Performance of Video Streaming in a Mobile IPv6 Environment

Prepared by:

Madhush Koruth Mathews

Supervised by:

Neco Ventura

Department of Electrical Engineering

University of Cape Town

2007

This dissertation is submitted to the Department of Electrical Engineering
in fulfillment of the academic requirements
for the Degree of Master of Science in Engineering

31 January 2007

Declaration

I declare that this thesis is my own work. Where collaboration with other people has taken place, or material generated by other researchers is included, the parties and/or material have been referenced or acknowledged as appropriate.

This work is being submitted for the degree of Master of Science in Engineering at the University of Cape Town. It has not been submitted to any other university for any other degree or examination.

_____ _____

Madhush Koruth Mathews Date

Acknowledgements

I would like to thank the following individuals and organisations for their contributions and assistance during the course of this project:

- Mr. Neco Ventura, my supervisor, for his guidance and encouragement throughout the project.

- Telkom SA, Siemens, the National Research Foundation (NRF) and the Department of Trade and Industry (DTI), for their financial contributions towards this research.

- David Waiting and George Kalebaila for reviewing and proofreading this thesis.

- My colleagues in the Communications Research Group (CRG) at UCT, for their input and the working environment they helped create.

- My family and close friends, for their love and support in every aspect of my life.

Synopsis

Real-time multimedia applications are becoming pervasive. This is partly due to the increasing bandwidths of wireless technologies, and partly due to the lowering costs associated with the use of these technologies. This has also been supported by the emergence of robust and more powerful session management and transport protocols for the Internet. Advances in cellular telephony occurred during the Internet evolution bringing the concept of mobility and reachability to every user and leading to the fixed mobile convergence of the Internet with wireless access technologies for seamless service delivery. Data-based mobile access networks are emerging with IP at their core.

Although mobility is the main feature that makes wireless networks attractive, it also poses the greatest challenges, particularly on IP networks. IP was not designed with mobility in mind. Mobile IPv6 was developed to handle IP mobility through the redirection of packets when a mobile node moved into a foreign network. It has emerged as the prevalent mobility management technique on IP networks. While the traditional approach of Mobile IPv6 provides a solution to global mobility, it does not take into account Quality of Service requirements. Therefore, there is no guarantee that the network will provide the resources required to deliver the application traffic. In addition, the connection disruption and packet loss that an application incurs during handover severely degrades its quality.

Enhancements have been proposed to the base Mobile IPv6 protocol to improve handover performance and reduce the impact of handovers on applications. The most valuable mobility management protocol discussed in the literature is Fast Mobile IPv6, which is able to effectively reduce packet loss to near zero and minimise handover latency.

A seamless video streaming handover scheme is formulated in this research to guarantee the continuity of a Video on Demand application in the context of IP

mobility. The scheme is analysed and an evaluation framework is developed to assess its feasibility.

This thesis presents an effective method of supporting real-time multimedia applications on IP-centric wireless networks using Mobile IPv6. From the results, the scheme shows a significant improvement in terms of the disruption time and packet loss that a Video on Demand application experiences during handover. The effects of Mobile IPv6 are mitigated by the use of the adaptive mobile video streaming scheme. Thus, it is feasible to use a combination of adaptation of video quality based on network conditions and Fast Mobile IPv6 to guarantee application continuity in the context of IP mobility.

Table of Contents

List of Figures

List of Tables

Abbreviations

AMVS – Adaptive Mobile Video Streaming

AP – Access Point

AR – Access Router

BAck – Binding Acknowledgement

BU – Binding Update

BUL – Binding Update List

CN – Correspondent Node

CoA – Care-of Address

DAD – Duplicate Address Detection

DNS – Domain Name Server

DVD – Digital Versatile Disk

FFHMIPv6 – Flow-based Fast Handover for Mobile IPv6

F-HMIPv6 – Fast handover over Hierarchical Mobile IPv6

FMIPv6 – Fast Mobile IPv6

FNA – Fast Neighbour Advertisement

HA – Home Agent

HAck – Handover Acknowledgement

HDTV – High Definition Television

HI – Handover Initiate

HMIPv6 – Hierarchical Mobile IPv6

HoA – Home Address

ICMPv6 – Internet Control Message Protocol version 6

IP – Internet Protocol

IPsec – Internet Protocol Security

IPv6 – Internet Protocol version 6

MAC – Media Access Control

MIPL – Mobile IPv6 for Linux

MIPv6 – Mobile IPv6

MN – Mobile Node

MPEG – Moving Picture Experts Group

NAAck – Neighbour Advertisement Acknowledgement

NAR – New Access Router

PAR – Previous Access Router

PrRtAdv – Proxy Router Advertisement

QoS – Quality of Service

RA – Router Advertisement

RS – Router Solicitation

RSSI – Received Signal Strength Indicator

RtSolPr – Router Solicitation for Proxy advertisement

SCTP – Stream Control Transmission Protocol

SIP – Session Initiation Protocol

TCP – Transmission Control Protocol

UDP – User Datagram Protocol

UMTS – Universal Mobile Telecommunications System

VoD – Video on Demand

WiMAX – Worldwide interoperability for Microwave Access

WLAN – Wireless Local Area Network

Chapter 1

Introduction

Providing ubiquitous Internet access has become increasingly important due to emerging new applications (e.g., networked games, real-time multimedia applications, and applications that require mobile information access). The use of real-time multimedia communication services such as telephony and videoconferencing over IP is already a reality in some corporate networks, and it is bound to expand to the global Internet [1, 2].

Real-time multimedia applications refer to the broad spectrum of applications where live or stored multimedia (i.e., voice, audio or video) is transmitted in real-time over a network from server to client. The popularity of these applications is growing especially over the Internet. This is partly due to the low cost of such services. As the available bandwidth on networks grows, there is also a greater demand for real-time multimedia applications. This has been supported by the emergence of robust and more powerful session management and transport protocols for the Internet, such as the Session Initiation Protocol (SIP) and Stream Control Transmission Protocol (SCTP), to specifically handle real-time traffic. Therefore, real-time multimedia applications are becoming pervasive.

Advances in cellular telephony occurred during the Internet evolution bringing the concept of mobility and reachability to every user. This led to the fixed mobile convergence of the Internet with wireless access technologies for seamless service delivery. The available bandwidth on wireless networks has also grown (e.g., over

UMTS, WiMAX and WLAN), fuelling this convergence. A universal data-based mobile access network is emerging with IP at its core.

Real-time multimedia applications typically have strict delay requirements. The volumes of data to be exchanged are often substantial, so the applications also have high throughput requirements. In addition, significant amounts of real-time processing are required for encoding and decoding the media [3]. Therefore, the level of quality the application can achieve depends not only on the performance of the underlying network but also on the user entity processing power. In order to provide a user with acceptable application quality, the application ideally requires Quality of Service (QoS) guarantees from the network [1].

Although mobility is the main feature that makes wireless networks attractive, it also poses the greatest challenges, particularly on IP networks. The Internet Protocol (IP) is a connectionless, best-effort packet switching protocol that provides packet routing and delivery services for the Internet and private networks. IP was not designed with mobility in mind. Mobile IP was developed to handle IP mobility through the redirection of packets when a mobile node moved into a foreign network. It has emerged as the prevalent mobility management technique on IP networks [4, 5].

IP version 6 (IPv6) [6] is the new protocol proposed to replace the current IP version 4 (IPv4). The main driver for IPv6 was the lack of address space, although some new features have also been included. The major changes to IPv6 are the redesign of the packet header and an increase in address size to 128 bits. There is a clear migration towards IPv6. The United States (US) and other countries are already in the early stages of IPv6 adoption [7]. The US Department of Defence has required its networking products to be IPv6 capable since 2003 and all US government agencies are required to transition their network backbones to IPv6 by 2008 [8]. Since Mobile IPv4 will soon be rendered obsolete, only Mobile IPv6 [9] is considered in this research.

There has been a proliferation of Internet connected mobile devices (such as laptops, personal digital assistants and mobile phones). However, IP mobility exacerbates the problem of supporting real-time multimedia over networks because of radio channel

characteristics and the complexity of mobility management. Thus, there are still many technical obstacles to be overcome before the reality of "multimedia communication anytime, anywhere, any style" can be achieved [10].

Mobile IPv6 provides connectivity to mobile nodes as they move from one wireless point of attachment to another in a different subnet. A mobile node is a node that changes its location within a network topology. This change may be due to physical movement of the node or due to changes in the network topology that cause the node to be attached to a different router. A Mobile IPv6 handover is required for a mobile node to successfully move from one IP subnet to another and maintain its ongoing sessions. The mobile node cannot receive IP packets at its new point of attachment until the handover is complete. The handover consists of a sequence of events, including movement detection, establishing a new care-of address for the mobile node and updating the mobile node's home agent and correspondents with its new location. The time taken to complete this handover is called the handover latency. The connection disruption at the network layer associated with handover is often too long to support real-time multimedia applications [11]. A mobile node's wireless channel characteristics such as signal strength and link layer bit-rate often deteriorate as the node moves within a wireless cell and this can also impede the delivery of multimedia applications. Further, mobility between wireless cells introduces disruptions to communication sessions at the link layer.

The core problem considered in this research is the disruption to IP streams caused by Mobile IPv6 handover, with the particular aim of supporting real-time multimedia applications. This research is limited to an investigation of a Video on Demand (VoD) streaming application. While the traditional approach of Mobile IPv6 provides a solution to global mobility, it does not take into account QoS requirements. Therefore, there is no guarantee that the network will provide the resources required to deliver the application traffic. In addition, the connection disruption and packet loss that an application incurs during handover severely degrades its quality.

Consider a scenario where a person is spending the day in a large shopping mall. The person would like to watch a movie at the theatre in the mall but would first like to view previews of the available shows before deciding which movie to watch. So he

connects to the Internet on his laptop via the WLAN in the mall. He accesses a VoD server and begins to stream the trailer for the latest blockbuster. However, as he walks through the mall, his laptop moves out of range of its current access point and must handover to a new access point. The connection disruption associated with the handover causes visible distortions and an interruption to his stream resulting in a poor user experience. This is an illustration of the sort of problem considered in this research.

Fast handovers aim to reduce the Mobile IPv6 handover latency, while smooth handovers aim to reduce the associated packet loss. A seamless handover is both fast and smooth enough to ensure the continuity of an application running on a mobile node without any significant degradation in its perceived quality.

The aim of this research is to formulate and analyse a seamless video streaming handover scheme. The scheme will be able to guarantee the continuity of a VoD application in the context of IP mobility.

1.1 Thesis Objectives

This study focuses primarily on an analysis, design and implementation of a scheme to support seamless Mobile IPv6 handovers for real-time multimedia applications. Traditional solutions to this problem have focused on improving the performance of Mobile IPv6. The most significant of these proposals are introduced and analysed in subsequent chapters.

Additionally, this study centers not only on Mobile IPv6 handover performance, but on system performance taking into account wireless link characteristics and network conditions. This investigation specifically addresses the integration of Mobile IPv6 and 802.11g WLANs [12] within a single network architecture. 802.11 networks indirectly impose certain restrictions on network layer mobility. Therefore, in order to fully analyse the performance of Mobile IPv6 handover, specific characteristics of 802.11 WLANs must be investigated.

The type of real-time multimedia application investigated also places restrictions on the network architecture. A brief survey of video streaming technology is presented to assist in determining the requirements of an ideal video streaming handover scheme.

This leads to the formulation of a scheme that supports seamless Mobile IPv6 handovers. Relevant research related to this study is reviewed. Useful concepts are extracted and used to develop a proposal for a seamless video streaming handover scheme.

A critical element of this research is to analyse the feasibility of the seamless video streaming handover scheme. A suitable architecture of an evaluation platform, fully IPv6 compliant and capable of supporting Mobile IPv6 handovers, is developed. This evaluation framework enables an analysis of the feasibility of the proposed handover scheme and the extent to which it would be effective in a practical scenario. The disruption time, packet loss, jitter and signalling load of Mobile IPv6 handovers is compared to Fast Mobile IPv6 handovers, and an assessment is made on whether the handover performance has been sufficiently improved to allow the proposed handover scheme to be used to mitigate the disruption associated with handover.

This research aims to present a constructive analysis of present work in this field together with pertinent evaluation data and an analysis of the proposal. The evaluation framework may be extended and used for future research into mobility management techniques.

1.2 Scope and Limitations

The field of mobility management in wireless networks is very broad. Several mobility management protocols beyond Mobile IPv6 exist. Some techniques use different layers of the protocol stack – an example is SIP which operates on the application layer. A full discussion of these protocols and other techniques to assist mobility is beyond the scope of this project. This thesis focuses on mobility management using Mobile IPv6 – a network layer protocol, and enhancements to this

protocol. All the possible techniques available to assist Mobile IPv6 handovers are not discussed exhaustively.

The ways to improve Mobile IPv6 handover are too numerous to be discussed in this thesis. The improvements focused on are those that have had the most significant impact on Mobile IPv6 and those that are extensively discussed in the literature. Peripheral issues affecting Mobile IPv6 such as security, authorisation, authentication, accounting, deployment issues, and ad hoc networking are also not addressed in this research.

Mobile IPv6 was designed to operate over different access technologies. Handovers depict different characteristics depending on the underlying technology. Therefore, in this research Mobile IPv6 handovers between different access technologies are not considered. Only handovers between homogenous 802.11g WLANs are considered.

The type of video streaming application to be supported by the network architecture significantly influences the design of a seamless handover scheme. This investigation is restricted to VoD applications. The main requirement of such applications is sustained high visual quality, not interactivity.

Simulations are an effective tool to predict network performance but they do not take into account real-world factors, so they often do not reflect the true behaviour of networks. Therefore, although implementing a physical framework is the best way to evaluate the solution proposed in this research, it limits the research since it is not as easy to adapt a physical test-bed to changes in topology or to new experiments.

1.3 Thesis Outline

The remainder of this document is organised as follows:

Chapter 2 introduces some relevant background theory and a review of research related to this study. It begins by introducing real-time video streaming applications, the demands they place on networks and factors that affect their performance. Next,

Mobile IPv6 is introduced. An analysis of the handover procedure is presented to illustrate the impact of handover on video streaming applications. The 802.11g WLAN technology, which defines the physical medium through which users access the network, is then introduced. This leads to the rationale for cross-layer solutions, and several cross-layer performance enhancements are introduced.

Chapter 3 discusses the requirements of a seamless video streaming handover scheme over Mobile IPv6. Useful material and concepts from the preceding chapter are extracted and applied to this work. An adaptive mobile video streaming scheme is then formulated and discussed.

In Chapter 4 the architectural design of an evaluation framework for the adaptive mobile video streaming scheme is given. Initially, the limitations, objectives and requirements of the test-bed are presented. Then the test-bed's topology and hardware implementation are described. Following that, the software used to implement various aspects of the framework is discussed.

Chapter 5 presents results from tests performed on the evaluation framework. However, the test-bed configuration, the tools used to perform the tests and the various test scenarios are discussed beforehand. The results obtained and an analysis of these results is then put forward.

Chapter 6 presents a set of conclusions that were drawn from these evaluations. Thereafter, recommendations and future work are presented.

Chapter 2

Literature Review

2.1 Introduction

In the previous chapter a few different network technologies were introduced, including Mobile IPv6 and 802.11 WLANs. This chapter presents a more thorough discussion on these technologies in an effort to understand the effect of network streaming on real-time multimedia applications.

Network architectures often incorporate a number of different inter-working technologies, so the task of managing them is a fairly complicated one. In order to facilitate this, network models that group functions which allow a node to communicate into several logical layers have been proposed. This approach makes the abstraction of detail at various levels possible. Consequently, the design of individual network technologies is simplified. The 802.11 WLAN technology independently defines both the physical layer (layer 1) and link layer (layer 2) and is used to establish wireless connectivity over relatively short distances. Mobile IPv6 is a network layer (layer 3) technology that manages IP mobility.

This chapter discusses theory on video streaming and introduces the Mobile IPv6 and 802.11 WLAN technologies. In particular, the degradation of IP streams due to Mobile IPv6 handover is discussed. Some proposals aim to support video streaming applications by fostering cooperation between applications and networks. These solutions tackle the problem by altering the quality requirements of applications.

However, more obvious techniques to reduce the effect of IP mobility on video streaming applications are those that aim to improve Mobile IPv6. These techniques address the problem at the network layer. Other proposals address the provision of video streaming applications at the link layer, with Quality of Service (QoS) guarantees. Each of these paradigms, driven at different layers of the protocol stack, is discussed. This leads to the rationale for cross-layer approaches to supporting applications [13, 14].

The aim of this chapter is to present the requirements of video streaming applications, the effect of network streaming on these applications, and in particular the impact of IP mobility on the delivery of these applications. The impact of Mobile IPv6 handover is discussed in the specific cases where either TCP or UDP is used as the transport layer protocol above the IP layer. The literature reviewed presents the origins of this work and contributes to the formulation of an effective video streaming handover scheme to guarantee the continuity of applications during handover.

2.2 Video Streaming

The transport of video allows for far more advanced forms of communication and entertainment compared to other media, but it also poses new and considerably higher demands on network resources. In trying to establish the requirements of an ideal video streaming handover scheme, it is first necessary to understand what application quality is and what the effect of network streaming is on application quality. Rate adaptation techniques to foster cooperation between the application and the network are then introduced. The rate adaptation entities are located in the streaming server. Thereafter, a discussion on other related research is presented.

2.2.1 Application Quality

Application quality is defined by performance characteristics such as response time, predictability and consistent perceptual quality. Factors that influence application quality include the task of the application, user expectations and experience, whether the application delivers the desired levels of performance, and less significant factors

like billing for the use of network resources [15]. These factors are difficult to quantify, making the task of determining quality parameters quite challenging. Another aspect of quality is the social behaviour of individual applications in an environment such as the Internet. Unresponsiveness and excessive use of resources by certain applications can lead to severe congestion that affects the entire network. Thus, the ability of an application to adapt to the state of the network may be considered an indirect aspect of its quality.

In order to sustain the perception of their quality, applications should employ some measure of tolerance to varying network conditions and built-in mechanisms that allow them to probe the network conditions and alter their transmission characteristics accordingly.

Metrics that characterise the performance of an IP network and are the most significant factors that affect end-to-end application quality are:

- *Delay* – the time taken for application data units to be carried by the network to the destination.
- *Delay variation* (*jitter*) – caused by the buffers built up on routers during periods of increased traffic and by changes in routing. The amount of jitter an application can tolerate depends on the nature of the application tasks. Applications with strict delay requirements cannot afford much jitter. Some applications make a trade-off by delaying presentation long enough to build up an initial buffer that smoothes out jitter.
- *Packet loss* – the fraction of IP data packets, out of the total number of transmitted packets, that are lost somewhere along the path from source to destination.
- The *loss pattern (loss period)* – losses usually appear in bursts. Depending on the encoding mechanism used, the type of data being transported and whether error correction techniques are used, loss may cause either minor distortions or significant deterioration in quality.

The application's perception of QoS parameters may not coincide with the network QoS metrics mentioned above. Application performance deterioration should be

expressed in terms of user-perceivable effects, rather than on their origin within the application end-to-end path [15]. Application-level performance metrics include latency (the end-to-end delay the application experiences), and availability and continuity of service (the requirement for uninterrupted service with acceptable quality). Factors that may disrupt the continuity of the service include information loss and jitter.

The bandwidth is the portion of the available capacity of an end-to-end network path that is accessible to the application. The effective share of bandwidth the application gets from the network is the throughput. The TCP transmission rate of the application will be throttled if the throughput is restrictive, resulting in increased end-to-end latency. The increased end-to-end latency negatively affects the end-to-end application quality.

2.2.2 Video Streaming

Video streaming refers to the real-time transport of live or stored video. Typical examples of video streaming applications include live video transmission, multimedia information broadcasting (webcasting), Video on Demand (VoD) and high definition television broadcasting (HDTV). A webcast uses streaming media technology to take a single content source and broadcast it to many clients simultaneously. On the other hand, VoD applications enable users to select stored video from a central server for viewing over a network. The server sends a unicast stream to each client. The main requirement of streaming video is for sustained high visual quality. Application interactivity is not a main feature, with the exception of VCR-like controls (e.g., play, pause and stop).

User expectations, desired quality and throughput requirements vary widely based on application usage scenarios. Some common throughput requirements are summarised below [15]:

- Streaming of modest-quality recorded or live content currently involves low expectations of throughput – from a few tens of Kbps to 1 Mbps.

- For applications such as VoD with equal to or better than VHS quality, throughput of 1-3 Mbps (using MPEG-1 codec) up to 10 Mbps or higher (using MPEG-2) is expected.
- HDTV is used for extremely high quality video. Depending on the compression used, throughput requirements range from 19.2 Mbps to 1.5 Gbps.

Some applications make use of an initial delay to build a receiver playout buffer that can accommodate quite high jitter values. This translates into bounded data loss rates. The user expectation of data loss is that it should be lower than 2-3 % [15]. Video streaming today is often restricted to modest-quality video with limited resolution (image size). The distortions introduced by encoding with comparatively low bit-rates and packet loss are generally tolerated by users in these situations.

On the other hand, encoding distortions or lost information is easily noticed by users of advanced video services over IP that use much higher resolutions. These applications need to maintain sustained high bandwidth and very low packet loss to meet user expectations that have been created by experiences with similar, non-IP, high quality video services (such as digital broadcast TV and DVD). Start-up delays of the order of seconds are often made use of to build up receiver buffers, since data loss tolerance is much lower. Nevertheless, many advanced video applications over IP suffer significant deterioration of visual quality.

The effect of network streaming has a direct impact on digital video quality. Bandwidth variability, increased delay, jitter and packet loss reduce the perceived quality of received video content [15]. The main factors that determine a video's visual quality are the image resolution and the frame rate of the video. Better image resolution and frame rates require higher encoding bit-rates for the video. This translates to a requirement for higher transmission bit-rates. Thus, the effective share of bandwidth the application receives from the network can be restrictive and result in increased delay and packet loss. Packet loss is particularly detrimental to video streams because image compression achieves reduction in the number of bits required

to transmit the stream by removing redundancies inherent in video data. Therefore, any loss of compressed data cannot be recovered.

2.2.3 Adaptation Techniques

Some form of cooperation between the application and the network is necessary to provide the required network stability and application quality. The idea of adaptation is that applications should be aware of underlying network conditions and react to them. Applications are capable of operating within a range of resources that are available to them. The audio and video applications that make up the majority of real-time traffic on the Internet today fall into this category. The idea of QoS provisioning is that an application should ask the network for the resources it needs and the network should respond by reserving them and offering the requested guarantees. This is very complex and does not scale effectively. The new demands of network communications also makes end-to-end guarantees very difficult to achieve. In this sort of environment, the ability of an application to gracefully adapt is an aspect of the application quality it can realise. The concept of adaptive video streaming is based on the widely accepted maxim that users prefer a reduced bit-rate to packet losses when streaming video [16].

Several kinds of adaptation techniques exist [15]:

- *Throughput adaptation* – multimedia streams are capable of adapting to the available bandwidth by varying their transmission rates.
- *Adaptation to delay and delay variation* – applications can adapt to variable delays within reasonable ranges. Certain levels of delay can be tolerated without conspicuous quality effects.
- *Adaptation to packet loss* – techniques such as robust encoding and use of error protection or reduction of the transmission rate to ease congestion can be used to improve the resistance of streams to packet loss.

There are various ways to achieve throughput adaptation (or rate adaptation – adapting the output bit-rate of video based on the available bandwidth) [15]:

- *Bandwidth negotiation* – the server-client system is able to estimate the available end-to-end capacity prior to transmission, and then transmit a stream that matches best the capacity characteristics of the path.
- *Multiple versions* – the server supports multiple versions of the stream encoded at different bit-rates, so that it may switch to the stream that best matches the client's capacity. Some servers support dynamic switching between the multiple streams. The drawbacks of this are that the encoding process is complicated and extra storage is required for the multiple encoded versions.
- *Control of encoding parameters* – a rate controller regulates the output bit-rate by approximately selecting the value of the quantisation parameter by means of rate-distortion optimisation. Other parameters that may be used are the frame skip rate, the resolution and dropping of chrominance components.
- *Dynamic rate shaping* – can be used to achieve reduction of a pre-encoded stream's bit-rate by efficiently eliminating several components of the encoded stream.
- *Scalable video encoding* – the encoding of a video stream into a set of layers in which each layer improves the visual quality of the previous layers. Layers can be added or dropped according to network conditions.
- *Multiple description coding (MDC)* – the video stream is encoded into two independent streams. In contrast to layered encoding, where the presence of lower layers is necessary for the higher layers to be decoded successfully, MDC streams can be independently decoded.

The specific technique used to achieve throughput adaptation in a given scenario is often determined based on the requirements and resource constraints of that architecture.

2.2.4 Discussion

Modern networks are extremely complex, varying both statically and dynamically. This is exacerbated when the network contains mobile elements. A number of solutions to these problems have been proposed based on dynamic adaptation to

changing network conditions and application requirements. The aim is for applications to employ built-in mechanisms that allow them to probe the network conditions and alter their transmission characteristics accordingly, in order to sustain their perceived quality. The research discussed in this section is within the context of video streaming applications.

Badrinath et al [17] summarise the results of several proposals based on dynamic adaptation and extract important general lessons about adapting data flows over difficult network conditions. This is then formulated into a conceptual framework for describing network and client adaptation. Several ways to adapt data flows over networks are highlighted: the underlying protocol can be altered to handle difficult conditions, the data can be compressed or encrypted in a lossless way, lossy adaptations can be used to obtain better compression over limited links, and data can be automatically converted to formats better suited to the end systems or the intermediate networks. The framework is presented in the context of several successful adaptation systems. It distinguishes the functionality of specific components of an adaptive application, allowing the various adaptation-related entities to be decoupled. The conceptual framework that is formulated is able to adequately describe the adaptive systems surveyed by the authors, and aims to be a foundation to describe all adaptive software systems. It provides a starting point for thinking about the general characteristics of software supporting network adaptation. The concepts surrounding dynamic adaptation presented by Badrinath et al highlight the need for network adaptation and provide the basis for the solution proposed in this research.

Cranley et al [18] present an adaptive delivery mechanism that takes into account user perception of quality. The authors propose an optimum adaptation trajectory that indicates how encoding quality should be adapted with respect to user perceived quality in response to network conditions. The quality of encoded material must be adapted to achieve an adaptation in the output bit-rate from a sender. In order to arrive at an encoding configuration which gives the user the best perceptual quality, the resolution of the image and the frame rate are adjusted. Subjective tests are performed that verified that it may be possible to adapt any content type to some generic optimum adaptation trajectory. However, no actual implementations of an adaptive

server based on these concepts have been built, making an objective analysis of the proposal difficult. Nevertheless, the proposal highlights the importance of taking user perception of quality into account. The technique of adjusting the image resolution and frame rate of video in response to network conditions to provide the best perceptual quality to the user is extracted and applied to the solution proposed in this research.

Li et al [19] make use of wireless link layer performance indicators to predict the streaming video quality that should be expected in WLAN environments. The wireless predictors employed include the received signal strength indicator (RSSI), wireless link capacity, MAC layer retry fraction, IP loss rate, round trip time (RTT) and throughput. They conclude that the wireless RSSI and average wireless capacity are the most effective predictors of video performance in WLANs. The effectiveness of individual predictors varies for different video configurations. Additionally, multiple level encoding for adaptation improves video performance over single level encoding in poor wireless conditions, and TCP streaming improves video frame rates compared with UDP streaming in the same regions. The results confirm that it is feasible to use rate adaptation techniques to improve video performance in dynamic WLAN environments. In addition, these results can be used to enhance rate adaptation techniques. The idea of monitoring wireless link characteristics is borrowed. However, instead of simply predicting the streaming video quality that should be expected, this information is used to determine the quality that the streaming video should be adapted to. In this way, video streams can be adapted for optimum performance for a node in a specific region of a WLAN.

Each of the proposals introduced in this section aim to improve the design of applications to promote increased cooperation with networks. The focus is on making adaptation techniques more robust, so that the perceived quality of applications can be sustained in hostile environments. The most important ideas that were borrowed include monitoring WLAN characteristics to determine the optimal video quality that can be achieved, and adapting the image resolution and frame rate of video to provide this optimal perceptual quality to the user.

2.3 Mobile IPv6

2.3.1 Introduction

Wireless networks pose increasing challenges to the delivery of real-time multimedia applications due to the mobility of nodes. Mobile IPv6 (MIPv6) was introduced as the prevalent solution to IP mobility. This section presents an introduction to MIPv6 handover, and how this affects the delivery of video streaming applications. The shortcomings of MIPv6 are illustrated and a discussion on enhancements to the base protocol as well as other network layer driven solutions is presented.

MIPv6 was designed to allow nodes to be reachable and maintain ongoing sessions while changing their location within a network topology. MIPv6 makes use of a stable IP address, the *home address* (HoA), which is assigned to *mobile nodes* (MNs). The HoA allows the MN to have a stable entry in the Domain Name System (DNS) and hides IP mobility from upper layers. The network layer hides mobility from upper layers so that ongoing sessions can be maintained while the MN changes its address. A result of keeping a stable address, independent of the MN's location, is that all *correspondent nodes* (CNs) try to reach the MN at that address, without knowing its actual location. If the MN is not in its home subnet, its *home agent* (HA) is responsible for tunnelling packets to its *care-of address* (CoA) – its current location.

When a MN moves to a new IP subnet, it must first establish a physical link to an *access point* (AP) in the new subnet. This is known as a link layer (layer 2) handover. MIPv6 (a network layer protocol) is not aware of link layer handover as it was designed to operate over heterogeneous networks (networks based on different technologies). The link layer handover may not necessarily have any relation to IP layer mobility. For example, 802.11 mobile devices can handover from one wireless access point to another. This is typically a result of degradation in the quality of the wireless link. The handover is handled by radio link protocols. If IP were used on top of the radio link, the handover may cause a change of the device's location in the network topology. This is illustrated in Figure 2.1.

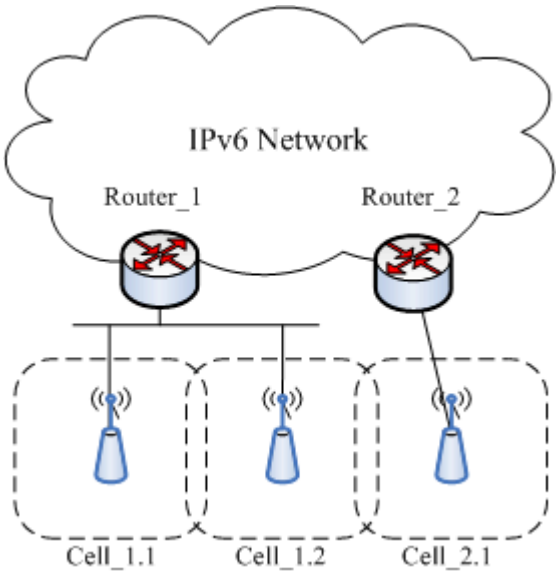

Figure 2.1: Link layer versus network layer handover.

A handover from Cell_1.1 to Cell_1.2 will be handled by radio link protocols. There is no change in the device's location within the network topology as it is still reachable via Router_1. Therefore, the IP layer is unaware of this handover. However, a handover from Cell_1.2 to Cell_2.1 results in a change in the device's location within the network topology, so an IP layer handover is also necessary.

In summary, when a MN moves to a new point of attachment on a new subnet, the MN needs to acquire a CoA in this foreign subnet. It can then inform its HA and CNs about this binding between its HoA and new CoA. Any packets that arrive on the home subnet addressed to the MN are intercepted by the HA and tunnelled to the MN at its current CoA. The MN may continue to send packets directly to its CNs. This type of routing is called triangle routing, since packets from a CN to the MN must be transmitted via the HA. When a CN receives a BU or when it receives packets from the MN's new CoA, it may cache this new CoA and begin routing packets directly to the MN. This is known as route optimisation. The routing of packets during MIPv6 handover is illustrated in Figure 2.2.

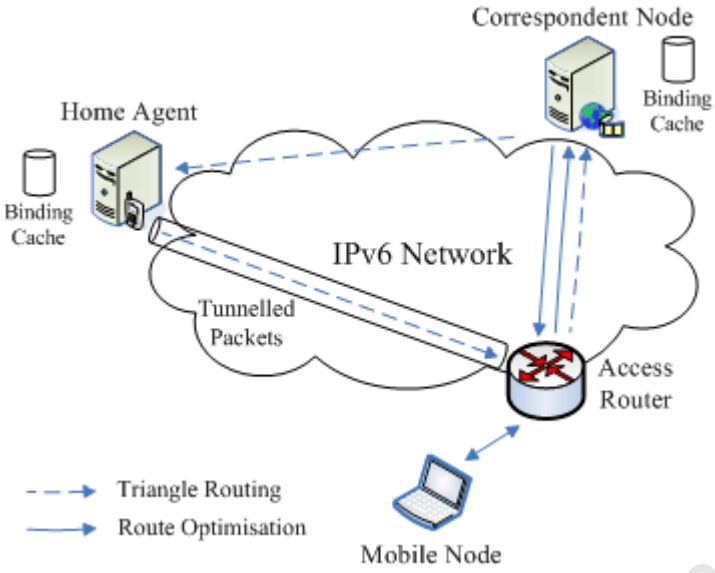

Figure 2.2: Mobile IPv6 packet routing.

The MN cannot receive packets at its new point of attachment until the MIPv6 handover is complete. This includes the time taken for movement detection, the new CoA establishment and handover registration (updating the MN's HA and CNs with its new location). This cumulative time is called the handover latency.

2.3.2 Mobile IPv6 Handover Analysis

The total time during which a MN is unreachable is the sum of the link layer handover latency and the MIPv6 handover latency. Several factors influence MIPv6 handover latency and packet losses. A brief analysis of these factors is presented in an effort to understand the shortcomings of MIPv6.

A MN cannot send or receive data packets for the duration of a handover. Since the MN cannot communicate during this period, the handover directly causes a disruption to a MN's IP streams. The longer the handover latency is, the greater the disruption that the MN experiences. Packets that are sent to the MN's old CoA during a handover cannot be delivered to the MN. In addition, the MN will only receive new packets at its new CoA once the handover is complete. Thus, the MN will experience considerable packet loss. Packets must also be re-routed to the MN once the handover is complete. This can cause increased delays in the delivery of packets and increase the jitter that applications experience. Video streaming applications can only tolerate

minimal packet loss and jitter. Thus, these applications will suffer a considerable deterioration in quality during a handover. This section discusses what causes the most substantial delays during handover, pointing out the inadequacies of MIPv6 and the areas that improvements should focus on. The effect of a handover when either TCP or UDP is used as the transport layer protocol above MIPv6 is then discussed.

MIPv6 handover latency consists of the movement detection time, the time taken to establish a new CoA and the handover registration time. This may be represented as:

$$T_{ho} = T_{mv} + T_{coa} + T_{bu} \tag{2.1}$$

where T_{ho} is the total handover latency, T_{mv} is the movement detection time, T_{coa} is the time taken to establish a new CoA and T_{bu} is the handover registration time.

MIPv6 handover begins with movement detection. The MN detects that it has moved when its current link becomes unreachable. After the link layer handover is complete and the MN is attached to a new *access router* (AR), it may make use of *router solicitations* (RSs) and *router advertisements* (RAs) to detect movement. A RS may be sent if the RA interval has expired and the MN has not received a RA. Alternatively, if the MN receives a layer 2 trigger informing it that a link layer handover has just taken place, it may send a RS to determine if IP movement has occurred and a MIPv6 handover is necessary. If the new router is sending advertisements frequently, the MN may receive an unsolicited RA, which would allow it to avoid sending a RS. These movement detection delays may be represented as:

$$T_{mv} = T_{rs} + T_{ra} \tag{2.2}$$

where T_{rs} is the time it takes the new router to receive the RS and T_{ra} is the period that starts after T_{rs} and ends when the MN receives the RA.

T_{coa} is the time it takes for the MN to form a new CoA and test it for duplication with Duplicate Address Detection (DAD). DAD could take up 1 second or more in the case of duplication, depending on how quickly a response is received [20].

After forming a new CoA, the MN must update its HA and CNs. The delay during this stage depends on the delay incurred by packets sent between the MN and the HA, between the HA and the CN and between the MN and the CN. The time it takes a

message to travel between the MN and the CN via the HA is always longer than or equal to the time it takes for a message to travel directly between the MN and the CN. This is the time it takes to send *binding updates* (BUs) and receive *binding acknowledgements* (BAcks) from the HA and the CN, so:

$$T_{bu} = T_{mn\text{-}ha\text{-}cn} + T_{mn\text{-}cn} \tag{2.3}$$

where $T_{mn\text{-}ha\text{-}cn}$ is the time it takes a message to travel between the MN and the CN via the HA, and $T_{mn\text{-}cn}$ is the time it takes for a message to travel directly between the MN and the CN.

Therefore,

$$T_{ho} = T_{rs} + T_{ra} + T_{coa} + T_{mn\text{-}ha\text{-}cn} + T_{mn\text{-}cn} \tag{2.4}$$

The procedures that cause the most substantial delays during handover are T_{rs}, T_{ra} and DAD delays during T_{coa} [20].

The handover latency and packet loss associated with MIPv6 handover can severely degrade the IP streams of MNs. As a result, application quality is negatively affected. In the case of video streaming applications, MIPv6 handover can completely disrupt the application. The total handover latency is visible only to the MIPv6 implementation in the MN. This is hidden from the upper layers. Therefore, applications and transport layers interpret this interruption as a change in the network's availability, and respond by changing their behaviour. The effect of MIPv6 handover on applications is illustrated in the cases where one of TCP or UDP (transport layer protocols) is used above the network layer. Depending on the transport layer protocol used by an application, the effect of the handover varies.

TCP is a connection-oriented reliable transport protocol. It achieves reliability by retransmitting lost segments and uses a checksum to discover errors in received packets. It transmits packets to applications in the same order in which they were sent. TCP also offers end-to-end flow-control mechanisms to allow connections to adapt to network congestion by changing their transmission rate. TCP assumes that packet losses are a result of network congestion and not due to transmission errors. Wireless links typically have high Bit Error Rates (BER), so the main reason for packet loss on wireless links is not necessarily congestion, but errors. During a MIPv6 handover, there is a handover period during which the MN is unable to receive new TCP

segments containing data or acknowledgements from the receiver. TCP assumes that this is due to congestion and responds by lowering the transmission rate. Once the handover is complete, it will take a significant amount of time for TCP to restore its original sending rate [20]. Thus, video streaming applications will experience relatively less packet loss using TCP, but the end-to-end latency that the applications experience will increase significantly. However, these applications require consistent low end-to-end latency, so they will suffer considerable deterioration in quality.

UDP is an unreliable transport protocol. It does not include any flow control or retransmission. Real-time multimedia applications would typically not want a transport layer to buffer packets or retransmit them if they are lost. It is more important for these applications to receive most of the traffic and present it to the user than to receive all the traffic. So, in the case of real-time traffic, if a packet experiences delays and arrives late, it is simply dropped by the receiver [20]. Therefore, during a MIPv6 handover, video streaming applications that use UDP will experience considerable packet loss. If the handover is very long, applications will experience a significant deterioration in quality, since the information lost due to packet losses is not recoverable.

2.3.3 Handover Enhancements

MIPv6 was designed to meet some important requirements. However, the rapid adoption of new applications (e.g., real-time multimedia applications), the desire to run traditional circuit switching emulation applications (e.g., Voice over IP), and the advances in cellular telephony together with the Internet evolution all contributed to adding new requirements to IP mobility [21, 22]. As discussed in the previous section, the latency and packet loss associated with MIPv6 handover causes too much of a disruption to support many real-time multimedia applications. Several enhancements to MIPv6 have been proposed to improve the handover process. Some have officially been adopted as extensions to the base protocol. Their aim is to make handovers faster (improve handover latency) and smoother (reduce packet loss). A seamless handover is both fast and smooth enough for an application to continue uninterrupted during and after the handover. Several enhancements to MIPv6 have been proposed. Hint-based movement detection, Hierarchical MIPv6 (HMIPv6) and Fast MIPv6 (FMIPv6)

are some of the enhancements that have had the biggest impact on MIPv6 and are extensively discussed in the literature. These are briefly discussed here.

Hint-based movement detection [23, 24] makes use of hints generated during link layer handovers. If the IP layer were made aware of such hints it could force the MN to broadcast a RS, effectively reducing the movement detection latency. Hint-based movement detection enables much faster handovers than traditional advertisement-based methods. Since T_{mv} is a major contributor to handover latency, the handover latency and packet loss associated with MIPv6 is significantly reduced and applications face a lesser disruption.

HMIPv6 [25] makes use of a hierarchy of ARs between the HA and the MN. The performance impact of mobility is reduced by handling local movement locally and hiding this from the HA. Thus, the signalling overhead and delay concerned with the handover registration process in MIPv6 is reduced. T_{bu} is reduced, improving the handover latency and providing crucial support to real-time multimedia applications.

FMIPv6 [26] sets up services for the MN on the new AR before the movement of the MN. This helps to minimise the delay associated with movement detection and removes MIPv6 signalling to the HA or CNs from the critical handover time. FMIPv6 achieves this by allowing MNs to anticipate network layer mobility. The MN makes use of link layer hints to determine whether a MIPv6 handover is imminent, so that the network layer handover may be initiated before the link layer handover is complete. Any packets sent to the previous AR (PAR) are tunnelled to the new AR (NAR) and buffered. When the MN completes the handover, buffered packets are forwarded to it. The signalling for this process is illustrated in Figure 2.3. This type of handover is called a predictive handover [20, 27].

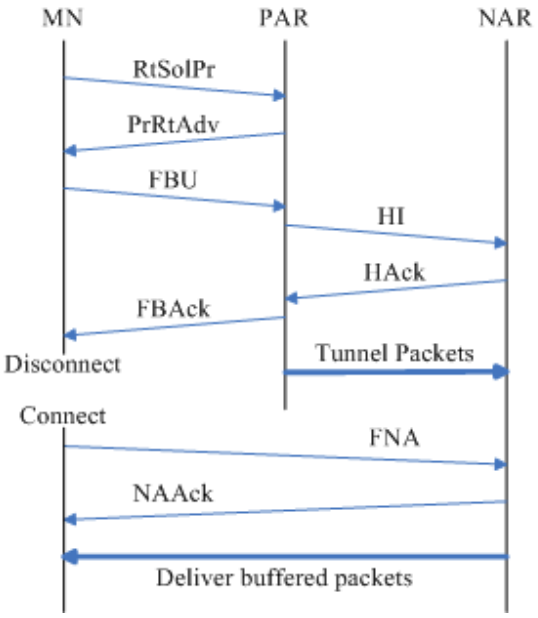

Figure 2.3: Predictive FMIPv6 signalling.

FMIPv6 also defines a reactive handover scenario, where the MN was unable to anticipate handover so it reacts when the handover is already in progress [20, 27]. In this case the NAR sends a fast binding acknowledgement (FBAck) to the MN once the link layer handover is complete. The PAR tunnels packets destined for the MN to the NAR (which then forwards the packets to the MN). The reactive handover signalling is illustrated in Figure 2.4.

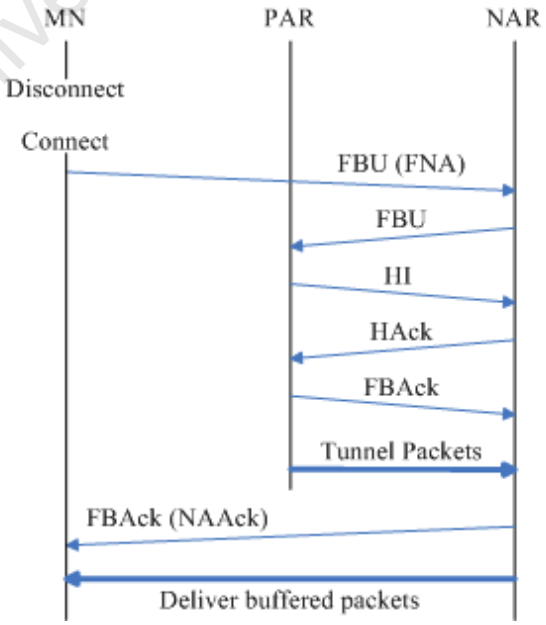

Figure 2.4: Reactive FMIPv6 signalling.

FMIPv6 handovers result in a marked improvement to the handover latency since T_{mv} and T_{coa} (which are the largest contributors to handover latency) are minimised. Furthermore, packet losses are minimised since packets addressed to the old CoA are forwarded to the MN through the NAR. FMIPv6 results in the greatest improvement to handover, as compared to other MIPv6 enhancements. Thus, it causes the least disruption to video streaming applications and enables the best application quality. However, the latency that persists during FMIPv6 can still be too long to allow seamless handovers for video streaming applications.

2.3.4 Discussion

Each of the enhancements to MIPv6 that were introduced is able to practically improve MIPv6 in some respect. FMIPv6 [26] and HMIPv6 [25] have both been adopted as official Internet Engineering Task Force (IETF) standards. Performance shortcomings introduced by IP mobility have been heavily researched. Many proposals to address these problems, beyond the few that will be introduced here, have been made (e.g., the protocols reviewed by Campbell et al [28], and Blondia et al [29]). These are not discussed here as they are not directly relevant to this research.

The simulation study performed by Costa et al [4] provides useful information on the performance of MIPv6 in a WLAN. The handover performance of standard MIPv6 was compared to that of a Fast Handover mechanism and route optimisation was investigated. The authors conclude that the Fast Handover mechanism successfully eliminates packet loss and that even with route optimisation the signalling load only contributes to a marginal portion of the overall handover latency.

Hsieh et al [30] present a performance analysis of HMIPv6 with Fast Handovers. Through simulation they show that Fast Handovers alone are capable of improving MIPv6 handover latency 15-fold. HMIPv6 is able to improve MIPv6 handover latency 7-fold. The combined HMIPv6 with Fast Handovers is able to reduce the overall handover latency by 18 times compared to standard MIPv6. However, while this study provides invaluable insight into designing better seamless handover architectures, the scheme itself is not seamless. Jung et al [31] propose a scheme to

support Fast handover over HMIPv6 (F-HMIPv6), similar to the scheme proposed by Hsieh et al. HMIPv6 reduces the signalling overhead and delay concerned with BUs in MIPv6, but still needs a further enhancement for supporting real-time applications. FMIPv6 is able to reduce handover latency and packet loss effectively. F-HMIPv6 is a scheme to integrate FMIPv6 and HMIPv6 without introducing unnecessary processing overhead. However, the proposal by Jung et al is still an early draft and is a long way from being adopted as a standard.

Sulander et al [32] propose a new method for fast handovers, called Flow based Fast Handover for MIPv6 (FFHMIPv6), and Puttonen et al [33] provide a performance analysis for it. The FFHMIPv6 method requires most of the routers in the IPv6 network to maintain the state information of traffic flows. This flow information is used to control the traffic when a MN moves between logical networks. The handover latency using the FFHMIPv6 method is considerably shorter than with standard MIPv6. However, since FFHMIPv6 is a relatively new proposal, performance comparisons to other MIPv6 improvements are not available.

Pan et al [34] propose an end-to-end multi-path transmission scheme for smooth handovers for streaming media. Multiple paths are acquired during the handover period to reach a single MN. In addition, a multi-layered encoding technique is applied to make the streaming media adaptive to heterogeneous network environments with different bandwidths. Through simulations, the authors conclude that the scheme protects important application data through redundant transmission on multiple paths, avoids drastic quality degradation and stream disruptions, and provides smooth video quality adaptation to the new available bandwidth. However, a network with many highly mobile nodes could become congested very quickly due to the redundant transmission of data on multiple paths. The associated network overhead is also considerable.

The connection disruption at the network layer associated with MIPv6 handover is often too long to support video streaming applications [11]. Network layer solutions typically aim to address the effect of IP mobility by improving the MIPv6 handover process in some way. The studies performed by Costa et al and Hsieh et al verify that FMIPv6 is the most effective enhancement to MIPv6 and results in the greatest

performance gains, as compared to other proposals. The study performed by Pan et al is a novel scheme that aims to minimise packet losses for smooth handovers and adapts the application to heterogeneous network conditions. While this concept has merit, the cost associated with minimising packet losses does not compare favourably to FMIPv6, which is also able to do this. It is important to minimise packet losses, because video streaming applications can only tolerate data loss of 2-3 % [15]. The improvements to handover latency and packet loss that network layer solutions try to accomplish translate to improved delivery of real-time multimedia content. This results in an enhancement in the user perceived quality of these applications. FMIPv6 is able to improve handover performance to a level that video streaming applications are more able to tolerate. Using FMIPv6, it is far more realistic to design an adaptive video streaming handover scheme that will guarantee the continuity of video streaming applications.

2.4 802.11 WLANs

IEEE 802.11 WLAN technology is the access technology considered in this research. Some important characteristics of 802.11 link layer handovers that affect handover performance are discussed below:

- 802.11 handovers are hard, "break-before-make" handovers. A node has to disconnect from a previous *access point* (AP) before connecting to a new AP. Therefore, during handover there will be a period when the node is temporarily disconnected and is likely to experience traffic loss.
- The delay introduced by active scanning for new candidate APs accounts for nearly all of the 802.11 handover time. The selection of an AP is only definite once the scan process is complete. Therefore, the prediction of the node's new AP is speculative at best until the completion of the scan process.
- When there are several nodes connected to an AP, the 802.11 handover latency increases considerably and the available throughput for each node becomes very restricted [11].
- In spite of how comprehensive the 802.11 standard is, manufacturers have a great deal of autonomy in designing and implementing 802.11 products. Thus,

hardware implemented by different manufacturers may result in varying handover delays [23].

These factors contribute to finite link layer handover latencies as nodes move between Basic Service Sets in WLANs. If the node's movement also results in a change in its location in the network layer topology, a MIPv6 handover is required. However, the link layer handover must be completed before the MIPv6 handover begins. Although some enhancements to MIPv6 (e.g., hint-based movement detection and FMIPv6) initiate the MIPv6 handover while the link layer handover is still in progress, some latency is introduced by link layer handover that does not overlap with MIPv6 handover. This has an indirect effect on the length of the MIPv6 handover. The link layer handover latency disrupts the delivery of video streaming applications, resulting in degradation in application quality. Therefore, it is important to take link layer handover into account when designing IP mobility management schemes.

2.4.1 Discussion

Link layer techniques to support video streaming applications predominantly aim to achieve this by guaranteeing some form of QoS to the application.

Garcia-Macias et al [1] consider a mobility management scheme optimised for QoS. Their proposal aims to integrate management of QoS and mobility based on a hierarchical architecture. Several parameters of a WLAN cell are controlled: the geographical span is limited to ensure the same high bit-rate for all nodes, the rate of traffic sources is constrained to limit the use of the channel as a function of the required QoS and the number of active nodes is limited to keep the load sufficiently low. This QoS management is coupled with mobility management at the IP layer. A micro-mobility scheme is implemented in the IP layer that avoids address translation and traffic tunnelling, and enables fast handovers. The results of this scheme show that it is able to provide substantially better performance to priority traffic, isolate different QoS classes and manage mobility efficiently.

More traditional approaches are focussed on methods to guarantee QoS in the MAC sub-layer of 802.11. Choi et al [35] provide a performance evaluation of the Enhanced

Distributed Coordination Function (ECDF) algorithm proposed for 802.11e [36]. EDCF is a contention-based channel access mechanism that provides differentiated channel access to frames with different priorities. Evaluations show that EDCF can successfully provide differentiated channel access for different types of traffic. Grilo et al [2] analyse and compare the Distributed Coordination Function (DCF) and Point Coordination Function (PCF) algorithms. In addition, mechanisms to enhance these algorithms were assessed. Experiments showed that Enhanced DCF and an improvement on PCF are both able to protect higher priority traffic in the presence of lower priority bursty data traffic, which standard DCF and PCF are unable to achieve. Lindgren et al [37] performed a similar study, evaluating PCF, Enhanced DCF, Distributed Fair Scheduling (DFS) and Blackburst. Their simulations showed that while Blackburst achieved the best performance, PCF and EDCF are also able to provide good service differentiation. Nevertheless, it is difficult to conclude that one QoS scheme is better than another, since it largely depends on the context where it is used.

QoS provisioning implementations in 802.11 are still in their early stages. The 802.11e standard that aims to provide QoS to legacy 802.11 standards has only recently been released. Therefore, it has not been possible to carry out a critical evaluation of 802.11e. In spite of proposals to guarantee QoS on the link layer, video streaming applications face mounting challenges in the context of IP mobility – link layer handover latency still persists and the delays associated with MIPv6 are not addressed.

The idea of QoS provisioning at the link layer is a key part of guaranteeing application quality. However, guaranteeing QoS is complex, and in the context of this research the gains to be achieved are limited. Furthermore, any link layer specific solution would restrict the proposal exclusively to 802.11 networks. Thus, QoS provision at the MAC layer of 802.11 is not integrated into the video streaming handover scheme proposed in this research. Nevertheless, the literature illustrates the impact of data transmission at the link layer and the importance of service differentiation. Wireless characteristics such as the RSSI and the wireless link capacity also provide useful information that can be used by upper layers to optimise the delivery of application data.

2.5 Cross-layer Approaches

Existing standard protocol stacks, which are designed and implemented in a layered manner, do not function efficiently in a mobile wireless environment. They have been criticised for being too rigid in the way they operate. A layer may only communicate with the layers directly above or below it and in a specified format. Many researchers have now adopted the philosophy that cross-layer feedback would be useful to improve the efficiency of these protocol stacks [13, 14]. Cross-layer feedback implies interaction between the layers of the protocol stack. Some examples of the interactions between specific layers of the protocol stack are summarised below.

The information available at the *link layer* includes the current forward error correction scheme in use, the number of frames retransmitted, frame length, the RSSI, wireless link capacity and handover related events. Link throughput information can be used as an indication of the kind of application performance to expect, so a user can decide what applications can be run. Retransmissions at the link layer when channel conditions are poor could cause TCP retransmissions at the transport layer, reducing throughput. To avoid this, TCP and the link layer may exchange retransmission information. Link layer handover information can be used to reduce MIPv6 handover latency at the network layer.

The information available at the *network layer* includes MIPv6 handover events and the network interface currently in use. Applications can control their sending rates based on MIPv6 handover indications. A device could have multiple wireless network interfaces, so based on application requirements, the network layer could select the appropriate interface to use. At the transport layer, MIPv6 handover latency could reduce TCP throughput. If TCP was informed about the handover, the retransmission latency could be reduced.

The information available at the *transport layer* is the transport protocol in use and its parameters. Transport protocols can provide packet loss and goodput information to applications, which may then adapt their sending rates.

The *application layer* can communicate its QoS needs (e.g., the acceptable jitter, required throughput, and acceptable packet loss rate) to other layers. Information about channel conditions from the physical and link layers is crucial in improving application performance. Link layer error control mechanisms may be adjusted according to application QoS requirements to improve application throughput.

The paradigm shift from strictly layered protocol stacks towards cross-layer designs has been triggered by the inadequacy of standard protocol stacks and the obvious benefits of cross-layer feedback. A combination of some the interactions mentioned above can be used to design a robust video streaming handover scheme to support video streaming applications. The feedback can improve the performance of existing protocols, and thus, the delivery of real-time traffic.

2.6 Chapter Discussion

This chapter began by introducing what application quality is, the effect of network streaming on video quality, and rate adaptation techniques to foster cooperation between applications and networks. Related research was reviewed, including a proposal to optimise the adaptation of video quality to maximise the user perception of quality and another that makes use of wireless link characteristics to predict the video quality to be expected in specific regions of WLANs. These ideas were extracted and applied to this work.

Thereafter, MIPv6 was introduced and a brief analysis of the handover process was given to illustrate the impact of handover on video streaming applications. The effect of handover was discussed in the cases where either TCP or UDP is used above the network layer. A few enhancements to MIPv6 that aim to reduce the impact of handover on applications were discussed. Of particular interest is FMIPv6, which is able to minimise the handover latency and packet loss experienced by a MN. Related research was presented that verifies the significant improvement to the handover disruption experienced when MIPv6 enhancements are used. A more effective video

streaming handover scheme can be designed by making use of a mobility management protocol such as FMIPv6 at the network layer.

Some important characteristics of 802.11 link layer handovers were then mentioned. These factors indirectly affect the MIPv6 handover and contribute to handover latency. Related research illustrated the importance of QoS provisioning to guarantee application quality. However, guaranteeing QoS at the MAC sub-layer of 802.11 is fairly complex and would restrict the proposed solution exclusively to 802.11 networks. Thus, this is not integrated into the solution proposed in this research.

This led to the rationale for cross-layer design optimisations to support applications. The improved communication between layers of the protocol stack can provide useful information that will allow applications to better adapt to changing network conditions. The proposed solution makes use of network layer information to alter application layer characteristics of video streams.

Chapter 3

Requirements and Analysis of a Video Streaming Handover Scheme

3.1 Introduction

This research investigates the effects of IP mobility on the continuity of video streaming applications. The greatest obstacle in achieving the required level of quality is the disruption to IP streams caused by MIPv6 handover. In this chapter, the requirements for a seamless video streaming handover scheme are discussed. A suitable scheme that meets these requirements and aims to guarantee application continuity is then formulated. Subsequent chapters focus on assessing whether handover performance can be improved to a level where such a scheme would be effective.

MIPv6 provides a solution to IP mobility. However, IP mobility still results in a broken data path from server to client. Thus, packets that have already been tunnelled to a MN's old CoA cannot be delivered to the MN and are therefore lost. In addition, packets sent to the MN's new CoA must first be intercepted by the HA before they are tunnelled to the MN resulting in communication routes that are significantly longer than the optimal routes and introducing extra delay for packet delivery. After movement, a MN must update its HA and CNs with its new location. If nodes are highly mobile, the signalling overhead associated with this will become quite significant as the number of MNs increases. Therefore, the handover latency and

packet loss associated with handover is often too long to support video streaming applications [11]. Several enhancements to MIPv6 (e.g., HMIPv6 and FMIPv6) have been proposed to improve the handover procedures and reduce the impact of MIPv6 handover on application quality. However, while the impact of handover can be reduced, MIPv6 cannot guarantee seamless handovers. Therefore, a more robust mobility management solution is necessary.

The type of real-time multimedia application studied is a Video on Demand (VoD) streaming application. VoD applications enable users to select stored video from a central server for viewing over a network. The client (in this case, a MN) requests a particular video stream from a server, and the server responds by sending a unicast stream of that video to the client. A VoD application was chosen mainly because no interactivity is required between the user and the application while the video is streaming. This simplifies the quality requirements of the application since only the downlink bandwidth needs to be managed, and allows the stream to be manipulated more easily. In addition, the architecture of the server is simplified since it only needs to store video and respond to client requests.

The seamless video streaming handover scheme formulated in this work is based on a cross-layer approach. A combination of existing techniques to enhance handovers and to adapt to changing network conditions will result in improved handover performance and enable application continuity to be guaranteed [13, 14]. The next section presents the requirements of an ideal video streaming handover scheme. A handover scheme that meets these requirements is then formulated and briefly discussed.

3.2 Requirements of a Video Streaming Handover Scheme

Providing seamless handovers for video streaming applications without any transmission disruption and quality degradation is a challenging proposition because MN movement results in a change in point of attachment to the network, and consequently in a broken data path from server to client. Taking this into

consideration, the requirements of an ideal video streaming handover scheme should be to:

- reduce the packet losses caused by handover to avoid quality degradation;
- minimise the handover latency to reduce the impact of the handover on the application;
- maintain the continuity of the application during the handover.

3.3 Overview of a Suitable Handover Scheme

The scheme that is proposed is based on adapting application quality so that the rate at which video frames are delivered to a streaming client (a MN) can be increased. The additional video frames are buffered to cover the disruption in the stream during the handover period. A mobility management protocol must be integrated into the scheme to handle IP mobility. An existing enhancement to MIPv6 may be used to minimise the impact of handover on the application and thus, the disruption that must be compensated for. This scheme will guarantee seamless handovers.

For the explanation of the scheme, the **transport bit-rate** of the stream is defined as the rate at which data is transmitted from the server to the client at the application layer. The **encoded bit-rate** of the stream is defined as the bit-rate at which the video is encoded to be streamed.

FMIPv6 is the most valuable enhancement to MIPv6 discussed in the literature, since it is able to effectively reduce packet loss to near zero and minimise handover latency. For this reason, FMIPv6 is the mobility management protocol used at the network layer in this scheme.

A streaming video server can adapt the encoded bit-rate and transport bit-rate of a video stream to a MN. Thus, the available bandwidth on the link can be managed based on predetermined requirements. If the MN can ensure that it always buffers sufficient video frames to allow for a disruption in the data path during a possible future handover, when the handover occurs the application will be able to tolerate the disruption.

This can be achieved by reducing the encoded bit-rate of the stream but maintaining the same transport bit-rate, increasing the rate at which video frames are delivered to the MN. However, the MN will continue to view video frames at the same rate (in real-time). The "extra" frames are buffered and may be viewed during the handover period. The quality of the video stream must be adapted to achieve the new encoded bit-rate. When the handover is complete the MN will continue to receive packets via the new AR. FMIPv6 ensures that any packets sent to the previous AR during the handover period would have been forwarded to the new AR. Therefore, packet loss should be near zero. If the completion of the handover is guaranteed before all the buffered frames are viewed, the video will continue streaming uninterrupted to the MN during and after the handover. Thus, the handover will appear seamless to the end user. This scheme relies on a combination of FMIPv6 and adaptive video streaming. It is referred to as the Adaptive Mobile Video Streaming (AMVS) scheme. Figure 3.1 illustrates how all the signalling in the scheme occurs at the application layer.

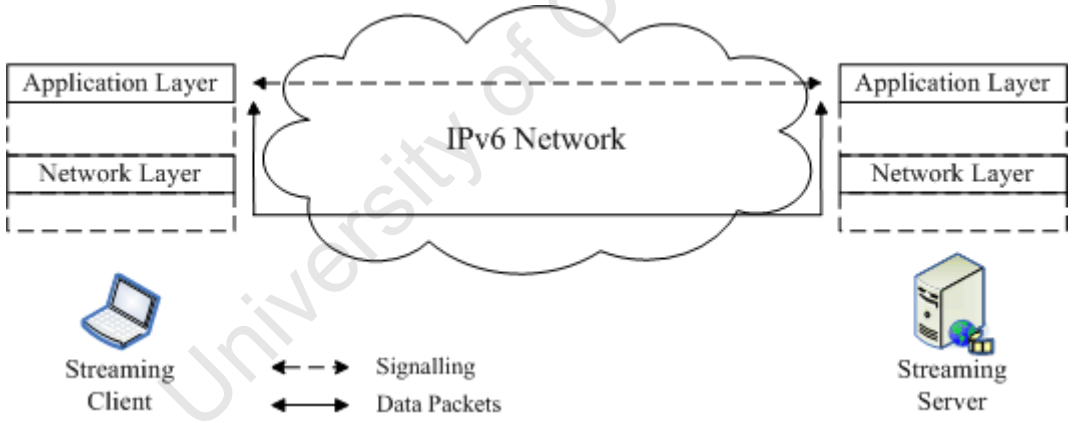

Figure 3.1: Overview of the Adaptive Mobile Video Streaming scheme.

The AMVS scheme depends on being aware of the handover latency a MN should expect. This is determined by the MN. Sufficient video frames must be buffered prior to handover to cover this period. Since the packet loss is kept to a minimum, the MN will continue receiving video frames normally after the handover.

3.4 Description of the Handover Scheme

Let the current transport and encoded bit-rate of a video stream to a MN be N. Let the handover latency to be expected be L and let a handover be imminent in t seconds. This scenario is illustrated in Figure 3.2. Thus, at $T < 0$, the transport bit-rate and the encoded bit-rate are both N.

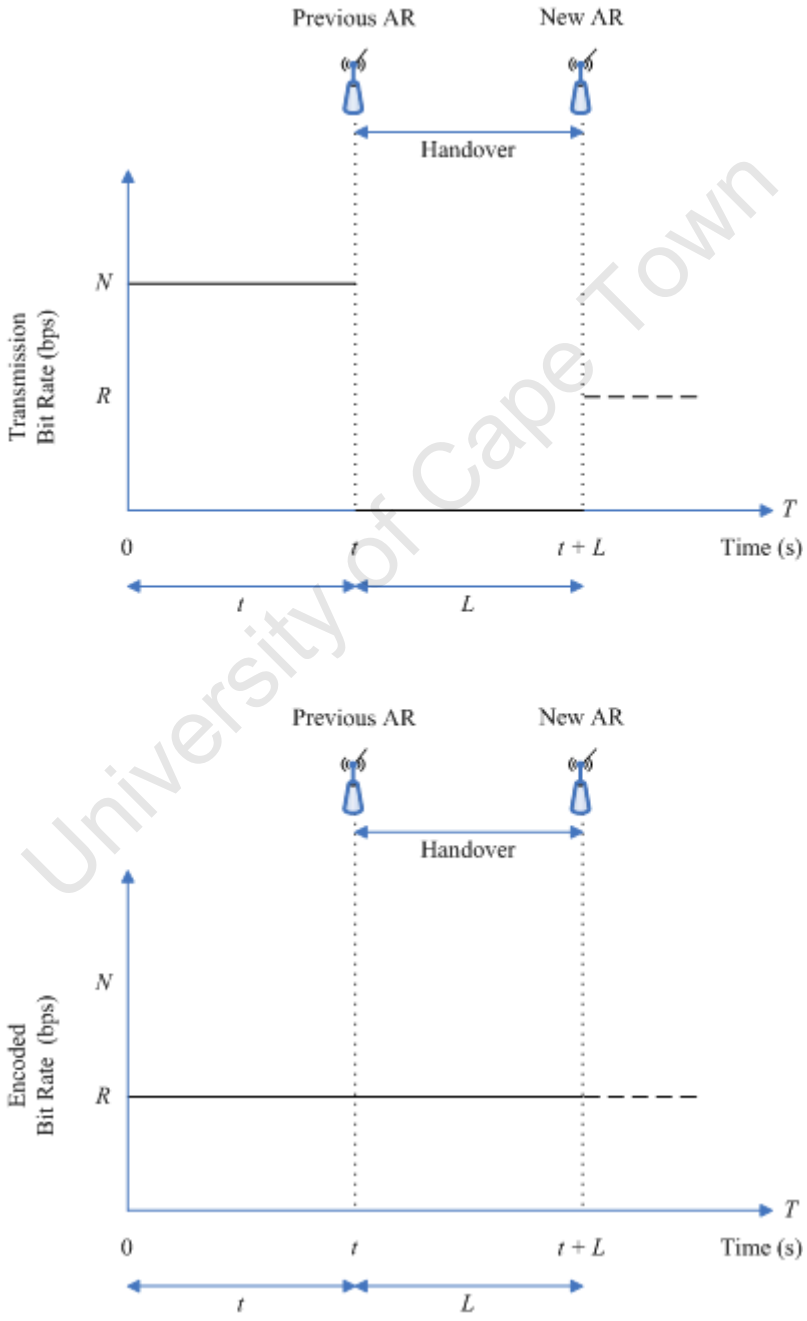

Figure 3.2: Illustration of the bit-rate adaptation.

In order to guarantee the continuity of the application, the MN should buffer sufficient additional video frames during t to cover the disruption that will be experienced during L. To send additional frames before t at the same encoded bit-rate (N), the server will have to send data to the MN at a higher transport bit-rate. However, the MN has no guarantee that the network will have the available resources to achieve a higher transport bit-rate. Thus, this is not a feasible option.

Therefore, the MN requests that the server sends the stream at a reduced encoded bit-rate of R (i.e., reducing the quality of the video stream) but at the same transport bit-rate of N. This is illustrated in Figure 3.1 during $0 < T < t$. During $t < T < t + L$, the MN receives no additional data so the transport bit-rate is zero. However, it views the frames that have been buffered, so the encoded bit-rate is still R. Thus, the same amount of data will be sent to the MN at a bit-rate of R to cover $t + L$ that would have been sent at a bit-rate of N to cover t:

$$Nt = R(t + L) \tag{3.1}$$

Then,

$$R = \frac{Nt}{t + L} \tag{3.2}$$

For example, if a handover is imminent in 5 s, the expected handover latency is 1.5 s and the current bit-rate of the stream is 400 kbps, then at $T = 0$, the encoded bit-rate of the stream should be adjusted to R:

$$R = \frac{400 \times 5}{5 + 1.5} = 307.7 \text{ kbps}$$

The transport bit-rate during t will remain at 400 kbps and during L it will be 0 kbps, but the encoded bit-rate during t and L will now be 307.7 kbps.

Since the same transport bit-rate is maintained, there are no additional resource demands on the network.

The MN requires a protocol to be able to communicate its requests to alter the encoded bit-rate of the stream while maintaining the same transport bit-rate to the server. A trigger is required at $T = 0$ to inform the server to alter the stream. However,

the purpose of this research is not to implement this scheme but to assess whether such an approach to the problem is feasible given the handover performance. Therefore, a protocol to do this has not been defined.

3.4.1 Determining the New Encoded Bit-rate

For the scheme to be feasible, it must be possible to determine *R*. In order to do this, *N*, *t* and *L* must all be known. *N* is the current bit-rate of the stream, which is known.

Handovers must be predictable, so that *t* can be determined. Therefore, some technique must be employed to establish whether a handover is imminent. A number of mobility prediction techniques exist. The most significant of these rely on an analysis of the MN's prior movements or locations [38]. However, predicting MN movements is an extremely complicated process. Since the focus of this research is on determining whether the AMVS scheme is feasible given the handover performance, integrating a mobility prediction algorithm into the scheme has been set aside as future work. Thus, it is assumed that a handover will definitely take place at some point in the future, so one is always imminent. This assumption does not add overhead to the network infrastructure, it simply implies that a user may be forced to view a video stream at a lower encoded bit-rate than the capacity on the link allows for. This cost is negligible for the analytical purpose of this work.

The expected handover latency (i.e., *L*) is the duration of the disruption in the stream that the buffer (of video frames) must compensate for. However, the handover latency is not an absolute value and varies depending on network conditions. The standard deviation is a gauge of the range of variation from an average group of measurements. If the handover latencies measured have a Gaussian distribution, at least 68 % of measurements will fall within one standard deviation of the average, and at least 95 % of measurements will fall within two standard deviations of the average. If *L* was defined as the maximum handover latency to expect, then the size of the buffer would be adequate to cover the entire range of possible values of the handover latency (i.e., 100 % of the handover latencies measured would be less than or equal to *L*). However, a very small percentage of measured handover latencies would occur near

the maximum, so on average the size of the buffer would be much larger than necessary. Therefore, *L* is defined as the average handover latency experienced by a MN in a given scenario plus one standard deviation, as illustrated in Figure 3.3 (i.e., at least 84 % of the measured handover latencies will be covered).

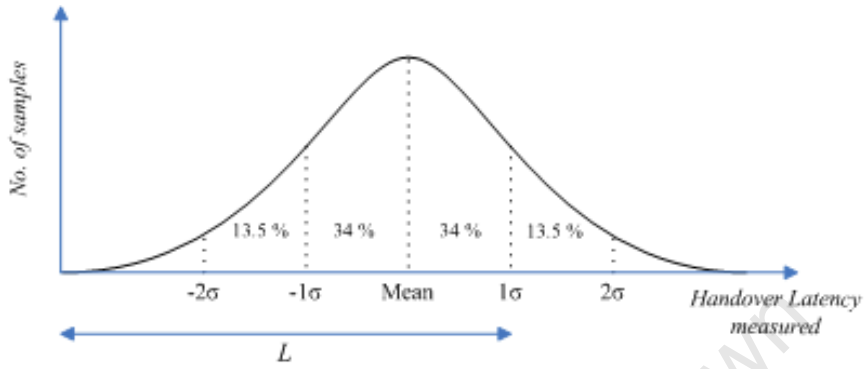

Figure 3.3: *L* is defined as the average handover latency plus one standard deviation.

The aim of the evaluations that follow is to determine *L* in a practical scenario and establish whether the AMVS scheme is feasible given the value of *L*. If the handover latency is too long, it is unrealistic for the scheme to create a large enough buffer and sustain it before the next handover occurs. This problem is exacerbated if a MN is highly mobile. In order for the scheme to be able to guarantee application continuity, *L* must have an acceptable upper bound. Since VoD applications require minimal interactivity, the delay requirements are fairly relaxed. A start-up delay of the order of seconds can be used to build up a buffer to improve the application's resistance to any variation in packet inter-arrival times and, thus, to minor disruptions in the stream. Higher values of jitter, even up to 500 ms, are considered acceptable [15]. However, the application will be unable to tolerate major disruptions of the order of seconds that are likely to occur during handover. The AMVS scheme is expected to only be able to build a large enough buffer to account for a disruption of up to 1 s. Therefore, for the scheme to be feasible, the upper bound for *L* is 1 s.

3.4.2 Requirements of the Streaming Server

A streaming video server typically encodes video and streams the encoded data in real-time. However, this requires significant amounts of real-time processing power and is a considerable overhead for a server supporting many clients. Since a VoD application is considered in this research, it is feasible for the server to make use of multiple versions of each video. Several copies of each video are encoded at different predetermined bit-rates and stored on the server. The main drawback of this method is that extra storage is required for the multiple encoded versions. In this research, only handovers between 802.11g WLANs in a fixed architecture are considered, resulting in a constant *L*. So, depending on the anticipated time before a handover and the current bit-rate of a stream, the server can simply refer to a lookup table to determine *R* and transmit the copy of the video with a bit-rate that closest matches *R*.

The server must also be able to support dynamic switching between the multiple versions of a video stream. In order to achieve this, it is necessary to encode the different versions of a video with synchronisation points in the streams. The drawback of this is that the encoding process becomes more complicated. However, since the multiple versions of the video are stored and not encoded during streaming, the encoding process does not contribute any additional overhead to the scheme.

3.5 Chapter Discussion

This chapter discussed the requirements of a seamless video streaming handover scheme over MIPv6. Thereafter, a suitable handover scheme was described.

FMIPv6 is able to reduce the packet losses caused by handover and minimise the handover latency, avoiding drastic quality degradation and minimising the impact of the handover on the application. The adaptive technique to buffer video frames in anticipation of handover helps to maintain the continuity of the application during the handover. The MN makes use of handover related events at the network layer to determine when a handover is in progress and when it is completed. The AMVS

scheme makes use of network layer information to alter application layer characteristics of the stream (the image resolution and frame rate of the video), changing the quality requirements of the stream in response to network conditions. Thus, it is able to guarantee the continuity of a VoD application during a MIPv6 handover.

The following chapters aim to analyse the feasibility of this scheme using a suitable evaluation framework. In order to verify that such an approach to guaranteeing application continuity is feasible, it is necessary to determine the handover latency to expect and ensure that it is within acceptable bounds. MIPv6 handover performance is compared to FMIPv6 handover performance and an assessment is made on whether the disruption to the application can be mitigated by using the AMVS scheme.

Chapter 4

Architecture and Implementation of an Evaluation Framework

In Chapter 3 a seamless video streaming handover scheme was formulated. This chapter presents the architectural design and implementation of a suitable test-bed to assess the feasibility of the AMVS scheme and the extent to which it would be effective in a practical scenario. First the objectives, requirements and limitations of the evaluation framework are presented. Then its architecture and hardware implementation are discussed. Thereafter, the software used to implement various aspects of the test-bed is discussed.

4.1 Objectives

The key objectives of the evaluation framework are:

- To determine the handover latency and disruption an application is likely to encounter during a MIPv6 handover. A suitable framework must adequately model a real-world scenario that a VoD streaming application may encounter.
- To compare FMIPv6 handover performance to traditional MIPv6 handover performance and determine whether FMIPv6 is able to significantly improve handover to support a VoD streaming application.
- To determine whether it is feasible to use the AMVS scheme to mitigate the disruption experienced during handover and, thus, guarantee the continuity of

a VoD streaming application in this context. The most significant parameter investigated is the handover latency experienced by a MN.

4.2 Requirements of the Evaluation Framework

In order to satisfactorily fulfil the objectives of an evaluation framework, a suitable test-bed architecture must be designed. A suitable test-bed should satisfy the following requirements:

- It must be fully IPv6 compliant.
- It must be capable of supporting MIPv6.
- The access technology considered in this research is 802.11g, so an 802.11g WLAN must be integrated within the architecture.
- It must be capable of supporting FMIPv6.
- In order to effectively test FMIPv6 handovers, at least two foreign subnets are required.
- A HA is required in the home subnet to maintain MN bindings.
- A streaming video server must be implemented as a CN, so that a MN can establish a communication session with it.
- At least one MN must be implemented to establish a session with the CN and move between the subnets.

4.3 Limitations

The evaluation of handover performance could be performed either in a simulation environment or on a physical test-bed. Simulations are an effective means to predict network performance. However, they do not always take real-world factors into account. So while they may allow experiments to be performed on complex network topologies, they often do not reflect the true behaviour of networks. A physical test-bed overcomes the limitations of simulations and allows a far more realistic analysis to be performed. Therefore, it was decided to implement a physical test-bed to analyse the feasibility of the AMVS scheme. However, a physical test-bed limits the research because it is not easy to adapt to changes in topology or to new experiments.

The focus of the evaluation is on determining the handover latency and disruption to a VoD streaming application that should be expected and whether it is realistic to use the AMVS scheme to guarantee the continuity of the application. For the purpose of the evaluation framework, it is not necessary to implement a video streaming server that satisfies all the requirements of the AMVS scheme. Therefore, a standard VoD server is used. The analysis of traffic at the network layer is sufficient to assess the feasibility of the proposed scheme.

The test-bed is designed so that the wireless cells in the topology almost entirely overlap. The MN is also stationary. Therefore, handovers between cells are initiated by forcing the MN to associate with a different AP. There are several ways to force this, and depending on the mechanism used, the link layer handover process (and consequently the link layer handover latency) may be affected.

4.4 Architecture of the Evaluation Framework

A test-bed is designed based on the requirements laid out in Section 4.2. It consists of an infrastructure mode 802.11g WLAN on a gigabit Ethernet wired backbone. The architecture (illustrated in Figure 4.1) consists of a home subnet and two foreign subnets. The HA, which also plays the role of a border router connecting the local network to the Internet, resides in the home subnet. The HA is implemented on a Linux Desktop Computer. An 802.11 AP is connected to the HA via an Ethernet switch, providing wireless connectivity in the home subnet. Each foreign subnet houses an AR. The ARs are implemented in software on Linux Desktop Computers. An 802.11 AP is attached to each AR, providing wireless connectivity in the foreign subnets. A MN equipped with an 802.11 wireless network adapter can attach to any of the subnets via the wireless APs. A CN is present to establish a communication session with the MN for testing purposes. The CN is implemented in the home subnet to simplify the architecture. Using this framework, the MN can establish a communication session with the CN and then move between the subnets. The MN is implemented on a stationary Linux Desktop Computer. Therefore, movement is emulated by forcing the MN to associate with a different AP.

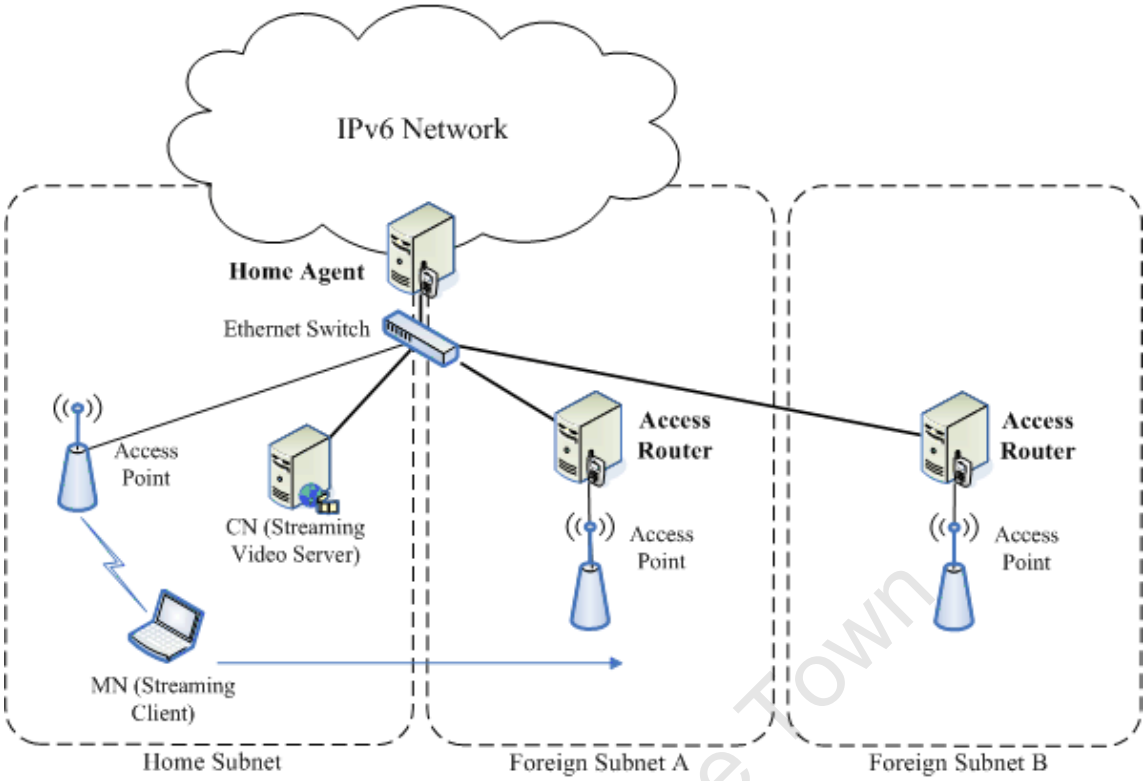

Figure 4.1: Evaluation framework architecture.

Commercially available APs and network adapters are used in the test-bed. Table 4.1 summarises the list of hardware used for the implementation. Each of the network nodes is implemented on an ordinary Desktop Computer, running on a 2.53 GHz Intel Celeron D processor with 256 MB RAM.

Table 4.1: List of hardware used in the evaluation framework.

Hardware	Model	Quantity
HA, ARs, MN, CN	Intel-based Desktop Computer	5
802.11 APs	D-Link DWL-7100 AP	3
8-port Gigabit Ethernet switch	D-Link DGS-1008 D	1
Gigabit Ethernet network adapters	D-Link DGE-528 T	6
Wireless 802.11 network adapter	Cisco AIRONET PCI350	1

4.5 Software Used for the Implementation

The test-bed is first set up as a fully functional IPv6 network. Since no commercial routers are used in the test-bed, IPv6 routing software runs on the Desktop Computers used as routers. In addition, software is required to send out periodic router advertisements. Once the IPv6 network is functional, software is necessary to extend the framework to support MIPv6 and FMIPv6. This section introduces the routing software used for the framework. Thereafter, a survey of available MIPv6 implementations is presented and the chosen implementation is briefly discussed.

Each of the network nodes is implemented on a Linux platform for several reasons. The Linux Operating System is open source software that is highly flexible and adaptable. These factors make it desirable for the implementation, as it is far easier to modify the source code of the various components of the test-bed. The decision to use a Linux platform was also influenced by the availability of suitable and robust MIPv6 and FMIPv6 implementations. This is discussed further in Section 4.5.2.

4.5.1 Routing Software

Quagga [39] is the routing software used on the test-bed. *Quagga* is a routing software package that provides TCP/IP based routing services with support for IPv4 and IPv6 routing protocols. A system with *Quagga* installed acts as a dedicated router. The system exchanges routing information with other routers using routing protocols. *Quagga* uses this information to update the kernel routing table so that the right data goes to the right place. The *Quagga* architecture consists of a core daemon, *zebra*, which acts as an abstraction layer to the underlying Linux kernel. Several other daemons implement specific routing protocols (e.g., *ripngd* implements RIPng, an IPv6 routing protocol). For detailed information on *Quagga*, refer to the *Quagga* documentation [40].

Stateless address autoconfiguration in IPv6 requires periodic broadcasts of router advertisements [41]. This needs to be implemented in the evaluation framework so that a MN can establish a new CoA when it moves to a new subnet. The router

advertisement daemon (*radvd*) [42] is software that implements this on Linux systems acting as routers. It sends router advertisements to a local Ethernet LAN periodically or when requested to by a node sending a router solicitation.

4.5.2 MIPv6 Implementations

The MIPv6 specification [9] is relatively recent (reaching RFC status in 2004), and is still an immature protocol. Further improvements and extensions to it are ongoing in the IETF MIPv6 working group. Consequently, implementations of MIPv6 are often incomplete or still developing. Additionally, many of the early implementations of MIPv6 have fallen out of date as the protocol has progressed. Nevertheless, there are a few implementations that are up to date with the MIPv6 specification. However, most of these implementations still differ slightly in their supported features and are notorious for poor interoperability. The 6NET group provides a survey of available implementations in their MIPv6 deployment guide [43]. A summary of some MIPv6 implementations that are known to comply with the MIPv6 specification and are publicly available is given in Table 4.2.

Mobile IPv6 for Linux (MIPL) [44] was chosen as the MIPv6 implementation to be used on the test-bed. MIPL was developed by the GO-Core project at the Helsinki University of Technology (HUT). MIPL 2.0.2, the version used in the evaluation framework, is a full implementation of the MIPv6 specification. It is freely available under a GNU license and operates on a Linux platform, a highly flexible, open source operating system. MIPL was selected for the evaluation framework primarily because it is a complete and robust implementation of MIPv6 and a significant amount of technical support exists. Furthermore, there are other open source software projects that aim to extend MIPL to support extensions to MIPv6, such as HMIPv6 and FMIPv6.

Table 4.2: MIPv6 implementations, adapted from the 6NET MIPv6 deployment guide [43].

	MIPL	Cisco	Microsoft	KAME	HP-UX
Platform	Linux 2.6.16	Cisco IOS	2000/XP/CE	FreeBSD	HP-UX-11i
Modes	MN/HA/CN	HA/CN	MN/HA/CN	MN/HA/CN	HA/CN
PND	Yes	Yes	Yes	Yes	Yes
IPv6-in-IPv6 tunnelling	Yes	Yes	Yes	Yes	Yes
DHAAD	Yes	Yes	Yes	Yes	Yes
Binding Management	Yes	Yes	Yes	Yes	Yes
HAO	Yes	Yes	Yes	Yes	Yes
Movement Detection	RAs	N/A	RAs and NDIS notifications	RAs	N/A
Smooth Handoff	Yes	Yes	Yes	Yes	N/A
IPsec	No (v1.1) Yes (v2.0)	No	Yes	Yes	Yes with HP-UX
Key exchange	MD5 or SHA-1	No	Manual	Manual	Unknown
MIPv6 built-in	No	Yes	No	Yes, but not enabled by default	No. Is a component of TOUR 2.0
Set-up tools	mip6d daemon	Command line tools	Auto-config, command line	Command line tools	Unknown
License	GNU	Commercial	Commercial	GNU	Commercial

One such project is fmipv6.org [45]. It is an FMIPv6 implementation based on HUT's MIPL 2.0.x and developed by the Network Research Team of the Louis Pasteur University. The goal of the fmipv6.org project is to provide a fully compliant implementation of the FMIPv6 specification [26]. Currently, it is the only open source software project that attempts to implement FMIPv6. The lack of availability of any other implementations of FMIPv6 influenced the decisions to develop the test-bed using MIPL on a Linux platform. However, the current fmipv6.org implementation is still quite experimental and required some modifications before it was able to operate correctly on the test-bed. The modified source files are available on the attached CD-ROM in Appendix C. The modified fmipv6.org implementation was used to extend the evaluation framework to support FMIPv6.

4.6 Implementation of the Evaluation Framework

4.6.1 The IPv6 Network

The first step in implementing the evaluation framework is ensuring that a functional IPv6 network is set up. Each of the Desktop Computers (the HA, ARs, MN and CN) is set up to run the Debian distribution of the Linux operating system, with kernel version 2.6.16. The Linux 2.6.x kernel has built in IPv6 capability. The network is set up according to the architecture described in Section 4.4.

For a functional IPv6 network each node in the network needs to be assigned an IPv6 address. While random IPv6 addresses could be statically assigned to each of the network entities, this would mean that the network remains isolated from the global IPv6 network. Instead, it is advantageous to apply for authorized IPv6 addresses, allowing each machine to be connected to the global IPv6 network. There are very strict regulations used for assigning these addresses. Numerous IPv6 address providers, known as brokers, are responsible for assigning individuals or organisations these addresses.

A global address was obtained for the border router between the local network and the Internet (i.e., the HA) from such a broker. Specific details on how this was done are available in Appendix B. The address was assigned with a 48 bit prefix. This prefix is split to form three subnets with 64 bit prefixes. Addresses are then assigned to all the nodes in the framework as illustrated in Figure 4.2.

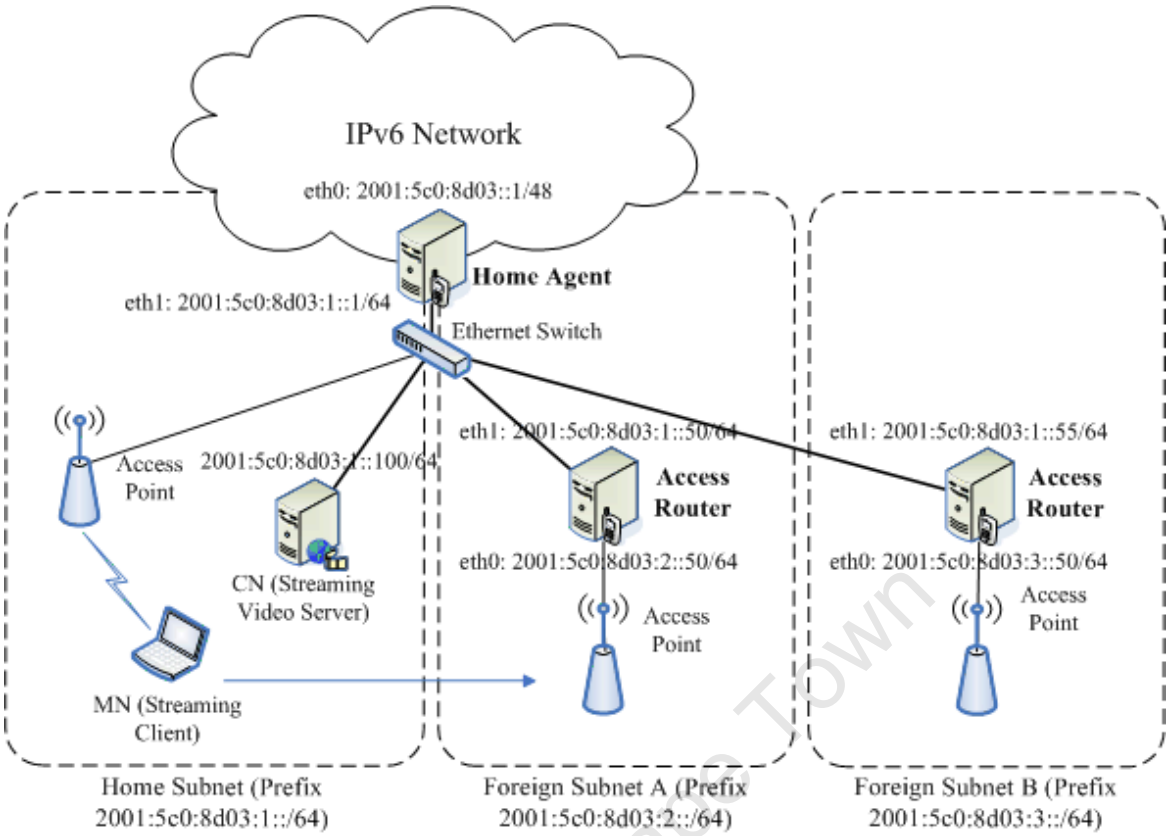

Figure 4.2: Evaluation framework IPv6 address assignment.

The *Quagga* routing software is then installed and configured on the HA and the two ARs. Thereafter, the *radvd* daemon is installed and configured on the machines. With the routing configured, IPv6 connectivity is tested using the Linux *ping6* and *traceroute6* tools. It is important to ensure that every node is reachable from every other node. Details on how the system was configured as a fully functional IPv6 network are provided in Appendix B. For a more detailed manual on how to set up an IPv6 network, refer to Peter Beiringer's *Linux IPv6 HOWTO* [46].

4.6.2 MIPv6 and FMIPv6

MIPL implements MIPv6 on a Linux platform. MIPL consists of a kernel part and a user space part. The kernel part is a patch for the Linux kernel, and is designed for a particular kernel. The source code for this kernel must be patched with MIPL, then configured and recompiled. Once this is complete, the user space part of MIPL can be installed and run as normal user software. Detailed instructions on how to install MIPL and configure it to make the network MIPv6 compliant are given in Appendix

B. A useful resource to help configure the test-bed is Lars Strand's *Linux Mobile IPv6 HOWTO* [47].

The fmipv6.org software implements FMIPv6. To install the fmipv6.org source, a kernel patch that allows the fmipv6 daemon to set IPv6 addresses without performing Duplicate Address Detection is required. Once the Linux kernel has been patched, another patch is needed for MIPL, which adds an IP security policy that allows a Fast Neighbour Advertisement to leave the MN before a Binding Acknowledgement is received. The user space part of FMIPv6 can then be installed. Modifications to the fmipv6.org source had to be made to adapt it to work correctly on the test-bed. Detailed information on the FMIPv6 installation and configuration is provided in Appendix B.

4.6.3 The Streaming Video Server

The VoD streaming server is placed in the home subnet to simplify the network architecture. It is set up as a HTTP web server with a selection of available video files. The MN is able to view the available files over HTTP and select one to stream. Several copies of each video file were encoded, each with a different bit-rate. Table 4.3 summarises the quality and corresponding bit-rate of each copy.

Based on the AMVS scheme, the MN can request the server to stream a copy of a video at a specific encoded bit-rate.

Open source video streaming software is required for the streaming server and client. Therefore, the *VideoLAN* [48] streaming software was chosen to stream video on the test-bed. *VideoLAN* is designed to stream MPEG videos on high bandwidth networks. The *VideoLAN Client* (VLC) is designed to support both server and client functionality. It can be used as a server to stream MPEG-1, MPEG-2 and MPEG-4 files, DVDs and live videos over a network in unicast or multicast, and as a client to receive, decode and display MPEG streams under multiple operating systems. For a detailed manual on how to stream using *VideoLAN*, refer to the *VideoLAN Streaming HOWTO* [49].

Table 4.3: Quality and corresponding encoded bit-rate of each version of a video file on the streaming video server.

	Quality	Frame Rate (fps)	Video Resolution	Bit-rate (kbps)
1	Low bandwidth video	10	160 × 120	83
2	Low bandwidth video	15	160 × 120	96
3	Low bandwidth video	29.97	160 × 120	166
4	VHS quality video	15	320 × 240	300
5	VHS quality video	29.97	320 × 240	400
6	DVD quality video	15	640 × 480	1073
7	DVD quality video	29.97	640 × 480	2073
8	High definition quality video	29.97	1280 × 720	5073

In the next chapter the experiments performed on the test-bed and the results obtained are discussed.

Chapter 5

Evaluation Results and Analysis

5.1 Introduction

Chapter 4 presents the architectural design of the evaluation framework. A number of tests are performed using the framework in order to determine feasibility of the AMVS scheme. The results from the tests are analysed and discussed in this chapter.

Initially, tests are carried out to determine the performance of MIPv6 handovers. Thereafter, the performance of FMIPv6 handovers is determined. These results are compared and discussed. Then the feasibility of using the AMVS scheme to alter the quality requirements and guarantee the continuity of a VoD application is assessed.

This chapter begins by describing the configuration of each of the network nodes and the effect of the configurations on the results obtained. The tools used to analyse the handovers are then briefly introduced. The experimental procedure and its impact on the results are also discussed. Thereafter, the handover performance evaluations and results are presented.

5.2 Test-bed Configuration

5.2.1 MIPv6 Configuration

This section discusses the MIPv6 configuration of the network nodes in the test-bed. There are several configuration parameters that can be altered for different types of operation. The most significant of these are briefly discussed below. Example

configuration files for the nodes are given in Appendix B and on the accompanying CD-ROM in Appendix C.

The debug level for all the MIPv6 nodes is set to the maximum value. This ensures that all the debug information about MIPv6 handovers is displayed to analyse and debug the software.

Route optimisation is disabled. Since the CN is in the home subnet and the HA acts as a router, any route optimised path between CN and MN will be the same as the normal path. Disabling this option reduces the overhead associated with route optimisation.

IP security (IPsec) is also disabled. IPsec enables encryption of all the MIPv6 signalling messages. However, in this research the security issues surrounding MIPv6 are not investigated. The overhead associated with IPsec is eliminated by disabling this option.

On the MN, the number of router probes is set to zero. This is the number of times the MN should send Neighbour Unreachability Probes to its old router after receiving a router advertisement from a new one. Since the option is set to zero, the MN will move to the new router immediately. This reduces the movement detection latency.

These configurations simplify the test-bed operation and remove any unnecessary overhead from the handover procedures.

5.2.2 Router Advertisement Configuration

RAs must be sent out periodically by the ARs so that a MN can create a new CoA using stateless address autoconfiguration when it moves to a new subnet. The frequency at which RAs are sent has a major impact on the movement detection latency. The higher the frequency of advertisements, the more likely a MN is to receive an unsolicited RA, reducing the movement detection latency. However, a higher frequency of RAs results in increased signalling overhead.

The minimum RA interval time was set to 3 s and the maximum RA interval time was set to 4 s. These are the minimum values that can be used according to RFC2461 [41]. The overhead associated with more frequent advertisements is negligible compared to the improvement in movement detection latency.

5.2.3 MIPv6 Handover Emulation

As stated in the previous chapter, the physical area that the evaluation framework occupies is limited. Furthermore, the MN is implemented on a stationary Desktop Computer. Thus, handovers between the wireless APs must be forced.

For the MIPv6 handovers, this was done using the Linux wireless tool "iwconfig." This tool can be used to force a wireless interface to disassociate with its current AP and associate with a specified new AP. The 802.11 link layer scanning for candidate APs happens in advance. Thus, the link layer scanning does not occur as part of the handover process, so the latency associated with the 802.11 scanning procedure is removed from the handover.

5.2.4 FMIPv6 Configuration

This section discusses the FMIPv6 configuration of the network nodes in the test-bed. Example configuration files for the nodes are given in Appendix B and on the accompanying CD-ROM in Appendix C.

The ARs are configured to ask for support for buffering packets. This means that during a handover the PAR can forward packets to the NAR and request that it buffers them until the handover is complete. The NAR can then forward the packets to the MN.

The ARs are also configured to enable network initiated handovers. FMIPv6 requires that a PAR sets up services for the MN on the NAR before handover. This means that handovers must be anticipated. If handovers are forced, as is done in the case of MIPv6, the FMIPv6 daemons will be unable to anticipate the handover and will be

ineffectual. By enabling network initiated handovers, the FMIPv6 daemons are made aware of the handover and are able to set up services for the MN on the NAR.

A protocol that enables candidate AR discovery is not implemented on the evaluation framework. Therefore, a list of other candidate ARs is manually configured for each AR.

The MN can monitor the link quality of the wireless interface it is using and triggers a handover if the quality falls below a specified threshold. This is an ideal way to anticipate movement and reflects real-world movement detection more accurately. However, this feature could not be implemented to operate correctly on the evaluation framework. Thus, only network initiated handovers are used to trigger handover.

5.2.5 FMIPv6 Handover Emulation

As discussed in the previous section, handovers must be anticipated so that FMIPv6 can set up services for the MN on the NAR prior to handover. If the "iwconfig" tool is used to force the MN to associate with a new AP, the network would be unable to predict the handover. Thus, network initiated handovers are employed when using FMIPv6.

The FMIPv6 implementation includes link layer scanning for candidate APs. Thus, real-world link layer handover is accurately modelled. This creates a slight disparity in the link layer handover latency experienced during MIPv6 and FMIPv6 handovers on the evaluation framework. However, the difference in link layer handover latency is in the order of milliseconds while the total handover latency is in the order of seconds. Therefore, the difference may be considered insignificant during the analysis of the protocols.

5.3 Network Analysis Tools

This section briefly discusses the tools used to generate traffic and analyse the handovers on the evaluation framework.

MIPv6 Tester [50] is a tool that was developed to simplify the testing of IPv6 mobility features. It opens a bidirectional TCP stream and two unidirectional UDP streams between two nodes on a network. These are designed as a "server" and a "client," even though both nodes act as client and server. The tool tests the connectivity between the nodes and reports the down time of each connection. This is the handover latency and is a measure of the disruption that the streams experience.

One node is configured as a server and the other as a client. When the server is started, it will wait for the client on its TCP socket and start to send UDP packets. The client can then be started. It should connect to the server node. The status of the application will indicate that both nodes are receiving and sending packets. When the application detects a handover (e.g., when a MN moves to a new network), it calculates the duration of the handover and displays it when the connection is restored.

The *Ethereal* network protocol analyser [51] is used to analyse handovers. It is open source software that runs on most popular computing platforms, including Linux. It allows data to be captured from a live network connection, or read from a capture file. Live data can be read from Ethernet, IEEE 802.11 and loopback interfaces (among others). It also supports MIPv6 and ICMPv6, allowing these protocols to be scrutinised. These are the signalling protocols used in the evaluation framework. The data display can be refined using a display filter. Display filters can also be used to selectively highlight packet summary information. These are useful tools to analyse the impact of handovers on a MN's packet streams.

5.4 Experimental Procedure

The MN is configured to move in a regular pattern between each of the subnets for the experiments. The MN first moves from the home subnet to the foreign subnet A, then to the foreign subnet B and then back to the home subnet. The movement pattern is illustrated in Figure 5.1. This pattern was repeated ten times for each set of results, and four sets of results were obtained in each case. Thus, a total of 120 handovers

were monitored to analyse each type of handover. This forms the results discussed later in the chapter.

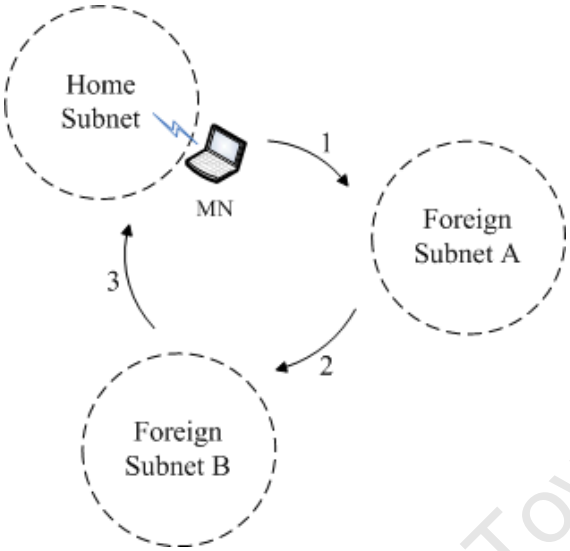

Figure 5.1: MN movement pattern.

The handovers were performed in quick succession of each other. On average, there was a 5 s interval between each handover. The high frequency of handovers is useful for analysis. The impact of a high frequency of handovers within a limited measurement period is that the signalling load is higher during this period.

A bidirectional TCP stream and two unidirectional UDP streams are started between the MN and the CN prior to movement. The packet streams are then monitored as the MN moves between the different subnets, using the *Ethereal* network analyser. The impact of handovers on the TCP and UDP streams is investigated.

5.5 Handover Performance Evaluations

The most important performance metrics used to compare MIPv6 and FMIPv6 are the handover latency, packet loss and jitter experienced in the MN's packet streams. The handover latency is the duration of a handover. This is the period of disruption in a MN's packet stream during a handover. The packet loss is the sum of all lost or dropped packets during the handover period. The jitter is the variance in arrival time of the received packet stream. A less significant metric is the signalling load. The

signalling load provides an indication of the overhead associated with each mobility management protocol.

Each of the performance metrics is discussed individually and the impact of a handover with respect to that metric is analysed. Based on the review of related literature, MIPv6 handover performance is expected to be inadequate to support video streaming applications [11]. FMIPv6 is expected to improve handover performance to a level where the AMVS scheme can be effectively used to guarantee application continuity. In the following sections, MIPv6 handover performance is assessed first and then FMIPv6 handover performance is compared to it. Thereafter, the feasibility of the AMVS scheme is discussed.

5.5.1 Handover Latency

The handover latency is the period of disruption in a MN's packet stream during a handover. The longer a handover, the greater the disruption that an application running on the MN experiences. The handover latency is measured using the *MIPv6 Tester* application.

The AMVS scheme requires that the handover latency be determined, so that it can compensate for this disruption by altering the quality requirements of the application. However, if the handover latency is too long, it is unrealistic for the AMVS scheme to adapt the quality of the application to a level that will guarantee its continuity.

The handover latencies discussed in this section are those experienced by the MN's received packet streams. This allows a direct comparison between the handover latencies to be made.

Figure 5.2 illustrates the average MIPv6 handover latency experienced by the TCP and UDP streams respectively. The average handover latency experienced by the TCP stream is 4.84 s and the average handover latency experienced by the UDP stream is 3.36 s. During handover, there is a period when the MN is unable to receive new TCP segments containing data or send acknowledgements. TCP assumes that this is due to congestion and responds by lowering the transmission rate. Thus, even when the

handover is complete and the connection is restored, it takes the MN some time before it receives the next TCP segment or sends an acknowledgement. However, the UDP stream maintains the same transmission rate in spite of the disruption caused by handover. So when the connection is restored, the MN will receive the next UDP packet relatively quicker. Therefore, the UDP stream experiences less latency during handover.

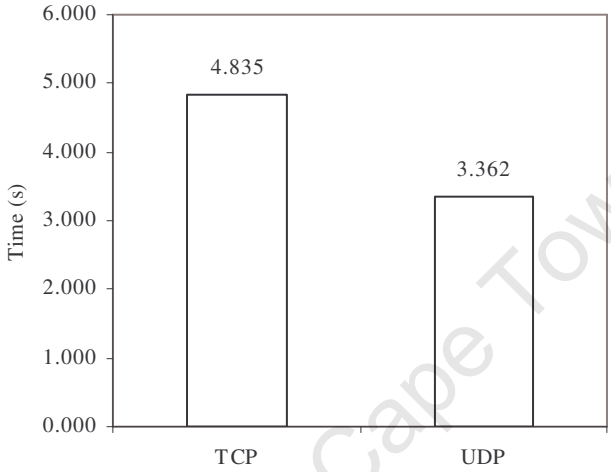

Figure 5.2: Average MIPv6 handover latencies experienced by the TCP and UDP streams.

Costa et al [4] performed a simulation study on the performance of MIPv6 in a WLAN. They determined a maximum MIPv6 handover latency of 4.2 s with a link delay of 1000 ms for constant bit rate UDP traffic. The handover latency drops to less than 1 s for link delays of less than 100 ms. Montavont et al [11] also performed an evaluation of MIPv6 handovers over WLAN. They determined the minimum value of the MIPv6 handover latency in the worst case to be 1.88 s. The maximum value in the same case was 3.08 s. The best case MIPv6 handover latency was as low as 151 ms. For the evaluations performed in this work, the minimum handover latency experienced by the TCP stream is 0.41 s and the maximum is 12.66 s, while the minimum handover latency experienced by the UDP stream is 0.12 s and the maximum is 11.81 s. These handover latencies are on average longer than those determined by similar simulation studies, such as those performed by Costa et al and Montavont et al. The reasons for the inconsistency are that the evaluations are performed on different frameworks with different assumptions. Simulation studies do

not always take real-world factors into account, in particular the factors that contribute to the movement detection latency and delays associated with Duplicate Address Detection. Different implementations of protocols are also used. The performance of an implementation depends on the design of that implementation and is influenced by the expertise of the implementer.

The MIPv6 handover latencies measured are unacceptable for video streaming applications. The average handover latency is too long for an application to recover from the disruption. Even video streaming applications that make use of an initial buffer of the order of seconds before presentation are unable to tolerate such long delays. In addition, video streaming applications require a consistently high data transmission rate at the application layer to deliver the expected quality, but such long latencies cause this to drop severely.

Figure 5.3 illustrates the average handover latency experienced by the TCP stream during MIPv6 and FMIPv6 handovers respectively. The use of FMIPv6 results in a clear improvement to handover latency.

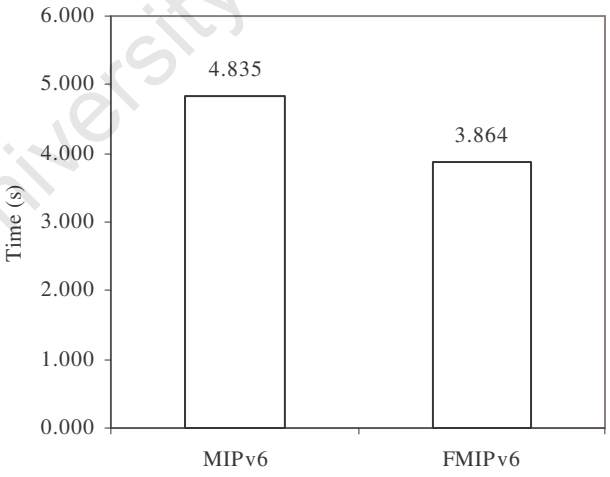

Figure 5.3: Average MIPv6 and FMIPv6 handover latencies experienced by the TCP stream.

The minimum MIPv6 handover latency experienced is 0.41 s and the maximum is 12.66 s, while in the case of FMIPv6 the minimum handover latency experienced is 1.65 s and the maximum is 6.17 s. MIPv6 has a wider distribution of handover

latencies. This is reflected in the standard deviation of the MIPv6 and FMIPv6 handover latencies, as illustrated in Figure 5.4. The larger standard deviation in the MIPv6 handover latency can be attributed to the movement detection latency. In the best case a MN will receive an unsolicited router advertisement almost immediately resulting in low handover latency. However, in the worst case a MN will have to send a router solicitation and wait the maximum period before it receives the router advertisement resulting in high handover latency. With FMIPv6, handover is anticipated, so the movement detection latency depends primarily on the duration of the link layer scanning procedure which is much shorter.

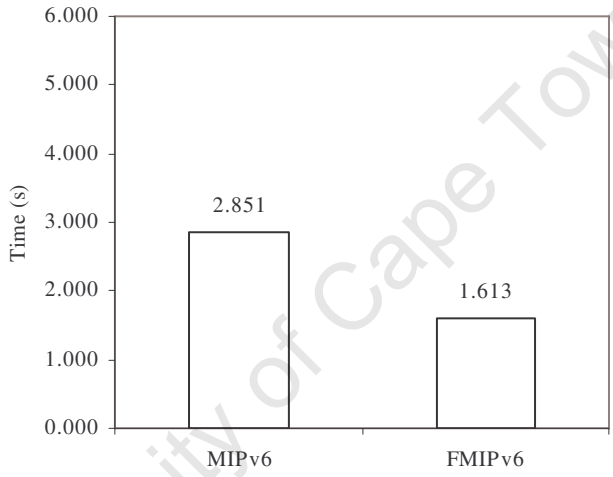

Figure 5.4: Standard deviation of MIPv6 and FMIPv6 handover latencies experienced by the TCP stream.

Figure 5.5 shows the average handover latency experienced by the UDP stream during MIPv6 and FMIPv6 handovers respectively. The use of FMIPv6 results in a major improvement to handover latency.

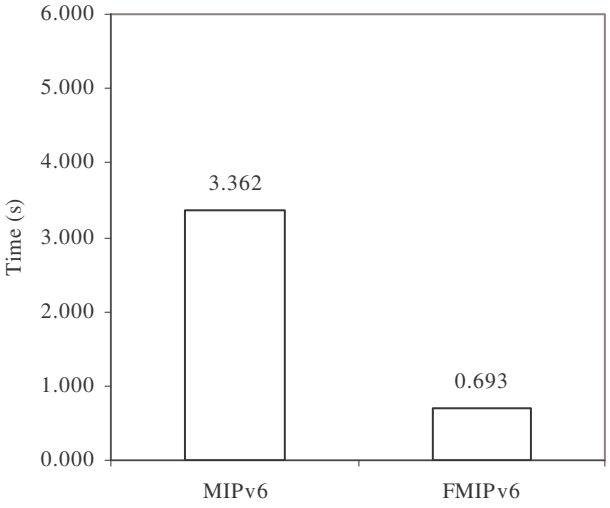

Figure 5.5: Average MIPv6 and FMIPv6 handover latencies experienced by the UDP stream.

The minimum MIPv6 handover latency experienced is 0.12 s and the maximum is 11.81 s, while in the case of FMIPv6 the minimum handover latency experienced is 0.30 s and the maximum is 0.91 s. Again, the MIPv6 handover latencies have a much wider distribution than FMIPv6. This is illustrated in Figure 5.6.

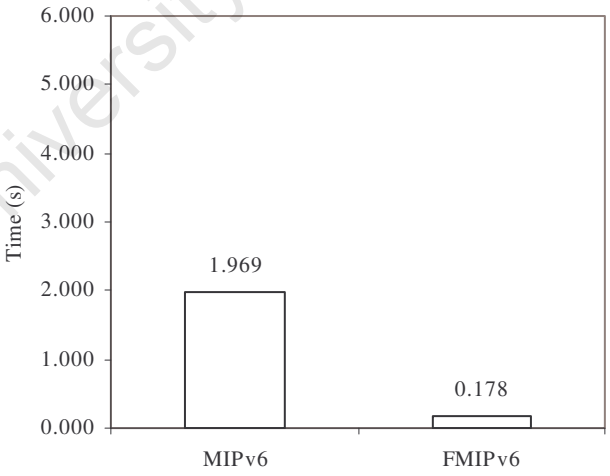

Figure 5.6: Standard deviation of MIPv6 and FMIPv6 handover latencies experienced by the UDP stream.

FMIPv6 results in considerably lower handover latencies than MIPv6. This can be attributed primarily to the better movement detection mechanisms in FMIPv6. The

delay associated with Duplicate Address Detection is also eliminated in FMIPv6. The maximum handover latency to be expected is significantly lower in the case of FMIPv6. The best case handover latencies were experienced by the UDP stream when FMIPv6 was used. The maximum handover latency to expect in this case is less than 1 s. Therefore, as expected, streaming video over UDP results in better performance than streaming over TCP.

Recall equation 3.2:

$$R = \frac{Nt}{t + L}$$

The main purpose of the evaluation is to determine L and assess whether the AMVS scheme is feasible based on the value of L. As discussed in Chapter 3, an acceptable upper bound for L is 1 s. The value of L for the UDP stream when FMIPv6 is used is:

$$L = 0.693 + 0.178 = 0.871 \text{ s}$$

L is less than 1 s in this case and falls within the range of acceptable values for the AMVS scheme. When TCP is used to stream the video under MIPv6, L is greater than 5 s. This is unacceptably high. Thus, FMIPv6 improves the handover performance to a level that makes the AMVS scheme feasible.

5.5.2 Packet Loss

The packet loss is the sum of all lost or dropped packets during the handover period. The greater the packet loss is, the more drastic the degradation in perceptual quality of an application running on the MN. The packet loss was measured as the difference of the total number of packets sent by the CN and the total number of packets received by the MN. The difference was attributed to handover.

Figure 5.7 illustrates the disruption to the MN's packet stream due to MIPv6 handovers. The handover indicated in the figure occurs at time 56 s (with three more handovers occurring at time 67 s, 83 s and 97 s). It is evident that the disruption caused by handover is detrimental to the throughput of applications. The difference in the packet loss for each handover can be attributed to the variation in handover latency for each handover.

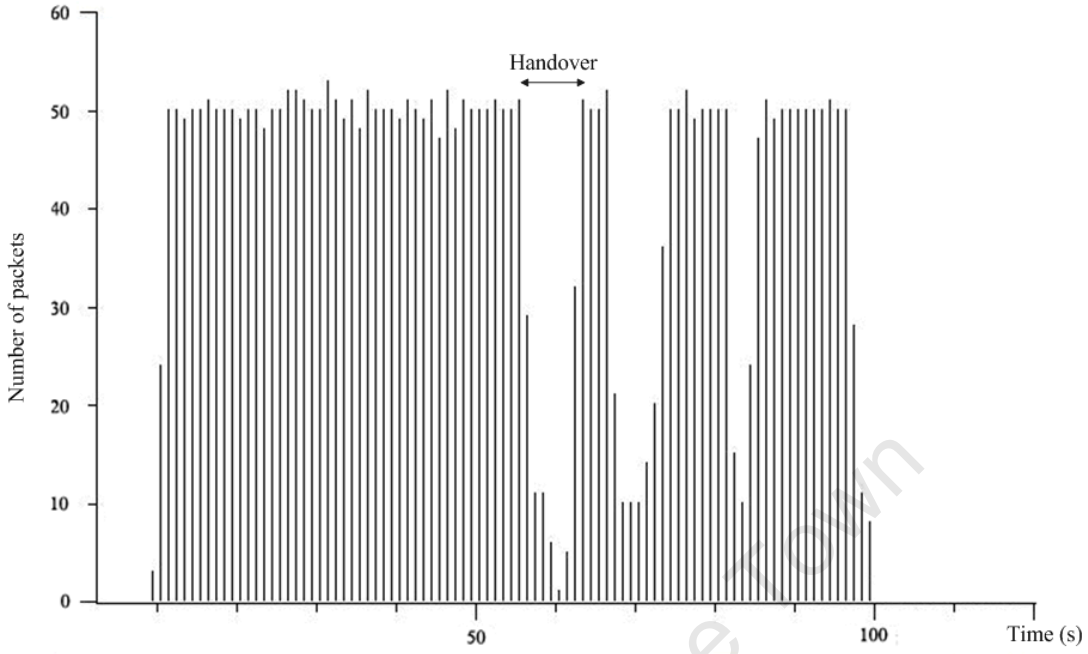

Figure 5.7: MIPv6 handover disruption.

Figure 5.8 illustrates the disruption to the MN's packet stream due to an FMIPv6 handover. The handover indicated in the figure occurs at time 17 s. Figure 5.9 illustrates the same handover at 10 times the resolution to view the impact of handover more clearly. The FMIPv6 handover is much faster than the MIPv6 handover. Thus, the MN's packet stream experiences less of a disruption. This is evident from a simple comparison of Figure 5.7 and Figure 5.8. The improvement results in far less packet loss and a smaller reduction in an application's throughput.

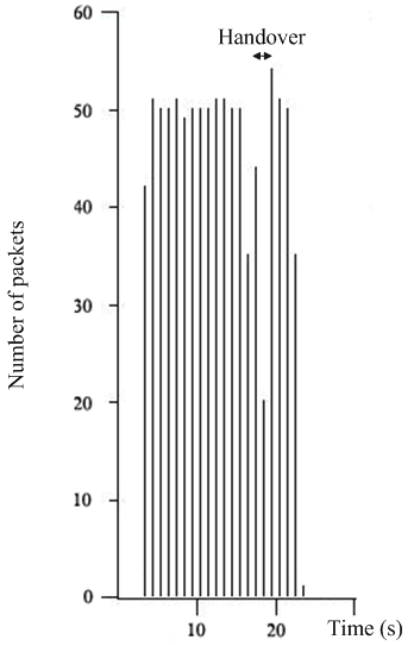

Figure 5.8: FMIPv6 handover disruption.

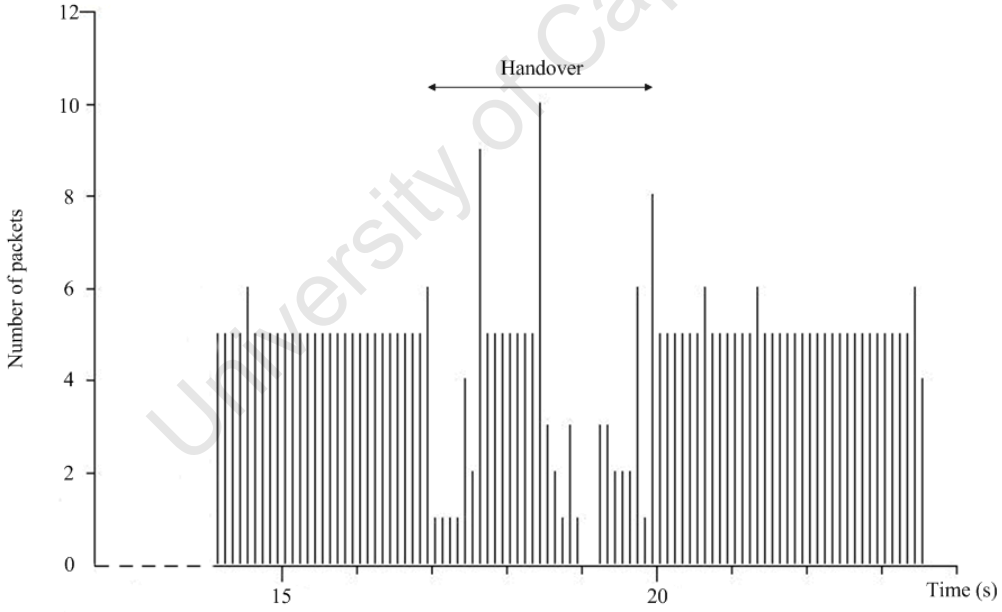

Figure 5.9: FMIPv6 handover disruption (higher resolution).

Figure 5.10 illustrates the average packet loss experienced by the TCP stream during MIPv6 and FMIPv6 handovers. FMIPv6 results in a 30.7 % improvement to packet loss. TCP assumes that packet losses are a result of network congestion and not due to transmission errors. During handover, there is a period during which the MN is unable

to receive new TCP segments containing data or send acknowledgements. TCP assumes that this is due to congestion and responds by lowering the transmission rate. Therefore, although the packet losses are comparatively low, the application throughput is significantly reduced. This is unacceptable for real-time multimedia applications that require consistent high throughput. Once the handover is complete, it will take a considerable amount of time for TCP to restore its original sending rate. For this reason, TCP is not used in delay-sensitive applications such as video.

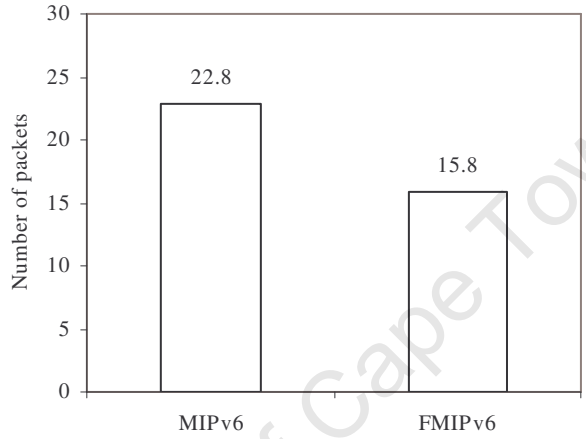

Figure 5.10: Average packet loss of the TCP stream.

Figure 5.11 illustrates the average packet loss experienced by the UDP stream during MIPv6 and FMIPv6 handovers. The packet loss experienced by the UDP stream is significantly more than that experienced by the TCP stream. This is because UDP continues to send packets at the same rate in spite of packet losses. Packets that arrive late are simply dropped. Thus, applications running on the MN will experience a drastic deterioration in quality. FMIPv6 results in an 81.9 % improvement to the packet loss experienced by the UDP stream.

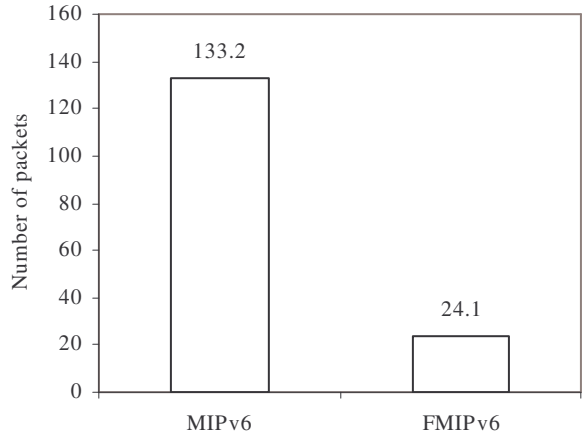

Figure 5.11: Average packet loss of the UDP stream.

FMIPv6 is able to improve the packet loss to a level that can be more easily tolerated by video streaming applications. The improvement in packet loss experienced by the UDP stream is quite drastic. This reinforces the assertion that it is most feasible to stream video over UDP using FMIPv6 to manage mobility.

5.5.3 Jitter

The jitter is the variance in arrival time of the received packet stream. Jitter can degrade the perceptual quality of video nearly as much as packet loss. Even a low amount of jitter severely degrades video quality [52]. However, jitter can be offset by a buffer in the receiver. It will only become detrimental if the buffer is insufficient to compensate for the variance in arrival time of the packets. Jitter causes visible distortions in the streamed video, which users cannot tolerate. The jitter was measured as the standard deviation of the difference in arrival time of the received packet streams.

Figure 5.12 shows the average jitter experienced by the TCP stream during MIPv6 and FMIPv6 handovers and Figure 5.13 shows the average jitter experienced by the UDP stream during MIPv6 and FMIPv6 handovers. FMIPv6 results in quicker handovers, so the variance in arrival time of the received packet stream should be less than with MIPv6, as can be seen in both figures.

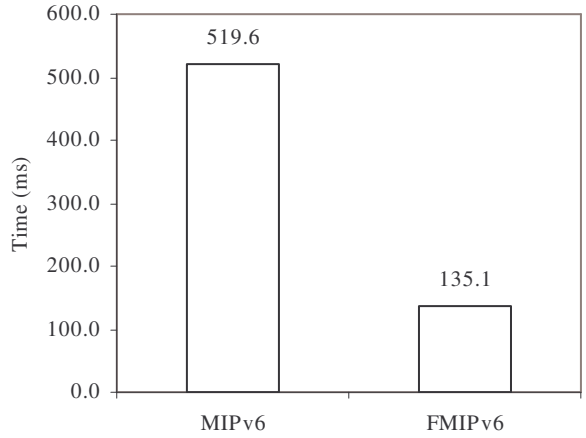

Figure 5.12: Jitter experienced by the TCP stream due to handover.

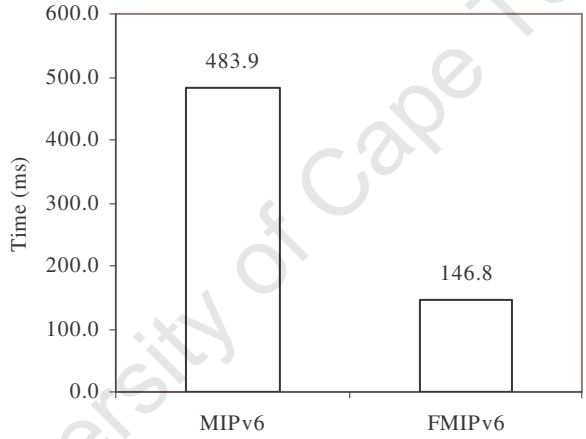

Figure 5.13: Jitter experienced by the UDP stream due to handover.

FMIPv6 provides noticeably better performance than MIPv6. The improvement to jitter experienced during FMIPv6 handovers provides vital support for video streaming applications. The jitter experienced during FMIPv6 handovers is within acceptable levels to allow the AMVS scheme to function effectively.

5.5.4 Signalling Load

The signalling load is defined as the fraction of signalling packets out of the total number of packets transmitted. This provides an indication of the overhead associated with each mobility management protocol.

The frequency of router advertisements (RAs) has a big impact on the signalling load. The more frequent RAs are, the greater the signalling load will be. As previously discussed, RAs are sent at a high frequency during the evaluations (every 3 to 4 s). This improves the movement detection latency. The associated overhead is considered negligible.

Another factor that influences the signalling load is the frequency of handovers within the measurement period. A high frequency of handovers results in a higher signalling load (e.g., the signalling load will be higher if 10 handovers occur within a measurement period, than if only 2 handovers occur within the same period).

The same RA frequency is used for both MIPv6 and FMIPv6. The MN also uses the same movement pattern for the MIPv6 and FMIPv6 evaluations, so the measurement period is roughly the same. Thus, the signalling load will allow a direct comparison of the overhead associated with each mobility management protocol to be made.

The signalling load is measured during the evaluations as the fraction of signalling packets out of the total number of packets transmitted. The average signalling load during a MIPv6 handover is 3.14 %, while the average signalling load during an FMIPv6 handover is 3.42 %. This is consistent with the expected results, since FMIPv6 makes use of 8 signalling messages during a handover and MIPv6 makes use of just 2 to register with its HA (and an additional 2 for every CN it must register with). The overhead associated with the AMVS scheme would be slightly higher than that of FMIPv6, because in addition to FMIPv6 signalling, signalling is required for the MN to communicate its requests to alter video quality to the server. However, the improved handover performance compensates for the increased signalling.

5.6 Chapter Discussion

The chapter began by describing how the test-bed configuration impacted the evaluations. The evaluation tools and experimental procedure were then briefly discussed. Thereafter, the evaluation results and analysis were presented.

The handover latency provides an indication of the disruption that a MN's packet stream faces during a handover. Longer handovers result in a greater reduction in an application's throughput and greater packet loss. The results showed that the MIPv6 handover latency was too long for video streaming applications to tolerate. FMIPv6 provides a significant improvement to the handover latency. It is more realistic for video streaming applications to tolerate this latency, although on its own, the improvement is insufficient to guarantee application continuity. The packet loss gives an indication of the degradation in perceptual quality that applications face during handover. MIPv6 handovers result in a significant packet loss. FMIPv6 results in a substantial improvement to packet loss. FMIPv6 provides a major improvement to the jitter experienced by a MN's packet stream. The signalling load of FMIPv6 is higher than that of MIPv6. However, the increased signalling is offset by the performance gains achieved.

The handover latency and packet loss caused by MIPv6 handovers causes too much disruption for the AMVS scheme to mask. The handover performance must be improved for the AMVS scheme to be effective. FMIPv6 is able to provide this enhancement to handover performance. The best handover results were obtained using FMIPv6 on the MN's UDP stream. This verifies that UDP streaming of video content provides better performance than TCP streaming. Additionally, FMIPv6 improves handover performance to a satisfactory level to conclude that it is feasible to use the AMVS scheme to guarantee application continuity.

Chapter 6

Conclusions and Recommendations

6.1 Summary

This research investigated the disruption to IP streams caused by Mobile IPv6 handover. The traditional approach of Mobile IPv6 provides a solution to global mobility. However, it does not take into account application QoS requirements. Therefore, there is no guarantee that the network will provide the resources required to deliver the application traffic. Furthermore, the connection disruption and packet loss that an application incurs during handover severely degrades its quality

The aim of this research was to formulate and analyse a seamless video streaming handover scheme to guarantee the continuity of a VoD application in the context of IP mobility.

Several enhancements to MIPv6 have been proposed in the literature to improve the handover procedures and reduce the impact of MIPv6 handover on application quality. However, while the impact of handover can be reduced, MIPv6 cannot guarantee seamless handovers. Therefore, a more robust mobility management solution is necessary. The design for the seamless video streaming handover scheme proposed in this research is based on a cross-layer approach. A combination of existing techniques to enhance handovers and to adapt video quality to changing network conditions results in improved handover performance and enables application continuity to be guaranteed.

A suitable evaluation framework was developed to assess the feasibility of the AMVS scheme proposed in this research. The test-bed was able to successfully perform the evaluations that were required. MIPv6 handover performance was compared to FMIPv6 handover performance and an assessment was made on whether the AMVS scheme was feasible in a practical scenario. The results that were obtained confirmed that FMIPv6 improves handover performance to a level that makes it feasible to use the AMVS scheme to guarantee application continuity.

6.2 Conclusions

Based on the findings in the preceding chapters, the following conclusions are drawn:

- The main contribution of this work is the performance assessment of MIPv6 and FMIPv6. Based on the results obtained, it is feasible to use a combination of adaptation of video quality based on network conditions and an existing mobility management protocol to guarantee application continuity in the context of IP mobility. The AMVS scheme that was proposed to guarantee application continuity relies on FMIPv6 to improve handover performance and on adaptive video streaming to mitigate the effects of handover on the application.

- MIPv6 provides a solution to global IP mobility. However, the connection disruption at the network layer associated with handover is often too long to support real-time multimedia applications. MIPv6 handovers are insufficient to guarantee seamless handovers and, thus, application continuity. This backs up original claims from the literature.

- Enhancements to MIPv6 can improve handover performance. FMIPv6 was implemented and investigated in this work. FMIPv6 is able to improve handover performance considerably and should be used to manage mobility wherever possible. However, on its own, it is inadequate to guarantee seamless handovers. Therefore, a more robust mobility management solution is necessary.

- The results obtained from the evaluation framework differ from those obtained from similar simulation studies. The main reasons for this are that the tests were performed on different frameworks with different assumptions and different implementations of the same protocols were used. However, physical implementations take into account real-world factors and model networks more accurately. This highlights the value of evaluations performed on physical platforms.

- TCP assumes that packet losses are a result of network congestion and not due to transmission errors. Therefore, during handover, when a MN is unable to receive new TCP segments containing data or send acknowledgements, TCP assumes that this is due to congestion and responds by lowering the transmission rate. Once the handover is complete, it will take a considerable amount of time for TCP to restore its original sending rate. Thus, TCP is an unsuitable transport protocol for delay-sensitive applications that require consistent high throughput. The evaluations verified that UDP provides far better performance for real-time multimedia applications. Therefore, UDP is the recommended transport protocol for the solution proposed in this research.

6.3 Recommendations and Future Work

This research delved into the issues surrounding mobility management in wireless networks and providing adequate support to real-time multimedia applications in the context of IP mobility. During the course of the work a number of avenues for further research came to light. Listed below are some of the most important recommendations that emerged.

- The evaluation framework that was developed in this work served to assess the feasibility of the AMVS scheme. A more complete analysis of the scheme would require a full implementation of a protocol for the MN to communicate its requests to alter the stream to the server, and a full implementation of a

server that supports dynamic adaptation of video quality. The framework developed in this work must be extended to implement this.

- The AMVS scheme is dependant on being able to anticipate handover. Thus, for best performance, it should be used in conjunction with a mobility prediction algorithm. The handover performance achieved should be compared to other prediction assisted handover schemes.

- This research performed a basic evaluation of MIPv6 and FMIPv6 handover and their effects on streaming video quality. Additional thorough research must be performed in order to extend these results.

- The evaluation framework was designed to allow the existing infrastructure to be extended. A larger and more realistic framework would be useful to test a broader range of scenarios and applications.

- The effect of altering the perceptual quality of video streams on the user experience was not investigated in this work. However, it is an important consideration. A more complete analysis of the effectiveness of the AMVS scheme should include techniques to assess the user-perception of altering video quality. Although such a method would be subjective, it would provide a crude indication of the usefulness of the scheme.

- The streaming server in the AMVS scheme sends a unicast stream to each client. This is a concern as it could become a considerable overhead for a server supporting many clients. Further research must be carried out to determine whether this problem can be mitigated.

- Security issues are a major concern in the context of network mobility. However, these issues were not investigated in this research. Further work must be carried out to assess the security mechanisms of 802.11 WLANs and MIPv6, and the impact of the AMVS scheme on these mechanisms.

Bibliography

[1] J. A. Garcia-Macias, F. Rousseau, G. Berger-Sabbatel, L. Toumi, and A. Duda, "Quality of Service and Mobility for the Wireless Internet," *Wireless Networks 9*, pp. 341-352, 2003.

[2] A. Grilo, M. M. Macedo, and M. S. Nunes, "IP QoS support in IEEE 802.11b WLANs," *Computer Communications 26*, pp. 1918-1930, 2003.

[3] A. Sukhov, P. Calyam, W. Daly, and A. Iliin, "Network Requirements for High-speed Real-time Multimedia Data Streams," *IPv6 Global Summit*, 2004.

[4] X. P. Costa, and H. Hartenstein, "A simulation study on the performance of Mobile IPv6 in a WLAN-based cellular network," *Computer Networks 40*, pp. 191-204, 2002.

[5] T. Choi, L. Kim, J. Nah, and J. Song, "Combinatorial Mobile IP: A New Efficient Mobility Management Using Minimized Paging and Local Registration in Mobile IP Environments," *Wireless Networks 10*, pp. 311-321, 2004.

[6] S. Deering, and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," *Internet Engineering Task Force (IETF), RFC2460*, 1997.

[7] "Technical and Economic Assessment of Internet Protocol, Version 6 (IPv6)," *IPv6 TASK FORCE, U.S. DEPARTMENT OF COMMERCE, National Telecommunications and Information Administration, National Institute of Standards and Technology, [Online] Available: http://www.ntia.doc.gov/ntiahome/ntiageneral/ipv6/final/ipv6final.pdf*, January 2006.

[8] "Solution Brief: Juniper Networks Enables a Secure and Assured Transition to Internet Protocol Version 6 (IPv6) in the Federal Government," *Juniper Networks, [Online] Available:*

http://www.juniper.net/solutions/literature/solutionbriefs/351137.pdf, October 2005.

[9] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," *Internet Engineering Task Force (IETF), RFC3775*, 2004.

[10] V. Marques, X. P. Costa, R. L. Aguiar, M. Liebsch, and A. M. O. Duarte, "Evaluation of a Mobile IPv6-Based Architecture Supporting User Mobility QoS and AAAC in Heterogeneous Networks," *IEEE Journal on Selected Areas in Communications*, 2005.

[11] N. Montavont, and T. Noel, "Analysis and Evaluation of Mobile IPv6 Handovers over Wireless LAN," *Mobile Networks and Applications 8*, pp. 643-653, 2003.

[12] ANSI/IEEE, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band," *IEEE Standard 802.11g*, 2003.

[13] J. McNair, T. Tugcu, W. Wang, and J. Xie, "A survey of cross-layer performance enhancements for Mobile IP networks," *Computer Networks 49*, pp. 119-146, 2005.

[14] V. T. Raisinghani, and S. Iyer, "Cross-layer design optimizations in wireless protocol stacks," *Computer Communications 27*, pp. 720-724, 2004.

[15] D. Miras, A. Sadagic, B. Teitelbaum, J. Leigh, M. El Zarki, and H. Liu, "A Survey of Network QoS Needs of Advanced Internet Applications – Working Document," *Internet2 QoS Working Group*, 2002.

[16] O. Verscheure, P. Frossard, and M. Hamdi, "MPEG-2 Video Services over Packet Networks: Joint Effect of Encoding Rate and Data Loss on User-Oriented QoS," *Proceedings of NOSSDAV 98*, pp. 257-264, 1998.

[17] B. Badrinath, A. Fox, L. Kleinrock, G. Popek, P. Reiher, and M. Satyanarayanan, "A conceptual framework for network and client adaptation," *Mobile Networks and Applications 5*, pp. 221-231, 2000.

[18] N. Cranley, L. Murphy, and P. Perry, "User-Perceived Quality-Aware Adaptive Delivery of MPEG-4 Content," *Proceedings of NOSSDAV 03*, 2003.

[19] M. Li, F. Li, M. Claypool, and R. Kinicki, "Weather Forecasting – Predicting Performance for Streaming Video over Wireless LANs," *Proceedings of NOSSDAV 05*, 2005.

[20] H. Soliman, *Mobile IPv6: Mobility in a Wireless Internet*: Addison-Wesley, 2004.

[21] H. Chaskar, "Requirements of a Quality of Service (QoS) Solution for Mobile IP," *Internet Engineering Task Force (IETF), RFC3583*, 2003.

[22] M. Eder, H. Chaskar, and S. Nag, "Considerations from the Service Management Research Group (SMRG) on Quality of Service (QoS) in the IP Network," *Internet Engineering Task Force (IETF), RFC3387*, 2002.

[23] A. Hasson, and N. Ventura, "An Evaluation of Mobile IP Movement Detection," *Proceedings of SATNAC 2004*, 2004.

[24] N. A. Fikouras, and C. Gorg, "Performance comparison of hinted- and advertisement-based movement detection methods for mobile IP hand-offs," *Computer Networks 37*, pp. 55-62, 2001.

[25] H. Soliman, C. Castelluccia, K. El Malki, and L. Bellier, "Hierarchical Mobile IPv6 Mobility Management (HMIPv6)," *Internet Engineering Task Force (IETF), RFC4140*, 2005.

[26] R. Koodli, "Fast Handovers for Mobile IPv6," *Internet Engineering Task Force (IETF), RFC4068*, 2005.

[27] E. Ivov, and T. Noel, "An Experimental Performance Evaluation of the IETF FMIPv6 Protocol over IEEE 802.11 WLANs," *Proceedings of WCNC 06*, 2006.

[28] A. T. Campbell, and J. Gomez-Castellanos, "IP Micro-Mobility Protocols," *Mobile Computing and Communications Review*, vol. 4, 2000.

[29] C. Blondia, O. Casals, L. L. Cerda, N. van der Wijngaert, and G. Willems, "Performance Evaluation of Layer 3 Low Latency Handoff Mechanisms," *Mobile Networks and Applications 9*, pp. 633-645, 2004.

[30] R. Hsieh, A. Seneviratne, H. Soliman, and K. El-Malki, "Performance analysis on Hierarchical Mobile IPv6 with Fast-handoff over End-to-End TCP," *Proceedings of GLOBECOM 02*, 2002.

[31] H. Jung, H. Soliman, S. Koh, and N. Takamiya, "Fast Handover for Hierarchical MIPv6 (F-HMIPv6)," *Internet Engineering Task Force (IETF), Internet Draft*, 2005.

[32] M. Sulander, T. Hamalainen, A. Viinikainen, and J. Puttonen, "Flow-based Fast Handover Method for Mobile IPv6 Network," *Proceedings of the 59th Semiannual IEEE Vehicular Technology Conference*, pp. 2447-2451, 2004.

[33] J. Puttonen, H. Suutarinen, T. Ylonen, A. Viinikainen, M. Sulander, and T. Hamalainen, "Flow-based Fast Handover Performance Analysis in Mobile IPv6 for Linux Environment," *Proceedings of the 61st Semiannual IEEE Vehicular Technology Conference*, pp. 2575-2579, 2005.

[34] Y. Pan, M. Lee, J. B. Kim, and T. Suda, "An End-to-End Multi-Path Smooth Handoff Scheme for Stream Media," *Proceedings of WMASH 03*, 2003.

[35] S. Choi, J. del Prado, S. Shankar, and S. Mangold, "IEEE 802.11e Contention-Based Channel Access (EDCF) Performance Evaluation," *IEEE International Conference on Communications (ICC)*, pp. 1151-1156, 2003.

[36] ANSI/IEEE, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements," *IEEE Standard 802.11e*, 2005.

[37] A. Lindgren, A. Almquist, and O. Schelen, "Quality of Service Schemes for IEEE 802.11 Wireless LANs - An Evaluation," *Mobile Networks and Applications 8*, pp. 223-235, 2003.

[38] G. Lui, and G. Maguire, "A Class of Mobile Motion Prediction Algorithms for Wireless Mobile Computing and Communications," *ACM/Baltzer MONET*, pp. 113-121, 1996.

[39] "Quagga Software Routing Suite," *[Online] Available: http://www.quagga.net/*, Accessed June 2006.

[40] K. Ishiguro, "Quagga - A routing software package for TCP/IP networks," *Quagga 0.99.3*, 2006.

[41] T. Narten, E. Nordmark, and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)," *Internet Engineering Task Force (IETF), RFC2461*, 1998.

[42] "Linux IPv6 Router Advertisement Daemon (radvd)," *[Online] Available: http://v6web.litech.org/radvd/*, Accessed June 2006.

[43] M. Dunmore, R. Ruppelt, and T. Saarinen, "Final MIPv6 Support Guide," *6NET*, 2005.

[44] "Mobile IPv6 for Linux," *[Online] Available: http://www.mipl.mediapoli.com/*, Accessed June 2006.

[45] "fmipv6.org," *[Online] Available: http://www.fmipv6.org/index.php/Main/HomePage/*, Accessed June 2006.

[46] P. Beiringer, "Linux IPv6 HOWTO (en)," *[Online] Available: http://www.tldp.org/HOWTO/Linux+IPv6-HOWTO/*, Accessed June 2006.

[47] L. Strand, "Linux Mobile IPv6 HOWTO," *[Online] Available: http://gnist.org/~lars/doc/Mobile-IPv6-HOWTO/Mobile-IPv6-HOWTO.html*, Accessed June 2006.

[48] "VideoLAN - Free Software and Open Source video streaming solution for every OS!," *[Online] Available: http://www.videolan.org/*, Accessed June 2006.

[49] A. de Lattre, J. Bilien, A Daoud, C. Stenac, A. Cellerier, and J. Saman, "VideoLAN Streaming HOWTO," *VideoLAN*, 2005.

[50] "MIPv6 Bull Project, MIPv6 Tester," *[Online] Available: http://www.bullopensource.org/mipv6/tester.php*, Accessed June 2006.

[51] "Ethereal: A Network Protocol Analyzer," *[Online] Available: http://www.ethereal.com/*, Accessed June 2006.

[52] M. Claypool, and J. Tanner, "The Effects of Jitter on the Perceptual Quality of Video," *Proceedings of ACM Multimedia*, vol. 2, 1999.

[53] ANSI/IEEE, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher Speed Physical Layer Extension in the 2.4 GHz Band," *IEEE Standard 802.11b*, 1999.

[54] B. Furht, and M. Ilyas, *Wireless Internet Handbook*: CRC Press, 2003.

Appendix A

Background Theory

A.1 Mobile IPv6

A.1.1 IPv6

The Internet Protocol (IP) provides packet routing and delivery services for the Internet and private networks. It is a connectionless, best-effort packet switching protocol. The version of IP currently in widespread use on the Internet is IP version 4 (IPv4). When IPv4 was designed, a 32-bit address space was considered more than adequate for the lifetime of the Internet. While in theory this allows for approximately 4 billion nodes on the Internet, in practice only about 1 billion nodes will be supported because addresses are not allocated in a consecutive manner. However, with the rapid growth and popularity of the Internet, and the emergence of many multimedia communication applications, there is a patent need for an even larger address space to support this evolution.

The new protocol proposed to replace IPv4 is IP version 6 (IPv6) [6]. IPv6 has the advantage of being an upgrade of an existing technology, allowing its developers to avoid many of the known problems with IPv4. In addition, it is not overly restricted by backward compatibility. Although the main driver for IPv6 was the lack of address space, some new features have also been included. The major changes to IPv6 are the redesign of the packet header and an increase in address size to 128 bits. IPv6 also allows recently developed IP security (IPsec) protocols to be included in the IP layer in IPv6 implementations.

With an address size of 128 bits, IPv6 addresses can be cumbersome to write. The accepted format is 8 blocks of 16 bits each, represented in hexadecimal format and separated by a colon (e.g., `ffff: ffff: ffff: ffff: ffff: ffff: ffff: ffff`). A sequence of 16 bit blocks containing only zeros can be replaced with "::" to condense the address representation (e.g., `2001:5c0:8d03:0:0:0:0:1` → `2001:5c0:8d03::1`).

Nodes are assigned two addresses when networking with IPv6. The IPv6 *global* address is a unique address, analogous to the single IPv4 address used for IPv4 networking, and is used for normal networking. The *link local* address is only valid on a link of an interface and is usually used for network configuration. A packet with this address as the destination would not be able to pass through a router.

IPv6 enables automatic address configuration using *stateless address autoconfiguration*. One way of doing this is to use *prefix discovery*. With prefix discovery, a router sends out periodic advertisements containing its network prefix. If the *autonomous* flag in this advertisement is set, any node receiving this prefix can append a random identifier onto this prefix and hence acquire an IPv6 address. There are numerous ways to obtain this random identifier, but the most common way is to use the MAC address of the network device. The node must then perform Duplicate Address Detection (DAD) to ensure that the address is unique. The details of the IPv6 protocol are not discussed any further here; for further details refer to the IPv6 specification [6].

Designing a mobility management protocol to fit into the existing IPv4 Internet brought with it many difficulties. This experience led the developers of IPv6 to consider mobility from the outset of the design process. Mobile devices were considered to be extremely important as they are expected to form the largest proportion of Internet connecting nodes in future.

IP mobility refers to the change in a node's IP address due to a change in its point of attachment to the network. Such changes are predominantly due to the node's physical movement, but can also be caused by a change in the network topology. IPv6

subnets are each allocated a short prefix. Nodes in that subnet use that prefix to configure their addresses. Routers on the link announce to other routers that packets with a destination address starting with their prefix be forwarded to them. The problems caused by IP mobility are illustrated with the help of Figure A.1 [20].

Assume that Host_D is communicating with Host_A in Figure A.1. When Host_D moves from Subnet_3 to Subnet_2, several problems arise:

- Host_A is unaware of the movement, so it continues to send packets to Host_D's address on Subnet_3. However, on arrival the packets will not be forwarded to any host and will be dropped.
- Host_D's address is no longer valid because it is not derived from the prefix assigned to Subnet_2, so it must form a new address. In the meantime, it will not receive any packets sent to the old address.
- If Host_D derives a new address on Subnet_2, it will have to update the socket containing information about its connection with Host_A. However, this is not possible since both nodes need to maintain the same set of information about the connection. Thus, the connection will be terminated.

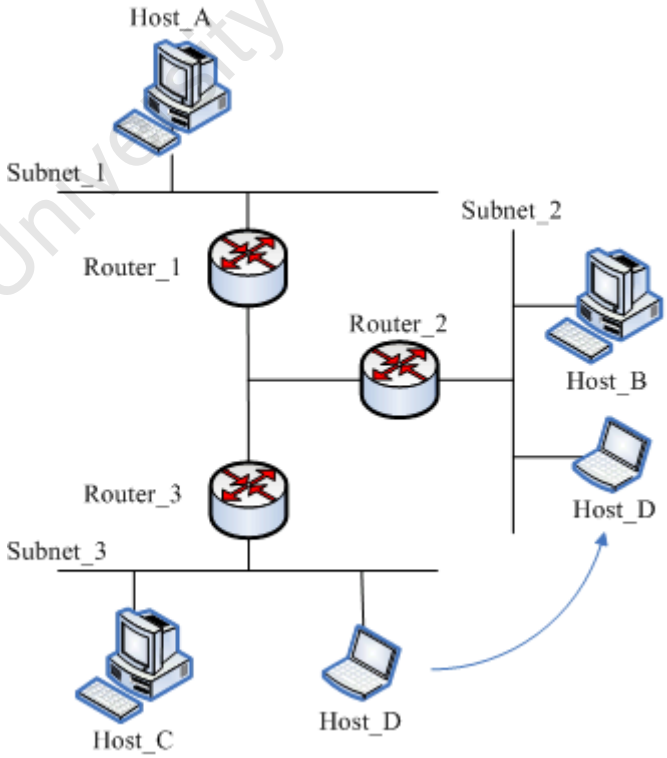

Figure A.1: An illustration of IP mobility.

These problems arise because the IP address of a node has changed while it has active connections with one or more other nodes. Therefore, the need for a mobility management protocol is apparent. Mobile IPv6 [9] solves the problem of IP mobility, allowing nodes to maintain active communication sessions and also ensuring that they are reachable by offering them a stable address.

A.1.2 Mobile IPv6

Some fundamental Mobile IPv6 (MIPv6) terminology is introduced here [9]:

- *Mobile Node (MN)* – a node that changes its location within a network topology. This change may be due to physical movement of the node or due to changes in network topology that cause the node to be attached to a different router.
- *Correspondent Node (CN)* – any node that communicates with the MN.
- *Home Address (HoA)* – a permanent, stable address that belongs to the MN and identifies it. IP packets addressed to the HoA are routed to the home subnet using standard routing protocols.
- *Home subnet* – the subnet to which the HoA prefix is assigned.
- *Home Agent (HA)* – a router located in the home subnet that acts on behalf of the MN while it is away from the home subnet. The HA redirects packets addressed to a MN's HoA to its current location.
- *Foreign subnet* – any subnet, other than the home subnet, visited by a MN.
- *Care-of Address (CoA)* – an address assigned to the MN when it is in a foreign subnet. This address defines the MN's current location when it is away from its home subnet. The MN retains a CoA only while it is in the particular foreign subnet indicated by the network prefix of the CoA.
- *Binding* – the association of the MN's HoA (stable address) and CoA (current location). This binding enables the HA and CNs to forward packets to the MN's current location. The binding is either refreshed (if its timer expires) or updated when the MN gets a new CoA (because it moved to a new subnet).
- *Binding cache* – a cache containing a number of bindings for one or more MNs. A binding cache is maintained by both the HA and CNs.

- *Binding Update List (BUL)* – a list maintained by the MN of all bindings that were sent to the MN's HA and CNs. This list is maintained so that the MN can determine when a binding needs to be refreshed and is used for selecting the correct CoA when communicating directly with a CN.
- *Binding Update (BU)* – a message sent by the MN to its HA and CNs to update their binding cache when the MN moves to a new subnet and forms a new CoA.
- *Binding Acknowledge (BAck)* – a response from the HA or CNs to a BU received from a MN.

The first stage of a MIPv6 handover is the movement detection stage. The MN detects that it has moved to a new subnet by analysing the *router advertisement* (RA) sent by the AR in the new IP subnet. The AR sends RAs periodically. If the MN does not receive a RA at the frequency indicated by the RA from the old AR, it can send a *router solicitation* (RS). The new AR will respond with an appropriate advertisement. Since the MN does not have to wait a full period for the next RA, the CoA establishment is accelerated. Configuration information may be included with the RA. A registration lifetime is also sent with the RA. This is the maximum time that the MN is allowed to remain registered to that AR. If RAs are broadcast at a higher rate, handover latency and packet loss are both reduced. However, bandwidth on the wireless link is wasted by the added signalling messages. This could be a significant drawback on a network with many access points and MNs.

The MN then forms a CoA using stateless address autoconfiguration. Once the MN has a new CoA it must inform its HA and CNs of its movement. This is called MIPv6 handover registration. The MN sends a *binding update* (BU), indicating the binding between its new CoA and its HoA, to its HA. The MN can request an acknowledgement. This reply from the HA is a *binding acknowledge* (BAck). The HA needs to store the information about the MN's current address so that it may forward packets addressed to the MN's HoA to the MN. The HA contains a binding cache which contains all bindings for the MNs it serves. When the HA receives a BU, it first validates the message. If the HA accepts the BU, it searches its binding cache to see if an entry already exists for the MN's HoA. If an entry is found, the HA updates the

entry with the new information from the BU, otherwise a new entry is created. The HA may now act as a proxy for the MN. The HA also defends the MN's HoA, so if another node in the MN's home subnet tentatively configures the MN's address and tests it using Duplicate Address Detection (DAD), the HA indicates that it is already assigned to another node. Now, should the HA receive packets addressed to the MN, it checks its binding cache to verify whether an entry for the MN exists. When an entry is found, the packets are tunnelled to the MN's CoA. The tunnel entry point is the HA and the tunnel exit point is the MN's CoA. The MN may continue to send packets directly to its CNs. This type of routing is called triangle routing, since packets from a CN to the MN must be transmitted via the HA. When a CN receives a BU or when it receives packets from the MN's new CoA, it may cache this new CoA and begin routing packets directly to the MN. This is known as route optimisation.

A.1.3 Handover Enhancements

Hint-based Movement Detection

There are two distinct types of movement detection methods: advertisement-based and hint-based. Advertisement-based movement detection methods are those that require periodic agent advertisements to determine MN positions. Hint-based methods are those that require handover hints to determine movement.

Lazy cell switching (LCS) and *eager cell switching* (ECS) are two advertisement-based movement detection algorithms [24]. The advertisement lifetime is the maximum time that an advertisement may be considered valid in the absence of subsequent advertisements (at most 3 times the interval between successive advertisements). In the case of LCS, if the MN does not receive any subsequent advertisements within the advertisement lifetime, it assumes that the MN has moved to a new subnet. If by this time the MN has not cached any advertisements from other agents it begins to actively search for one by broadcasting a RS. ECS, on the other hand, assumes *any* unheard advertisement to be an indication that handover has taken place. For this reason, ECS performs faster handovers than LCS.

Hint cell switching (HCS) is a hint-based movement detection algorithm [24]. HCS makes use of hints generated during link layer handovers. If the IP layer were made

aware of such a hint it could force the MN to broadcast a RS. This effectively reduces the movement detection latency, allowing HCS to enable faster handovers than its advertisement-based counterparts. However, it wastes bandwidth due to increased signalling [23]. Since the movement detection latency is a substantial contributor to total handover latency, the total handover latency is significantly reduced.

Hierarchical Mobile IPv6 (HMIPv6)

HMIPv6 [25] was designed to allow MNs to move within a particular domain without having to update their HAs or CNs every time they move. A single BU is sent to a Mobility Anchor Point (MAP) in the visited network. A MAP essentially serves as a local home agent located in the visited network. When a MN visits a foreign network it will discover the IPv6 address of the MAP and request a temporary home address in the MAP's domain. This temporary home address is called the *regional care-of address* (RCoA). To distinguish a MN's address in the MAP's domain from its CoA in a foreign subnet, the term *on-link care-of address* (LCoA) is used to refer to the latter. The performance impact of mobility is reduced by handling local movement locally and hiding this from the HA [20, 30]. A hierarchy of MAPs is created between the HA and the MN. Tunnelling can be changed in the MAP hierarchy without the MN having to register with the HA. Thus, the signalling overhead and delay concerned with the BU (and consequently the handover registration process) in MIPv6 is reduced. HMIPv6 allows MIPv6 to scale better for quick handovers.

Fast Mobile IPv6 (FMIPv6)

FMIPv6 [26] sets up services for the MN on the new AR before the movement of the MN. This helps to minimise the delay associated with movement detection and removes MIPv6 signalling to the HA or CNs from the critical handover time. FMIPv6 achieves this by allowing MNs to anticipate IP layer mobility. The MN makes use of link layer hints to determine whether a MIPv6 handover is imminent, so that the network layer handover may be initiated before the link layer handover is complete.

Once a MN has anticipated the handover, it requests information on neighbouring ARs. A MN does this by sending a *router solicitation for proxy advertisement* (RtSolPr). This solicitation may contain the identity of one or more access points (APs, e.g., obtained from a link layer scan procedure), thus requesting subnet

information corresponding to the AP. It may also contain a request for all nearby AP-AR couples. The currently default AR responds with a *proxy router advertisement* (PrRtAdv) resolving the specified AP identifier. The information is in the form of an tuple. AR-Info may include the AP link layer address, the AR link layer address, the AR subnet prefix and the prefix length. The MN sends a *fast binding update* (FBU) to its current AR (PAR) with its current CoA and the new AR (NAR) it plans to switch to. The PAR sends a *handover initiate* (HI) message to the NAR, containing the identity of the MN. The NAR responds with a *handover acknowledge* (HAck) message and the PAR then sends a *fast binding acknowledgement* (FBAck) back to the MN. After receiving the FBU and HAck messages, the PAR starts forwarding the MN's packets to the NAR. After receiving the FBAck message, the MN has the necessary information about the new subnet and a new CoA, and is ready to actually switch links. If an FBAck was received before the MN moved, then the MN assumes that data will be forwarded to its new location. However, if the MN detached from the PAR before receiving the acknowledgement, it will not have this information. If the MN advertises its presence on the new subnet, the NAR can forward buffered messages to it. The *fast neighbour advertisement* (FNA) message is used to inform the NAR that the MN is now attached to its subnet and requests two things: that the NAR forward buffered packets to it (including the acknowledgment), and that the NAR tell the MN if the new CoA is invalid. The latter information is delivered to the MN with a *neighbour advertisement acknowledgement* (NAAck). This type of handover is called a predictive handover [20, 27].

FMIPv6 also defines a reactive handover scenario, where the MN was unable to anticipate handover so it reacts when the handover is already in progress [20, 27]. In this case the NAR sends the FBAck to the MN once the link layer handover is complete. The PAR tunnels packets destined for the MN to the NAR (which then forwards the packets to the MN).

A.2 802.11 Wireless LANs

The 802.11 standard is a family of specifications for wireless local area networks (WLANs) developed by a working group of the Institute of Electrical and Electronics

Engineers (IEEE). The 802.11 standard independently defines both the physical layer and data link layer (layers 1 and 2 of the OSI model). The original 802.11 standard provides 1-2 Mbps transmission in the 2.4 GHz band using either frequency hopping spread spectrum (FHSS) or direct sequence spread spectrum (DSSS). Due to the fairly low data rates, this standard is largely outdated. Extensions have been made to the original standard:

- The 802.11b ('baseline') standard [53] is backward compatible with 802.11. Its frequency band is subdivided into several channels. Standard DSSS together with advanced coding techniques allow data rates of up to 11 Mbps to be achieved.
- The 802.11a ('another band') standard is another extension to 802.11 that allows data rates of up to 54 Mbps in the 5 GHz band using orthogonal frequency division multiplexing (OFDM).
- 802.11g ('going beyond b') [12] allows data rates of up to 54 Mbps to be achieved in the 2.4 GHz band using OFDM. It is compatible with 802.11b.
- 802.11e [36] adds QoS features and multimedia support to the legacy 802.11 wireless standards, while maintaining full backward compatibility with these standards. QoS and multimedia support are critical to WLANs where voice, video and audio are to be delivered.

A few other specifications exist in the 802.11 family. They will not be discussed here.

802.11 supports physical technologies that communicate at multiple data rates. For example, the 802.11g physical layer is able to support data rates of 1, 2, 5.5, 6, 9, 11, 12, 18, 24, 36, 48 and 54 Mbps using different modulation schemes. When the link quality deteriorates, lower rates are used to offset poor channel conditions and minimise bit-error rates. The capacity of a wireless link is greatly reduced when lower transmission rates are used.

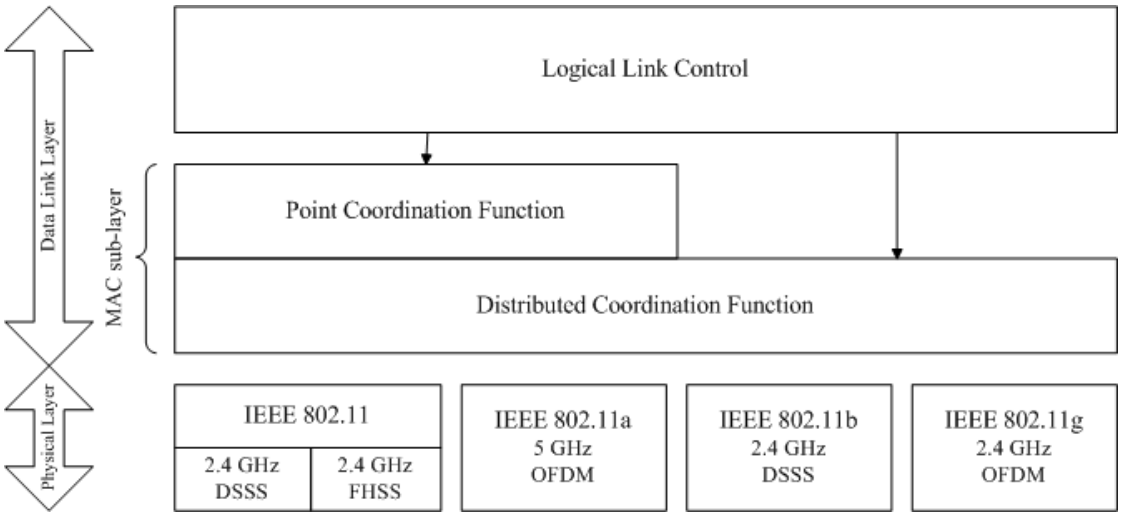

Figure A.2: The 802.11 protocol stack.

The 802.11 protocol stack is illustrated in Figure A.2. 802.11 specifies the physical and data link layers. The physical layer defines the frequency band at which transmission occurs, supported data rates, and data encoding and modulation techniques. The data link layer is composed of two layers, the Media Access Control (MAC) layer and the Logical Link Control (LLC) layer.

The MAC layer prevents interference between adjacent stations by regulating access to the shared radio frequency band. This is achieved through the use of two sub-layers, the Distributed Coordination Function (DCF) and the Point Coordination Function (PCF). The DCF is a contention algorithm that provides access to all data traffic. The wireless node either waits for the channel to become idle before it begins its transmission, or it uses a request-to-send/clear-to-send system to determine whether it can use the channel. The PCF is a centralised MAC algorithm that provides a contention-free service. It is dependant on the priority of data transmissions, based on particular timing requirements. 802.11e does not make use of DCF or PCF. Instead, it defines a Hybrid Coordination Function (HCF). There are two access mechanisms within the HCF, contention-based channel access, referred to as Enhanced Distributed Channel Access (EDCA), and controlled channel access, referred to as HCF Controlled Channel Access (HCCA). HCF provides differentiated channel access to frames of different priorities as labelled by the higher layer [2, 54].

The LLC layer provides an interface to higher layers of the protocol stack and performs simple functions such as error control.

The 802.11 standard enables two operational modes, an ad hoc mode and an infrastructure mode. Nodes in an ad hoc topology communicate directly with each other. There is no use of an *access point* (AP) or any connection to a wired network. On the other hand, all communication between nodes in an infrastructure topology occurs via an AP. An AP has at least one wireless interface and one wired interface. It serves as a bridge between the wired network and the WLAN. In order to establish network connectivity, a node must first authenticate and then associate with an AP. 802.11 concepts considered in this research are within the context of infrastructure networks. A Basic Service Set (BSS) is a collection of communicating nodes within a particular localised area. In infrastructure mode, these nodes will communicate through the same AP. Two or more BSSs can be interconnected through a wired network. The BSSs are grouped to form an Extended Service Set (ESS). An ESS network allows larger and more complicated wireless networks to be formed [54]. 802.11 allows wireless nodes to maintain connectivity and ongoing sessions as they move between BSSs. This is achieved through 802.11 handovers between APs.

A.2.1 802.11 Handover

A node in a WLAN can move within a single BSS, remaining in the coverage area of its associated AP. This node can continue communicating without any change in its AP association. However, if a node moves from one BSS to another, it will have to associate with a new AP to carry on communicating.

The case of a node moving from one BSS to another within the same ESS is termed a link layer handover, because only link layer mechanisms are involved. However, a movement from a BSS in one ESS to a BSS in another ESS may result in the node being located on a new network. In this case, the upper layers are affected by the movement. Therefore, in addition to a link layer handover, a MIPv6 handover is necessary.

During a link layer handover, a node changes its physical layer connectivity from one AP to another. The quality of a wireless link gradually deteriorates as the communicating nodes move further apart. If a particular node is associated with an AP and the link quality falls below a certain threshold, the node will initiate handover to another AP offering a higher quality link (assuming such an AP exists). Parameters such as received signal strength or signal-to-noise ratio (SNR) are used as an indication of the quality of a wireless link.

The node then needs to discover new candidate APs. It does this by performing a scan. Neighbouring APs usually operate on different frequencies to avoid interference. Therefore, the node performs a scan on all channels to discover any new APs. A wireless node can perform active or passive scans. On an active scan, the node broadcasts a *probe request* message on a channel and starts a timer. APs using that channel respond with a *probe response*. When the timer expires, the node scans the next channel. Probe responses are then prioritised according to signal strength to determine the best possible candidate AP. On a passive scan, the node passively waits for beacon messages on a particular channel. After a delay it scans the next channel until all channels are scanned.

Once the node has chosen a new AP, it needs to associate with it. 802.11 specifies that a wireless node must dissociate with its old AP before attempting to associate with a new one. The node must then authenticate itself before associating with the AP.

Appendix B

Evaluation Framework Implementation

B.1 System Configuration

B.1.1 Kernel Configuration

Before a MIPv6 framework could be created, a fully functional IPv6 network had to be developed. Debian Linux was installed on each of the five Computers acting as network nodes, running kernel version 2.6.16. The Linux 2.6.x kernel has built in IPv6 capability. To enable this, the following settings must be in the `.config` file:

```
CONFIG_EXPERIMENTAL=y
CONFIG_SYSVIPC=y
CONFIG_NET=y
CONFIG_INET=y
CONFIG_IPV6=y
CONFIG_XFRM=y
CONFIG_XFRM_USER=y
CONFIG_IPV6_TUNNEL=y
```

The `.config` file is a file usually stored in the Linux source code directory. It contains all the kernel settings for that particular compilation.

To test for IPv6 support in the running kernel, type:

```
test -f /proc/net/if_inet6 && echo "kernel is IPv6
   ready"
```

With each node IPv6 enabled, IPv6 link local addresses are automatically created for each interface. For the framework to function as a network, each interface needs to be assigned a global IPv6 address.

B.1.2 Network Configuration

Debian Linux uses the `/etc/network/interfaces` file to hold the network settings for all the network adapters on that Computer. An example of what the `/etc/network/interfaces` file may look like is:

```
# The primary network interface
auto eth0
iface eth0 inet6 dhcp
```

This file will be used later to statically assign addresses and routes for the various network adapters. To reset the network settings (e.g., if the interfaces file has been changed) type:

```
/etc/init.d/networking restart
```

B.1.3 Access Points

The D-Link APs are link layer devices, and each device will assume the IP address of the adapter it is attached to (IPv4 or IPv6). To allow for configuration, the device is equipped with an IPv4 address, *192.168.0.50/24*. To log into the device, open a web browser and type *http://192.168.0.50* into the address bar with:

Username: *admin*
Password: *password*

To configure the AP, the IP address of the Computer being used will have to be changed so that both the Computer and the AP are on the same network. To do this, type:

```
ifconfig eth0 192.168.0.49 netmask 255.255.255.0
  broadcast 192.168.0.255
```

To reset the network to its original settings type:

```
/etc/init.d/networking restart
```

A script on the HA and ARs was written to automatically setup the IP addresses for AP configuration:

```
/home/mipha/Desktop/Scripts/ap_admin_config
```

B.1.4 IPv6 Tunnel

A global IPv6 address was obtained from *Hexago/Freenet6*, an address broker. Not only does *Hexago/Freenet6* provide an official IPv6 address, but it also assigns a network prefix meaning that all nodes within a network can be assigned IPv6 addresses.

The evaluation framework (an IPv6 network) is located within an IPv4 network on the university campus. Therefore, to connect to the global IPv6 network, an IPv4/IPv6 tunnel must be created between the node acting as the border router (the HA) and the global IPv6 network. This allows the evaluation framework to be IPv6 connected.

A global IPv6 address was acquired for the HA and then used together with a network prefix to assign unique addresses to all the remaining network elements. This can be done as follows:

- Create a user account at *http://www.hexago.com*
- Download the Tunnel Server Protocol Client

  ```
  apt-get install tspc
  ```

- Edit `/etc/tsp/tspc.conf`

  Ensure the *passwd* and *userid* are the ones received from *Hexago/Freenet6*

  Also ensure the following options:

  ```
  server=broker.freenet6.net
  host_type=router
  prefixlen=48
  ```

- Type:

  ```
  tspc -vvv
  ```

A tunnel is now established between the HA and the IPv6 backbone (known as *6Bone*). This tunnel can be seen in the network interfaces list. Type `ifconfig`, there should be an entry called `tun`.

The HA was assigned the global IPv6 address `2001:5c0:8d03::1`, with a network prefix of `48`. The HA can now connect to the IPv6 backbone. This can be shown by going to *http://www.kame.org* where the animated image of a dancing turtle indicates IPv6 connectivity.

A script was written to automatically configure the IPv4v6 tunnel:

`/home/mipha/Desktop/Scripts/v4v6_tunnel_config`

B.1.5 Routing Configuration

Once the HA has been assigned a global IPv6 address, all the other nodes can acquire there own unique addresses. To do this, stateless address autoconfiguration with prefix discovery is used. The HA needs a RA daemon program: *radvd*. Once this program is installed and configured, it sends out periodic advertisements on specified interfaces containing the prefix of that network. The nodes on that network can then receive these advertisements and generate their own unique global IPv6 addresses.

An IPv6 address and prefix have already been assigned (`2001:5c0:8d03::1/48`). This prefix must be split to allow three subnets:

Home subnet: `2001:5c0:8d03:1::/64`

Foreign subnet A: `2001:5c0:8d03:2::/64`

Foreign subnet B: `2001:5c0:8d03:3::/64`

Each of the HA interfaces must first be configured manually. The network must be bootstrapped at some point (i.e., all IP address configuration is not automatic). The HA has two interfaces, one to the home network and one to the Internet. The interface to the Internet has already been assigned an address by *Hexago/Freenet6* (This address is `2001:5c0:8d03::1/48`). To statically assign addresses to the other interfaces the `/etc/network/interfaces` file is edited to include:

```
# Home network
iface eth1 inet6 static
address 2001:5c0:8d03:1::1
netmask 64
gateway 2001:5c0:8fff:fffe::4044
```

The gateway address is the address of the tunnel server. To determine this address, type:

```
grep type=\"ipv6\" /var/log/tspc/log
```

The output will look something like:

```
<server> <address type="ipv4">206.123.31.114</address>
  <address type="ipv6">
  2001:05c0:8fff:fffe:0000:0000:0000:4044</address>
  </server>
```

`2001:5c0:8fff:fffe::4044` is the IPv6 address of the tunnel server and that is the default gateway.

The following lines were added to the `/etc/network/interfaces` file of the HA to enable all nodes to reach one another and the outside world:

```
# Home network
up ip -6 route add 2001:5c0:8d03:1::/64 dev eth1
down ip -6 route del 2001:5c0:8d03:1::/64 dev eth1
```

Quagga is a routing software suite that comes with many routing daemons. RIPng is an IPv6 routing protocol that is implemented in the *ripngd* daemon. The *zebra* daemon is a core daemon, which acts as an abstraction layer to the underlying Linux kernel. The *zebra* daemon configuration file can be found at `/etc/quagga/zebra.conf`. An example of this file is given below:

```
! -*- zebra -*-
! zebra configuration file
```

```
!
hostname HA
password zebra
enable password zebra
!
! Interface's description
!
interface eth0
 ipv6 address 2001:5c0:8d03::1/64
 ipv6 nd suppress-ra
!
interface eth1
 ipv6 address 2001:5c0:8d03:1::1/64
!
 ipv6 nd suppress-ra
!
interface lo
!
interface sit0
 ipv6 nd suppress-ra
!
 ip forwarding
 line vty
 log stdout
```


The *ripngd* configuration file can be found at `/etc/quagga/zebra.conf`, an ecample:

```
! -*- rip -*-
! RIPngd configuration file
!
hostname HA
password zebra
!
```

```
! debug ripng events
! debug ripng packet
!
router ripng
 network eth0
 network eth1
 redistribute connected
 redistribute static
 redistribute kernel
!
! network sit1
! route 3ffe:506::0/32
! distribute-list local-only out sit1
!
 log stdout
```

To run the *zebra* daemon, type `zebra`, and to run the *ripngd* daemon, type `ripngd`.

B.1.6 Router Advertisement Configuration

The *radvd* program was installed on each of the machines in the network. The configuration file is found at `/etc/radvd.conf`. Advertisements must be sent to the home network from one interface.

For this type of behaviour, the *radvd* configuration file should look as follows:

```
# Advertisements to home network
interface eth1 {
   AdvSendAdvert on;
   AdvIntervalOpt on;
   MinRtrAdvInterval 3;
   MaxRtrAdvInterval 4;
   AdvHomeAgentFlag on;
   AdvHomeAgentInfo on;
   HomeAgentLifetime 1800;
```

```
    HomeAgentPreference 10;

    prefix 2001:5c0:8d03:1:: /64
    {
       AdvAutonomous on;
       AdvOnLink on;
       AdvRouterAddr on;
       };
};
```

To run the *radvd* daemon type:

```
/etc/init.d/radvd restart
```

This is automatically done when running the `v4v6_tunnel_config` script. Once the daemon is up and running, other machines on the network may acquire unique global IPv6 addresses allowing the framework to act as a fully functional IPv6 network.

B.1.7 Testing IPv6 Connectivity

To test the IPv6 connectivity of each node the `ping6` and `traceroute6` tools were installed on each machine:

```
apt-get install iputils
```

Using these tools, it was possible to test that each node could communicate successfully with each other node.

B.2 MIPv6 Extension

B.2.1 Patching the Kernel

MIPL works as a patch for the Linux kernel. The version of MIPL used on the test-bed, MIPL 2.0.2, is designed for use with the 2.6.16 kernel. Therefore, the source for this kernel must be downloaded and installed.

Copy the tar ball to the `/usr/src/` directory. Unpack the tar ball:

```
tar xfvj /usr/src/linux-2.6.16.tar.bz2
```

This creates a `linux-2.6.16/` directory with the necessary install files.

The MIPL 2.0.2 patch may be downloaded from *http://www.mobile-ipv6.org*, and copied to the `/usr/src/` directory. Move to the kernel source directory and patch the kernel:

```
cd /usr/src/linux-2.6.16/
zcat mipv6-patch.gz | patch -p1
```

(Replace `mipv6-patch.gz` with the actual path of the patch file downloaded.)

The fmipv6.org implementation also requires a kernel patch:

```
cat fmipv6-linux-2.6.16.20.patch | patch -p1
```

(Replace `fmipv6-linux-2.6.16.20.patch` with the actual path of the patch file downloaded.)

Now the kernel is ready for configuration and recompilation. The `.config` file stores all the settings for that particular kernel. It is very difficult to know what to include in the kernel and what not to. For this reason it is best to use a previous `.config` file and make minor adjustments as required. The previous `.config` file is stored in `/boot` and is called `config-2.6.8-2-686`. Copy this file to the Linux source directory and rename it to `.config`.

```
cp /boot/config-2.6.8-2-686 /usr/src/linux-2.6.16
cd /usr/src/linux-2.6.16/
mv config-2.6.8-2-686 .config
```

There are numerous applications that allow the `.config` file to be manipulated (e.g., xconfig, menuconfig, etc.). To adjust the `.config` file, type:

```
make menuconfig
```

Ensure that at the `.config` file has the following settings:

```
CONFIG_EXPERIMENTAL=y
```

```
CONFIG_SYSVIPC=y
CONFIG_PROC_FS=y
CONFIG_NET=y
CONFIG_INET=y
CONFIG_IPV6=y
CONFIG_IPV6_MIP6=y
CONFIG_XFRM=y
CONFIG_XFRM_USER=y
CONFIG_XFRM_ENHANCEMENT=y
```

The Home Agent and Mobile Node also need:

```
CONFIG_IPV6_TUNNEL=y
CONFIG_IPV6_ADVANCED_ROUTER=y
CONFIG_IPV6_MULTIPLE_TABLES=y
```

The Mobile Node also needs:

```
CONFIG_IPV6_SUBTREES=y
```

For some additional movement indicators on the Mobile Node, set:

```
CONFIG_ARPD=y
```

For IPsec support:

```
CONFIG_INET6_ESP=y
```

For use of the IPsec tunnel:

```
CONFIG_NET_KEY=y
CONFIG_NET_KEY_MIGRATE=y
```

`CONFIG_IPV6_MIP6_DEBUG` must not be turned on, unless the user is specifically debugging the MIPv6 extensions in the kernel. This option will degrade system performance and output too many debug messages. For more information see the documentation provided with MIPL package.

Most IPv6 functionality is compiled as a module, rather than as part of the kernel. For MIPv6 functionality it is advantageous to compile all MIPv6 and IPv6 functionality as part of the kernel.

A new `.config` file has now been created (i.e., a new set of kernel settings). Now the new kernel must be compiled and installed. Type:

```
make bzImage
make modules
make modules_install
```

These steps will take quite some time. An Initial RAMDisk may need to be created, depending on the kernel settings:

```
mkinitrd -o /boot/initrd-2.6.16.img 2.6.16
```

Now that the kernel is created it must be prepared for use. The created kernel and the `System.map` file must be copied to the `/boot` directory:

```
cd /usr/src/linux-2.6.16/
cp arch/i386/boot/bzImage /boot/bzImage-2.6.16
cp System.map /boot/System.map-2.6.16
ln -s /boot/System.map-2.6.16 /boot/System.map
```

Next, the Grub loader must be adjusted. This program is responsible for loading the kernel at boot time. The following must be added to the `/boot/grub/menu.lst` file:

```
title mobile_ipv6 kernel (2.6.16)
      root(hd0,0)
      kernel /boot/bzImage-2.6.16 root=/dev/hda1 ro
      initrd /boot/initrd-2.6.16.img
```

Adding this entry above the previous ones makes it the default. The Computer may now be rebooted and a MIPv6 enable kernel will be running.

B.2.2 User Space MIPL Part

The user space part of MIPL can be downloaded from *http://www.mobile-ipv6.org*. Unpack the tar ball to `/usr/src/`. This will create a MIPL directory. Change to the main MIPL directory.

The fmipv6.org implementation requires a patch for MIPL:

```
cat fmipv6-mipv6-2.0.2.patch | patch -p1
```

(Replace `fmipv6-mipv6-2.0.2.patch` with the actual path of the patch file downloaded.)

Then type:

```
CPPFLAGS='-isystem /usr/src/linux-2.6.16/include'
   ./configure
make
make install
```

For IPSec support, refer to documentation provided with the MIPL package.

B.2.3 Node Configuration

The MIPL configuration file can be found at `/usr/local/etc/mip6d.conf`. An example of the configuration file on the HA is given below:

```
NodeConfig HA;

DebugLevel 10;
DoRouteOptimizationCN disabled;

## List of interfaces where we serve as Home Agent
Interface "eth1";

## IPsec configuration
UseMnHaIPsec disabled;
```

```
KeyMngMobCapability disabled;

IPsecPolicySet {
   HomeAgentAddress 2001:5c0:8d03:1::1;
   HomeAddress 2001:5c0:8d03:1:211:20ff:fe48:246/64;

   IPsecPolicy HomeRegBinding UseESP;
   IPsecPolicy MobPfxDisc UseESP;
   IPsecPolicy TunnelMh UseESP;
}
```

An example of the configuration file on the MN is given below:

```
NodeConfig MN;

DebugLevel 10;
DoRouteOptimizationCN disabled;
DoRouteOptimizationMN disabled;
SendMobPfxSols enabled;
UseCnBuAck enabled;
Interface "eth0";
MnRouterProbes 1;

MnHomeLink "eth0" {
   HomeAgentAddress 2001:5c0:8d03:1::1;
   HomeAddress 2001:5c0:8d03:1::99/64;
}

## IPsec configuration
UseMnHaIPsec disabled;
KeyMngMobCapability disabled;

IPsecPolicySet {
   HomeAgentAddress 2001:5c0:8d03:1::1;
```

```
    HomeAddress 2001:5c0:8d03:1::99/64;

    IPsecPolicy HomeRegBinding UseESP;
    IPsecPolicy MobPfxDisc UseESP;
    IPsecPolicy TunnelMh UseESP;
}
```

For further information on MIPL configuration, consult the `man` page of the configuration file:

```
man mip6d.conf
```

B.3 FMIPv6 Extension

The FMIPv6 implementation can be downloaded from *http://www.fmipv6.org*. Unpack the tar ball to `/usr/src/`. This will create an FMIPv6 directory. The Linux kernel patch and MIPL patch should already have been applied. Modifications had to be made to the `iwmanage.c` and `fmipv6-mn.c` files for them to operate correctly on the test-bed. Replace the default files provided with the fmipv6.org source with the modified files.

Change to the main FMIPv6 directory. Prepare the environment by typing:

```
aclocal
autoheader
autoconf
automake --add-missing
```

Then, to compile and install the software, type:

```
CPPFLAGS='-isystem /usr/src/linux-2.6.16/include'
   ./configure
make
make install
```

B.3.1 Node Configuration

The configuration file for an AR can be found at `/usr/local/etc/fmipv6-ar.conf`. An example of this configuration file is given below:

```
DetachFromTTY = false

DebugLevel = 3

AskSupportForBuffering = true

BufferLength = 20

Interface {
   IfName = "eth0"
   preference = 1
}

NIHover {
   do_ni_hover = true
   ap_lla = 00:0D:88:50:70:17
}

nap fmipar1 {
   nap_lla = 00:0D:88:50:70:17
   nr_lla  = 00:13:D3:65:F4:31
   nr_addr = 2001:5c0:8d03:3::50/64
   nr_pfx  = 2001:5c0:8d03:3::/64
}
```

The configuration file for the MN can be found at `/usr/local/etc/fmipv6-mn.conf`. An example of this configuration file is given below:

```
DetachFromTTY = false

DebugLevel = 3

ScanningInterface {
   IfName = "eth0"
}

LinkQuality {
   UseLinkQualityTriggers = false
   Threshold = -50
}
```

For further information on FMIPv6 configuration, consult the `man` pages of the configuration files:

```
man fmipv6-ar.conf
man fmipv6-mn.conf
```

Appendix C

Accompanying CD-ROM

The following information may be found on the CD-ROM that has been included with this document:

- **Software**
  All the source code that was used to develop the evaluation framework can be found in the "Software" directory.

- **Configuration Files**
  Example configuration files for each of the network nodes in the evaluation framework can be found in the "Config Files" directory.

- **Research Literature**
  Electronic copies of research papers and other literature used in this research can be found in the "Research Literature" directory.

- **Thesis**
  This document, in PDF format, can be found in the "Thesis" directory.