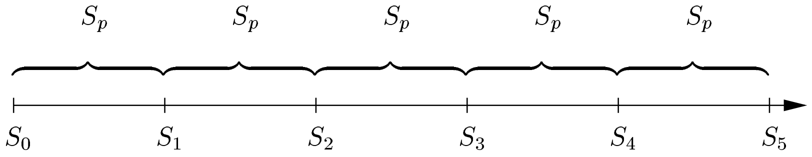
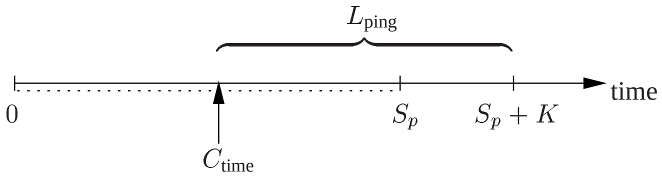


Figure 5.12: Periodic pings.

In figure 5.12 the period between two pings S_n and S_{n-1} is constant and denoted S_p. The primary network can get out of range in any of the intervals, therefore it is sufficient to consider the interval, where the primary network did get out of range. If the ping timeout occurs after the arrival of a router advertisement from a new network the handoff latency is denoted L_{ping} and is defined as illustrated in figure 5.13.



0: Ping sent prior to S_p.
S_p: First ping sent to primary network after C_{time}.
K: Timeout interval for all pings.
$C_{\text{time}} \in [0, S_p]$
$L_{\text{ping}} = S_p + K - C_{\text{time}}$

Figure 5.13: Handoff latency if the ping timeout occurs after the arrival of a router advertisement from the new network.

In figure 5.13 C_{time} denotes when the primary network gets out of range and a new network gets within range. S_p denotes the first ping sent after C_{time}, K denotes the timeout period of a ping which is constant, and $S_p + K$ denotes when the ping times out. The handoff latency is defined as

$$L_{\text{ping}} = S_p + K - C_{\text{time}} \tag{5.9}$$

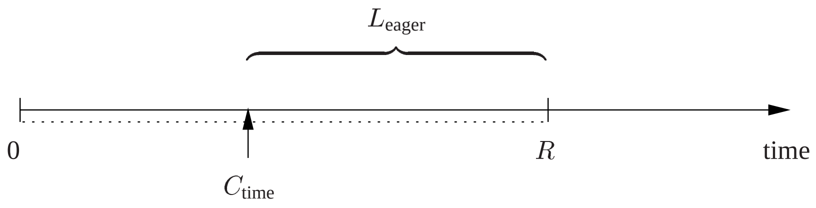
Determining $P_{L_{\text{ping}}}(l_{\text{ping}})$

In formula 5.9 S_p and K are constants and C_{time} is uniformly distributed in the interval $[0, S_p]$. This is expressed by $P_{C_{\text{time}}}(c_{\text{time}}) = \frac{1}{S_p} \cdot 1_{c_{\text{time}} \in [0, S_p]}$. As C_{time} is the only variable in L_{ping} and $c_{\text{time}} = S_p + K - l_{\text{ping}}$ we get that the probability distribution $P_{L_{\text{ping}}}(l_{\text{ping}})$, which expresses the probability that $L_{\text{ping}} = l_{\text{ping}}$, is given by

$$\begin{aligned} P_{L_{\text{ping}}}(l_{\text{ping}}) &= P_{C_{\text{time}}}(S_p + K - l_{\text{ping}}) \\ &= \frac{1}{S_p} 1_{(S_p+K-l_{\text{ping}})\in[0,S_p]} \\ &= \frac{1}{S_p} 1_{l_{\text{ping}}\in[K,S_p+K]} \end{aligned} \tag{5.10}$$

Determining L_{eager}

Now we consider the situation in which the first router advertisement from a new network arrives after the timeout of a ping to the default router at the primary network as depicted in figure 5.11b. In this situation the handoff latency is defined as for the *Eager Cell Switching* handoff initiation algorithm as illustrated in figure 5.14.



0: Router advertisement sent prior to router advertisement R

R: First router advertisement received after C_{time}

$C_{\text{time}} \in [0, R]$

$L_{\text{eager}} = R - C_{\text{time}}$

Figure 5.14: Handoff latency if the arrival of the first router advertisement from a new network occurs after the timeout of a ping to the default router at the primary network.

From figure 5.14 we note, that L_{eager} is defined as

$$L_{\text{eager}} = R - C_{\text{time}} \tag{5.11}$$

Determining $P_{L_{\text{eager}}}(l_{\text{eager}})$

Recall from formula 3.6 that for the *Eager Cell Switching* handoff initiation strategy the probability distribution $P_{L_{\text{eager}}}(l_{\text{eager}})$ is given by

$$P_{L_{\text{eager}}}(l_{\text{eager}}) = \begin{cases} \frac{2}{R_{\text{max}}+R_{\text{min}}} & \text{if } 0 \le l_{\text{eager}} \le R_{\text{min}} \\ \frac{2(R_{\text{max}}-l_{\text{eager}})}{{R_{\text{max}}}^2-{R_{\text{min}}}^2} & \text{if } R_{\text{min}} < l_{\text{eager}} < R_{\text{max}} \\ 0 & \text{otherwise} \end{cases} \tag{5.12}$$

Combining L_{ping} and L_{eager} into L_{param}

Now that we have showed how to determine the handoff latency in both situations, we can summarize the expression for the handoff latency in the *Parametric Cell Switching* handoff initiation algorithm, which we denote L_{param}, to

$$L_{\text{param}} = \begin{cases} L_{\text{eager}} & \text{if } L_{\text{eager}} > L_{\text{ping}} \\ L_{\text{ping}} & \text{if } L_{\text{eager}} \leq L_{\text{ping}} \end{cases} \tag{5.13}$$

Determining $P_{L_{\text{param}}}(l_{\text{param}})$

In formula 5.13 we note, that the probability that $L_{\text{eager}} > L_{\text{ping}}$ corresponds to the probability that the ping sent to the primary network times out before the arrival of the first router advertisement from a new network. This relation can be observed from figure 5.13 and from figure 5.14 where we have shown the same C_{time} in two different contexts. If L_{eager} is larger than L_{ping} this must be because the first router advertisement from the new network has arrived after the timeout of the ping to the primary network. Similarly, the probability that the first router advertisement from the new network arrives before the timeout of the ping is the same as the probability that $L_{\text{eager}} \leq L_{\text{ping}}$.

To determine the probability distribution for L_{param}, $P_{L_{\text{param}}}(l_{\text{param}})$, we must calculate the probability that L_{eager} represents L_{param} and the probability that L_{ping} represents L_{param}. We start by deriving the probability that L_{param} is represented by L_{eager}, which according to formula 5.13 is the probability that $L_{\text{eager}} > L_{\text{ping}}$. This probability, which we denote $P_{L_{\text{eager}}>L_{\text{ping}}}$, can be obtained by combining the probability that $l > L_{\text{ping}}$ with the probability that $P_{L_{\text{eager}}} = l$ which we denote $P_{L_{\text{eager}}}(l)$. This can be seen from figure 5.15. The probability that $l > L_{\text{ping}}$ is the area under the probability curve for $P_{L_{\text{ping}}}(l_{\text{ping}})$ from $-\infty$ up to the value of l.

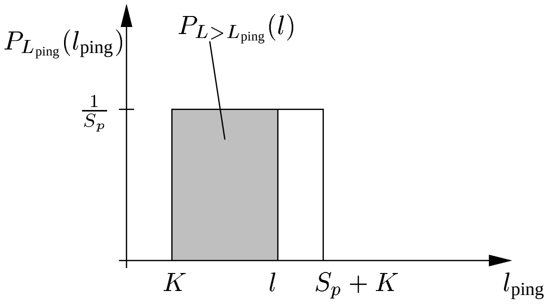


Figure 5.15: The probability that $l > L_{\text{ping}}$.

From figure 5.15 we get that the probability that $l > L_{\text{ping}}$ is the area below the graph for $P_{L_{\text{ping}}}(l_{\text{ping}})$ up to l which can be expressed as

$$P_{L>L_{\text{ping}}}(l) = \int_0^l P_{L_{\text{ping}}}(l_{\text{ping}}) dl_{\text{ping}} \tag{5.14}$$

The same deduction goes for $l \geq L_{\text{eager}}$, which therefore can be expressed as

$$P_{L\geq L_{\text{eager}}}(l) = \int_0^l P_{L_{\text{eager}}}(l_{\text{eager}}) dl_{\text{eager}} \tag{5.15}$$

Now we can determine the probability distribution for L_{param}, $P_{L_{\text{param}}}(l_{\text{param}})$, which expresses the probability that the handoff latency is l_{param} in the *Parametric Cell Switching* handoff initiation algorithm, and is given by

$$\begin{aligned} P_{L_{\text{param}}}(l_{\text{param}}) &= P_{L_{\text{eager}}}(l_{\text{param}}) \cdot P_{L>L_{\text{ping}}}(l_{\text{param}}) \\ &\quad + P_{L_{\text{ping}}}(l_{\text{param}}) \cdot P_{L\geq L_{\text{eager}}}(l_{\text{param}}) \\ &= P_{L_{\text{eager}}}(l_{\text{param}}) \cdot \int_0^{l_{\text{param}}} P_{L_{\text{ping}}}(l_{\text{ping}}) dl_{\text{ping}} \\ &\quad + P_{L_{\text{ping}}}(l_{\text{param}}) \cdot \int_0^{l_{\text{param}}} P_{L_{\text{eager}}}(l_{\text{eager}}) dl_{\text{eager}} \end{aligned} \tag{5.16}$$

We choose not to show the expansion of the result for $P_{L_{\text{param}}}(l_{\text{param}})$ as it is very complex and span several pages[2]. It is composed of numerous functions each restricted to certain values of l_{param}. Instead we plot the probability distribution $P_{L_{\text{param}}}(l_{\text{param}})$ for two different settings of R_{max}, R_{min}, S_{p} and K in figure 5.16. These particular settings are chosen, as $K = 0.1$ seconds and $S_p = 0.5$ seconds are the values used in the implementation of *Parametric Cell Switching* which we present in chapter 6.

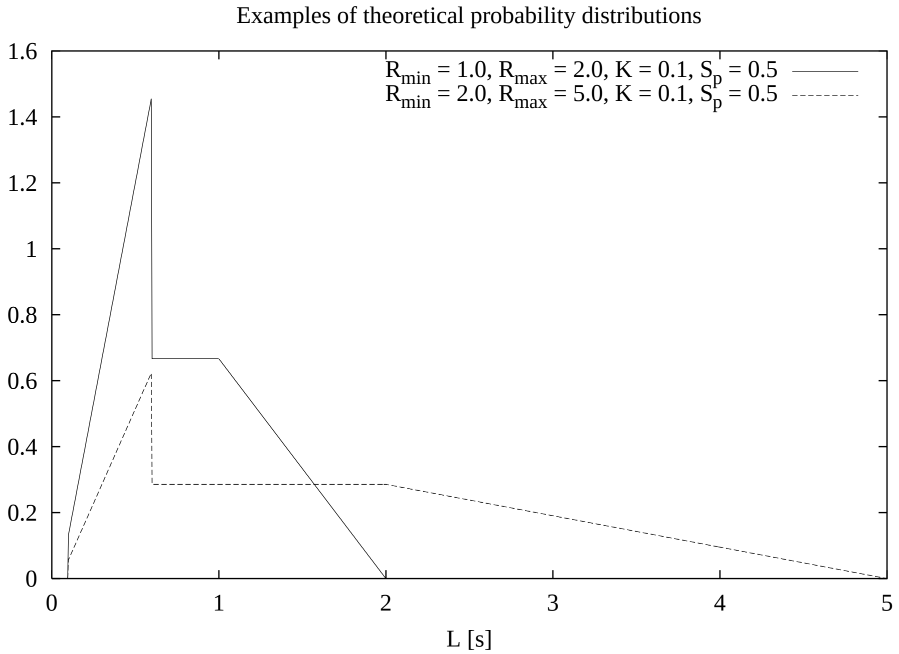


Figure 5.16: Probability distribution of the handoff latency in the proposed *Parametric Cell Switching* handoff initiation algorithm for different settings.

Note in figure 5.16 that the probability distributions above a handoff latency of 0.6 seconds ($S_{\text{p}}+K$) are equal to the probability distribution for the *Eager Cell Switching* handoff initiation strategy, as the ping timeout will always occur before the arrival of a router advertisement causing this handoff latency. Below 0.6 seconds the probability distribution is influenced by

---

[2] The expansion was calculated using Maple V.

the timeout of pings. The probability is zero for latencies below 0.1 seconds as a ping cannot timeout before this. Above a value of 0.1 the overall probability increases as the probability of router advertisement arriving before the timeout increases.

Determining $\overline{L}_{\text{param}}$

To determine the average handoff latency for the proposed *Parametric Cell Switching* handoff initiation algorithm we use the standard form when the probability distribution is known

$$\overline{L}_{\text{param}} = \int_{-\infty}^{\infty} l_{\text{param}} \cdot P_{L_{\text{param}}}(l_{\text{param}}) dl_{\text{param}} \tag{5.17}$$

As the result for $\overline{L}_{\text{param}}$ is also rather complex we only present results for some settings of R_{min} and R_{max}. In figure 5.17 the theoretical handoff latency for the recommended settings of R_{min} and R_{max} of the *Parametric Cell Switching* handoff initiation algorithm is compared to results for *Eager Cell Switching* and *Lazy Cell Switching*.

Handoff	**Configuration**					**L [s] (theory)**		
strategy	R_{min}	R_{max}	T_{l}	Q_{min}	Q_{max}	**Avg**	**Min**	**Max**
Eager	0.5	1.5	N/A	N/A	N/A	0.54	0	1.5
Lazy	0.5	1.5	4	0	1	3.96	2.5	5.0
Param	0.5	1.5	N/A	N/A	N/A	0.61	0.1	1.5

Figure 5.17: Theoretical values for L_{param} for recommended configuration.

In figure 5.18 the theoretical handoff latency for our proposed settings of R_{min} and R_{max} of the *Parametric Cell Switching* is compared to *Eager Cell Switching* and *Lazy Cell Switching*.

Handoff	**New configuration**					**L [s] (theory)**		
strategy	R_{min}	R_{max}	T_{l}	Q_{min}	Q_{max}	**Avg**	**Min**	**Max**
Eager	0.9	1.1	N/A	N/A	N/A	0.50	0	1.1
Lazy	0.9	1.1	2	0	1	2.00	0.9	3.0
Param	0.9	1.1	N/A	N/A	N/A	0.57	0.1	1.1

Figure 5.18: Theoretical values for L_{param} for our proposed configuration.

Note that both with the recommended configuration and with our proposed configuration the theoretical responsiveness of *Parametric Cell Switching* is almost identical with the theoretical responsiveness of *Eager Cell Switching*. This indicates, that should it be necessary the *Parametric Cell Switching* handoff initiation algorithm is able to react almost as fast as *Eager Cell Switching*.

5.7 Summary

In this chapter we have presented the concept of *Parametric Cell switching*, proposed a *Parametric Cell Switching* handoff initiation algorithm and determined its expected performance in a restricted scenario. Theoretically, the *Parametric Cell Switching* handoff initiation algorithm is able to initiate a handoff almost as fast as the *Eager Cell Switching* handoff initiation algorithm.

This result indicates that the *Parametric Cell Switching* handoff initiation algorithm has the responsiveness of the *Eager Cell Switching* handoff initiation algorithm should it be required. It is expected, that the *Parametric Cell Switching* handoff initiation algorithm will not initiate as many unnecessary handoffs as the *Eager Cell Switching* handoff initiation algorithm, when applied in a wireless scenario. As an additional feature, it can be configured to account for other criteria such as network availability duration and price of using a network.

It should be noted that it is not possible to achieve a faster handoff initiation than with the *Eager Cell Switching* handoff initiation algorithm, as this algorithm is only limited by how fast new networks can be discovered.

However, to decide whether the *Parametric Cell Switching* handoff initiation strategy is also an improvement in practice, empirical evidence from a scenario where the signal-to-noise ratio influences the handoff decision must be collected. We therefore consider such an experiment in chapter 6.

Chapter 6

Empirical study of Parametric Cell Switching handoff initiation strategy

In this chapter we present the results of an empirical study of the handoff performance for the *Parametric Cell Switching* handoff initiation algorithm devised in chapter 5.

First we present the design of a prototype implementation which we have merged into the KAME Mobile IPv6 software. Then we describe a set of experiments performed in the Mobile IPv6 testbed using this implementation. Finally we present the results of an informal building wide experiment, whose purpose it is to compare the handoff performance of the *Eager Cell Switching*, *Lazy Cell Switching* and *Parametric Cell Switching* handoff initiation algorithms in a more realistic scenario than the Mobile IPv6 testbed.

6.1 Motivation

In chapter 5 we proposed the *Parametric Cell Switching* handoff initiation algorithm and developed a theoretical model describing its expected handoff performance for a restricted class of scenarios.

We find it important to provide an actual implementation of the *Parametric Cell Switching* handoff initiation strategy for several reasons:

- Through experiments with an implementation the conformance between our mathematical model and an actual implementation can be investigated.
- To demonstrate the feasibility of using link layer information in a proactive handoff initiation algorithm. By providing an implementation we get the opportunity to demonstrate the performance of the *Parametric Cell Switching* handoff initiation strategy in a realistic scenario.
- The *Parametric Cell Switching* handoff initiation strategy can be configured through a number of constants. In chapter 5 we proposed a configuration which appears to be reasonable. An implementation can confirm whether the proposed configuration is also reasonable in real scenarios.

- An implementation is needed in order to empirically compare *Parametric Cell Switching* with implementations of *Eager Cell Switching* and *Lazy Cell Switching*.

6.2 Design and implementation of prototype

In order to study the performance of the proposed *Parametric Cell Switching* handoff initiation algorithm we have devised a prototype implementation merged into the KAME Mobile IPv6 software. The prototype is designed to be implemented based on the following heuristics:

- The prototype should mainly be implemented in user-land[1].
- The prototype should reuse as much as possible of the existing implementation of Mobile IPv6 in the KAME distribution.

KAME Mobile IPv6 is implemented as a part of the kernel for FreeBSD 4.1. Our strategy in implementing *Parametric Cell Switching* is to instrument KAME Mobile IPv6 to make received router advertisements available to user-land processes and allow user-land processes to set the primary network for Mobile IPv6. This allows for a user-land implementation of *Parametric Cell Switching*. Thereby alterations of the kernel is kept at a minimum resulting in less debugging of kernel processes.

Testing and debugging kernel-land[2] code is very time consuming compared to testing and debugging user-land code, as errors are harder to detect and fix. In user-land a dangling pointer often causes a segmentation fault, which halts the program. As kernel-land processes has unlimited access, dangling pointers in kernel-land seldom causes segmentation errors, but more often causes the whole system to crash and automatically reboot. Besides there exists better debugging tools for user-land programs.

The disadvantage of an user-land implementation of *Parametric Cell Switching* is degraded performance. User-land programs are subject to a scheduler, which determines when the *Parametric Cell Switching* process runs. The performance of the prototype is limited by system load caused by other activities at a host and by how fast the scheduler can reschedule. In FreeBSD 4.1 the default behavior of the scheduler is to reschedule once every 10 ms. For use in our Mobile IPv6 testbed this is adequate for two reasons:

- We control which programs runs on the mobile node.
- The precision of the measurements in the Mobile IPv6 testbed is ±100 ms due to the way experiments are conducted.

The implementation of *Parametric Cell Switching* performs the following tasks:

- Intercepts all received router advertisements.
- Keeps track of which networks are available and which are not.

---

[1] User-land is a BSD expression for the term user space.

[2] Kernel-land is a BSD expression for the term kernel space.

- Determines the round trip time to the default routers at available networks.
- Determines the signal-to-noise ratio for links to available networks.
- Finds the best network and makes sure it is chosen as the primary network by KAME Mobile IPv6.

Figure 6.1 shows how the components of the prototype of *Parametric Cell Switching* interact and distribute information.

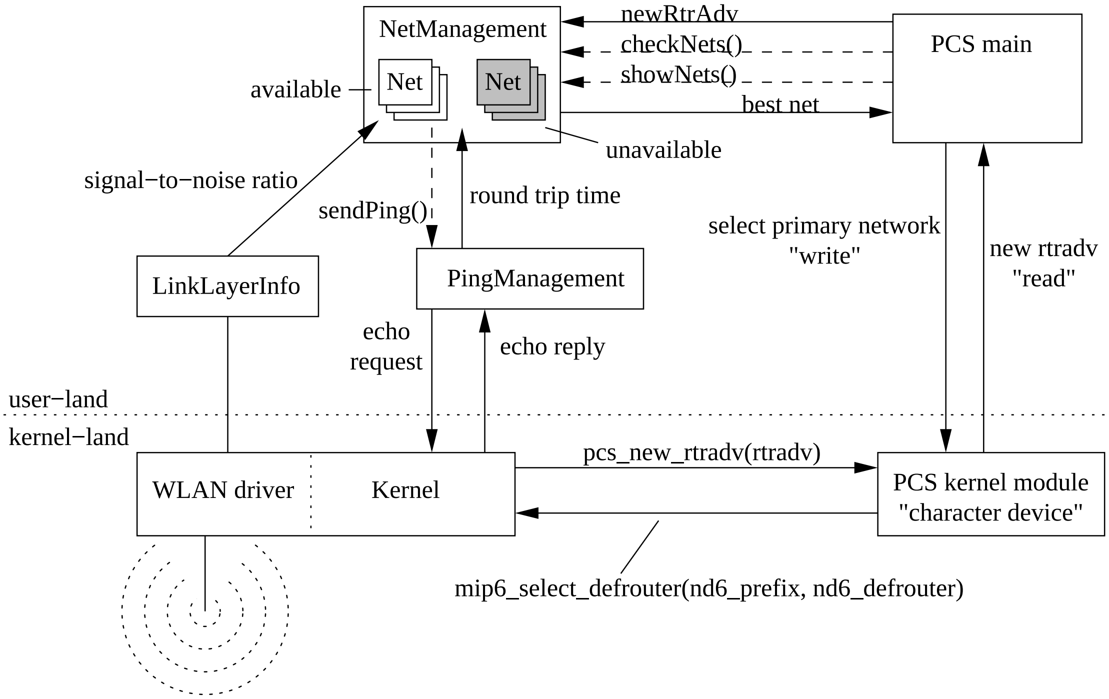


Figure 6.1: Interaction and flow between components in the implementation of *Parametric Cell Switching* (PCS).

Each component in the prototype is described in further detail in the following:

PCS kernel module is a module in the implementation of *Parametric Cell Switching* (PCS), which resides in kernel-land. It is necessary to have some part of the prototype in kernel-land as there is no user-land interface available for interception of router advertisement and for setting the primary network for Mobile IPv6 at the mobile node.

In FreeBSD a kernel module can be loaded and unloaded dynamically. This allows for fast modifications/bug-fixes, which can be tested immediately. The alternative is to make it part of the kernel. This means that modifications require a re-compilation and a re-installation of the kernel and a reboot to run/test the modification. The PCS kernel module supplies a simple interface for user-land processes through which they can:

- Receive a copy of new router advertisements.
- Instruct KAME Mobile IPv6 software to choose a certain network as the primary network.

The PCS kernel module implements the interface of a character device. This means that user-land processes can *read* a copy of the newest received router advertisement, *write* the prefix of the network selected to be the primary network, and *poll* for the arrival of new router advertisements. The KAME kernel is instrumented by adding a hook in the implementation of IPv6 neighbor discovery. The hook is a method call which is performed only when the PCS kernel module is loaded. When the PCS kernel module is unloaded the hook is disabled. The hook is inserted where neighbor discovery has detected a router advertisement and has updated its lists of prefixes and default routers. This means that the PCS kernel module is invoked every time a router advertisement is received. When a router advertisement is received by the PCS kernel module, the content of the router advertisement is copied and user-land processes are notified, that a new router advertisement has arrived.

LinkLayerInfo supplies a simple interface to information from the Wireless LAN driver. LinkLayerInfo is a simple wrapper around the non-intuitive API to the Wireless LAN driver.

PingManagement has two tasks:

- Send ICMPv6 echo requests (ping) to a router's link-local address.
- Listen for ICMPv6 echo replies and determine the round trip time.

During initiation PingManagement spawns a thread, which does nothing but listens for ICMPv6 echo replies. When an echo request is sent the current system time is stored in the data part of the echo request. When the echo request arrives at its destination, an echo reply containing a copy of the data part of the incoming echo request is generated. Therefore the time of sending the request can be found in the echo reply and the round trip time can be determined.

If an echo reply to an echo request of a network is not received within 100 ms, the echo request or reply is considered lost and the value of 100 ms is set as the network's round trip time. The value of 100 ms is an ad hoc value.

NetManagement keeps track of which networks are available and which are not. Conceptually it has two lists: one for currently available networks and one for networks declared unavailable. The unavailable list is maintained to prepare the implementation of *Parametric Cell Switching* to the use of the handoff criteria for duration of network availability. A network is declared unavailable if the lifetime of the prefix times out, because the lifetime has not been renewed by a new router advertisement for that prefix. NetManagement periodically checks if any of the networks in the list of available networks has timed out and moves them to the list of unavailable networks. It is necessary to keep track of the unavailable networks, if we need to store information about each network's average availability period.

When NetManagement receives a router advertisement, NetManagement performs the actions illustrated in figure 6.2. Note that a score for a new network is determined immediately upon the reception of the first router advertisement from that network. This enables a fast handoff if necessary.

The score for available networks is determined on a regular basis initiated by the PCS main. This involves initiating the sending of an echo request to the default router at a network and fetching the current signal-to-noise ratio.

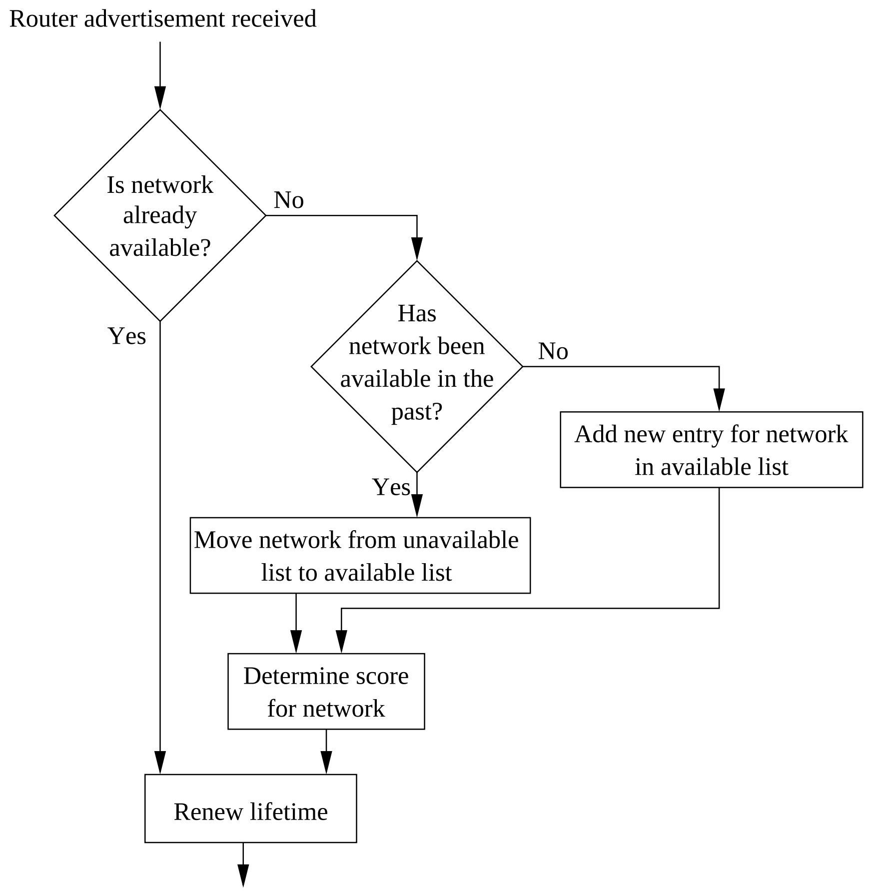


Figure 6.2: Flowchart for acting on reception of a new router advertisement in NetManagement.

PCS main is the controlling component. It has two threads. One thread, which performs the following tasks:

1. Wait for the arrival of a new router advertisement.
2. Fetch the router advertisement from the PCS kernel module.
3. Make NetManagement update the lists of networks with the information contained in the new router advertisement according to figure 6.2.
4. Goto 1.

Another thread, which performs the following list of actions:

1. Make NetManagement check for timed out networks and determine the score for all available networks, which involves the measurement of signal-to-noise ratio and the round trip time for each network.
2. Make NetManagement show the information about the known networks on the screen.
3. Retrieve the score of the network which has the highest score from NetManagement.

4. If the network with the highest score has a score of at least 2 points better than the primary network, instruct the PCS kernel module to make it the new primary network. The threshold of 2 points was determined in section 5.4.
5. Check for and process user input. The user can manually choose a network as the primary network.
6. Sleep for 0.5 seconds.
7. Goto 1.

The prototype of *Parametric Cell Switching* has the property, that *it declares a network unavailable within 0.6 seconds from the time the network got out of range*. Each network is send an echo request once every 0.5 seconds and the timeout period for an echo reply is 0.1 seconds. This makes the prototype able to perform a handoff to another available network within 0.6 seconds. If no other networks are available it must wait until a router advertisement is received from a new network.

The prototype implementation is fully operational as described, is stable and has no known bugs.

6.3 Experiments conducted with Parametric Cell Switching

The experiments conducted using the prototype implementation of *Parametric Cell Switching* falls within two categories.

First, we have conducted two experiments in the Mobile IPv6 testbed. The objective of these experiments is to reveal if the theoretical model proposed in chapter 5 is in conformance with the prototype implementation. These experiments are described in section 6.3.1.

Secondly, we have deployed three Wireless LAN base stations, each at a different network, within the Department of Computer Science at Aalborg University. These base stations are used in an informal building wide experiment with the objective of demonstrating the feasibility of using link layer information in a proactive handoff initiation algorithm. In this experiment we also compare the performance of the three different handoff initiation strategies in a realistic scenario. These experiments are described in section 6.3.2.

6.3.1 Experiments in Mobile IPv6 testbed

The following experiments are performed in the Mobile IPv6 testbed as presented in chapter 4 with the implementation of the *Parametric Cell Switching* handoff initiation algorithm:

Default settings using wired connections: In this experiment the router advertisement interval and the network prefix lifetime is set as recommended in [Johnson *et al.*, 2000]. This means an interval between subsequent router advertisement randomly chosen between 0.5 and 1.5 seconds and a network prefix lifetime of 4 seconds.

New proposed default settings using wired connections: In chapter 3 we proposed a new configuration of access routers, which improves the handoff performance of *Eager Cell*

> *switching* and *Lazy Cell Switching* without increasing network load. The objective of this experiment is to reveal if these settings produce the expected reduction of the handoff latency.

Note that these experiments are performed using Ethernet as link media, why the signal-to-noise ratio does not influence the handoff decision.

In figure 6.3 the frequency of values for handoff latency measured in an experiment using the *Parametric Cell Switching* handoff initiation algorithm with default router configuration is shown.

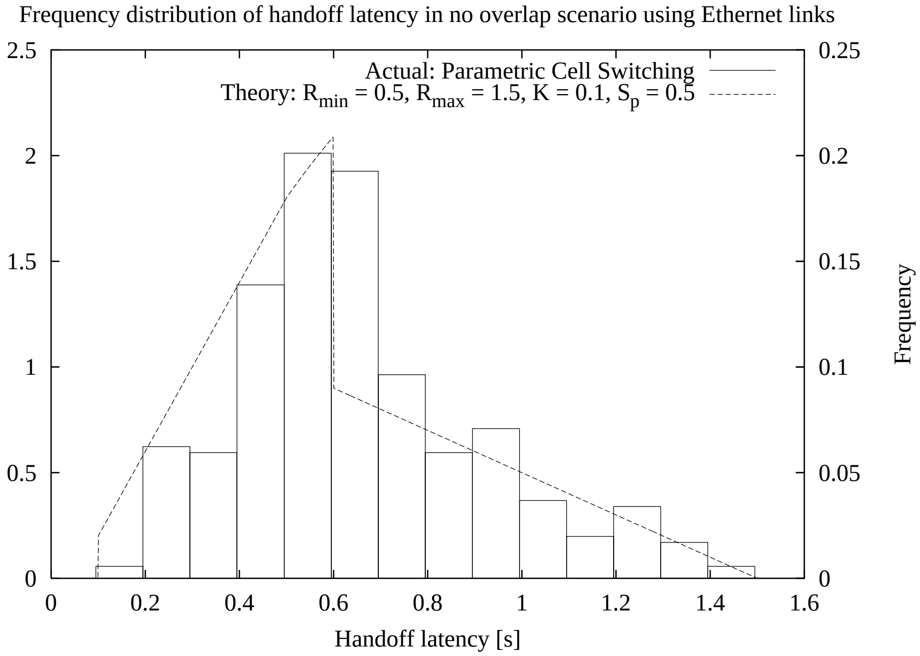


Figure 6.3: Frequency distribution for handoff latency using *Parametric Cell Switching*, Ethernet links and default router configuration in the no overlap scenario.

From figure 6.3 we see that the theoretically predicted distribution of handoff latencies apparently conforms with the realities measured in the testbed. Note that at least 50% of the handoff latencies occurs in the interval between 0.4 seconds to 0.7 seconds. It is a general property of *Parametric Cell Switching*, that a large portion of the handoff latencies will assume values around $K + S_p$ seconds, no matter the setting of R_{min} and R_{max}.

In figure 6.4 the frequency of values for handoff latency measured in an experiment using the *Parametric Cell Switching* handoff initiation algorithm with our proposed router configuration is shown.

In figure 6.4 we see that also for our proposed router configuration the theoretical model conforms well with the frequency distribution measured in the testbed.

Figure 6.5 contains a summary of the results for *Parametric Cell Switching* depicted in figure 6.3 and figure 6.4. We have included both the theoretical and empirical results for *Eager Cell Switching* and *Lazy Cell Switching* as derived in chapter 3 and chapter 4.

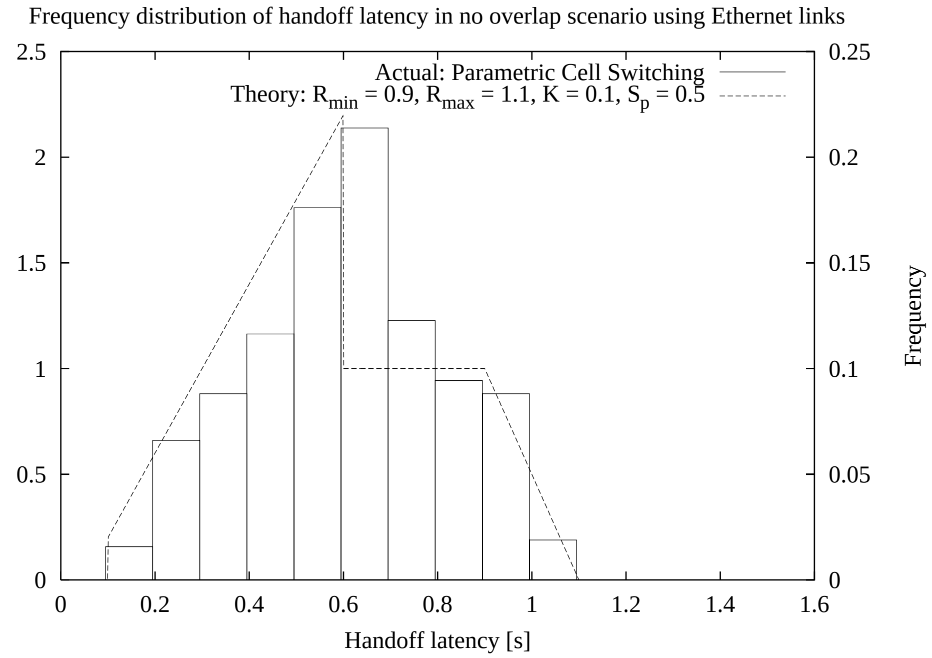


Figure 6.4: Frequency distribution for handoff latency using *Parametric Cell Switching*, Ethernet links and proposed router configuration in the no overlap scenario.

Handoff	**Router settings**			L **[s] (Theory)**			**Latency [s] (Ethernet)**		
strategy	R_{min}	R_{max}	T_{l}	**Avg**	**Min**	**Max**	**Avg**	**Min**	**Max**
Eager	0.5	1.5	4	0.54	0	1.5	0.54	0.10	1.52
Lazy	0.5	1.5	4	3.95	2.5	5.0	3.97	2.54	4.97
Param	0.5	1.5	4	0.61	0.10	1.5	0.62	0.10	1.42
Eager	0.9	1.1	2	0.5	0	1.1	0.53	0.10	1.12
Lazy	0.9	1.1	2	2.0	0.9	3.0	1.96	0.91	2.94
Param	0.9	1.1	2	0.57	0.10	1.1	0.57	0.10	1.02

Figure 6.5: Summary of expected and actual results using *Parametric Cell Switching*, *Eager Cell Switching* and *Lazy Cell Switching*.

Figure 6.5 shows that the measured values for average, minimum and maximum handoff latencies for *Parametric Cell Switching* are almost identical to the theoretical values. Figure 6.5 also shows that *Parametric Cell Switching* has a performance, when not utilizing link layer information almost identical to *Eager Cell Switching* as the theoretical models also predicted. We also note, that the maximum possible handoff latency of *Parametric Cell Switching* is identical to the maximal possible handoff latency of *Eager Cell Switching*. As *Eager Cell Switching* represents the theoretical limit for how fast an algorithm can respond to the reception of router advertisements from a new network (by performing a handoff), this indicates that *Parametric Cell Switching* will be fast in real scenarios. The novelty of *Parametric Cell Switching* is that it is expected to perform informed handoff decisions when link layer information is utilized.

When link layer information is utilized we expect *Parametric Cell Switching* to outperform both *Eager Cell Switching* and *Lazy Cell Switching*.

6.3.2 Building wide experiment

The objective of this experiment is to reveal the handoff performance of the *Eager Cell Switching*, *Lazy Cell Switching* and *Parametric Cell Switching* handoff initiation strategies in a more realistic scenario. Here we expect the *Parametric Cell Switching* handoff initiation algorithm to exhibit the better performance due to its use of link layer information.

Description of experiment

For this experiment we have configured a topology of networks as presented in figure 6.6

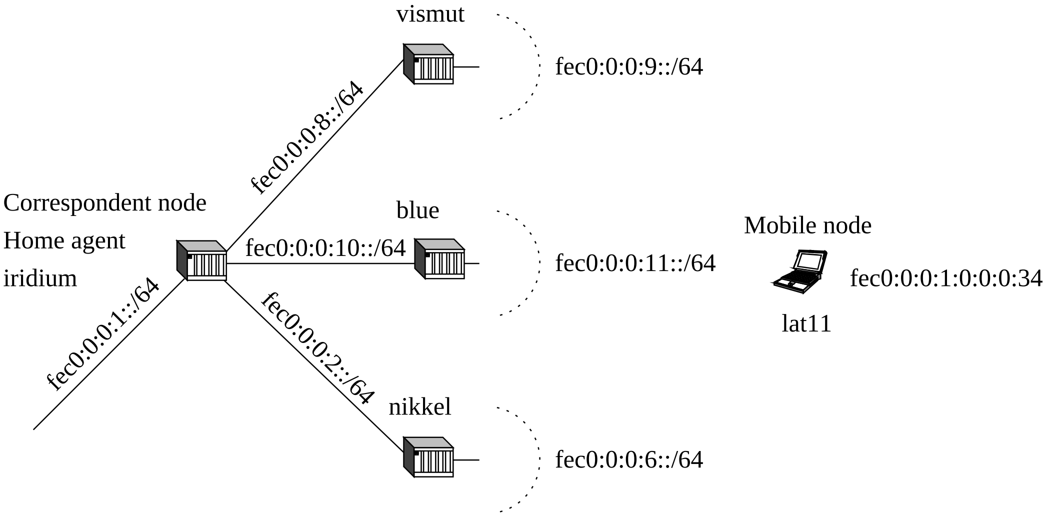


Figure 6.6: Scenario of networks used in the building wide experiment.

The three access routers *nikkel*, *blue* and *vismut* are all installed with Wireless LAN network devices and are configured with our proposed default router configuration: $R_{\text{min}} = 0.9$ seconds, $R_{\text{max}} = 1.5$ seconds and $T_{\text{l}} = 2$ seconds. These three access routers have been deployed at the Department of Computer Science at the locations depicted in figure 6.7. Note that the positions of the access routers is **not** the result of trying different locations for the access routers attempting to obtain more feasible results of the experiments. The access routers were deployed and the experiments were conducted once.

Figure 6.7: Deployment of access routers within the Department of Computer Science.

To determine which access routers are reachable from different locations we have measured the signal-to-noise ratio for each of the three networks once every second as we walked along the path illustrated in figure 6.7. In figure 6.8 we have plotted the signal-to-noise ratios measured along the path.

The numbers along the x-axis in the plot in figure 6.8 corresponds to the numbered locations in figure 6.7. Structures in the buildings cause the radio signals to propagate diversely. This causes each wireless network to supply high signal quality only in areas close to the access router. For example *nikkel* has a high signal-to-noise ratio in the area from location #1 to location #3. At other locations no network supply significantly better signal quality than the other networks. At these locations two or three networks are available but with moderate signal quality, as in the

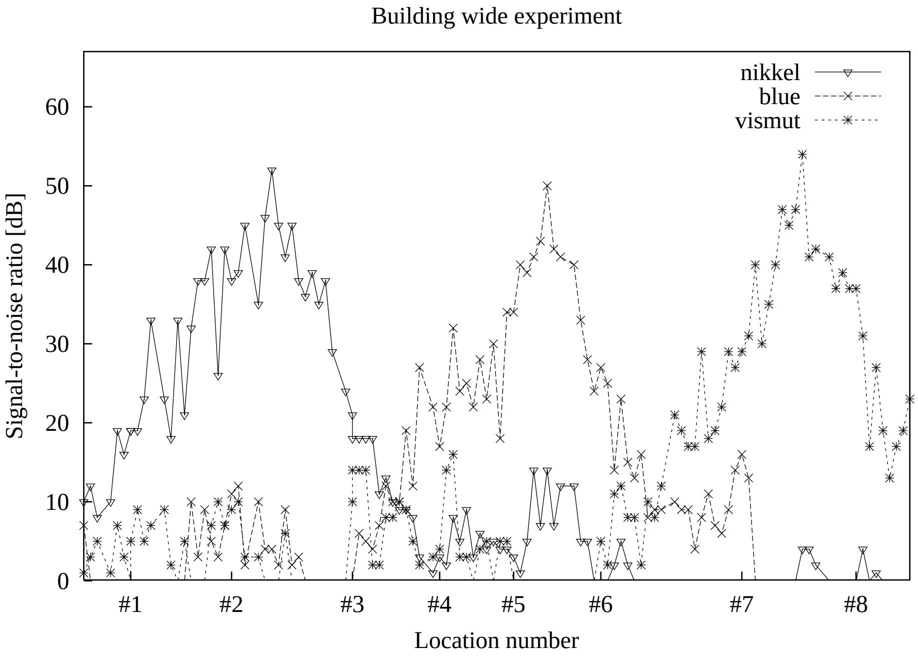


Figure 6.8: Measured signal-to-noise ratios from the three access routers when the mobile node is moved along the path depicted in figure 6.7.

area from location #3 to location #4 or in the area from location #6 to location #7. Finally, if a network is not dominant at a location it is often sporadic available, meaning that it becomes available for a short period of time and then disappears. For example *blue* and *vismut* are reachable but offers unstable connections in the area from location #1 to location #2.

The building wide experiment is performed by carrying at normal walking velocity the mobile node *lat11* along the path illustrated in figure 6.7. Along the path the mobile node travels both through areas where it is obvious which access router the mobile node should utilize and through areas where it is not at all obvious which access router the mobile node should utilize.

During the experiment a program at *iridium*, which acts as home agent and correspondent node, sends UDP packets with intervals between 95 ms and 105 ms. A program at the mobile node *lat11* receives the packets and generates a log file like in the experiments performed in the Mobile IPv6 testbed described in chapter 4. As the mobile node travels along the path the received packets are logged along with the number of handoffs. It takes approximately 2 minutes to walk the path from *Start* to *End* and approximately 1200 UDP packets are sent from the correspondent node in this period of time. We perform the experiment for all three handoff initiation strategies.

Results for building wide experiment

Figure 6.9 shows a summary of the results obtained in this informal experiment. The column *Number of attempted handoffs* is the number of times a handoff initiation algorithm tried to initiate a handoff. Whether the home agent received the binding update or not is not considered.

The columns *Average latency* and *Maximum latency* are latencies of all kinds experienced by the mobile node. That is, latencies caused by both handoffs to unstable networks and latencies caused by the mobile node maintaining attachment to networks which has become unstable.

Handoff strategy	**Number of attempted handoffs**	**Number of lost packets**	**Average latency [s]**	**Maximum latency [s]**
Eager	24	48	0.14	0.41
Lazy	3	10	0.10	0.10
Param	2	0	0.00	0.00

Figure 6.9: Summary of results for building wide experiment.

From figure 6.9 we surprisingly observe that *Eager Cell Switching* shows the poorest performance. *Eager Cell Switching* initiates many handoffs as each network disappears and reappears several times along the path. Often it tries to perform handoffs to networks which are only sporadicly available. This results in a very poor performance as can be seen from the following scenario:

1. The mobile node has selected a stable network as primary network.
2. A router advertisement is received from a new network. The *Eager Cell Switching* algorithm initiates a handoff to the new network.
3. The new primary network becomes unreachable, but the lifetime of this network has not yet expired.
4. A router advertisement is received from the stable network, but this is not taken as an indication of a new network as an entry for this network already exists in the network prefix list.
5. The entry for the new primary network in the network prefix list eventually expires as no new router advertisements from that network are received.
6. A handoff to the stable network is initiated as this is the only reachable network.

In this scenario packets are lost in the period from 3) until 6). In the building wide experiment networks often become sporadicly available, why the above described scenario is likely to occur. We expect that networks will become sporadicly available in many types of scenarios and when using many different link layer technologies. Therefore we believe that *Eager Cell Switching* seldom will constitute a good choice of a handoff initiation algorithm.

Lazy Cell Switching performs much better. Compared to *Eager Cell Switching* it has reduced the number of handoffs significantly and lost only 10 packets. From the log file it was observed that these packets were single packets dropped when approaching the maximum range of the primary network.

Parametric Cell Switching shows excellent performance. It keeps the number of handoffs at a minimum of two handoffs in this scenario. The number of handoffs is optimal because *Parametric Cell Switching* only initiates a handoff when a network is present which has significantly better signal-to-noise ratio than the primary network and has a stable low round trip time to the access router. In this scenario the signal-to-noise ratio behaves in a way that gives *Parametric Cell Switching* opportunity to handoff to a new network before the connection to the primary network becomes unstable.

Whether the locations of the access routers favors *Parametric Cell Switching* is difficult to say. Other scenarios might result in different performance for all three handoff initiation algorithms. To determine, whether the observed relative performance of the three handoff initiation algorithms is what can be generally expected, we need to conduct experiments in other scenarios. However, we are confident, that due to its use of link layer information *Parametric Cell Switching* will always perform better than both *Eager Cell Switching* and *Lazy Cell Switching*.

The cost of using our implementation of the *Parametric Cell Switching* algorithm, is that it causes a little extra network load. Every 0.5 seconds the algorithm sends an echo request to the default router at all available networks. The default routers are expected to reply the echo. These echo requests are sent for three reasons:

- We cannot measure the signal-to-noise ratio of a link to a network if there is no traffic at the network. By periodically performing pings we ensure that there is traffic at all the available networks.
- By performing periodic pings to each available network we can determine faster that a network has become unreachable than by monitoring the lifetime of network prefixes.
- We want an indication of the capacity of a network. The round trip time determined by performing a ping can give us this indication.

As the last reason is of least importance we suggest the following approach to reduce the number of pings. The mobile node can monitor how often it receives packets from each available network. If it has not received a packet for 0.5 seconds it sends a ping to the default router at the network to cause traffic and to investigate whether the network is still available.

6.4 Summary

In this chapter we have presented the design and implementation of a working prototype of the *Parametric Cell Switching* concept. The prototype is stable and there are no known bugs.

We have subjected the prototype to two experiments in the Mobile IPv6 testbed to determine whether the model developed in chapter 5 conformed with the realities of the implementation and to compare the performance of *Parametric Cell Switching* with the performance of *Eager Cell Switching* and *Lazy Cell Switching*. We have found that the theoretical model conforms to the results obtained in the testbed. This result also confirms that *Parametric Cell Switching*, in a scenario where link layer information is unavailable, is almost as fast as *Eager Cell Switching*.

Finally we introduced all three handoff initiation strategies to a more realistic scenario using wireless interfaces at access routers deployed at the Department of Computer Science, Aalborg University. In this scenario we have showed that by the use of link layer information the implementation of *Parametric Cell Switching* is able to perform seamless handoffs and cause no lost packets. In the same scenario both *Eager Cell Switching* and *Lazy Cell Switching* looses packets and attempts to perform more handoffs.

Chapter 7

Conclusions and future work

This thesis concerns handoff initiation strategies in the Mobile IPv6 protocol. The objective of our work has been to suggest a handoff initiation strategy enabling a roaming mobile node to perform seamless handoffs. We have focused on *proactive handoff initiation* as a method of reducing the handoff latency by initiating a handoff before the primary network becomes unreachable. It has also been an objective that the proposed handoff initiation strategy should require no assistance from special network entities. To approach these objectives we applied the following strategy:

Theoretical assessment of existing handoff initiation algorithms: We decided to perform a theoretical study of the performance of the two most common existing handoff initiation strategies, *Eager Cell Switching* and *Lazy Cell Switching*. This choice was made as these two handoff initiation strategies have never been thoroughly investigated and the influence of various protocol settings on handoff performance were not well understood. Our goal was to clarify the relation between handoff performance and protocol configuration.

Empirical assessment of existing handoff initiation algorithms: In parallel with theoretical reasoning we decided to obtain empirical evidence regarding the performance of *Eager Cell Switching* and *Lazy Cell Switching*. This was undertaken as a method of verifying theoretically obtained results and to suggest which configuration parameters were essential to the theoretical study and which were not. Our goal was to investigate the conformance between theory and practice.

Specification of new proactive handoff initiation algorithm: Based on obtained results from the theoretical and the empirical study we decided to specify a new proactive handoff initiation algorithm denoted *Parametric Cell Switching*. This choice was made, as both *Eager Cell Switching* and *Lazy Cell Switching* were found to have serious generic performance lacks. Our goal was to show the feasibility of a proactive handoff approach able to consider link layer information.

In chapter 3 we developed mathematical models for both the *Eager Cell Switching* handoff initiation strategy and the *Lazy Cell Switching* handoff initiation strategy. Using these models, we are able to predict the handoff latency as a function of the frequency of broadcasting unsolicited router advertisements from access routers and of the lifetime of broadcasted network prefixes.

We see several applications for these mathematical models including assisting in the choice of a handoff initiation strategy and to predict in what range handoff latencies can be expected. An important application of the mathematical models lies in optimizing protocol parameters. This leads us to propose a new and optimized configuration of Mobile IPv6 access routers resulting in a significant reduction of handoff latency without increasing network load. A novelty of the mathematical model proposed for *Eager Cell Switching* is that it reflects how fast a new network is discovered upon entering a link. The outcome of this model is thus applicable for all handoff initiation strategies discovering new networks by the reception of unsolicited router advertisements.

To investigate the relationship between theoretically predicted handoff latency and handoff latency experienced in an actual network we decided to setup a testbed running Mobile IPv6. In chapter 4 we designed a Mobile IPv6 testbed, suggested a set of experiments and presented the results of an extensive empirical study of handoff latency for both the *Eager Cell Switching* and the *Lazy Cell Switching* algorithms. We found, that the handoff latencies obtained in the testbed essentially conformed to those predicted using the mathematical models for a wide range of different configurations of access routers. The empirical experiments also confirmed, that our optimized Mobile IPv6 protocol configuration did in fact yield the expected reduction of handoff latency. Based on this empirical evidence we conclude that our mathematical models essentially conforms to the implementations of the *Eager Cell Switching* handoff initiation algorithm and the *Lazy Cell Switching* handoff initiation algorithm provided by the KAME Mobile IPv6 software.

Even when using our optimized configuration, both *Eager Cell Switching* and *Lazy Cell Switching* were found to have serious performance lacks. The *Lazy Cell Switching* handoff initiation strategy yields an unacceptable handoff latency in the range of several seconds. The *Eager Cell Switching* strategy was able to avoid packet loss by initiating a handoff before the current primary network becomes unreachable. Initially this strategy seems like a better choice, but it initiates far too many handoffs resulting in increased network load and in the selection of unstable networks as primary network. We conclude, that neither of the two handoff initiation strategies are capable of providing a satisfying handoff performance.

In chapter 5 we pursued the idea that a handoff initiation strategy should be proactive, so that it is able to initiate a handoff before the current primary network becomes unavailable. We investigated how link layer information can assist in a handoff initiation strategy and proposed a strategy also accounting for special metrics such as price and indications of available bandwidth. We showed, that theoretically this advanced handoff initiation algorithm is able to initiate a handoff almost as fast as the *Eager Cell Switching* algorithm in scenarios where link layer information is unavailable, but without the disadvantage of initiating as many unnecessary handoffs.

Chapter 6 contains the implementation effort needed to instantiate the advanced handoff initiation strategy. We implemented *Parametric Cell Switching* into the KAME Mobile IPv6 software as a working prototype. We performed experiments which showed, that the theoretically predicted handoff latency for this strategy conforms to the handoff latency experienced in the testbed. In an informal building wide experiment using three base stations at three different IPv6 networks and Wireless LAN as the link media, we demonstrated the feasibility of using link layer information in an advanced handoff initiation algorithm. In this realistic experiment *Parametric Cell Switching* was able to reduce packet loss as well as the number of handoffs to a minimum, while both *Lazy Cell Switching* and especially *Eager Cell Switching* performed poorly. We conclude, that using link layer information to assist in a handoff decision is feasible, and that by using link layer information it is indeed possible to combine the responsiveness of *Eager Cell Switching* with the ability of *Lazy Cell Switching* to reduce the number of handoffs.

To summarize, we have made two main contributions to the field of handoff initiation strategies in Mobile IPv6:

- We have established the relationship between the configuration of essential protocol parameters and handoff latency using either the *Lazy Cell Switching* or the *Eager Cell Switching* handoff initiation strategies. Based on this relationship we propose a new default Mobile IPv6 protocol configuration which for both handoff initiation strategies, compared to the current default configuration, results in significantly lower handoff latency while maintaining an equal network load.
- We have showed the feasibility of using a more advanced *proactive handoff initiation* strategy. By means of a prototype implementation, we demonstrate a mobile node roaming between several networks without losing either packets or connections.

Our work suggests that taking a proactive approach to handoff initiation, in zones where two or more networks are reachable, can result in seamless handoffs. A further novelty of proactive handoff initiation is, that a proactive handoff initiation strategy is still able to take full advantage of a reduction of the handoff execution time, should research in this area prove successful.

When deploying Mobile IP in the Internet it is essential, that AAA actions are performed to ensure that only authorized mobile nodes connect to a network and to address billing issues. AAA is likely to add a substantial overhead to the time it takes to perform a handoff, an overhead which must be compensated for by the handoff initiation algorithm. Only a handoff initiation algorithm that acts proactively will be able to compensate for this overhead and perform seamless handoffs.

We believe that by means of proactive handoff initiation it is indeed possible to achieve low latency and even seamless handoffs when Mobile IPv6 is deployed in the Internet. Should this prove correct, we are looking at the next Internet revolution, as omnipresent access to the Internet will result in a whole range of new services. For example, possibilities are that research in the field of quality of service protocols will eventually allow e.g. Voice over IP to achieve adequate quality. When combined with Mobile IP this becomes an application with a huge potential.

We therefore strongly recommend that proactive handoff initiation receives further attention and that future work should be dedicated to this particular field.

Future work

There are numerous aspects of our work which requires further attention. Below we have listed some of the most urgent tasks which need to be completed:

Redo local area experiments: To further verify the conformance between the mathematical models and the KAME Mobile IPv6 implementation, a larger amount of samples must be collected in experiments. Currently, experiments conducted in the Mobile IPv6 testbed only include between 300 and 400 handoffs for each configuration. This is due to a memory leakage in the KAME software which causes a correspondent node to crash after a certain number of sent packets. We suggest that at least 5000 samples are collected in each experiment.

New building wide experiments: Our building wide experiment should be performed with the access routers placed at different locations in order to further investigate the performance of *Parametric Cell Switching*.

To gather further information about the behavior of proactive handoff initiation in Mobile IP more experiments in realistic scenarios must be performed. Currently we have mostly been concerned with the behavior of handoff initiation algorithms within a local area testbed. The effects of deploying Mobile IP in a larger environment, with possibly many mobile nodes simultaneously attached through the same access router or access point must be studied.

Handoff criteria: We have not engaged in a study of how the handoff criteria currently used in *Parametric Cell Switching* should be weighted. The currently suggested weights are a mere ad hoc setting. Experiments with different weight settings therefore need to be conducted in order to optimize performance.

More handoff criteria which can contribute to the decision of handoff initiation must also be studied. This should include criteria at all layers in the protocol stack.

Besides from these short term tasks, we list some important issues which must be addressed in order to ensure future development of the Mobile IPv6 protocol:

Deployment: The Mobile IPv6 protocol has never been deployed in an actual set of networks. A large scale deployment of Mobile IPv6 would allow for the collection of important empirical evidence as to whether the generic Mobile IPv6 specification is scalable or whether it needs adjustments.

Authentication, Authorization and Accounting: It is not reasonable to expect that Mobile IPv6 should have any success before AAA issues are solved. Currently many AAA issues is not yet agreed upon, including the specifics of how a mobile node should authenticate itself at a foreign network.

Other interesting issues regarding Mobile IPv6 which could be addressed are the following:

Network assisted handoff initiation: In this thesis we have put forward the view, that a handoff initiation algorithm should not *require* that special entities within networks are present to assist in a handoff decision. It could be interesting to investigate to what extend such a service could improve proactive handoff initiation if available.

Quality of service protocols: In the deployment of time sensitive applications quality of service protocols are essential. It is important that Mobile IP and quality of service protocols can interact without any disparities. If this can be achieved a wide range of applications becomes attractive in the market of mobile computing. These applications include Voice over IP and mobile video conferencing.

Simulations: In our work we have experienced, that even conceptually simple handoff initiation algorithms requires rather complex theoretical reasoning to determine performance. It would be interesting to investigate the possibilities of simulating different handoff initiation algorithms in large scale scenarios. For instance a Mobile IPv6 extension for the network simulator ns-2 has become available just prior to the publication of this master thesis.

Bibliography

[Andersen *et al.*, 2000] Torben W. Andersen and Anders Lildballe, *’Handoff in Mobile IPv6’*, 9th semester report, Department of Computer Science, Aalborg University, January 2001.

[Anderson *et al.*, 1999] David R. Anderson, Dennis J. Sweeney and Thomas A. Williams, *’Statistics for business and economics’*, Seventh edition, South-Western College Publishing 1999, ISBN 0-538-87593-3.

[Atkinson, 1995] R. Atkinson *’Security Architecture for the Internet Protocol’*, RFC 1825, August 1995.

[Bhatia] Medhavi Bhatia, *’QoS over the Internet: The RSVP protocol’*. Available at http://www.rvs.uni-hannover.de/people/neitzner /Studienarbeit/literatur/more/rsvp.html.

[Braden *et al.*, 1997] R. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin, *’Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification’*, RFC 2205, September 1997.

[Bradner, 1996] S. Bradner, *’The Internet Standards Process – Revision 3’*, RFC 2026, October 1996.

[Comer, 1997] Douglas E. Comer, *’Computer Networks and Internets’*, Prentice-Hall, 1997, ISBN 0-13-239070-1.

[Deering, 1991] S. Deering, *’ICMP Router Discovery Messages’*, RFC 1256, September 1991.

[Dummynet] Dummynet flexible IP bandwidth manager. Available at http://www.iet.unipi.it/∼luigi/ip_dummynet/

[Ericsson GSM] Ericsson GSM systems homepage. Available at http://www.ericsson.com/gsm.

[Ericsson Press] Ericsson Press Releases, *’Ericsson and Viasat drive mobile Internet in the car with automotive GPRS’*. Available at http://www.ericsson.se/press/20001107-0128.html.

[Flanagan, 1999] William Flanagan, *’Technologies Leading the Communications CommVergence’*, October 1999. Available at http://www.tbg.com/Public/WhitePapers/CommVergence.html.

[FreeBSD] FreeBSD homepage. Available at http://www.freebsd.org.

[Granbohm *et al.*, 1999] Håkan Granbohm and Joakim Wiklund, *’GPRS – General packet radio service’*, in *Ericsson Review No. 2*, pages 82-88, 1999.

[Guardini *et al.*] I. Guardini, P.D’Urso and P. Fasano, *’The role of internet technology in future mobile data systems’*. Available at http://www.cselt.it.

[Huitema, 1997] Christian Huitema, *’IPv6 - The new Internet protocol’*, second edition, Prentice-Hall, 1997, ISBN 0-13-850505-5.

[IGEB] Interagency GPS Executive Board homepage. Available at http://www.igeb.gov/.

[IP, 1981] Information Sciences Institute University of Southern California, *’Internet Protocol’*, RFC 791, September 1981.

[Johnson *et al.*, 2000] David B. Johnson and Charles Perkins, *’Mobility Support in IPv6’*, Internet-Draft, draft-ietf-mobileip-ipv6-13.txt, November 2000. Work in progress.

[KAME] KAME homepage. Available at http://www.kame.net

[Kotz *et al.*, 1999] David Kotz and Robert S. Gray, *’Mobile Agents and the Future of the Internet’*, in *ACM Operating Systems Review, 33(3)*, pages 7-13, August 1999. Available at http://www.cs.dartmouth.edu/∼dfk/papers/kotz:future2/.

[La Porta *et al.*, 2000] T. La Porta and L. Salgarelli, *’IP micro-mobility support using HAWAII’*, Internet-Draft, draft-ietf-mobileip-hawaii-01.txt, July 2000. Work in progress.

[Mouly *et al.*, 1992] Michel Mouly, Marie-Bernadette Pautet *’The GSM System for Mobile Communications’*, 1992, ISBN 2-9507190-0-7.

[Narten *et al.*, 1998] T. Narten, E. Nordmark, W. Simpson, *’Neighbor Discovery for IP Version 6 (IPv6)’*, RFC 2461, December 1998.

[ORiNOCO] ORiNOCO homepage, *’ORiNOCO PC Card (Silver/Gold)’*. Available at http://www.orinocowireless.com/.

[Pahlavan *et al.*] Kaveh Pahlavan, Prashant Krishnamurthy, Juha Mäkelä and Mika Ylianttila, *’Inter-technology Mobility Testbed’*, Centre for Wireless Communications at University of Oulu, Center for Wireles Information Network Studies at Worcester Polytechnic Institute.

[Perkins, 1996] Charles E. Perkins, *’IP Mobility Support’*, RFC 2002, October 1996.

[Perkins, 1998] Charles E. Perkins, *’Mobile IP - Design Principles and Practices’*, Addison-Wesley, 1998, ISBN 0-13-850505-5.

[Perkins, 2000A] Charles Perkins, *’IP Mobility Support for IPv4, revised’*, Internet-Draft, draft-ietf-mobileip-rfc2002-bis-04.txt, 20 February 2000. Work in progress.

[Perkins, 2000B]	Charles E. Perkins, *’AAA for IPv6 Network Access’*, Internet-Draft, draft-perkins-aaav6-03.txt, November 2000. Work in progress.
[Ramjee *et al.*, 1999]	R. Ramjee, T. La Porta, S. Thuel, K. Varadhan and S.Y. Wang, *’HAWAII: A Domain-Based Approach for Supporting Mobility in Wide-area Wireless networks’*, 1999. Available at http://redefine.dyndns.org/∼andyr/documents_and_writings/cs497rhk /areitz-Hawaii.html
[Rose *et al.*, 2000]	L. Rose and V. Magret, Alcatel, *’Simple Multicast Mobility Protocol’*, Internet-Draft, draft-rose-mobileip-smm-00.txt, July 2000. Work in progress.
[Scourias]	John Scourias, *’Overview of the Global System for Mobile Communications’*. Available at http://ccnga.uwaterloo.ca/∼jscouria/GSM/gsmreport.html.
[Seymour, 2000]	Jim Seymour, *’The Key to Connectivity: Packet vs. Circuit Switching’*, In *TheStreet.com*, 3th of August 2000, Available at http://www.thestreet.com/comment/techsavvy/1026391.html
[Soliman *et al.*, 2000]	Hesham Soliman, Claude Castelluccia, Karim El-Malki and Ludovic Bellier, *’Hierarchical MIPv6 mobility management’*, Internet-Draft, draft-soliman-mobileip-hmipv6-03.txt, February 2000. Work in progress.
[Tanenbaum, 1996]	Andrew S. Tanenbaum, *’Computer Networks’*, Third Edition, Prentice-Hall, 1996, ISBN 0-13-394248-1.
[Tsirtsis, 2001]	G. Tsirtsis, A. Yegin, C. Perkins, G. Dommety, K. El-Malki, M. Khalil, *’Fast Handovers for Mobile IPv6’*, Internet-Draft, draft-ietf-mobileip-fast-mipv6-01.txt, April 2001. Work in progress.
[Varshney *et al.*, 2000]	Upkar Varshney and Ron Vetter, *’Emerging Mobile and Wireless Networks’*, Communications of the ACM 43, 6 (June 2000), pages 73-81.
[Walpole *et al.*, 1998]	Ronald E. Walpole, Raymond H. Myers and Sharon L. Myers, *’Probability and Statistics for Engineers and Scientists’*, Sixth edition, Prentice Hall 1998, ISBN 0-13-095246-X.
[Yegin *et al.*, 2000]	Alper E. Yegin, Mohan Parthasaraty and Carl Williams, Sun Microsystems, *’Mobile IPv6 Neighborhood Routing for Fast Handoff’*, Internet-Draft, draft-yegin-mobileip-nrouting-01.txt, November 2000. Work in progress.
[Zakon]	Robert H’obbes’ Zakon *’Hobbes’ Internet Timeline v5.2’*. Available at http://www.zakon.org/robert/internet/timeline/.

Appendix A

Mobile IPv6

This appendix describes the main parts of Mobile IPv6 based on the specification in [Johnson *et al.*, 2000]. The following topics are presented:

- Notation of IPv6 addresses.
- Features of IPv6 that Mobile IPv6 utilizes.
- Message types in Mobile IPv6.
- Data structures in Mobile IPv6.
- Operation of Mobile IPv6, which includes home agent registration, triangle routing, route optimization, binding management and dynamic home agent address discovery.
- Handoffs in Mobile IPv6.
- Requirements to correspondent nodes, mobile nodes and home agents in Mobile IPv6.

But first an important definition. A **binding** in Mobile IPv6 is the term for a relation between two IP addresses. The binding

$$(\text{address}_1, \text{address}_2)$$

expresses, that address_1 should be used instead of address_2. A typical binding is a binding between a mobile node's care-of address and its home address. Often a lifetime is associated with a binding.

A.1 IPv6 addresses

An IPv6 address is composed of 128 bits. These 128 bits are written as eight 16-bit integers separated by colons. Each integer is represented by four hexadecimal digits, as in

fedc:ba98:7654:3210:fedc:ba98:7654:3210

Even though hexadecimal notation is relatively compact, it is still tedious to enter the numbers. Abbreviations are allowed, as all 128 bit might not be used. Consider a number as

1080:0000:0000:0000:0008:0800:200c:417a

It is allowed to skip leading zeros of each hexadecimal component, that is 0000 becomes 0, 0008 becomes 8 and 0800 becomes 800. This leaves us with

1080:0:0:0:8:800:200c:417a

A further possibility for abbreviation is the double-colon notation. A set of consecutive null 16-bit numbers can be replaced by two colons. The previous example becomes

1080::8:800:200c:417a

A double colon can only be used once in an address. Expanding an abbreviation is simple. Align whatever is at the left of the double colon to the left of the address. Then align whatever is at the right of the double colon to the right of the address and fill up with zeros. Some examples

fedc:ba98::7654:3210 → fedc:ba98:0:0:0:0:7654:3210
fedc:ba98:7654:3210:: → fedc:ba98:7654:3210:0:0:0:0
::fedc:ba98:7654:3210 → 0:0:0:0:fedc:ba98:7654:3210

Some IPv6 addresses is obtained by prepending 96 zero bits to an IPv4 address. To reduce the risk of making errors in the transformation of the dot-decimal notation of IPv4 to the colon-hexadecimal notation of IPv6, there is a specific format for these addresses. Instead of writing

0:0:0:0:0:0:a00:1

the last 32 bits may be written in dot-decimal form as in

::10.0.0.1

The notation for prefixes, i.e., the high order bits of addresses used by routing protocols, is derived from IPv4. The notation is a regular IPv6 address followed by a slash and a number of bits. For example

fedc:ba98:7600::/40

describes a 40-bit long prefix whose hexadecimal value is

fedcba9876

Some IPv6 addresses have been reserved for special purposes. Among these are the link-local addresses and the site-local addresses. The prefixes in IPv6 for link- and site-local addresses are

Link-local	fe80::/10
Site-local	fec0::/10

Link-local addresses are addresses, that only can be used to address other nodes at the local network. Local means, that the nodes are directly connected to the same bus. Packets to or from link-local addresses are never routed to other networks. A link-local address can be composed of the prefix for link-local addresses and an unique host ID based on the unique IEEE-802 media address for the network device. IEEE has defined a set of standards for Local Area Networks (LAN) known as IEEE-802. Examples are 802.3, which is also known as Ethernet, 802.4, which is Token Ring and 802.11, which is a wireless data link layer protocol. A **media address** is an unique 48 bit number, that is assigned to each IEEE-802 device according to the standards for LANs.

Site-local addresses are addresses local to the site. This means, that a set of connected local networks can use site-local addresses. Site-local addresses must not be routed outside the site. Site-local addresses are also used in IPv4.

A.2 IPv6 features

Mobile IPv6 is based on version 6 of the IP protocol. Therefore it is natural, that Mobile IPv6 exploits a set of features present in IPv6. The main features from this set is [Huitema, 1997]:

Router advertisements are messages, that routers transmit on the networks they serve to inform hosts about their presence and IP address. A router advertisement among other things contains the network prefix of the network and the address of the router that sent the advertisement [Huitema, 1997].

Mobile IPv6 assumes that routers on a network transmits router advertisements periodically. Mobile IPv6 further assumes that routers that intend to serve mobile nodes, transmit router advertisements with smaller intervals (0.5 to 1.5 seconds [Johnson *et al.*, 2000]) than normal (7 to 10 minutes [Deering, 1991]). The router advertisements are used by a mobile node to determine the prefix of a network and thereby to discover new networks.

Neighbor discovery is a mechanism defined in IPv6 that ensures that a host knows the media addresses of other nodes directly attached to the host. When a host connects to a network it multicasts a **neighbor solicitation** message to other nodes at the network. The solicitation contains the media address of the host while the source of the solicitation is set to the host's link local address. Each node at the network replies to the host with a **neighbor advertisement** message. The advertisement contains the media address of the node's interface to the network and the source is the IP address of the node. If a node is a router it indicates this in its neighbor advertisement to the host. In this way all other nodes at the network should have received the media address and the IPv6 address of the host, while the host has received the media and IPv6 address of all other nodes at the network.

Mobile IPv6 exploits this feature to let a home agent intercept packets for a mobile node at the mobile node's home network. The home agent performs **proxy neighbor discovery** to emulate that a mobile node is directly connected to the home network.

The home agent sends neighbor solicitations and neighbor advertisements at the mobile node's home network on behalf of the mobile node. However the home agent puts its own media address in the neighbor solicitation and neighbor advertisements, so that hosts and routers at the home network that address packets for the mobile node, will send the packets to the home agent. This means that all packets for the mobile node will be sent to the home agent.

Neighbor discovery is used by mobile nodes to locate routers when they attaches to foreign networks.

Auto-configuration is a mechanism that allows a host to automatically discover and register parameters needed to connect to the Internet. IPv6 provides two types of auto-configuration:

Stateless auto-configuration. In stateless auto-configuration a host generates its own IP address based on the network prefix and the IEEE-802 address of its network interface. No server at the network must be consulted to form an IP address. However it may be necessary to contact servers for other configuration information like the address of the Domain Name Server (DNS).

Stateful auto-configuration uses the IPv6 version of the Dynamic Host Configuration Protocol (DHCP). DHCP operates in the following way: The host multicasts a message to all DHCP servers on the network. DHCP servers reply with the parameters that the host should use to configure itself. The parameters include the host's IPv6 address on the network, the domain name of the network, the address of the DNS server at the site, the address of the file server, etc.

In Mobile IPv6 mobile nodes use auto-configuration whenever they move to a foreign network.

A.3 Mobile IPv6 messages

Mobile IPv6 requires additional information to be exchanged between nodes. By additional is meant more information than is supplied in the standard IPv6 protocol. In IPv6 extra header information is supplied in so called **header extensions**. The header extensions form a daisy chain inserted between the IPv6 header and the payload data. Figure A.1 shows an IPv6 packet without header extensions and two packets with header extensions.

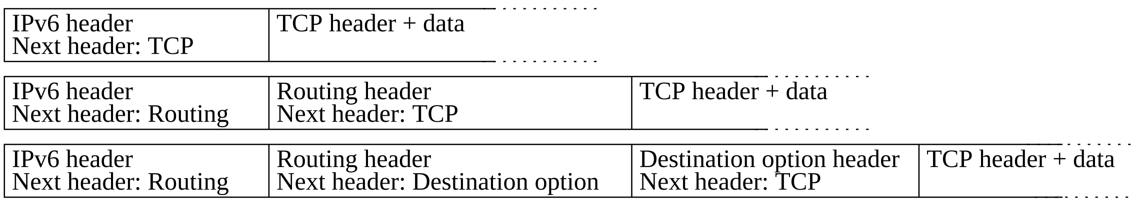


Figure A.1: Examples of packets without and with header extensions.

IPv6 offers an header extension named **destination option** [Huitema, 1997]. A feature of destination options is that they only need processing at the destination of the packet. Thus it is only the source and destination of a packet including a destination option that must understand the content of the destination option. Intermediate nodes ignore destination options.

Four new destination options are defined in Mobile IPv6 [Johnson *et al.*, 2000]. They carry the extra information that Mobile IPv6 nodes exchange. The destination options are:

Binding update. The *binding update* option is used by the mobile node to inform its home agent or any other correspondent node about its current care-of address.

Binding acknowledgment. The *binding acknowledgment* option is used to acknowledge the reception of a *binding update*, if an acknowledgment was requested.

Binding request. The *binding request* option can be used by any node to request a mobile node to send a *binding update* with the mobile node's current care-of address.

Home address. The *home address* option is used in a packet sent by a mobile node to inform the receiver of this packet about the mobile node's home address. A mobile node cannot set its home address as the source of a packet, when it is attached to a foreign network. The packet would be dropped by routers, that perform ingress filtering.

A router that performs **ingress filtering** checks the source of packets to see if it could be reached over the interface, that received the packet. If the packet could not origin from a node behind that interface it is dropped. This ensures, that a node cannot send a packet with a false source address. The mobile node will set its care-of address as source of the packet and provide its home address in a home address option. As the care-of address is a valid address at the foreign network, the packet will not be dropped in an ingress filter.

All destination options can be piggy-bagged[1] on a data packet. This reduces the overhead of exchanging mobility specific information. The formats of these options are specified in [Johnson *et al.*, 2000]. Examples of how they are used in Mobile IPv6 can be found in section A.5.

Besides the four destination options Mobile IPv6 defines two new Internet Control Message Protocol (ICMP) message types. ICMP messages are used in IP to report unexpected events, to send administrative messages and to test the network. An ICMP message is carried in an IP packet. Typical types of messages in ICMPv6 [Huitema, 1997] are:

- Destination unreachable.
- Packet too big.
- Time exceeded.
- Echo request.
- Echo reply.
- Router solicitation.
- Router advertisement.

---

[1] In a few cases it is not allowed to piggy-bag an option [Johnson *et al.*, 2000].

- Neighbor solicitation.
- Neighbor advertisement.

The two new ICMP message types defined for Mobile IPv6 are:

Home agent address discovery request, this message is used by a mobile nodes to initiate the *dynamic home agent address discovery mechanism* described in section A.5.5, in order to be able to register with a home agent at its home network.

Home agent address discovery reply, this message is used by a home agent at a mobile node's home network to reply to a *home agent address discovery request* message from the mobile node. The reply contains a prioritized list of routers providing home agent services at the mobile node's home network.

The use of the two message types is described in section A.5.5.

A.4 Mobile IPv6 data structures

Three conceptual data structures are used in the specification of Mobile IPv6 [Johnson *et al.*, 2000]. They are:

Binding cache. Every IPv6 node has a binding cache. A binding cache is used to hold the bindings for mobile nodes. If a node receives a binding update, it will add this binding to it's binding cache. Whenever a node is about to send a packet, it checks the binding cache. If there is an entry for the destination of the packet, the packet is instead sent to the care-of address found in the binding cache.

Binding update list. A mobile node maintains a binding update list of nodes that must receive binding updates. Every time a mobile node sends a *binding update* to a correspondent node or a home agent it adds or renews an entry for that node in the binding update list.

Home agent list. For each network directly attached to a router for which this router serves as a home agent, the router generates a home agent list. If a router serves as home agent for three networks, it generates three home agent lists; one for each network. A home agent list contains information about all home agents present at a network and these home agents' individual preference.

The information in a home agent list is learned from multicasted router advertisements. The information in the home agent lists is used by the dynamic home agent discovery mechanism presented in section A.5.5.

The use of the data structures just presented is illustrated by examples in section A.5.

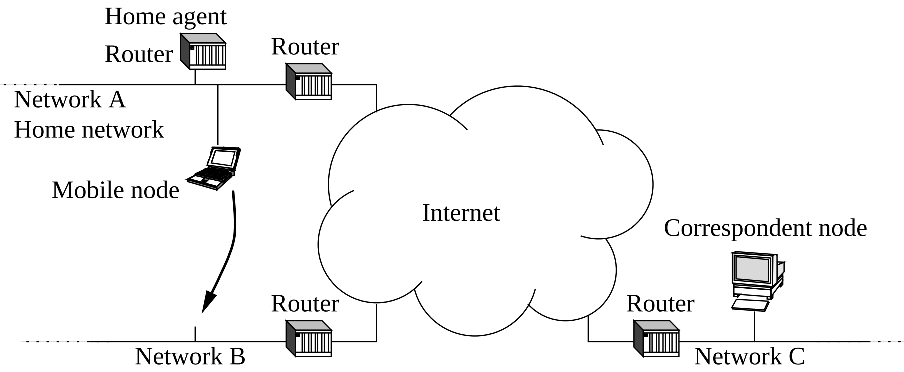


Figure A.2: Operation scenario for Mobile IPv6.

A.5 Mobile IPv6 operation

The mechanisms of Mobile IPv6 presented in this section are illustrated using the scenario in figure A.2.

Figure A.2 shows three networks, a home agent, a correspondent node and a mobile node. On network *A* resides a router serving as home agent for network *A*. Network *A* is also the home network for the mobile node. The mobile node leaves network *A* and moves to network *B*, which is a foreign network. On network *C* resides a correspondent node, that communicates with the mobile node. All three networks are connected to the Internet.

In the following subsections, which presents aspects of Mobile IPv6 operation, it is first assumed that the mobile node knows the IP address of its home agent. This address can be manually or automatically configured. The final subsection, subsection A.5.5, presents a mechanism, where the mobile node automatically learns its home agent's IP address while it is away from home.

A.5.1 Home agent registration

Movement is defined as a mobile node changing its point of attachment. While a mobile node is away from home, it selects one router as its default router and thereby the network prefix advertised by that router as the network prefix of its primary care-of address.

Mobile IPv6 specifies, that a mobile node can use any combination of mechanisms to detect that it has moved to another network [Johnson *et al.*, 2000]. Two possibilities have been presented in section 1.4.3, the *Eager Cell Switching* handoff initiation strategy and the *Lazy Cell Switching* handoff initiation strategy.

When a mobile node detects that it has moved from one network to a new network and it has discovered a new default router, the mobile node generates a new care-of address. The care-of address is generated using either *stateless address auto-configuration* or *stateful address auto-configuration* as described in section A.2. The prefix of the new care-of address is the prefix of the new network. This means that packets addressed to the care-of address will reach the mobile node at the new network.

In order to register its new care-of address with its home agent, the mobile node creates a *binding update* option message containing the new care-of address and the mobile node's home address. The *binding update* is sent to the mobile node's home agent. The home agent registers the binding by adding the binding to its *binding cache* and replies with a *binding acknowledgment* option message to the mobile node. These actions are illustrated in figure A.3.

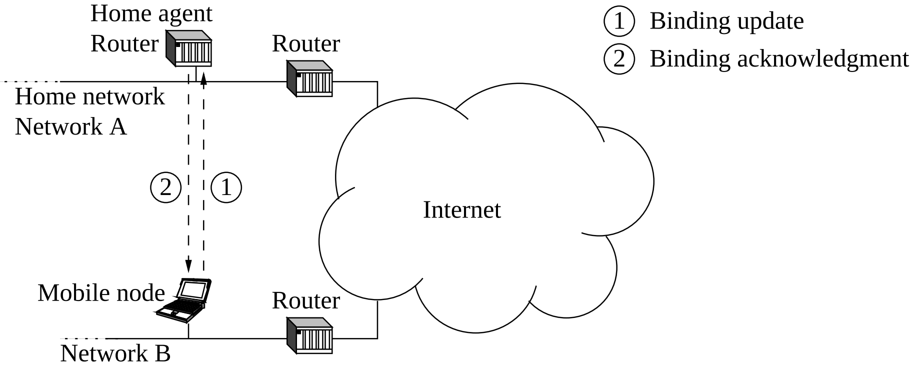


Figure A.3: The mobile node registers the new care-of address with its home agent.

When a mobile node moves between two foreign networks the registration of a new care-of address is performed in a similar way.

A.5.2 Triangle routing

When the mobile node has registered with the home agent, the home agent should intercept any packet addressed to the mobile node's home address. To intercept these packets the home agent uses proxy neighbor discovery as described in section A.2. When the home agent intercepts a packet to the mobile node, the home agent tunnels the packet to the registered care-of address of the mobile node using encapsulation. **Encapsulation** means that the original IPv6 packet destined for the mobile node is inserted as payload in another IPv6 packet before it is sent. The home agent is set as source and the mobile node's care-of address as destination of the packet. This concept is illustrated in figure A.4.

IPv6 header Next header: IPv6 Dst: Mobile node's care-of address Src: Home agent	IPv6 header Next header: TCP Dst: Mobile node's home address Src: Correspondent node	TCP header + data

Figure A.4: Headers for an intercepted packet, that is tunneled from the home agent to the mobile node using encapsulation.

When the mobile node sends packets to another node, it sends the packets directly to the node. The mobile node sets the source of this packet to the care-of address and includes its home address in a *home address* option. This is to avoid dropping the packet in an ingress filter as described in section A.3. When a packet with a *home address* option arrives at a node, the packet's source is exchange with the home address before the packet is passed on to the layer above the network layer. This concept hides the mobile node's care-of address from the transport and application layer protocols. In this way TCP connections and other sessions are not terminated, because a mobile node changes its point of attachment and thereby its care-of address.

When a mobile node communicates with a correspondent node while being away from home, packets are routed from the correspondent node to the home agent and from the home agent to the mobile node, while packets from the mobile node is routed directly to the correspondent node as illustrated in figure A.5.

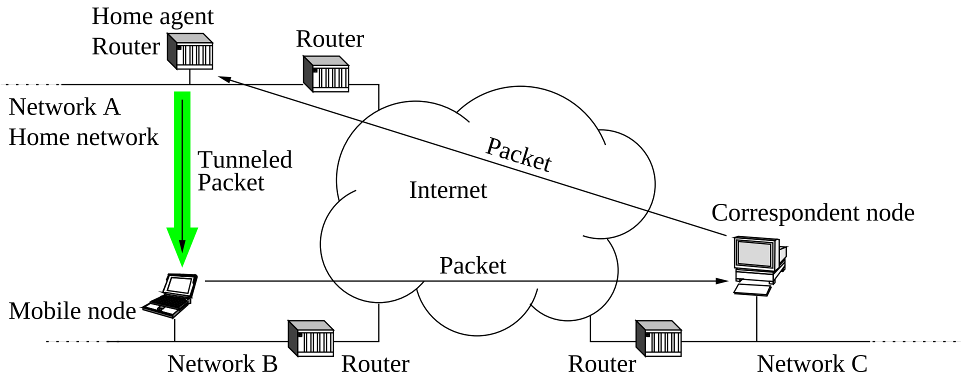


Figure A.5: Triangle routing.

This phenomenon is called **triangle routing**. If a mobile node's point of attachment is far from the home agent, triangle routing can cause a significant overhead compared to the direct route between a correspondent node and a mobile node.

A.5.3 Route optimization

To avoid triangle routing a mobile node can send *binding updates* to a correspondent node. This allows a correspondent node to cache the mobile node's current care-of address and send packets directly to the mobile node's point of attachment and not via the home agent. This is called **route optimization**. The initial steps in route optimization are illustrated in figure A.6.

Only the first, few packets from the correspondent node to the mobile node need to travel via the mobile nodes home agent. When the correspondent node receives the *binding update*, it can send the packets directly to the mobile node.

Any IPv6 node wanting to send a packet must first check its *binding cache* for the packet's destination address. If an entry is found, a routing header containing the mobile node's care-of address is added to the packet. The **routing header** indicates, that the packet should first be routed to the mobile node's care-of address and then should be routed to the mobile node's home address. The packet arrives at the mobile node which swap the destination address with the address in the routing header. The mobile node discovers, that the destination now is its home address and pass the packet on to the transport layer. Figure A.7 shows how the packet is looped back at the mobile node.

One may ask why routing headers are used, why can the mobile node not just replace the care-of address in the destination field with its home address and then pass the packet on to

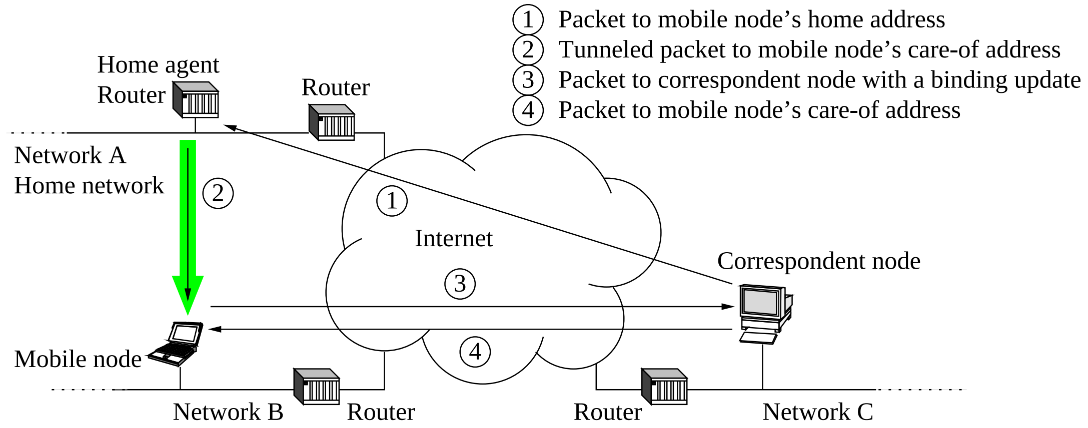


Figure A.6: The initial steps in route optimization.

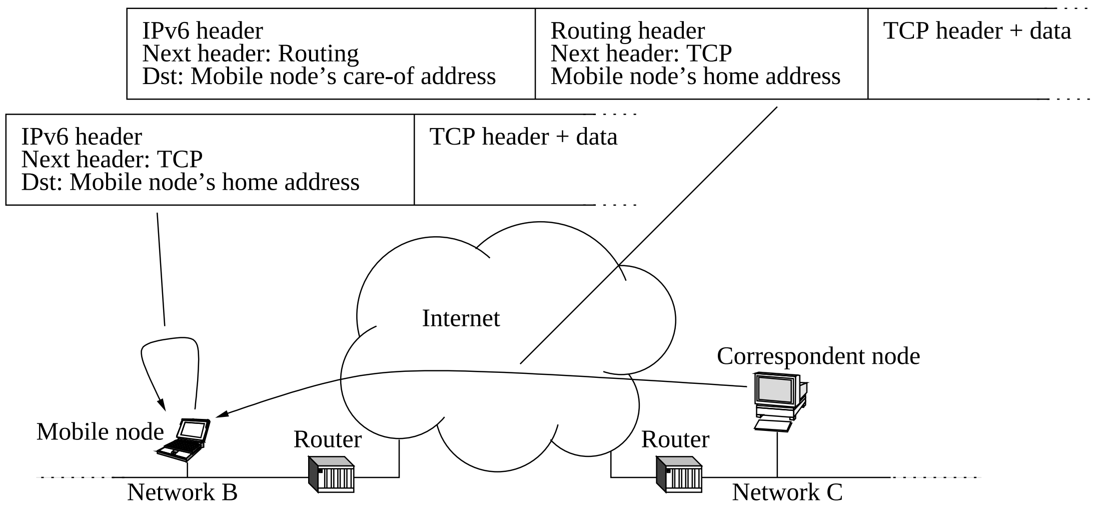


Figure A.7: A packet sent from a correspondent node directly to a mobile node using a routing header.

the transport layer the first time the packet arrives. If *stateless auto-configuration* is used and the mobile node's care-of address happens not to be unique at the network, the mobile node may receive packets, which is not intended for it. The same situation can also occur if *stateful auto-configuration* is used and a set of addresses reserved for mobile nodes is recycled. Then a mobile node could be assigned a care-of address, to which a correspondent node has associated another mobile node. In these situations the routing header should ensure, that the mobile node does not pass the packet on to the transport layer protocols, but either drops the packet or sends it back out on the network, where it probably will be dropped by an ingress filter. The reason for using a routing header instead of encapsulation is that an IP packet with a routing header introduces less extra header bytes than an IP-in-IP encapsulated packet.

If no entry is found in the *binding cache* the packet is sent to the mobile node's home address. When the packet arrives at the mobile node's home network, the home agent intercepts the packet and tunnels it to the mobile node's care-of address.

A.5.4 Binding management

A mobile node, that has configured a new care-of address as its primary care-of address must register the care-of address with its home agent and all the correspondent nodes that has an entry for the mobile node's home address. The new care-of address is provided to the home agent and correspondent nodes through a *binding update*. When the mobile node sends a *binding update* to a node, it can enforce the receiver to send an acknowledgment on receipt of the *binding update*. This is enforced by setting the acknowledgment bit in the *binding update*. If the mobile node does not receive a *binding acknowledgment*, it should retransmit the *binding update*.

When a mobile node away from home receives a packet from a correspondent node it can detect whether the correspondent node has an entry for the mobile node in its *binding cache* or not. If the packet was sent directly to the mobile node using a routing header, the correspondent node does have an entry in its *binding cache*. If the packet was tunneled from the home agent, the packet is encapsulated, which means that the correspondent node does not have an entry in its *binding cache*. The mobile node may choose to send a *binding update* to the correspondent node to enable it to send future packets directly to the mobile node.

The mobile node should always be reachable through its home address. This means that the home agent should always know the mobile node's current point of attachment. Therefore a mobile node **must** set the acknowledgment bit in a *binding update* addressed to a home agent. The home agent must thus acknowledge the receipt of a *binding update*. An acknowledgment is not required from correspondent nodes. The mobile node can detect whether a correspondent node has inserted an entry for the mobile node in its binding cache, when the mobile node receives a packet directly from the correspondent node.

Bindings have a lifetime specified by the mobile node. When the lifetime of a binding expires, that binding is removed from the *binding cache*. The correspondent node may choose to send a *binding request* to the mobile node, before the lifetime of the binding expires. The mobile node may choose to reply with a *binding update*. If the mobile node sends a binding to a correspondent node with lifetime set to zero, any binding for the mobile node should be removed. If the mobile node sends a binding with lifetime set to infinity, that binding should never expire. An infinite binding can only be removed by a *binding update* with lifetime set to zero.

A.5.5 Home agent discovery mechanism

If a mobile node away from home does not know the IP address of its home agent, Mobile IPv6 provides a mechanism that allows a mobile node to dynamically discover the IP address of a home agent at its home network.

Mobile IPv6 defines an anycast address [Huitema, 1997] that is reserved for home agents. When a packet for an **anycast address** at a network is received by a router at the network, the router forwards the packet to the nearest, operational node that provides the service associated with the anycast address.

The mobile node sends a *home agent address discovery request* to the anycast address reserved for home agents at its home network using its care-of address as source. The request packet is routed to the nearest operational home agent at the mobile node's home network. When a home agent receives a *home agent address discovery request*, it replies with a *home agent address discovery reply*, which contains the *home agent list* that the home agent has generated for that network. This is illustrated in figure A.8.

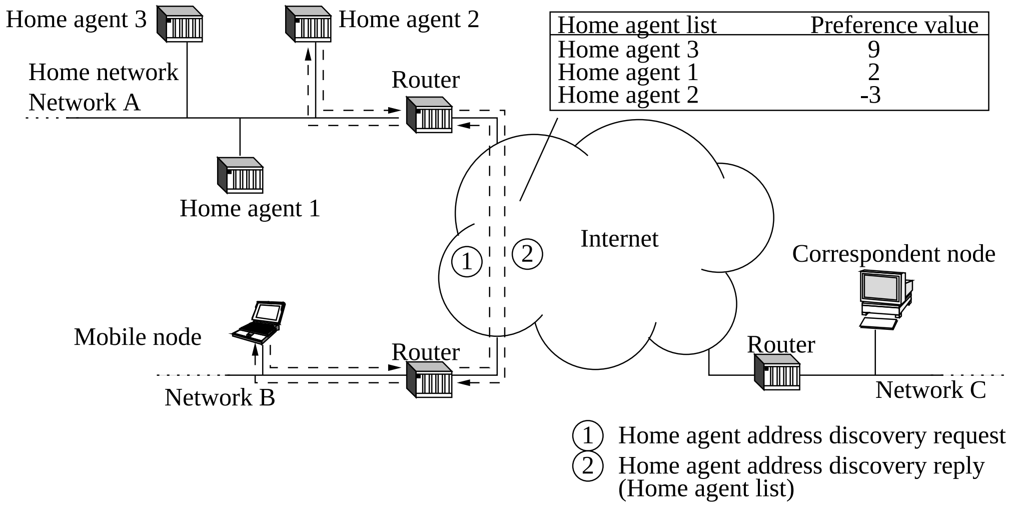


Figure A.8: Home agent address discovery. The mobile node sends a request and a home agent replies with a *home agent list*.

The home agent list contains a prioritized list of the home agents at the network. When the mobile node receives the home agent list, it should send a *binding update* to the first home agent in the list to register its care-of address as described in section A.5.1. If the mobile node does not get a *binding acknowledgment* it may try to register with other home agents in the list.

The purpose of this mechanism is to make Mobile IPv6 robust to single point of failures. Dynamic home agent address discovery makes it simple to have several home agents serving the same network. If a mobile node has registered its care-of address with a home agent and this home agent fails, the mobile node can register with another home agent.

A.6 Handoff in Mobile IPv6

In the previous sections the general mechanisms of Mobile IPv6 has been presented. This section presents the details of handoff in Mobile IPv6.

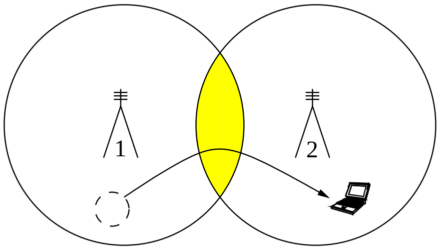


Figure A.9: Scenario for handoff description.

To illustrated the details of a handoff a simple scenario is set up. The scenario is depicted in figure A.9. The scenario depicts two base stations that provide access to two different networks. In this example it is assumed that the mobile node uses the *Eager Cell Switching* handoff initiation strategy described in section 1.4.3. A mobile node moves from the area covered by base station *1* to the area covered by base station *2*. First the mobile node is attached to the network connected to base station *1*. As it gets in range of base station *2*, it performs a handoff to the network connected to base station *2*.

Figure A.10 shows a diagram over the activities and events during the handoff from the network connected to base station *1* to the network connected to base station *2* as experienced by the mobile node.

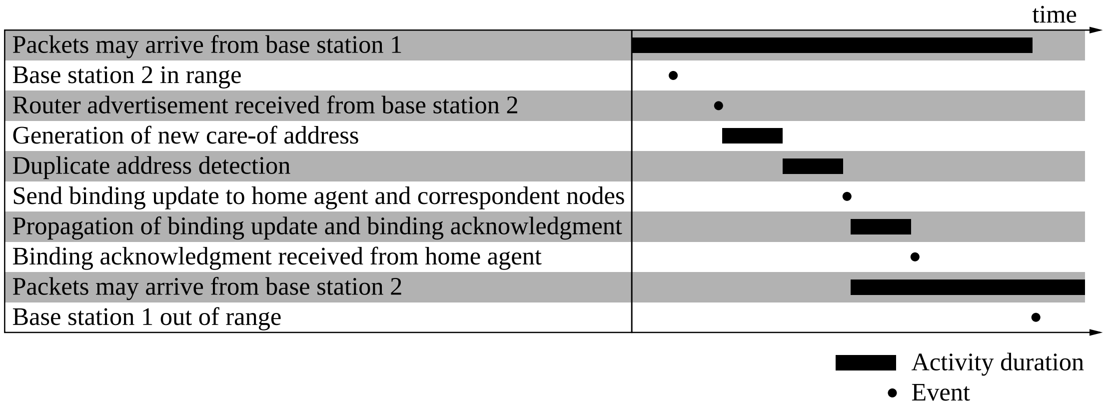


Figure A.10: Activities and events during a handoff.

In figure A.10 bars indicate the duration of an activity and dots indicate the occurrence of an event. Note that the mobile node can receive packets from base station *1* as long as it is within range, while it should not receive data packets from base station *2* before it has transmitted a *binding update* to its home agent and correspondent nodes.

There are three main time consuming activities during a handoff, apart from discovering a new network. They are the generation of a new care-of address, *duplicate address detection* and the propagation of a *binding update* and a *binding acknowledgment*.

Generation of a new care-of address happens through *stateless* or *stateful auto-configuration*. If *stateless auto-configuration* is used, a new care-of address can be generated in milliseconds as only local processing is needed. If *stateful auto-configuration* is used the duration can expand to seconds as a DHCP server is contacted. Authentication, Authorization and Accounting (AAA) also consumes time in the generation of a new care-of address. It is suggested that AAA actions are performed during the process of determining a new care-of address [Perkins, 2000B]. In this case the duration can expand to several seconds as an AAA server on the mobile node's home network is contacted for authentication.

Duplicate address detection. When a new address has been generated through *stateless auto-configuration*, it may not be unique. Manufacturers of low cost network devices are known to use unregistered IEEE-802 addresses. Thus an IP address can be based on an unregistered IEEE-802 address and therefore duplicate addresses can occur at a network [Huitema, 1997]. *Duplicate address detection* is the process of checking if a new care-of address is an unique address at the network. In practice the mobile node asks if any nodes at the network has a media address for the mobile node's new care-of address. If the mobile node receives a reply, some node is already using that address and the mobile node must generate a new address. If no reply is received, the mobile node must retransmit at least once after a random delay between zero and three seconds before it may assume that the new address is unique [Huitema, 1997]. *Duplicate address detection* must be performed when a new care-of address is generated, independent of whether the address is obtained through *stateless* or *stateful auto-configuration* [Tsirtsis, 2001]. This introduces a latency in the range of three seconds, which is not desirable. In [Tsirtsis, 2001] Charles E. Perkins suggests some alternatives:

- Perform *stateful auto-configuration* without *duplicate address detection*. In *stateful auto-configuration* the DHCP server can ensure that all hosts have an unique IPv6 address and that time is not spend on *duplicate address detection*.
- Generate the care-of address and assume that it is unique, but perform *duplicate address detection* later. This alternative does not perform *duplicate address detection* until after other nodes have been notified of the new care-of address. If the new address is not unique packets for the mobile node may be picked up by another node at the network[2], which makes the mobile node unreachable until it generates and registers a new care-of address.
- Perform *stateless auto-configuration* without *duplicate address detection*. Optimistically assume that the interface IEEE-802 address can form an unique host ID for the network. This alternative is not valid for link layer technologies, that does not supply an IEEE-802 address. Besides as earlier mentioned: There are unregistered interfaces out there. While the previous alternative will recover when it has performed *duplicate address detection*, this alternative may never recover, as it will never discover, that something is wrong.

Duplicate address detection may cause an unacceptable delay, but can we live without it. In the process of reducing handoff latency, the overhead of performing *duplicate address detection* may outweigh the benefits.

---

[2] The *routing* and *encapsulation* headers should ensure that a packet is dropped at other host, than the intended destination.

Propagation of binding updates is the process of routing *binding updates* from the mobile node to the home agent and correspondent nodes. The propagation delay depends on the number of hops and load on the routes. The duration is in the range of 1 ms to 10 seconds. As described in section 1.7 it is the focus of many extensions for Mobile IP to reduce the propagation delay of *binding updates*.

In figure A.10 the generation of a care-of address is initiated on the receipt of a periodic router advertisement from base station *2*. If link layer information indicates, that base station *2* is within range, this could trigger the mobile node to multicast a router solicitation. The router solicitation should cause routers behind the base station to transmit a router advertisement, which is the premise for generation of a care-of address.

A.7 Requirements for supporting Mobile IPv6

In this section the requirement for mobility support in Mobile IPv6 is summarized. There are different requirements for correspondent nodes, mobile nodes and home agents.

A.7.1 Requirements for correspondent nodes

A mobile node should be able to communicate to all nodes in the Internet. This means that all IPv6 nodes should be able to act as correspondent nodes. The requirements for a correspondent node to support Mobile IPv6 are few [Perkins, 1998]. A correspondent node must:

- Be able to insert routing headers into a packet for a mobile node. An IPv6 node should already be capable of using a routing header for other purposes than mobility.
- Have a *binding cache* to hold bindings from mobile nodes. Any IPv6 node already has a destination cache used in *neighbor discovery* procedures. An entry in a destination cache is the relation between the media address and the IP address of a node at the network. A destination cache normally only contains local neighbor information. If bindings for mobile nodes is also added to the destination cache it must be capable of handling nonlocal neighbor information.

  Bindings for mobile nodes also have a lifetime associated with them. When the lifetime expires the binding must be removed from the destination cache/binding cache. When a correspondent node wants to send a packet to a mobile node it checks its destination cache for an entry for the mobile node. If a binding exists, a routing header is added to the packet. The routing header ensures, that the packet is routed to the mobile node's care-of address. If no entry is found, the packet is sent without additions to the mobile node's home network.
- Be able to process a *home address* option. A correspondent node receives a *home address* option with all packets sent from a mobile node. The address in the *home address* option, the home address of the mobile node, must be exchanged with the source address of the packet. This exchange must be made to ensure the transparency of mobility to transport protocols and applications.

As just described a correspondent node already knows how to add a routing header. It already have a destination cache, which with few alterations can manage bindings. A hurdle is to enable a correspondent node to handle a *binding update* option. It is hoped that all IPv6 nodes will be able to handle a *binding update* option. Any node that does not recognize a *binding update* should return an ICMP parameter problem (code 2) message to the source and drop the packet [Perkins, 1998]. Should a correspondent node be incapable of processing a *binding update*, only *route optimization* cannot be performed. The correspondent node can still reach a mobile node via the home agent. It is essential to enable a correspondent node to process a *home address* option, but this requires only few modifications to an IPv6 node.

We therefore conclude that only few alterations are required for correspondent nodes to support Mobile IPv6.

A.7.2 Requirements for mobile nodes

An IPv6 node must perform several additional functions to be mobile. Apart from fulfilling the requirements of correspondent nodes, a mobile node must:

- Be able to detect when it can attach or is attached to a new foreign network.
- Be able to form a new care-of address for a foreign network. This involves both *stateless* and *stateful auto-configuration*.
- Be able to maintain its permanent home address even in the absence of all neighbor discovery messages. Normally a node will assume, that it has been disconnected from a network if it does not receive for example router advertisements periodically. When a node has assumed, that it is disconnected from a network it will normally remove the prefix for that network. This must not happen for a mobile node's home network. If the prefix of the home network is lost while the mobile node is away from home, the mobile node will not be able to locate a home agent at its home network. If the mobile node cannot register with a home agent at its home network, other nodes will not be able to reach the mobile node using its global IPv6 address.
- Be able to determine when to send *binding updates* to its correspondent nodes and its home agent. It must keep a list of active correspondent nodes; the *binding update list*. This involves registering when a correspondent node sends a packet directly to the mobile node and when it sends via the home agent. When a correspondent node sends directly to the mobile node using a routing header it means the correspondent node has an unexpired binding in its destination cache. The binding was earlier sent by the mobile node, so the mobile node knows the time of transmitting and the lifetime of the binding. Based on this information the mobile node can approximate when the binding will expire at the correspondent node. The mobile node may choose to renew the binding at the correspondent node before it expires.
- Be able to process *binding acknowledgments*.
- Be able to process *binding requests*.
- Be able to insert *home address* options in packets to correspondent nodes.

- Be able to decapsulate a packet tunneled from a home agent. Decapsulation is a standard procedure in IPv6.
- Be able to process a routing header, as all IPv6 nodes must be capable of.

The mobile nodes in Mobile IPv6 carry most of the added requirements of mobility.

A.7.3 Requirements for home agents

Home agents are routers, that fulfill some extra requirements. These requirements are few, so it is expected, that all routers in the future will be able to act as a home agent [Perkins, 1998]. The set of requirements to a home agent, apart from fulfilling the requirements to correspondent nodes, is:

- It must be able to encapsulate packets for a mobile node, that has registered a care-of address with the home agent. Encapsulation is also used in other circumstances, so routers are already capable of encapsulating packets.
- It must be able to do *proxy neighbor discovery* on behalf of mobile nodes that has registered a care-of address with the home agent. *Proxy neighbor discovery* enables the home agent to intercept packets for a mobile node as presented in more detail in section A.2. *Proxy neighbor discovery* is also used in other circumstances unrelated to mobility.

It is advised, but not required, that home agents keep their *binding cache* in a non-volatile storage, so bindings will persist in spite of crashes or power failures. If a home agent looses a binding for a mobile node, that mobile node is unreachable through the home agent. A correspondent node, which initiates a communication with a mobile node, sends the first packets to the mobile node's home network and thereby the mobile node's home agent, if the mobile node's care-of address is unknown to the correspondent node.

If neither home agent or correspondent nodes know the care-of address of the mobile node, no one but the mobile node will know its care-of address and it cannot be reached. The mobile node will send its care-of address to the home agent, when some significant event happens like the mobile node attaches to a new network. The mobile node also sends a *binding update*, when it expects, that the binding at the home agent (the one that was lost) is about to expire. As the lifetime of a binding is as default set to 600 seconds, a mobile node can be unreachable for an unacceptable period of time, as the result of a failure at the home agent.

A majority of the required capabilities of a home agent is also used in circumstances unrelated to mobility. This means, that the effort of implementing a home agent on top of IPv6 router software is limited.

A.7.4 Summary of requirements

Very few modifications are needed for correspondent nodes. This an advantage as mobile nodes should be able to communicate with all IPv6 nodes and fewer modifications improves the possibility, that commercial implementations of IPv6 network stacks will fulfill the requirements of correspondent nodes in Mobile IPv6. The situation is the same for home agents. It requires few modifications for an IPv6 router to supply home agent capabilities. It is the IPv6 stack at mobile nodes that requires most additions.

This represents a nice distribution of new software in the deployment of Mobile IPv6. If the system administration of a network decides to provide the service of mobility to their users, the system administration needs to install a home agent at a local router and install mobility software at nodes, that often leave the premises.

Appendix B

Installing FreeBSD and KAME

This appendix provides a brief description of how we installed FreeBSD 4.1 and the KAME extension providing mobility for IPv6.

FreeBSD can be obtained from their website at http://www.freebsd.org/. KAME is a joint effort to provide a free implementation of IPv6 and IPsec (for both IPv4 and IPv6) for BSD variants. It is a collaboration between a number of Japanese companies including NEC Corporation, Hitachi Ltd. and Toshiba Corporation. Others may contribute to the KAME software too, for example is the Mobile IPv6 implementation in KAME supplied by Ericsson Radio System Research Lab.

FreeBSD 4.1 come integrated with a stable KAME release that includes IPv6 but not Mobile IPv6. A newer release of KAME is thus installed as described in section B.2.

B.1 Installing FreeBSD

The following apply to the installation of FreeBSD 4.1.

- Prepare the installation floppies as described at http://www.freebsd.org/handbook/install.html.
- Install FreeBSD via FTP. To avoid a bug in the installation program first select "active FTP"in the installation program. After this has failed (due to security restrictions in the s.cs.auc.dk network) then select installation via "HTTP proxy".
- Select "Standard"installation.
- Select the "X-user"distribution.

If "active FTP"is not first selected, the installation hangs.

B.2 Installing KAME

The KAME source are supplied in different variants. Every Monday a new "snap"is released. A "snap"is a release that includes all code present within the KAME development project. Thus much experimental and undocumented code is included. The mobility support for IPv6 included with the used "snap"is experimental code.

Installation of the release from 2000-09-25 is described below. The installation of KAME involves the build and installation of a customized kernel and a set of tools. A `%` indicates that the command can be executed by any user, a # that user must be root:

- Download KAME from http://www.kame.net/
- Unpack it

```
% tar -xvzf kame-20000925-freebsd41-snap.tgz
```

- Create symbolic links before building

```
% make TARGET=freebsd4 prepare
```

- Change dir to kame/freebsd4/sys/i386/conf
- Make a kernel configuration file CONFIGFILE by using the GENERIC.KAME as a template. Activate mobile IPv6 by uncommenting "MIP6"and "MIP6_DEBUG". Deactivate support for IPsec because it might conflict with mobile IPv6 (according to a README file supplied with KAME).
- Invoke `/usr/sbin/config CONFIGFILE`
- Build the kernel

```
% cd ../../compile/CONFIGFILE
% make depend
% make
```

- Install the kernel file to the root directory as root

```
# make install
```

- Change dir to kame/freebsd4
- Install the updated userland binaries supplied with KAME

```
% make includes
# make install-includes
```

- Perform make

```
% make
```

- Install userland tool into /usr/local/v6/{bin, sbin, whatever}

```
# make install
```

- Reboot the system

```
# fastboot
```

Then update the path in /etc/rc with /usr/local/v6/{bin, sbin} so they are searched **before** the existing paths. This makes sure that userland tools supplied with KAME is invoked instead of the normal FreeBSD versions.

Appendix C

Configuring IPv6 and KAME Mobile IPv6

This appendix describes how we have configured the nodes in the Mobile IPv6 testbed described in chapter 4.

C.1 Configuring IPv6

This section briefly describes how to configure the routers and the host to run IPv6.

C.1.1 Routers

Edit the /etc/rc.conf file such that:

- Configure the machine as an IPv6 router by uncommenting the appropriate lines in the file /etc/rc.conf. It should include the lines

  ```
  ipv6_enable="YES"
  ipv6_gateway_enable="YES"
  ipv6_router_enable="YES"
  ipv6_router="/usr/local/v6/sbin/route6d"
  prefixcmd_enable="YES"
  rtadvd_enable="YES"
  ```

- Configure the IPv6 address for each interface by specifying the prefix, e.g.

  ```
  ipv6_prefix_fxp0="fec0:0000:0000:0001"
  ipv6_prefix_xl1="fec0:0000:0000:0002"
  ```

  where fec0 indicates a site local address.

- Give route6d the -l option to make the router exchange site local addresses in route tables

  ```
  ipv6_router_flags="-l"
  ```

- Enable the interfaces for IPv6, e.g.

  ```
  ipv6_network_interfaces="fxp0 xl1"
  ```

C.1.2 Hosts

At hosts it is only necessary to enable the IPv6 stack. An IP address will be assigned automatically using *stateless auto-configuration*. To the file /etc/rc.conf the following lines should be added:

```
ipv6_enable="YES"
ipv6_network_interfaces="auto"
```

C.2 Configuring Mobile IPv6

This section describes how to configure Mobile IPv6 once the network is running IPv6.

After installing KAME as described in appendix B, a node has the correspondent node functionality activated automatically. Home agent functionality and mobile node functionality must be configured and activated as described below.

C.2.1 Access routers

The configuration described below only applies to routers, to which a mobile node can directly attach (access routers).

- Add the -m option to rtadvd in file /etc/rc.network6
- Configure /usr/local/v6/rtadvd.conf with the lines

  ```
  ether2:\
       :mtu#1500:maxinterval#2:mininterval#1:
  fxp0:\
       :tc=ether2:
  ```

  in order to increase the frequency with which router advertisements are transmitted. Remember to create an entry for each interface. The `maxinterval` should be set to 1.5 s and the `mininterval` should be set to 0.5 s according to [Johnson *et al.*, 2000]. However, due to a bug no values below 2 and 1 respectively can be specified. The router advertisement daemon have been modified in order to be able to specify the correct values, see appendix E.

C.2.2 Routers hosting a home agent

In addition to the configuration done to all routers, the router acting as a home agent should also be configured with the following:

- Add the -m option to rtadvd in file /etc/rc.network6
- Make a file /usr/local/v6/etc/mip6conf with the lines

  ```
  debug
  enable_ha
  ```

- Edit the /usr/local/v6/etc/rtadvd.conf so the interfaces that can serve as home agents gets the following properties

  ```
  ha:\
       :hatime#100:hapref#10:\
       :raflags#32:\
       :pinfoflags#224:\
       :maxinterval#4:mininterval#3:
  fxp0:\
       :tc=ha:
  ```

- The home agent can be invoked from the command line by

  ```
  #mip6config -f /usr/local/v6/etc/mip6.conf
  ```

C.2.3 Mobile nodes

The mobile node is configured with a static home agent, as using *dynamic home agent address discovery* is not recommended in KAME.

- Make a file /usr/local/v6/etc/mip6conf with the lines

  ```
  debug
  homeaddr fec0::1:0:0:0:44/64@de0%fec0::1:290:27ff:fe90:6a76
  eager_md 0
  ```

  where `homeaddr` takes the following arguments:

  ```
  homeaddr <home addr>/<plen>@<interface>%<home agent addr>
  ```

  `eager_md` configures which handoff strategy should be used. The meaning of the values 0, 1 and 2 is as described in chapter 4.

- The mobile node can be invoked from the command line by

```
#mip6config -f /usr/local/v6/etc/mip6.conf
```

We have configured the kernel variable `net.inet6.icmp6.nd6_prune` (by using the `sysctl` tool) to have a value of 0 instead of its default value of 1. This kernel variable determines how often IPv6 neighbor discovery polls the default router list and the prefix list. If set to 1, the lists are only scanned for expired entries once every second resulting in an increased handoff latency. If set to 0, the lists are polled at every opportunity.

Appendix D

Bugs in FreeBSD and KAME

This appendix describes the bugs which have been encountered in FreeBSD and KAME.

D.1 Bugs in FreeBSD

As described in B.1 there is a bug in the installation procedure when installing FreeBSD 4.1 via FTP at a computer at the s.cs.auc.dk network, which is a site-local network where access to the Internet is supplied through a HTTP proxy server. The bug can be reproduced in the following way:

- Prepare the installation floppies as described at http://www.freebsd.org/handbook/install.html
- Boot the floppies.
- Select installation via "HTTP proxy".

Then the installation program hangs. We have discovered the following work around to the bug:

- Instead of selecting installation via "HTTP proxy"select "active FTP". This fails due to security reasons (there is a firewall between the Internet and the s.cs.auc.dk network).
- After this has failed, "HTTP proxy"installation can be successfully selected.

The bug has been submitted to the FreeBSD community on the 13th of December 2000. Status on the bug is available at http://www.freebsd.org/cgi/query-pr.cgi?pr=23516

D.2 Bugs in KAME

The following bugs apply to the KAME "snap"released on the 25th of September 2000.

D.2.1 Bug in Mobile IPv6

When conducting the tests described in chapter 4, UDP packets are sent from a correspondent node (located at the same machine as the home agent) to the mobile node.

When the total number of sent UDP packets reaches somewhere between 35800 and 35900 the machine sending the UDP packets crashes and has to be rebooted.

The bug has been discussed via email with Conny Larsson from Ericsson Radio Systems AB in Sweden, who has written part of the source code for KAME Mobile IPv6. He has experienced an identical problem, and believes that the bug is caused by using up all the memory. Each time a packet is sent, some memory is allocated that is not properly deallocated. The actual number of packets that one can send from a correspondent node is probably dependend on the amount of RAM on the machine. Ericsson is currently working at a fix for this bug.

In general, once the home agent functionality or the mobile node functionality has been invoked, it cannot be successfully halted. The solution is to reboot.

D.2.2 Bug in mobile node software

It happens that the computer running the mobile node crashes. The problem seems to be emerging whenever the `root` user is running other code than the mobile node code.

The bug occurred during the tests, when we logged the received UDP packets to a file as `root`. When doing this as an ordinary user no problems has been experienced.

Performing handoffs with a high frequency can also cause the mobile node to crash. One time a period of four seconds between handoffs continuously crashed the mobile node. However, this is not a situation which we have been able to reconstruct since.

D.2.3 Bug in router advertisement daemon

KAME supplies an updated version of the router advertisement daemon `rtadvd`. This supposedly should be compliant with new requirements for sending router advertisements introduced with Mobile IPv6, but this is not the case. The `rtadvd` is stable, but we have had to add some functionality. This is described in appendix E.

Appendix E

Modifications to FreeBSD software

This appendix describes how the router advertisement daemon `rtadvd` has been modified.

E.1 Router advertisement daemon

The Mobile IPv6 specification recommends that unsolicitated router advertisements should be sent with an interval between 0.5 and 1.5 seconds [Johnson *et al.*, 2000].

FreeBSD 4.1 comes with a router advertisement daemon named `rtadvd`. However `rtadvd` does not permit a lower interval than between 3 s to 4 s to be specified[1]. For that reason the KAME mobility support package supplies a slightly modified version of `rtadvd`. This version presumably allows the interval between router advertisements to be configured to a random number between 0.5 s and 1.5 s. However the lowest interval, which can be specified in practice, is between 1 and 2 seconds.

Only integer values are allowed to be specified. This is because that the minimum and maximum value specified in the router advertisement daemon configuration file, `rtadvd.conf`, are stored in integer variables. We have modified the `rtadvd` implementation in the following way:

Locate the `rtadvd` source code under `kame/rtadvd/`

- To `rtadvd.h` the following modifications have been made
  - We change the defines `MIN_MAXINTERVAL` and `MIN_MININTERVAL` from 1.5 and 0.5 respectively, so that both have the value 0
  - In the struct `rainfo` two variables of type long, `maxinterval_usec` and `mininterval_usec`, are added. They are used to specify the interval between router advertisements in μs
- To `rtadvd.c` the following modifications have been made (line numbers refer to the original file)

---

[1] This is due to the IPv6 Neighbour Discovery Protocol [Narten *et al.*, 1998], which does not permit an interval below 3 s between sending unsolicitated multicasted router advertisements.

  - In line 1480 a condition has been added in order to avoid the possibility of making modulus with 0.
  - In line 1494 `tm->tv_usec` is set to a random value between `mininterval_usec` and `maxinterval_usec`. Before it was always set to 0.

- `config.c` has been modified to read two extra values from the router advertisement configuration file
  - After

    ```
    tmp->maxinterval = (u_int)val;
    ```

    in line 151, the following code to read in `maxinterval_usec` from the configuration file has been added

    ```
    MUSTHAVE(val, "maxinterval_usec");
    tmp->maxinterval_usec=val;
    ```

  - After

    ```
    tmp->mininterval = (u_int)val;
    ```

    in line 161, the following code to read in `mininterval_usec` from the configuration file has been added

    ```
    MUSTHAVE(val, "mininterval_usec");
    tmp->mininterval_usec=val;
    ```

- The `Makefile` is modified. The flag `MIP6` needs to be defined. This is ensured by

  ```
  CFLAGS=+-DMIP6
  CPPFLAGS=+-DMIP6
  ```

To compile the router advertisement daemon enter

```
make
```

It can then be invoked with

```
./rtadvd -m [interfaces]
```

The newly compiled `rtadvd` program can be set as the default router advertisement daemon by changing the path to `rtadvd` in the file `/etc/rc.network6`

When the variables `maxinterval` in the `rtadvd.conf` file is set to 0, the `rtadvd` sets the lifetime of the unsolicitated router advertisements to 0. Thus this value need to be specified explicitly in the `rtadvd.conf` file. This can be done by adding the line

```
  rltime#4
```

to the appropiate interfaces. Normally the lifetime should be set to 3 times the maxAdvertisementInterval. The minimum value is 1 second due to the design of the ICMP Router Advertisement [Deering, 1991].

The modified version of `rtadvd` requires that in rtadvd.conf the variables `mininterval_usec` and that `maxinterval_usec` is specified and `maxinterval_usec` must be greater or equal to `mininterval_usec`. The variables `maxinterval` and `mininterval` should be set to 0. Similarly, the `rltime` must now be explicitly specified.

The following is an example `rtadvd.conf` file:

```
ether:\
        :mtu#1500:maxinterval#0:mininterval#0:\
        :maxinterval_usec#1500000:mininterval_usec#500000:\
        :rltime#4:
fxp1:\
        :tc=ether:
fxp0:\
        :mtu#1500:maxinterval#4:mininterval#3:\
        :maxinterval_usec#0:mininterval_usec#0:
xl0:\
        :mtu#1500:maxinterval#4:mininterval#3:\
        :maxinterval_usec#0:mininterval_usec#0:
```

In this configuration file, the interface `fxp1` is the one to which the mobile node can connect. A router advertisement is specified to be sent once every 0.5 to 1.5 seconds.