Prediction Assisted Fast Handovers for Seamless IP Mobility

Prepared by:
Andre E. Bergh

Supervised by:
Neco Ventura

Department of Electrical Engineering
University of Cape Town
2006

This dissertation is submitted to the University of Cape Town
in fulfilment of the academic requirements
for the Degree of Master of Science in Engineering

October 2006

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 are indicated in the acknowledgements or references as appropriate.

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

Signed by candidate

06/10/2006

Andre E. Bergh

Date

Acknowledgements

I would like to thank the following people for their assistance during the course of this project.

Mr. Neco Ventura, for his supervision and guidance throughout this project.

Mr. David Waiting, for his technical assistance and invaluable criticism.

The past and present members of the Communications Research Group at UCT, for their advice and feedback.

Synopsis

In wired IP-based networks, the single biggest challenge in providing high quality real-time media is achieving a Quality of Service (QoS) consistent with user expectations. In wireless networks, one of the principal additional factors affecting QoS is the inherent service disruption during the handovers of mobile nodes. This research investigates the techniques used to improve the standard Mobile IP handover process and provide proactivity in network mobility management. Numerous fast handover proposals in the literature have recently adopted a cross-layer approach to enhance movement detection functionality and make terminal mobility more seamless. Such fast handover protocols are dependent on an anticipated link-layer trigger or pre-trigger to perform pre-handover service establishment operations. This research identifies the practical difficulties involved in implementing this type of trigger and proposes an alternative solution that integrates the concept of mobility prediction into a reactive fast handover scheme. A data-mining approach is used to predict a users network-layer mobility based on its previous mobility history. It is shown that the data-mining prediction algorithm is simple, accurate and effective. The proposed scheme, called Prediction Assisted Fast handover protocol for Mobile IP or PA-FMIP, enables the nodes' incoming packet stream to be tunneled to the predicted next access router during its link-layer handover. This essentially improves the seamlessness of the nodes' subnet transition by reducing the overall handover latency and packet loss. The work is evaluated through simulation to determine the accuracy capabilities of the prediction algorithm, as well as the achieved improvement in handover performance when compared to Mobile IPv6, reactive FMIPv6, proactive FMIPv6 and Simultaneous Binding for FMIPv6.

Contents

List of Tables

Glossary

This section defines some of the commonly used terms and abbreviations that appear throughout this document.

1G The first generation of analog mobile phone technologies, including AMPS, TACS and NMT.

2G The second generation of wireless communication systems (including GSM, CDMA IS-95 and D-AMPS IS-136) using digital transmission and advanced control techniques to improve the performance of voice communications, provide special features and limited digital messaging capabilities, such as GSM.

3G The third generation of mobile phone technologies covered by the ITU IMT-2000 family. It allows greater bandwidth and opens the way to increased data-over-wireless solutions.

3GPP The 3rd Generation Partnership Project is a global collaboration between 6 partners: ARIB, CWTS, ETSI, T1, TTA, and TTC. The group aims to develop a globally accepted 3rd-generation mobile system based on GSM.

Access Point (AP) An entity that bridges information between the wireless medium and the distribution medium on behalf of its associated stations. Access points are only used in infrastructure wireless LANs.

Access Router (AR) An IPv6 router on the the periphery of a network that provides mobility services to the visiting mobile nodes.

Application Program Interface (API) A set of routines provided in libraries that extends a language's functionality.

Binding The association of the home address of a mobile node with a care-of address for that mobile node, along with the remaining lifetime of that association.

Binding Cache A cache of bindings for other nodes. This cache is maintained by home agents and correspondent nodes. The cache contains both "correspondent registration" entries and "home registration" entries

Care-of-Address (CoA) A temporary IP address assigned to a mobile node while it is visiting a foreign network.

Corresponding Node (CN) Any network node that is communicating with the mobile node.

Data Mining The process of analysing data to identify patterns, associations, significant structures or relationships, from information stored in a data repository or database.

Foreign Link Any link other than the mobile node's home link.

GPRS The General Packet Radio Services is a GSM Packet Based bearer for the delivery of data services. With GPRS charges are based on the amount of information downloaded rather than the duration of the connection.

GSM The Global System for Mobile Communications is one of the leading digital cellular systems. It uses narrowband TDMA, which allows eight simultaneous calls on the same radio frequency.

Home Agent (HA) An IPv4 or IPv6 router residing on a mobile node's home network. It forwards the mobile node's traffic to its current point of attachement while it is away from its home network.

IP Internet Protocol provides for the transmission of datagrams from a source to a destination. The source and destination are hosts identified by fixed-length IP addresses.

Link Entities that share a link are physically connected through a communication channel. These entities are able to communicate directly using a link layer protocol.

Link-layer Address A link-layer identifier for an interface, such as IEEE 802 addresses on Ethernet links.

Mobile Node (MN) A network node that is able to communicate while moving through different networks.

Movement Detection IP layer mechanism that allows a mobile node to detect its arrival on a new IP network, i.e. as part of a Mobile IP handover.

QoS Quality of Service is the idea that transmission rates, error rates, and other characteristics of a communications network can be measured, improved, and, to some extent, guaranteed in advance.

Signal-to-Noise Ratio (SNR) The difference in decibels between the received signal strength and the noise level.

UMTS Universal Mobile Telecommunications Service, part of the IMT-2000 initiative, is a 3G standard supporting a theoretical data throughput of up to 2 Mbps.

VoIP Voice over IP is the two-way transmission of voice information over a packet-switched TCP/IP network. (Also known as "IP telephony".)

Chapter 1

Introduction

1.1 Background Information

A new generation of wireless computing devices is emerging. Technological developments have miniaturised digital hardware and brought about a wave of notebooks, handheld PDAs and data-ready cellular devices to satisfy the demand for increasing user mobility and pervasive communication needs. More and more users are communicating online through the Internet using Internet telephony and video conferencing applications.

A significant percentage of the Internet now comprises of wireless access networks, allowing Internet users freedom from fixed desktop systems. However, the Internet is not tuned to allow mobility in the midst of data transfers because protocols used in the Internet are not conceived for devices that frequently change their point of attachment in the Internet topology. This poses a problem, as users expect to connect to the Internet from anywhere, at anytime and to remain permanently connected without any disruption of service.

The Internet Protocol (IP) was released in 1980s for the purpose of routing packetised data between fixed nodes attached to the Internet at static points. It serves as a network layer inter-networking protocol designed to function independently from underlying hardware, physical media or data-link technology. This permits heterogeneous networks such as ATM, Ethernet, etc. to inter-operate and route data on a common level. This includes wireless link-layer technologies like 802.1x,

List of Figures

Bluetooth, UMTS, CDMA2000 and GPRS. The convergence of fixed and wireless networks at the network layer is still the focus of current research.

Support for network layer mobility was not catered for until the release of the Mobile IP protocol [1]. Mobile IP is a mobility management protocol that allows a mobile node to migrate between wireless links of different IP networks, while keeping its mobility functionality transparent to the upper layers. This transparency ensures the upper-layer sessions (e.g TCP) are maintained in spite of any network-layer change that may occur.

Mobile IP[1] adapts the standard IP routing mechanism such that a mobile node's traffic is forwarded to its current location as it moves between different networks. However, Mobile IP suffers from serious performance drawbacks, specifically during a handover. A handover is executed when a mobile node moves from one IP network / subnet to another. During a handover, a mobile node must reconfigure its IP address, and the forwarding of the node's traffic must be adjusted to reflect its new topological position. During this time, the node is temporarily disconnected from all IP networks, causing subsequent packets to be misrouted or dropped. The latency of a Mobile IP handover can therefore lead to serious performance degradation on upper-layer applications. It is the task of any mobility management scheme to enable network applications to continuously operate at the required quality of service (QoS) throughout an IP handover.

The type or class of application defines the level of disruption each can tolerate. Delay sensitive applications such as Voice Over IP (VoIP) require strict end-to-end latency and jitter boundaries to maintain an intelligible two-way conversation. TCP applications on the other hand, can tolerate and recover from handover latencies and packet loss, but at the expense of a decrease in overall data rate. There are a number of methods of optimising a IP handover, although the overall performance is limited by network conditions and topology. The Internet Engineering Task Force (IETF) is at the forefront in this field with a number of active working groups researching specific aspects related to IP mobility. The Mobile IP Handover Optimisation (MIPSHOP) working group has proposed a method of improving handover latencies by minimising the IP signalling procedures in the Mobile IP protocol, called *Fast Handovers for Mobile IPv6* (FMIPv6) [2]. The FMIPv6 protocol defines the signalling procedure that enables a mobile

[1]From hereon, Mobile IP and IP refer to version 6 (v6) unless specified otherwise.

node to send and receive data almost immediately after associating with a new link.

FMIPv6, and other fast handover protocols rely on the timeous reception of certain link-layer event information to solve the problem of movement detection. Traditional movement detection mechanisms rely purely on periodic router advertisements. Studies have shown that a cross-layer approach has a significant performance advantage over traditional methods. For example, a link-layer event or "trigger" indicating that a handover is about to occur, gives the network layer sufficient time to proactively perform address negotiation and registration operations prior to the link change. Although there are implementation issues regarding triggers, it is envisaged that future fast handover techniques will incorporate this type of inter-layer communication to minimise latencies and data loss, and provide seamless IP mobility.

1.2 Problem Description

The management of wireless connections in electronic computing devices is controlled by the data-link layer of its protocol stack. In most cases, the decision to choose one wireless service over another is based on signal strength or other radio parameters. This means that upper-layer protocols are forced to blindly react to the actions of the link-layer. Fast handover protocols such as FMIPv6 (in proactive mode) require the underlying link layer to anticipate link changes, and subsequently notify the network layer (and higher layers) of this anticipation. This proactive link layer notification or pre-trigger is also expected to carry the link layer address of the base station that the node will handover to. This address is obtained by methods specific to the link technology. Consider this process in a common 802.11b enabled mobile node operating in infrastructure mode. A number of practical difficulties arises:

- In order to scan for neighbouring access point (AP) beacons, the node must invoke the MLME-SCAN.request primitive [3]. This scanning procedure may take hundreds of milliseconds to complete. Because the node is in infrastructure mode, it cannot send or receive packets on it own link during the scan time.

- The node may however schedule this scan for a time prior to any real-time traffic to minimise any interruption. If the interval between the scan and handover is too long, the set of discovered neighbours may become out of date.

- A change in signal strengths through interference or fading may cause the link layer to handover to an AP that was not necessarily discovered by the earlier scan. The mobility characteristics of the mobile node are another variable in this scenario.

The need to predict the mobility of a mobile user is justified with the following advantages:

- Service providers may reserve sufficient resources at upcoming network points for the user, preventing possible call blocking.

- The establishment of security associations or trust relationships usually involves key distribution and lengthy authentication procedures. Performing these actions proactively may save time during network handovers.

- QoS is an important aspect of IP networks. Users often have a service level agreements with the network provider, defining parameters such as header compression, and expected bit rate. Mobility prediction allows the transfer of this context / state information to the next wireless access point to occur prior to a link-layer handover.

- With the exception of cellular technologies like CDMA2000 and UMTS, most link layer handovers are break-before-make or hard handovers. Knowing the next point of attachment in advance allows a mobile node to decouple the link-layer and network-layer handovers, and redirect or multi-cast packet streams to the new location. This softens the effect of link switching delay and reduces packet loss.

The number of benefits of proactive service establishment and the practical difficulties of pre-triggers suggests that an alternative approach may be suitable here. FMIPv6 caters for reactive type handovers, i.e. a network layer handover that

occurs (reacts) immediately after the link change. The inherent problem of this type of handover is that they are more prone to higher latencies and packet loss than proactive ones, and are not eligible for the benefits listed above. It is plausible then, if a separate mechanism was able to predict the mobility of a mobile node, then one could improve the performance of reactive handovers, to a degree that they rivalled proactive handovers.

1.3 Thesis Objectives

On wired IP-based networks, the single biggest challenge in providing high quality real-time media is achieving a Quality of Service (QoS) consistent with user expectations. In wireless networks, one of the principal additional factors affecting QoS is minimising service disruption during handovers of mobile nodes. This study investigates the techniques used to improve the standard Mobile IP handover process and provide proactivity in network mobility management. In the literature, a number of fast handover protocols and Mobile IP optimisations are presented. These protocols aim to mitigate the factors that contribute to the overall handover latency and packet loss. This research aims to examine these factors and uncover the particular mechanisms required to perform fast handovers in both proactive and reactive handover scenarios. In the past, mobile nodes had no means for performing pre-handover service establishment operations such as address negotiation or home registration. Numerous proposals in the literature have recently adopted a cross-layer approach to enhance movement detection functionality and make terminal mobility more seamless.

This thesis investigates the fundamental issues surrounding link-layer triggers and their use in fast handover protocols. The diversity of link-layer technologies and the lack of a standard trigger model creates a number of practical challenges related to the implementation of fast handover protocols such as FMIPv6.

Mobility prediction in mobile users can be achieved through a number of means. The work in this thesis introduces and explores the viability of using a mobile user's mobility history to determine their future locations. There are many techniques and algorithms that may be used to analyse this data, the most significant of these algorithms are reviewed and discussed. A novel approach to this method

of mobility prediction involves the use of data mining algorithms. This research aims to determine the suitability of this prediction algorithm when integrated into a reactive type fast handover scheme.

Using a simulation framework, the proposed prediction assisted fast handover protocol is evaluated in a number of experiments. FMIPv6 and Simultaneous Bindings [4] are used as a benchmark for the comparison of results. As with most fast handover systems, the forwarding / multicasting of data is used to reduce packet loss for a smoother handover. Indeed, this results in a network bandwidth overhead. The analysis of performance results weighs the performance improvement against the cost of overhead in a number of application scenarios.

1.4 Scope and Limitations

Several limitations have been set to reduce the scope of this study. Mobility functionality can be implemented on many layers of the network protocol stack. This work investigates cross-layer communication between the link-layer and network-layer. The exact link-layer mobility control mechanisms depends on the type of wireless access technology in question, thus any analysis of specific link-layer connection procedures are out of the scope of this work. This includes wireless issues such as dynamic rate scaling in 802.11 networks and other capacity related issues for wireless networks.

We do not attempt to reduce the link-switching delay of a handover or generalise an equation or heuristic for the optimal timing of a pre-trigger. The scope is also limited to infrastructure based networks, and does not consider ad-hoc networks.

One of the factors responsible for latency during handover is long round trip delays in the communication path. This thesis does not attempt to mitigate this problem as done in Hierarchical Mobile IP and other schemes.

The advantage of multicast over unicast is acknowledged for certain applications, for simplicity however the only transmission method used in the evaluation of this work is unicast. Also, the fragmentation of large IP packets is common in the Internet. The evaluation procedure in this work does not consider the effects this type of packet size changes. The evaluation also only considers data flowing in one direction at a time. This is based on the assumption that only

one party in a VoIP conversation is speaking at a time, and that a handover will only affect one party's traffic [5]. This is true for most streaming content such as Video On Demand (VoD), IPTV, and digital radio. However in the case of video conferencing applications, the source application would have to manage or buffer the outgoing data during each handover. This is an application buffering issue and is out of the scope of this study.

The field of movement prediction is very broad. A full analysis of all the available techniques, algorithms and mobility models is out of the scope of this work. This includes hardware dependent mobility tracking and prediction methods that involve GPS or radio sensing devices.

Seamless mobility management schemes often rely on active buffering mechanisms to reduce packet loss. Accurate buffering requires knowledge of the type of application being used *a priori*; for example, large buffers are suitable for large packet size (delay-tolerant) TCP streams but unsuitable for high data rate VoIP streams. Buffers, including jitter buffers are discussed in general, however a detailed investigation is out of the scope of this study.

1.5 Thesis Outline

The remaining sections of this document are structured as follows:

- Chapter 2 introduces some basic and fundamental networking concepts and terminologies. It then reviews the standard Mobile IPv6 protocol operation and its related handover issues. Special attention is then given to fast handover protocols and the link-layer trigger mechanisms. Link-layer and network-layer mobility issues are discussed in detail before the concept of mobility prediction is introduced. The different mobility prediction techniques in the literature are then reviewed, including schemes that utilise prediction to improve handover performance.

- Chapter 3 is separated into two main sections that make up the crux of this thesis. Firstly, the concept of data mining is introduced and the details of how it is used as the basis for the prediction algorithm are given. The motivational reasons for this approach are also discussed. The second

part of this chapter explains the exact signalling procedures involved in the mobility prediction assisted fast handover proposal (PA-FMIP). A simple example is given illustrating the PA-FMIP operation.

- Chapter 4 describes the evaluation platform used in this research. The design of the network topology and mobility details needed to adequately evaluate the proposed work in a real-world environment are described. It details exactly how the algorithms and theoretical handover protocols described in chapter 3 are implemented.

- Chapter 5 describes the tests performed for the quantitative evaluation. The results of each component of the proposed scheme are analysed and compared to suitable benchmark schemes.

- Chapter 6 presents a set of conclusions that were drawn up from these evaluations. This chapter also contains concluding remarks on several issues raised in preceding chapters. Recommendations are then given for future work and developments that should be done.

Chapter 2

Literature Review

This chapter provides a detailed overview and review of the literature pertaining to IP based networking and mobility issues in the Internet and mobile communication networks. It also provides the foundation for future chapters, and identifies the origins and objectives of this research. Section 2.1 begins by introducing the reader to basic network concepts and terminology.

2.1 Networking basics

Network architectures usually incorporate a number of inter-networking technologies. To manage the inter-working complexities of heterogeneous networks and protocols, the functionalities of each technology and protocol are grouped according to the Open Systems Interconnect (OSI) model. As shown in Figure 2.1, the OSI model is a layered logical network model that is used to simplify the design of individual network technologies. It describes a hierarchical networking protocol stack where the details of a given layer are abstracted to upper layers. This structure allows the complexities associated with transmitting information over a network to be localised at specific layers. The OSI model allows implementations of specific layers to be changed without having to replace the entire stack.

A mobile node typically has two points of attachment to a wireless network: A Base Station (BS) and an Access Router (AR). A base station is a link-layer device that provides connectivity between wireless hosts and the wired network.

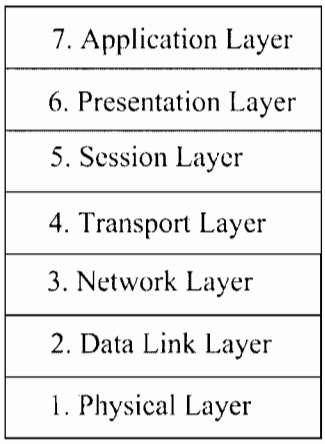

Figure 2.1: OSI Network Model.

Depending on the nature and purpose of the wireless network, base stations are usually strategically positioned to maximise the total coverage area. Wireless networks have a number of fundamentally different characteristics than wired networks. viz.,

- **low bandwidth** : Due to the limits on the physical layer design and transmission medium.
- **high link error rate** : This is a result of the nature of the transmission medium. Attenuation of the signal-noise ratio (SNR) with distance, inter-symbol interference, the Doppler shift and multipath fading are common causes of bit errors in wireless links.
- **mobility of end hosts resulting in handovers** : Besides blackouts whereby the mobile host loses connectivity with a base station, a handover to a new base station is a process that may cause a perceptible interruption in the host's connection causing a number of packets to be dropped.

An access router is the edge router in the wireless IP access network that provides routing services for one or more base stations. Figure 2.2 illustrates a simple reference topology of two separate IP subnets. Each subnet consists of one access router serving two base stations. A mobile node traversing path 1 is forced to disconnect from BS_1 and connect to BS_2 (using link specific methods). This

process is called a layer 2 or link-layer handover and does not involve AR_1 or any IP signalling. As the mobile node traverses path 2 it undergoes a link-layer handover from BS_2 to BS_3. Since it is now attached to a different subnet it must perform a network-layer handover from AR_1 to AR_2. A network-layer handover is the process of negotiating a new IP address from the new AR and adjusting the routing of all of its active packet flows to its new position.

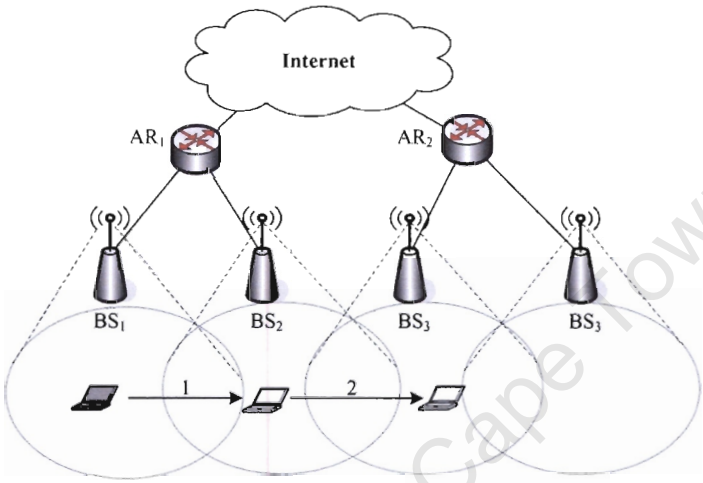

Figure 2.2: Intra-subnet and inter-subnet mobility.

Figure 2.2 also illustrates the paradigm of fixed-mobile convergence. Here, the technology of wired networks meets the wireless network at the edge of the Internet. Comparing wired and wireless systems, it is clear that they have very different characteristics. Traditionally wired protocol, services and applications now have to handle link errors, bandwidth bottleneck, and also have to support mobility on a number of layers of the network stack. Application layer mobility for example, may be managed by the Session Initiation Protocol (SIP). The focus of this work however only involves aspects of link-layer mobility and network-layer mobility.

2.2 Link-layer Mobility

Link-layer mobility is presented in a number wireless technologies and standards. These technologies vary in terms of cost, bandwidth, radio range and supported

services. They can also be divided into two logical groups: Cellular and Non-Cellular access technologies.

2.2.1 Cellular access technologies

The most common example of link-layer mobility present today is perhaps the GSM cellular network. GSM is the most frequently used cellular telephony standard in the world, with over 670 GSM mobile operators serving in more than 200 countries [6]. Mobile phones are served by a network of base stations. All link-layer connections and hence handovers are network initiated and controlled in order to manage network resources. With additional infrastructure, a new General Packet Radio Service (GPRS) was introduced on top of GSM in the late 1990s. GPRS offers packet-switched wireless data services for the mobile user at a data rate up to approx 40kbps.

3G wireless networking standards such as UMTS (3GPP) and CDMA2000 (3GPP2) are CDMA based allowing an end user to simultaneously communicate with old and new base stations, permitting soft handovers to take place. An important property of soft handovers is a handover latency of zero.

2.2.2 Non-Cellular access technologies

In 1990, the Institute of Electrical and Electronic Engineers (IEEE) established a committee with the main goal of developing a standard for wireless LANs [7]. To this end, the IEEE 802.11 or Wireless LAN (WLAN) standard was approved in 1997, defining two different data rates: 1 and 2Mbps, operating at 2.4GHz. The evolution of data rates continued with the release of additional standards in the 802.11 family: 802.11a/b/g offering up to 54Mbps, while 802.11n offers over 100Mbps.

WLAN technology enables users to establish wireless connections within a restricted area, and also replaces the need for physical cabling in networks. Thus the main use for WLAN is providing portable computing devices with wireless access to corporate or domestic LANs, including University campuses, airports and retail hotspots. WLAN has quickly become the most popular wireless LAN standards today.

The 802.16 (WiMax [8]) family offers broadband wireless access covering distances of up to 8km, while 802.15 (Zigbee, UWB) offers low data rate, low power wireless personal area network access to mobile computing devices.

As can be seen, there are a number of link layer standards available, each with differing wireless properties. In handover scenarios, link layers often have to perform lengthy scanning, association and authentication procedures depending on the type of access network. Link layer handovers can range from tens to hundreds of milliseconds, even higher when considering inter-technology or vertical handovers. Therefore, when combined with a network-layer handover (in the case of a subnet change), this latency effects the overall period of service disruption experienced by the mobile user.

The issue of detecting an actual link change or "movement detection" is the main focus of the IETF DNA[1] working group. Traditional techniques rely on network-layer indications such as a change in the advertised router prefixes. Generally, the reliance on such indications does not yield rapid detection, since these indications are not readily available upon a node changing its point of attachment. Mobile IPv6 Router Advertisements are typically broadcast with periodic intervals between 50ms and 5 seconds. This means that a mobile node is only able to detect a subnet change once it receives a new advertisements on its link. Furthermore, the bandwidth consumed by the frequent broadcast of these advertisements can be substantial for multiple access routers. Figure 2.3 illustrates the wasted bandwidth for a single access router. It is observed that the resources wasted at low broadcast rates are negligible but grow exponentially for shorter intervals. Most important issue here is the advertisement interval itself, and how it hinders the performance of fast handover schemes.

[1] "Detecting Network Attachment"

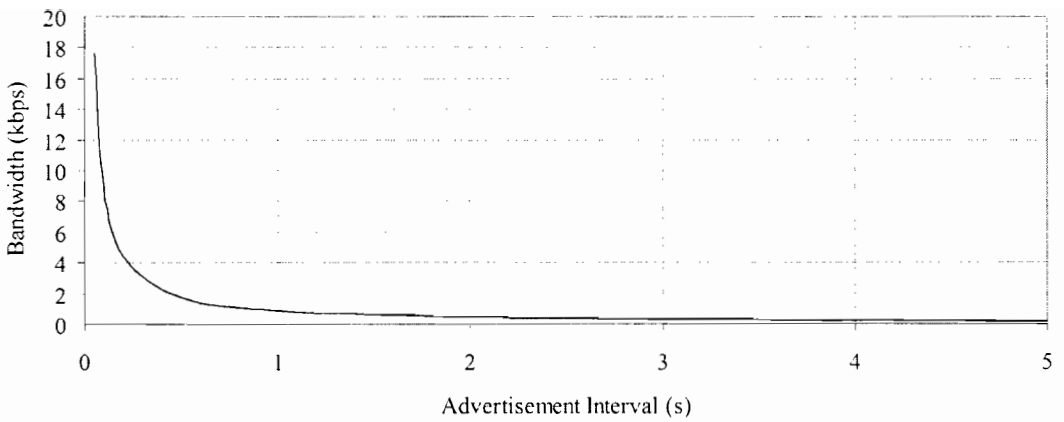

Figure 2.3: IPv6 Router advertisement bandwidth usage for different broadcast intervals.

IP mobility protocols rely heavily on timeous movement detection to maintain a suitable QoS level.

2.2.3 Link-layer triggers

This section first introduces the concept of link-layer triggers and then reviews the details of their use in mobile communication systems and protocols.

Link-layer trigger: A link-layer trigger is defined as an abstraction of a notification from the link-layer that a certain event has happened, or is about to happen.

Changes to the underlying link-layer connection status can be relayed to IP in the form of triggers. Such triggers provide information about certain events which can help the network and higher layers better streamline their handover related activities [9]. These "hints" or triggers are a cross layer form of communication that have been quickly adopted as a means for fast movement detection instead of the router advertisement method.

Although the use of cross layer communication of this nature seems relatively new, the Hayes AT command set is one example that has been in use for many years. AT commands are used extensively in mobile telephony systems such as

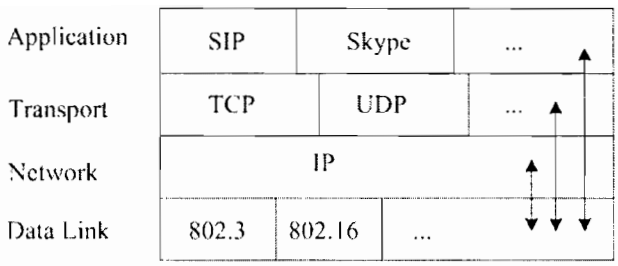

Figure 2.4: Link layer triggers in the OSI model.

GSM / GPRS / Bluetooth, enabling the management and control of radio or modem capabilities by an upper layer application or host. The AT command set is a standardised protocol built into these systems, where as link-layer triggers are not defined nor standardised into a protocol or API. Although seemingly similar, link-layer triggers are still in their infancy in terms of unification and use, and are a very popular topic in the literature called cross-layer design. A number of working groups have released preliminary technical drafts outlining similar link-layer trigger models. Before these models are discussed, it would be important to describe the basic underlying link-layer requirements for optimal performance, as generalised by the IETF in [2]:

- **Link Up (LU) trigger**: This is an indication that a new link-layer connection has been established and IP may send and receive packets.
- **Link Down (LD) trigger**: A link down trigger is an indication that the current wireless link has been disconnected. Both the LU and LD triggers replace the need for conventional movement detection mechanisms. Normally, triggers are delivered to co-located upper layers via an internal mechanism or interface. For devices operating separately, such as a base station and an access router, a transport mechanism is needed to convey L2 trigger information between the two nodes. The Link-Layer Trigger Protocol [10] is one example. It is a UDP based client-server protocol to transport L2 event information from access devices/points to other IP nodes such as mobile hosts and access routers.
- **Link Pre-Trigger**: This trigger indicates the completion of a scan[2] pro-

[2]"scan" refers to a link-specific method of discovering a new access point or base station.

cess to discover any available access points. The execution of such a scan is usually at the instant at which the current AP's signal-to-noise ratio (SNR) falls below a certain threshold level. For a device travelling at constant speed, the point at which a link change will occur is deterministic. However, radio signal fluctuations and interferences are variables in this theory and may cause timing problems, affecting the performance and execution of the dependent protocol. In 802.11 networks for example, a pre-trigger (see Figure 2.5) would fire at time T_1 when $SNR(AP_1)$ crossed a threshold value SNR_{thresh}. A link-layer switch would occur at approximately T_2 where the difference between the two SNR values is large enough, or $SNR(AP_2) \geqslant SNR(AP_1) + \triangle$. The resulting period T_2-T_1 is therefore subject to variations in the node's velocity and possible signal fading or interference.

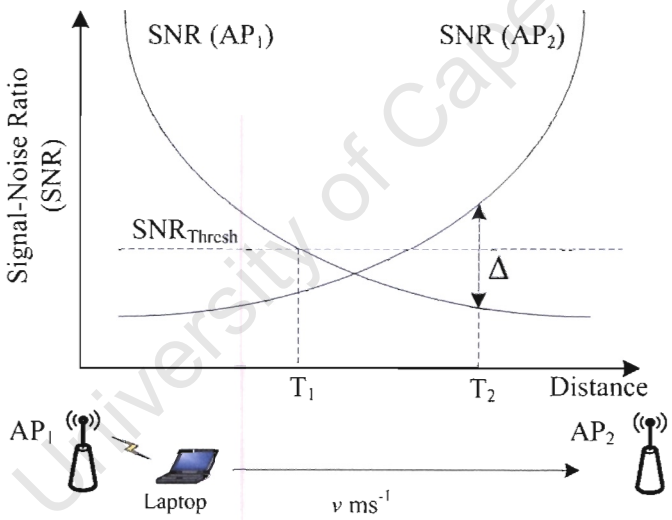

Figure 2.5: Illustration of SNR change as node moves between two IEEE 802.11 access points.

The IEEE 802.21 working group released a proposal for a generalised trigger model aimed at supporting link-layer triggers in IEEE 802 networks. This model defines a set of events and commands that upper-layer mobility protocols can read or issue to synchronise their handover related activities with the link layer. These generic triggers include: Link Up, Link Down, Link Going Down, Link Going Up, Link Quality crossing threshold, Trigger Rollback and Better Signal

Quality AP Available [9]. A full description of this model is available in Appendix A.

The IETF MIPSHOP working group has released informational RFCs describing how to implement FMIPv6 in 802.11 [3], 802.16e [11] and CDMA (3G) [12] access technologies. This is an important step in the deployment of FMIPv6 in to areas that standard Mobile IPv6 is not suitable for.

The IEEE 802.21 working group also published a draft for a Media Independent Handover (MIH) service to support seamless handovers between heterogeneous networks [13]. This document describes an architecture which specifies a layer 2.5 to reside above the link-layer in the OSI protocol stack. L2.5 aims to provide an additional link-layer abstraction to support triggers from any type of link-layer. The MIH service would significantly simplify handovers between IEEE 802 type networks and cellular 3GPP/3GPP2 networks. Similar work is presented by Gollum et al. [14] proposes a unified API called a Universal Link Layer API (ULLA) to solve the complexities related to the interoperability of multiple wireless interfaces and standards. ULLA however is still in the research phase.

2.3 Network-layer Mobility

In this section, the basic operation of Mobile IPv6 is introduced and the factors influencing its performance are discussed. Thereafter, Mobile IPv6 extensions and other significant network-layer mobility schemes proposed in the literature are introduced and reviewed.

2.3.1 Mobile IPv6

Mobile IPv6, the successor to Mobile IPv4, is a network-layer mobility management protocol which allows nodes to remain reachable while migrating through different IP subnets in the Internet. It is designed to operate alongside existing network layer entities and be fully compatible with existing IP end-systems and routers, requiring no mobility enhancements to protocol stacks. Mobility support in IPv6 is particularly important, as mobile computers are likely to account for the majority of the population of the Internet during the lifetime of IPv6.

The central element that supports IP mobility in this protocol is the allocation of more than one IP address to a mobile node. A MN is assigned a primary IP address called a home address, which is essentially the MN's identity on the network. It is also used to define upper layer connections such as TCP. When a node moves to a foreign network, it is allocated a second globally-routable IP address, called a care-of-address (CoA) which is valid on the visited network. The CoA represents the MN's current network point of attachment and reflects its topological position in the foreign network. The CoA is thus a transient address, changing as the MN traverses through different foreign networks. The MN can acquire its care-of address through conventional IPv6 mechanisms, such as stateless or stateful auto-configuration. Stateless address auto-configuration involves the MN appending its 64-bit interface identifier to the foreign link's 64-bit prefix. Duplicate address detection (DAD) is performed by the AR in the new network during the auto-configuration process to verify the uniqueness of the CoA. DAD however, may take up to 1 second to complete without suitable optimisations.

As long as the MN stays in its current location, packets addressed to its care-of address will be routed to the MN. The association between a MN's home address and care-of address is known as a binding. While away from home, a node registers its primary care-of address with a router or home agent (HA) on its home link. This process is called home registration. The MN performs this registration by sending a Binding Update (BU) message to the home agent. The home agent thereafter uses proxy Neighbour Discovery to intercept any IPv6 packets addressed to the mobile node's home address on the home link, and tunnels each intercepted packet to the mobile node's primary CoA. The round trip time (RTT) between the MN and its home agent accounts for a substantial part of the overall Mobile IPv6 handover latency. Home registration takes at least 2 x RTT to complete.

Thus the particular area of focus of this section is when a MN moves to another subnet, causing a Mobile IP handover to occur. The ability to immediately send packets from a new subnet link depends on the overall handover latency summarised by the following equation:

$$t_{handover} = t_{L2} + t_{localIP} + t_{BU} \tag{2.1}$$

Here, the handover latency comprises of t_{L2} the link switching delay, $t_{localIP}$ the local IP address reconfiguration and movement detection delay, and t_{BU} the total binding update delay. The following four performance metrics pertaining to a handover are of interest here:

- Handover latency: This is usually defined as the total period of service disruption during a handover.
- Packet loss: The total number of packets dropped or mis-routed during a handover. This metric is often proportional to the handover latency and packet arrival rate.
- Jitter: This is the variance in arrival time of the received packet stream.
- Overhead: The signalling used in a protocol and the routing of additional packets both contribute to bandwidth overheads and cost.

The standard Mobile IPv6 protocol has been described as not suitable for real-time protocols [2, 3, 15, 16]. Analytical studies of the MIPv6 handover by Schmidt [17] and by Costa [18] reveal that one of the major factors influencing the overall latency is the binding update procedure. As previously mentioned, the delay of such registration procedures is proportional to the RTT between the home and foreign networks.

Real-time protocols and applications require stringent QoS parameters. One such parameter for VoIP is a handover latency (or maximum break in the packet stream) of less than 150ms, and packet jitter (delay variation) of less than 50ms. Transport protocols such as TCP, on the other hand, are more affected by packet loss. The literature is full of Mobile IP extensions and optimisations aimed at improving different areas affecting the performance of Mobile IP in a number of environments. The most common and notably important areas of interest remain the design of a fast or seamless handover scheme.

2.3.2 Fast Handovers for Mobile IPv6

Koodli outlines the Fast Handovers for Mobile IPv6 (FMIPv6) protocol in an IETF Internet Draft [2]. FMIPv6 optimises the standard Mobile IPv6 signalling

and incorporates link-layer triggering to improve handover performance. The protocol is defined for two modes of operation depending on the type of handover scenario the mobile node is involved in, namely: a "predictive[3] handover" and a "reactive handover" scenario. This section describes the intricate protocol operation for each scenario.

Proactive Handovers:

The fast handover procedure consists of 3 main phases: handover initiation, tunnel establishment and packet forwarding. For a proactive handover to occur, FMIPv6 states that a link-specific event needs to notify the network layer that a link-layer handover is imminent. As discussed before, a pre-trigger usually occurs when the current link's SNR has crossed a certain threshold, prompting the underlying layer to scan for an access point with a better signal strength. The parameters contained in the trigger include the details of the newly discovered access point and access router. This trigger is delivered to the network-layer a period of time $t_{proactive}$ (see figure 2.6) before the link switch. This marks the beginning of the handover initiation phase whereby the MN transmits a Router Solicitation for Proxy (RtSolPr) message to the previous access router (pAR) indicating that it wishes to perform a fast handover to a new access point. The pAR then resolves the next access router's (nAR's) corresponding IP address from the link layer address contained in RtSolPr, and returns it together with any control information in a Proxy Router Advertisement (PrRtAdv) message. At this point, the MN constructs a new CoA (NCoA) out of its interface identifier and the nAR's subnet prefix. It subsequently uses this NCoA to send a Fast Binding Update (FBU) to the pAR. A Fast Binding Acknowledgement (FBack) is returned to the mobile node on both the pAR's and nAR's interface to indicate a successful binding.

The tunnelling phase begins with the transmission of a Handover Initiate (HI) message that informs the nAR of the MN's intent to handover, and establishes a bi-directional tunnel between the two routers. The Handover Acknowledge (Hack) message indicates that the nAR accepts the NCoA. The reception of Hack also initiates the packet forwarding phase and hence the forwarding of the

[3]referred to as "proactive" to eliminate any confusion related to mobility prediction

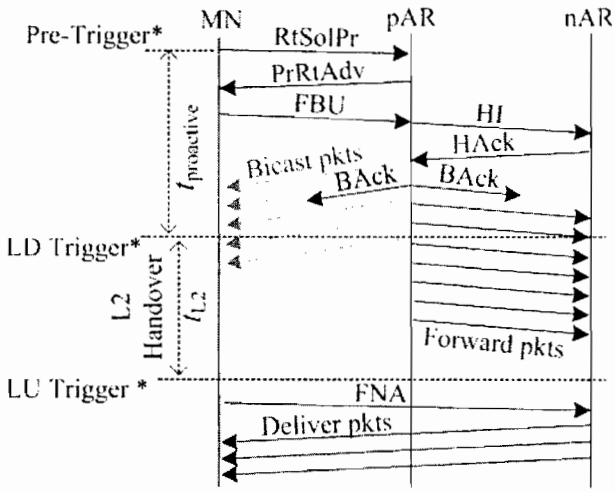

Figure 2.6: Proactive FMIPv6 handover timing diagram with bicasting.

MN's data through the tunnel to the nAR. Packet forwarding allows the MN to continue to receive real-time services while it performs post-handover home and corresponding node registrations. It also smooths out the disruption caused by the link-switching delay.

The timing of the pre-trigger is extremely important: If the pre-trigger occurs too soon, the proactive signalling would have completed and the forwarding of data would continue for longer than necessary, during which time the MN is unable to receive packets on its current link. Too late and the link-layer handover occurs unpronounced, preventing the FMIP signalling from completing. In this case all of the MN's packets would get dropped, and the node would have to fall back to a reactive handover. In practice, configuring this trigger timing is very difficult since the actual link-layer handover decision is controlled internally by the specific firmware.

FMIPv6 recommends the use of Link-Up and Link-Down triggers when available instead of the traditional router advertisement method. Immediately after the completion of a link-layer change, as indicated by a link-up trigger, the MN sends a Fast Neighbour Advertisement (FNA) to announce its presence to the nAR. This prompts the nAR to deliver any buffered or incoming packets to the MN. The advantage of this is that the MN is now able to receive packets which have been forwarded by pAR at sometime before or during the link-layer handover.

Reactive Handovers:

Reactive handovers differ from proactive handovers, as the FBU message is transmitted after the link change has occurred. Once the FBU is received by the nAR, it is forwarded to the pAR. The FBU/FBack messages establish the bi-directional tunnel used to forward packets from the pAR.

This scenario is inherently subject to an increase in packet loss because of the unforeseen link-layer handover. A study by Schmidt [17] compares the performance of reactive versus proactive FMIPv6 handovers. The study highlighted the relationship between packet loss and the distance between pAR and nAR. It was found that the packet loss rates of proactive handovers was lower than the reactive case only if that distance (measured by transmission delay) was greater than 13ms.

The buffering of incoming data packets at the pAR has been suggested to minimise packet loss. However, the implementation and configuration of buffers is a complicated issue which has an impact on jitter. For simplicity, buffers are kept out the the scope of this study.

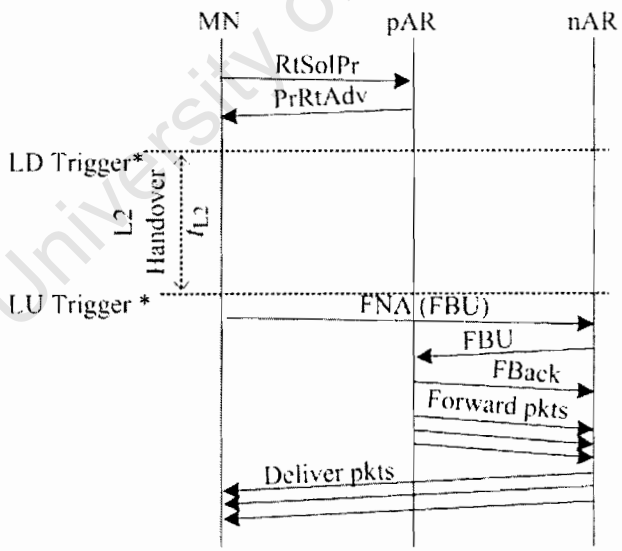

Figure 2.7: Reactive FMIPv6 handover timing diagram.

A number of factors may force a mobile node into a reactive handover scenario:

- A pre-trigger is unavailable in the node's current hardware implementation.

- The MN has moved to an AR that is different to the proposed one. This may occur if the pre-handover scanning procedure is inaccurate, incomplete or has returned multiple target access point link layer addresses. It is possible that the MN has aborted the change altogether, however this may be solved at the link-layer itself.
- The pAR is unable to resolve the target nAR's identity in a timely manner.
- In the case of very frequent cell hopping, the node may not have sufficient time to execute a pre-trigger or the signalling that must occur before a link layer handover. This is common for nodes moving at high velocities such as vehicular ad-hoc networks, or mobile nodes moving in densely populated wireless overlay networks. Irregular / erroneous movement such as "ping-pong" movement must also be considered.

Reactive handovers is the default fall-back mode of operation for proactive FMIPv6. Given the number of factors that can cause a fall-back to occur, it seems only intuitive that it should performs adequately enough. Perkins and Koodli [19] investigated the "context transfer" problem in FMIPv6 for both reactive and proactive scenarios. Context transfer / state relocation is a concept that has risen from research on mobility management, with the objective of complementing the handover procedure. As explained in [19], the transfer of context is actually managed by a context transfer protocol (such as CXTP), although part of the signalling or request for context may be included in HI / FBU messages. Additional signalling also means additional round trip delays. Perkins and Koodli explored the problem of trying to achieve a seamless handover of a packetised VoIP stream that requires header compression. The size of the IPv6/UDP/RTP header with Mobile IPv6 home address option was approximately 150 bytes. It was estimated that this header compression context re-establishment would take approximately 400ms in a cellular type IP network. An alternative to this was proposed that involves the transfer of current context information from the pAR to the nAR during the handover. This alternative was tested for proactive and reactive handover scenarios.

It was found that the proactive handover latency (with context transfer) was less than 100ms, perfectly suitable for supporting real-time applications. The link-switching delay in fact accounted for approximately 90% of this time, meaning

that the overhead incurred by IP layer signalling overhead is very small. The reactive handover and context transfer, on the other hand, achieved a latency of approximately 77ms, including a 66.4ms link-switching delay. The context transfer only accounts for only 1.1ms, this is a 10.3% contribution to the IP signalling overhead.

The main objective of these experiments was to investigate the effect of context transfer on the handover latency, thus the effects of packet loss and jitter were ignored. It can be concluded that context transfer within FMIPv6 is indeed useful, especially for high data-rate multimedia applications.

Context packets that included other types of context such as QoS policies, AAA profiles and security (encryption keys and IPSec state) information would certainly be larger in size, require more round trips times in terms of protocol interactions at the nAR or home networks, and thus further impact the performance of fast handover schemes. Pagtiz et al. [20] argues that IP mobility management cannot rely on the reactivity of the upcoming visited network when real-time packet flows need to be guaranteed. Instead, they propose a scheme that supports the proactive management and distribution of a MN's IP connectivity state well in advance of its handover transition.

2.3.3 Simultaneous Bindings

Simultaneous Bindings for Mobile IPv6 Fast Handovers addresses the "timing ambiguity" problem [4]. In many wireless networks it is impossible to know in advance exactly when a mobile node will detach from its current wireless link. It is therefore not simple to determine the correct time to start forwarding between pAR and nAR. This timing ambiguity has an impact on how smooth the handover will be [4]. Simultaneous bindings proposes to mitigate this problem by bicasting or n-casting packets destined for the MN from the pAR to one or more potential nARs before the MN actually moves there. Choosing multiple destinations also compensates for a link-layer prediction error. Referring to Figure 2.6, one copy of the packets is sent to the MNs previous on-link CoA, and another copy to the NCoA. The MN is thus able to receive traffic independently of the exact link-layer handover timing during the handover period.

Although this method drastically reduces packet loss and improves the overall smoothness of the handover, studies have shown that the n-casting of packet streams increases the handover latency [15]. Another issue is the increased probability of receiving duplicated packets. This is generally only a problem in TCP, as some TCP congestion avoidance schemes react negatively to the reception of 2 or 3 duplicate acknowledgements. Wireless TCP implementations such as TCP Eifel [21] and TCP Jersey [22] address this problem as well as the losses caused by noisy or congested wireless links.

Further investigation on the effects of TCP-handover behaviour is pursued in later chapters.

2.3.4 Seamless Handover

Hsieh et al. [15] performs a comparison of five current fast handover techniques, including an original Seamless handover (S-MIP) proposal. He evaluates the performance of each handover[4] scheme through simulation, and discusses their impact on end-to-end TCP applications. S-MIP shows the best results, however it requires a network entity called a Decision Engine to determine when and how the MN is to handover, depending on network conditions and movement patterns. These results highlight the importance of a seamless handover for TCP applications. Ignoring wireless / lossy link effects, a more seamless handover significantly improves the throughput of a file download session.

2.3.5 NeighborCasting

Shim et al. [16] proposes NeighborCasting, a pre-trigger MIPv4 based low latency handover scheme. This scheme is similar to FMIPv6 however it does not assume that the link-layer technology can determine or predict the next AR. It therefore multicasts the MNs incoming data streams to all of its neighbouring ARs during a handover. Performance results indeed show a low latency handover but with significant overhead due to the multicasting.

This "over-provisioning" approach may not be affordable in expensive or condensed networks. The trade-off between handover performance and the cost of

[4]Hsieh et al. only evaluates proactive handovers

network resources is evident in most seamless handover schemes due to their packet forwarding or multicasting. Mobility prediction would benefit NeighborCasting by reducing the set of possible handover targets and thus the improve the network load. NeighborCasting is used for comparison in later chapters.

2.4 Mobility Prediction

Previous sections have discussed both link-layer and network-layer mobility issues. This section introduces and reviews the literature pertaining to mobility prediction techniques. The different predictors are investigated and their uses in mobile communication systems are discussed.

Mobility prediction[5] effectively facilitates pro-activity in mobile communications, allowing expensive operations to be performed prior to a link / subnet change. This has benefits for resource reservation, call admission control, network preconfiguration, and security and header compression context transfer. A number of prediction techniques have been proposed in the literature. The most significant of these involve an analysis of the mobile user's prior movements, locations, or trajectories, or apply stochastic and topological information to a mobility model.

2.4.1 Pattern-matching

Pattern-matching techniques constitute an important class of mobility prediction. This approach requires the visited locations (and often handover times) of mobile users to be recorded. This information is readily available to every network operator that supports roaming, however the mobile user itself may also record its own movement.

Lui and Maguire [23] are pioneers in this field through their Mobile Motion Prediction (MMP) algorithms used to predict future locations of a mobile user according to the user's movement history patterns. This was one of the first of many techniques in the literature used to proactively connect services at the new location before the user's arrival. The MMP algorithms are based on the fact that human movement generally consists of regular and random movement. These algorithms

[5]also referred to as "location prediction" in the literature

use correlation analysis to match movement sequences in a movement database. The problem facing pattern-matching algorithms is that random behaviour, or new movement segments, increase the probability of a prediction miss. Results of this approach show that the MMP algorithm is highly accurate for regular movements but decreases linearly with increasing random movement.

Song et. al [24] investigate two families of predictors, Order-k Markov predictors and LZ-based predictors. The accuracy of each of the location predictors is evaluated using a large set of real mobility data. This mobility data comprises of the sequence of Wi-Fi access points frequented by more than 6000 users over a period of two years. It was found that the simple low-order Markov predictors worked well or better than the more complex compression-based (LZ) predictors, and better than the high-order Markov predictors. The $O(2)$ Markov predictor obtained a median accuracy of about 72% for users with trace lengths of more than 1000 movements (cell crossings). It is also acknowledged by the authors that this result is based on the observations of over 2000 users and that for an individual user the outcome may be quite different.

Yavas [25] proposes a novel *data mining* approach for the prediction of user movements in mobile environments. It involves a three stage prediction algorithm based on the Apriori algorithm [26, 27] and a web-prefetching algorithm by Nanopoulos in [28, 29], to predict the mobility of a user travelling between the cells of a PCS (Personal Communication System) network. Simulation results reveal the optimal prediction parameters for the PCS topology. A moderate prediction accuracy was achieved, decreasing only minimally with an increase in random movement. The authors focus primarily on prediction recall and precision results, and make no practical use of the movement predictions.

This approach by Yavas forms the basis for the proposal in this study. Data mining, the Apriori algorithm and its associated components will be discussed in detail in Chapter 3.

2.4.2 Mobility tracking

A more complex approach to mobility prediction involves recording the position and velocity of each user. This information can be obtained by real-time signal

triangulation techniques or client-carried GPS devices. Levine et al. [30], for example, use the shadow cluster concept. Depending on the velocity of the user, its current base station determines a set of neighbouring cells that can be visited by the mobile user. The past residence history, call duration statistics, and direction of the user are all used to estimate handover probability at a future time from the current cell, for purposes of resource reservation and admission control. Similarly, Aljadha et al. [31] uses a direction-estimation based method to form a most likely cluster (MLC) of future cells. Shen et al. [32] uses a Recursive Least Square algorithm to predict the next cell using location inference obtained by fuzzy logic. Besides the disadvantage of being reliant on physical hardware, the accuracy of mobility prediction in all these models depends on how well the position, velocity, and acceleration are estimated.

2.4.3 High level mapping

High-level topological information derived from road and building maps that describe the distinguishing, designating or limiting properties of each cell, is used in many prediction schemes. Soh et al. [33] argues that road maps, which show road segments and their intersections, can help estimate the path of a caller in a vehicle and thus predict the time and place of the next handover. Rather than modelling the transitions between cells, they model the transitions between road segments. The road topology, positioning information and precise velocity estimates allow this scheme to predict the next most likely handover occurrences and efficiently reserve resources at the target base stations.

Mishra et al.[34] describes a novel data structure called Neighbor Graphs which dynamically captures the topology of the wireless network as a means for pre-positioning the mobile node's context information at the next (neighbouring) access points. This proactive context-caching scheme was shown to significantly improve the layer-2 handover latency in an 802.11 testbed by minimising the re-association delay. Pack et al. [35] suggests a similar scheme called Selective Neighbor Caching (SNC) also for 802.11 networks. SNC proactively propagates a mobile node's context to only selected neighbouring APs whose handover probabilities are higher than a threshold value. This threshold value is set to minimise the overhead due to the context transfer, but also influences the cache-hit prob-

ability.

2.5 Mobility Prediction Based Fast Handover Approaches

Several authors have specifically used mobility predictors to improve the network-layer handover performance in wireless networks.

Feng et. al. [36] proposes a prediction scheme based on the MMP algorithms [23] that uses actual movement traces taken from the campus-wide 802.11b wireless network. Data streams are duplicated and forwarded to predicted subnets resulting in a network-layer handover latency that is close to a link-layer handover. Results show a reduced handover latency and packet loss rate compared to NeighborCasting.

Van den Wijngaert et al. [37] proposes a prediction mechanism that learns the mobility patterns of a mobile node according to an urban mobility model. The model attempts to capture realistic node movement in an urban environment, characterised by the MN's speed, direction, pause time and street coordinates. A weighted road selection process uses these parameters to predict the node's next hop, pre-emptively setting up tunnels and estimating tunnel activation times, consequently eliminating the need for a pre-trigger. This approach achieves 100% prediction accuracy only after 3000 seconds of movement over a small Manhattan-style street topology.

Mobility models play an important role in evaluating mobility prediction algorithms as they define how a mobile entity physically moves over a topology. For example, a Random Waypoint Model will force a node to continuously choose random coordinates and travel (linearly) to them at a certain speed, while a Restricted Random Waypoint Model [38] will restrict the random movement of the node to defined boundaries, say within the streets of a street map.

2.5.1 Discussion

In this chapter, work has been identified that is closely related to the topic of this research. It is clear that improving the performance of the handover procedure

is a current topic in the literature. The use of link-layer notifications in this regard has also been discussed; however, the difficulties of interoperability and implementation given the diversity of access technologies available, have also been highlighted. Standardising or unifying cross-layer communication has been suggested as a means to deprecate this issue.

Mobility prediction is a broad field. In the context of mobile networks, it has certainly proven its applicability. Relying on external hardware to measure speed, direction or even signal strengths to predict movement may be effective, but is not always feasible. Using a node's own mobility history to determine its next hop is far more attractive. A number of prediction algorithms have taken this approach, using different techniques to determine regular patterns in the data. A data mining approach in the literature has shown to be remarkably simple compared to other schemes [25], and performs well even with an increase in random movement. The next chapter discusses in more detail the use of a data mining approach as part of a prediction algorithm. Thereafter, the design of a fast handover protocol that incorporates this prediction algorithm is detailed.

Chapter 3

Mobility Prediction Assisted Fast Handover Design

3.1 Introduction

Chapter 1 discussed the benefits of mobility prediction to mobile communication networks. Chapter 2 reviewed the literature related to layer-2 and layer-3 mobility and found a number of fast handover proposals that incorporated mobility prediction to improve their system performance. The mobility prediction techniques themselves varied in their approach, complexity and accuracy. A simple and hardware-free approach involved detecting regular patterns from a mobile user's movement history.

The work presented in this thesis essentially comprises of two modular units, a data mining based approach to mobility prediction algorithm, and a fast handover scheme. Together, these two units form the proposal that is central to this thesis, i.e. a Prediction Assisted Fast Handovers for Mobile IP (PA-FMIP). Later chapters thoroughly evaluate the performance of this proposal. This chapter begins by detailing the specific aspects of data mining which form the basis of the mobility prediction algorithm. It is important that these details be explained so that the functionality of the prediction algorithm is clearly understood.

3.2 Data Mining based Mobility Prediction

3.2.1 Data Mining

Data Mining can be defined as the process of analysing data to identify patterns, associations, significant structures or relationships from information stored in a data repository or database [25].

Association Rule Mining (ARM) [26] is one of the most important and well researched techniques of data mining. It aims to extract interesting correlations, frequent patterns or associations among sets of items in transactional databases. ARM can be divided into two phases. In the first phase, all frequent itemsets are mined from the given data. The second phase consists of the generation of all frequent and confident association rules from the result of the first phase [39].

Sequential pattern mining (SPM) [40], a special subset of Frequent Itemset Mining (FIM), is the process of extracting sequential patterns from a transactional database or dataset [41]. In SPM, the order of items in a database is always considered and used to determine the most frequently occurring sequential patterns. This type of data mining is used in business to study customer behaviour, in telecommunication networks to analyse system performance, stock market trend analysis and even in DNA sequencing. For the purposes of this mobility prediction, SPM is used instead of FIM since we are interested in the regularity of sequences within a user's mobility history.

A common example of ARM is found on Amazon.com. While browsing for a specific product, the site often displays information similar to "customers who bought this product also bought ..." This association makes no use of which product was bought first, however it implies that these products were bought during the same transaction (session). A good example of sequential pattern mining is found in a web pre-fetching algorithm by Nanopoulos [28, 29]. To effectively predict the users' future web requests, users' access patterns are mined from the web logs of previous requests. These patterns are used for the prefetching of web documents.

Most algorithms used for sequential pattern and association rule mining are all variations based on the Apriori Algorithm [26]. Each algorithm attempts to reduce the number of times the database is scanned, and explore different candidate

pruning techniques so as to minimise computational time, space and memory [39]. The Apriori algorithm by Bodon [40], is a level wise algorithm and makes multiple passes over the data to discover large (frequent in terms of *support*) sequences. Support (introduced by Agrawal [26]) is defined on itemsets / sequences and gives the proportion that they are contained in the data. It is used as a measure of significance of an itemset. An itemset with a support greater than a set minimum support threshold is called a frequent or large itemset. Supports main feature is that it possesses the down-ward closure property (antimonotonicity) which means that all subsets of a frequent set are also frequent. The fact that no superset of a infrequent set can be frequent, prunes the search space in level-wise algorithms.

In the Apriori algorithm the sequences (called candidates) range from LENGTH(1) to LENGTH(k). Supports for these candidates are counted at each pass of the algorithm. The largest candidates of LENGTH(k-1) are then used to determine the candidate set for LENGTH(k) during the next pass. This process repeats until no new large sequences are found.

3.2.2 The Algorithm

Before the design of the algorithm is discussed, it would be important to re-address some of the challenges facing mobility prediction. The objective of any prediction scheme is to achieve perfect accuracy. Hardware-based mobility tracking methods monitor the physical trajectory and behaviour of a mobile node while history-based methods only record the logical network transitions. In the latter case, the process of determining the next-hop location of a node may only be achieved with the record of its previous locations leading up to its current one.

Radio properties introduce irregularities into a users mobility path. With the effects of cell layout, cell bouncing and signal fading, a seemingly regular movement trajectory is observed to be irregular or random. Referring to Figure 3.1, the sequence of network cell changes is not as expected. This "noise" in the mobility history is overcome by data mining based predictors as they do not rely on exact sequence matching, but rather frequent sequences. Thus, as will be explained in more detail later, random entries in a node's mobility history are essentially filtered out. This is a significant proponent of these types of schemes. Mobility tracking schemes, such as the shadow cluster technique [30] , are at the mercy of

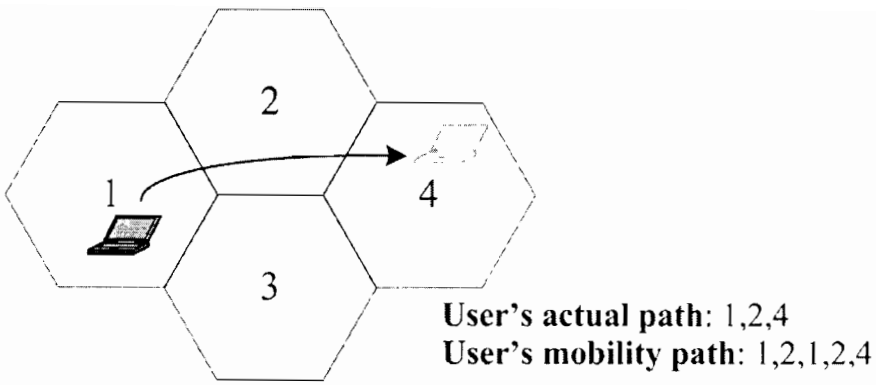

Figure 3.1: An illustration of a users actual path differing from the network mobility path.

physical terrain and the radio characteristics. As explained in Section 2.4.2, the shadow cluster compensates for erroneous predictions by selecting a set of surrounding cells. Tracking and predicting the logical network-layer mobility rather than the actual mobility path may therefore be simpler and more effective. This approach also supports mobility prediction in heterogeneous and multi-domain overlay networks.

The prediction algorithm begins by appending the identity *id* (say p) of the new access router (nAR) to a transactional database D following a successful handover. Consecutive *id*s in a transaction form a trajectory, and represent the movement between neighbouring cells in a network. The respective ARs may not necessarily be spatial neighbours or even routing neighbours; we describe them as handover neighbours. If the mobility history contains n entries, then D_n=p_1,p_2,p_3,...,p_{n-1},p_n. The last of the comma deliminated entries in the database, p_n, identifies the node's current position on the network (see Table 3.1). And for any $0 < i < n$, it also happens that $p_i \neq p_{i+1}$ since entries are only recorded following a successful handover to a new subnet. D is also parsed into transactions. These transactions represent mobility sessions. After a threshold period of T seconds of active mobility, a new transaction is started. Thus after sufficient time, D contains a number of mobility transactions of varying length.

The Apriori algorithm by Bodon [40] is applied to D with a min_{supp} value of say 10%. A set of the most frequent sequential patterns is outputted, ranked in descending order of support. These frequent sequential patterns, or itemsets,

range from LENGTH(1) to LENGTH(k), i.e. $C = <c_1, c_2, c_3, ..., c_{k-1}, c_k>$. The corresponding support value ($supp$) for each itemset here is greater than the minimum support threshold (min_{supp}) value of say 10%. This itemset-support tuple represents the sequence of handovers that the user has repeated a certain number of times. Thus any random items distributed within D are essentially filtered out as they do not earn enough support, granted the value of min_{supp} is not too low. A value of min_{supp} that is too high would result in too few itemsets to be found, increasing the possibility of a prediction-miss.

The next step in the prediction algorithm requires the generation of association rules from these sequential patterns. As the name says, association rule mining finds the association or relationship between a set of items in a transaction and forms a rule. The objective is to determine the probability of a single item in the itemset given a preceding sequence. The Apriori association rule mining algorithm is used here. For each transaction from the previous phase, a set of rules are generated. The term preceding the arrow is termed the *head* or *consequent*, while the term following the arrow is the *tail* or *antecedent*. The following set of rules are derived from the sequence $C = <c_1, c_2, c_3, ..., c_{k-1}, c_k>$

$$\begin{aligned} c_1 &=> c_2, c_3, ..., c_{k-1}, c_k \quad & conf_1 \\ c_1, c_2 &=> c_3, ..., c_{k-1}, c_k \quad & conf_2 \\ &... \\ c_1, c_2, c_3, ..., c_{k-1} &=> c_k \quad & conf_3 \end{aligned}$$

The result is that the ARM algorithm is a set of mobility rules [25]. Each rule is coupled with a confidence value ($conf$), which is the conditional probability of the term's support values [39]. The rules are then ranked in descending order of confidence. The minimum confidence of this set is limited by a pruning parameter min_{conf} which reduces the overall number of rules to be generated. The form of the mobility rules are generalised as R: $r_1, r_2, r_3, ..., r_{j-1}, r_j => r_{j+1}$ for $0 < j < k$-1. The number of items in the tail term is limited to one since we only wish to predict one hop into the future. The confidence of each rule R may be expressed by the conditional probability

$$P(r_{j+1} | r_1, r_2, r_3, ..., r_{j-1}, r_j) = \frac{support(r_1, r_2, r_3, ..., r_{j-1}, r_j \cup r_{j+1})}{support(r_1, r_2, r_3, ..., r_{j-1}, r_j)}$$

Table 3.1: a) The structure of a mobility history or database. b) This table shows the typical output of the sequential pattern mining algorithm. c) A list of mobility rules that are generated by the Apriori association rule mining algorithm. d) The rules relevant to the current position are filtered out. e) List of the final predictions made by the algorithm.

Database D_n
1,2,3,6,7
2,3
6,7,8,1
…
$..,p_{n-3},p_{n-2},p_{n-1},p_n$

a)

Frequent Patterns	*Supp.*
<2,3,6>	w
<6,7>	x
<1,2,3>	y
…	…
$<..,c_{k-2},c_{k-1},c_k>$	$>min_{supp}$

b)

Mobility Rules	*Conf.*
1,2=>3	95%
2,3=>6	92%
6,7,8=>9	91%
…	…
$..,r_{j-2},r_{j-1},r_j=>r_{j-1}$	$>min_{conf}$

c)

Rules matching $r_j = p_n$ for $p_n = 2$, all $r_{j-1} \in N$, $min_{conf} = 85\%$	*Conf.*
1,2=>3	95%
2=>4	91%
4,3,2=>1	90%
…	…
$..,r_{j-2},r_{j-1},r_j=>r_{j-1}$	>85%

d)

Algorithm Outputs for M=2	AR*id*
1st Prediction	3
2nd Prediction	4

e)

The final step in the algorithm searches the set of rules for items immediately before the arrow that match the condition $r_j{=}p_n$. These matching rules are related to the node's current position and are collected and ranked according to their confidence values. An online list of the current AR's handover neighbours is maintained by the MN through the use of the Candidate Access Router Discovery (CARD) [42] protocol.

Through a simple handover reporting mechanism, CARD facilitates network entities to build and maintain a list (topology) of neighbouring ARs and their associated base stations. In IEEE 802.11 networks for example, a MN may determine the AR-AP map for the wireless neighbourhood. CARD was originally developed to assist mobile hosts to choose the new access routers based on their capabilities and available resources. It is a lightweight protocol requiring the exchange of only two ICMP messages: AR-AR CARD Request and AR-AR CARD Reply. This is a useful tool for effecting seamless handovers, however it assumes the handover decision algorithm embedded within the MN is directly accessible. In this work, CARD is used to populate a set of all possible AR neighbours (N) surrounding the current AR.

In the set of matching mobility rules, any rule with $r_{j+1} \notin N$ is removed from the set. All that remains in the list are valid and confident mobility rules. The most confident r_{j+1} value in the set is essentially the next-hop prediction and algorithm output. In practice, it is possible that more than one rule may have identical confidence values. This is a factor that affects the algorithms prediction accuracy. A simple tie-breaker, or additional means to differentiate the similar predictions may help, although there is still a probability that the tie-breaker is not accurate at all [24]. This algorithm compensates for possible prediction error by introducing a user-defined parameter M, which is the number predictions that are outputted. Here, the M highest (most confident) r_{j+1} values are selected as the mostly likely next hop predictions. M may range from 1 to $\max(N)$ or the maximum number of valid neighbours. The user may set this value depending on a required prediction accuracy or QoS agreement.

The limitations of this prediction algorithm are that it requires an initial mobility history before any prediction can be made. This history also has to be related to the users current mobility scenario. That is, if a user is travelling from home to work, the algorithm will only be able to accurately predict subsequent network

attachment points if this route has been previously traversed. This issue is similar to the learning period that artificial intelligence algorithms require to make accurate decisions.

3.3 Prediction Assisted Fast Handovers

Handover latency is the primary cause of packet loss and performance degradation for mobile nodes, especially for end-to-end TCP [15]. The previous chapter discussed the differences between standard Mobile IPv6 and FMIPv6. It was shown that the (proactive and reactive) FMIPv6 handover latency was not affected by the round-trip-time (RTT) between the home and visited networks, as is the case for Mobile IPv6. It is this RTT delay in all home and CN registrations that hampers its performance in real-time applications. Proactive FMIPv6 offers promising results but requires careful setup and configuration of trigger timings. Reactive FMIPv6 on the other hand, is the fall-back procedure for a failed proactive handover. It requires fewer signalling messages to complete but suffers from an inherent packet loss problem due to the unforeseen wireless link change. PA-FMIP is proposed here to improve reactive FMIPv6. With the addition of two new messages, and a coordinated tunnelling agreement with the pAR, PA-FMIP is able to further hide the (reactive) link-switching latency and minimise the number of lost packets.

The PA-FMIP protocol operation is as follows:

The prediction algorithm of Section 3.2.2 is run as an application at the application layer at some time between handovers. To avoid disruption, it should be executed immediately after a successful attachment to a new subnet. Recall that the prediction process is the responsibility of the MN, not the AR. It then notifies the current AR (pAR) of its M prediction targets by sending it an *AR Notice* message. This message is forwarded by the pAR to all predicted nARs. In Figure 3.2, bi-directional tunnels are setup between points a and b, but only activated at point c. The acknowledgement of the AR Notice message (*AR Notice Ack*) notifies the pAR and MN of the acknowledging nARs and completes the handover preparation. The link-down (LD) trigger indicates the start of the link-layer handover and the instant that the pAR begins forwarding the node's incoming data

stream to the notified nARs. This is the main advantage this proposal.

A number of schemes in the literature have employed active buffering mechanisms to prevent packet loss and effect seamlessness in the handover. Since buffering is usually customised to a particular type of application, this protocol does not directly specify its usage.

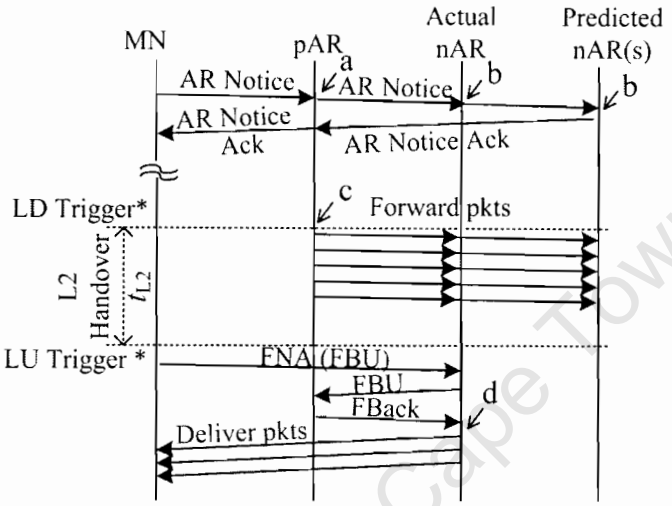

Figure 3.2: Handover timing diagram of PA-FMIP proposal.

A link-up (LU) trigger indicates that the node has completed its association with a new base station. The node then begins the fast registration process by sending a Fast Neighbour Advertisement (FNA) to the nAR. Once the nAR receives a fast binding acknowledgement (FBACK) from the pAR, the nAR delivers any en-route or buffered packets to the node. The node continues to receive its forwarded data with no interruption until it completes its home registration of its new CoA, upon which time, the tunnel(s) between pAR and nAR is deactivated.

Context transfer and security association negotiation operations usually occur at point d in reactive handovers. For small amounts of context, this delay is not significant although it does directly affect the overall handover latency [19]. PA-FMIP solves this by proactively transferring any context in the AR Notice messages.

In the case where the prediction algorithm predicts the incorrect nAR, the handover procedure falls back to the original reactive FMIPv6. The only expense of a prediction-miss is the packet forwarding overhead to the other M ARs, and the

poorer performance of the fall-back handover scheme. This is explained in more detail in chapter 5.

In the case where link-layer triggering is unavailable and traditional movement detection mechanisms are used, the performance of this (or any fast handover) protocol will be severely affected. One can expect an additional delay up to 1 second per unavailable trigger, depending on the frequency of the router advertisements (beacons). Regarding the deployment of this protocol, MN and AR entities need to be able to support its operation. Like FMIPv6, PA-FMIP may be deployed as a software patch for existing networking stacks.

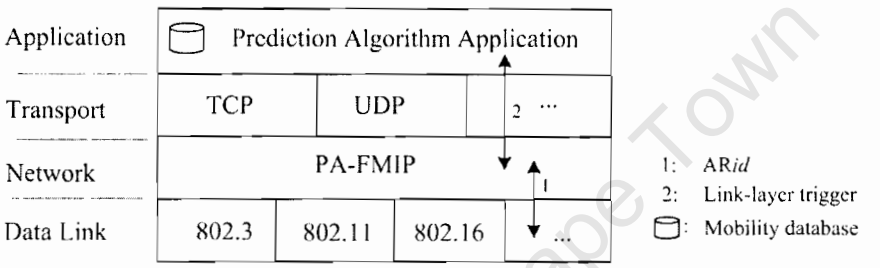

Figure 3.3: The mobile node's basic network protocol stack showing PA-FMIP on the network layer and the mobility prediction algorithm as a user application.

Figure 3.3 illustrates how the proposed PA-FMIP protocol is designed to be implemented on a mobile node. AR target information is passed to the network layer. The Link-up and link-down trigger communication is also shown. As specified, the mobility prediction algorithm is executed as an application after every subnet change. The self contained prediction algorithm (application) passes its M number of AR target *id*s to the network layer. Upon reception of the link-down trigger by the pAR the PA-FMIP signalling begins.

An example of the PA-FMIP scheme in action is shown in figure 3.4. The MN and ARs are all PA-FMIP enabled. The link and network layers are displayed separately. The routing topology is made up of the access routers positioned without any specific spatial reference to each other. The base stations however are layed out in physical cells, and the radio range of each is indicated. Note that the wireless cells could consist of heterogeneous technologies. The MN is currently connected to BS_4 and hence AR_4 (pAR in Figure 3.2). Its current

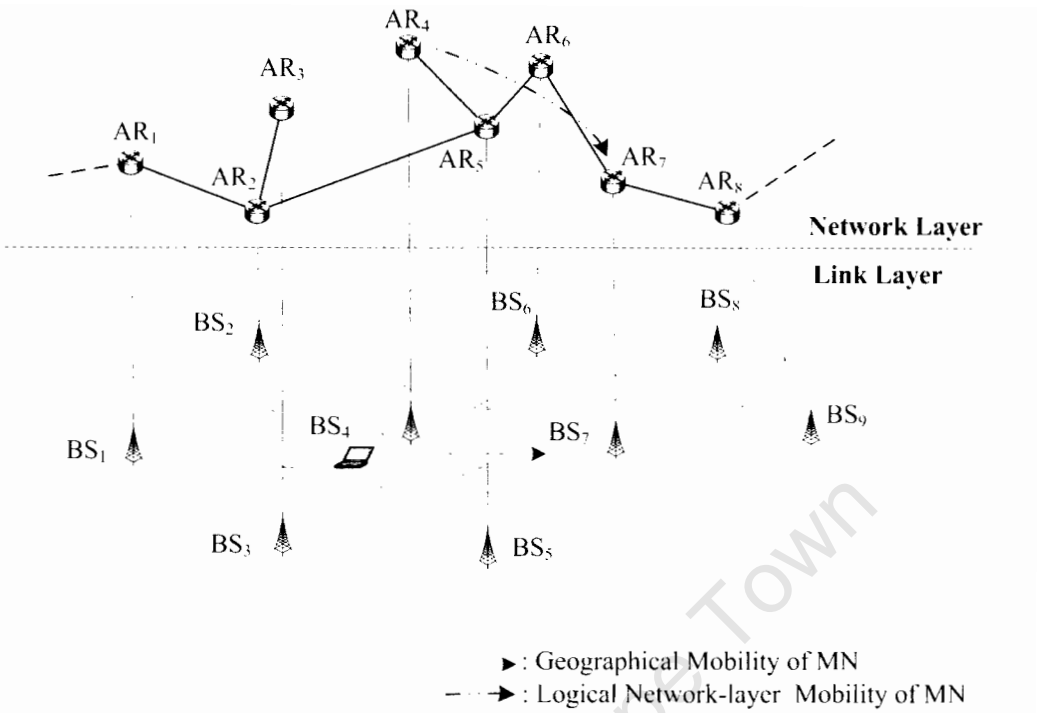

Figure 3.4: Example showing the BS-AR mapping on a typical topology.

geographical trajectory is shown. Logically, it would handover to AR_7 (actual nAR in Figure 3.2).

Part of PA-FMIP includes the use of the CARD protocol whereby the MN reports the identity (and possibly the wireless resources) of its previous AR attachment. This enables each AR to maintain a current list of its neighbouring ARs. New neighbours added accordingly so as to keep the list up-to-date. Recall that they are not necessarily spatial or routing neighbours, but handover neighbours. This list would be stored as a datastructure on each AR. The handover neighbours of AR_4 are AR_1, AR_2, AR_3, AR_5, AR_6, AR_7. The underlying geographical terrain usually determines which cells are accessible by the MN. For this example assume the terrain is flat and that each AR is equally accessible by any MN.

Let us say that the output of the prediction algorithm with M=3 (three predictions per hop) identifies AR_2, AR_6 and AR_7 as likely next-hop targets. Remember that this is derived from the MN's previous mobility history over this area. Upon reception of a link-down trigger by AR_4, it begins forwarding the MN's data through the three tunnels setup between AR_4 and the predicted ARs. The tun-

nelled packets are encapsulated and use the MN's interface identifier appended to the predicted AR's 64-bit routing prefix. In this example the predicted AR matches the actual nAR (AR_7). Thus when the link-layer handover completes, the link-up trigger (which contains the MN's interface identifier) is received by the MN and nAR, the nAR delivers the correct packet stream down through the BS the MN is attached to.

It is also clear here that the BS-AR mapping does not have to be one-to-one for PA-FMIP to work. An AR may of course have more than one serving BS connected to it. Mobility between BSs with a common AR would only invoke a link-layer handover. Evaluation topologies in the next chapter simplify the evaluation process by assuming that each AR is co-located with a single BS. This is common practise in all fast handover related IETF publications as they focus specifically on network-layer handovers [3, 11, 12].

As mentioned in Chapter 1, the actual target BS selection decision is controlled entirely by the link-layer of the MN. If the prediction algorithm had to randomly choose 3 ARs from the 6 neighbours surrounding AR4, it would have a 50% chance of a prediction hit. Since this is not the case, and it chooses the 3 most likely nARs based on the MN's mobility history, it would have a prediction hit probability of $\geq 50\%$. Given a mobility history containing a certain amount of regularity, the probability of a prediction hit is extremely high. A prediction hit would mean the MN receives the ideal performance, however if the next nAR is actually AR_1, AR_3 or AR_5, then the MN would experience the (fall-back) performance of an ordinary reactive FMIPv6 handover.

3.4 Discussion

This chapter introduced the theory behind data mining and its usefulness to mobility prediction. The details of the prediction algorithm were discussed. It was shown how data mining based prediction is tolerant to random mobility, which is an important characteristic of this proposal. Compared to similar prediction schemes in the literature, our approach is simple and effective for its purpose within PA-FMIP. The focus of this thesis is centred around PA-FMIP and its improvement over FMIPv6. With only minor additions to reactive FMIPv6, PA-

FMIP experiences the advantages of proactive FMIPv6 in providing seamless service continuity to mobile users.

The important properties of PA-FMIP may be summarised as follows:

- The pAR redirects the MN's packet stream to its predicted next AR. In this reactive type scenario, the packet loss rate is expected to be lower, making the handover more seamless.
- The pre-handover signalling between the pAR and the predicted nARs allows context and information to be transfered prior to the handover, effectively eliminating any associated delay during a subnet change. The algorithm itself also eliminates the need for a pre-trigger as in proactive FMIPv6.
- An incorrect next-hop prediction would force a reactive FMIPv6 handover to occur, leading to an increase in the number of lost packet and additional delays if context transfer are to occur.
- The user has control over the prediction parameters min_{supp}, min_{conf} and M to customise the performance depending on the user's active mobility scenario.

The following chapters address the evaluation of this work and how it performs compared to similar schemes.

Chapter 4

Evaluation Framework for PA-FMIP

4.1 Introduction

This chapter discusses the design of the platform that is used to evaluate the performance of the PA-FMIP proposal. With the requirements of this evaluation framework in mind, the implemented network topologies, components and protocols used are introduced and discussed. At the outset, the reasons for the choice of evaluation platform are discussed. This chapter is structured in such a way that the reader may get a clear understanding of the implemented processes, and gauge the level of complexity involved at each stage of the evaluation.

4.2 Choice of Platform

The evaluation of an experimental handover scheme is perhaps better performed in a simulation environment rather than a hardware-based testbed. The reason for this is primarily due to the limitations of a physical testbed. Comparing network simulation to a network testbed or emulation, the following facts about simulation become clear:

- The scale and complexity of the network topology or experiments is not limited by cost or availability of resources.

- Commercial hardware is also affected by the availability and compatibility of software and drivers. Open source implementations of a desired network or network technology are often simple emulations at best.

- Modifying drivers and protocol stacks on network components is also very difficult. Researchers are often faced with "black-box" components with limited access to driver code or APIs.

A major aspect of the PA-FMIP proposal involves network mobility. Evaluating a mobility-centred handover scheme for wireless networks requires a particularly large scale topology.

An example of a full test-bed evaluation of a mobility prediction (and context transfer and caching) scheme is NeighborGraphs [34] by Mishra et al. It spanned 2 floors of office building with 9 APs. A mobile node (laptop) was required to physically traverse the topology and record the results of hundreds of inter-AP handovers. Besides the fact that running this experiment multiple times is extremely tedious, there is no focus on the mobility characteristics of the mobile node (i.e. whether is it regular or purely random).

Song et al. [24] on the other hand, evaluated mobility predictors using huge amounts of mobility data collected from the Wi-Fi network of Dartmouth College campus over a period of 2 years. Intuitively, this would be the most accurate representation of human mobility characteristics for a physical test-bed. Unfortunately, real-life mobility datasets are usually very specific to one type of topology or environment, and are not easily incorporated into a fast-handover scheme for example.

It was therefore decided that the *Network Simulator 2 (ns-2)* [43] would be most suitable for this research. Ns-2 has the following important characteristics:

- Ns-2 is a discrete event simulator designed specifically for network research. It is written in C++ and an object oriented version of Tcl. The distribution is entirely open source and is freely available for most operating systems. The published software packages are up to date and are supported by sufficient documentation. For this project Ns-2 version 2.27 was installed on a Dell Pentium4 3.4GHz PC running Debian Linux 2.6.8.

- New protocols, applications, mobility models and wireless modules for ns-2 are regularly contributed by the research community. These components, including a number of Mobile IP type implementations, are widely used.
- It allows for complex simulation environments to be created in a shorter time and with fewer practical difficulties than in a physical test-bed. Ns-2 network animation, graphing and other tools allow for the simple collection and analysis of results.

4.3 Objectives of Simulation

The following are the main objectives of the proposed simulation testbed:

- To quantitatively evaluate the accuracy of the proposed mobility prediction algorithm in a real-world scenario. The evaluation architecture therefore needs to resemble a practical, real-world environment.
- To determine the performance of the PA-FMIP protocol, and compare its results to similar fast handover schemes. All implemented fast handover protocols need to be precisely implemented according to their technical specifications. The implementation of PA-FMIP must especially be accurate to the details of Section 3.3.

Chapter 5 discusses the simulation experiments to be performed on the simulation architecture detailed below.

4.4 Simulation Testbed Requirements

To achieve the objectives of the simulation testbed, a suitable network topology needs to be formulated. Firstly, it should have finite mobility boundaries. The mobility terrain should be covered by overlapping wireless subnets, and together forming a large, routable IPv6 network. A mobile user should be able to roam freely around the area such that it does not lose connectivity except during a handover. It should also move according to a particular mobility model that

simulates real mobility. For the purposes of mobility prediction, the mobile users mobility paths should have some degree of regularity.

4.5 Mobility Model

(a) (b)

Figure 4.1: (a) A street map of West University Place (source: http://maps.google.com), and (b) the corresponding ns-2 simulation topology covered by 36 access routers, identified as **Topology 1** for future reference.

The choice of mobility model here in fact defines the simulation terrain. Typical mobility models such as the Random Walk or Random Waypoint model are often used in the literature, however they create entirely random motion. They have no means for any sort of repetitive motion. The Restricted Random Waypoint (RRWP) model [44] only permits motion within defined boundaries / domains[1]. RRWP is used here with digital data captured from the TIGER (Topologically Integrated Geographic Encoding and Referencing) [45] database which contains selected geographic and cartographic information on road maps in the USA. This information is typically used to provide the digital map base for Geographic Information Systems or mapping software [38]. In this work, the TIGER data for a particular city section is used to generate (using the application in [44]) ns-2 compatible RRWP movement patterns. A 1200m by 1200m city section of West

[1]The RRWP mobility domain [44] is restricted to the line segments of the connected edges.

University Area in Houston Texas is chosen. It consists of 383 intersections and 594 road segments as shown in Figure 4.1.

This area is covered by a network of 36 access routers, positioned in a uniform grid pattern. The RRWP model [44] is able to generate the mobility patterns for any number of nodes and according to specific mobility characteristics. Here it can be seen how the RRWP model is able to accurately extract the street topology (a) into the ns-2 mobility topology shown in (b).

As will be shown in the next chapter, the mobile node is given vehicular mobility characteristics. The main reason for using this model is that it generates patterns such that a wireless node is made to move within the streets of the chosen city section. With the wireless access routers covering the entire area, the node is able to handover from AR to AR as it moves. The crux here is that the mobility of the node is no longer purely random. i.e. the node's mobility is restricted to the boundaries of the streets such that it has to move linearly for a certain period before it can (randomly) turn a corner, thus creating a certain degree of regular mobility. To the best of the author's knowledge, no one in the literature has used the RRWP model for the purpose of mobility prediction.

4.6 Implementation of the Prediction Algorithm for PA-FMIP

The experiments performed on topology 1 (Figure 4.1 (b)) involve the prediction algorithm of PA-FMIP. Recall that this algorithm is a direct implementation of the process detailed in Section 3.2.2. The algorithm itself is incorporated into the application layer of the MN, and executed after every successful handover to a new subnet. Remember that the prediction algorithm is designed to be a modular unit, allowing other algorithms / prediction applications to replace it if needed. Here, it is written mainly in C++, and for the purposes of the simulation is compiled into ns-2. This process is structured into a single function and executed entirely with call to `PredictionAlgorithm()`. The C++ pseudo code of this function is shown as follows:

```
void PredictionAlgorithm ()
{
1:GetCurrentPosition(); //Read database file and identify last entry. This
entry is the current position on the IP network.
2: system(FSM, D_input_data, min_supp, fsm_output_data);
//Frequent sequential itemset mining algorithm called using a Linux "system"
function. This function allows the user to execute external programs during the
simulation.
3: system(RULES, fsm_output_data, min_conf, rule_output_data);
//The rule mining algorithm is executed with the output of 2 as an input. The
resulting rules are written to file rule_output_data.
4: Filter out the matching and valid rule-confidence tuples.
5: Select M predictions from step 4.
6: Write predictions to file.
7: Pass predictions to PA-FMIP protocol (network-layer).
}
```

Firstly, the mobility database D is implemented as a text file containing the previous mobility of the MN. It was found that storing D in RAM meant that all data was lost at the end of each simulation. The node's current attachment to the network is found by reading the last ARid that has been appended to the database. In practice, D should be implemented using a legitimate database structure. For this purpose however, a text file is sufficient.

The sequential pattern mining stage of this algorithm is implemented using a pre-compiled version of the Apriori algorithm by Bodon in [40] called FSM (Frequent Sequential Itemset Mining). This stage is executed using a Linux `system` call with the database D and $\min_{supp}$ value as inputs. It outputs the frequent itemsets / mobility patterns (with their support values) to a separate text file.

In step 3, a pre-compiled version of the Apriori rule mining algorithm by Goethals [46] is used to determine the mobility rules from the previous output. At this stage the output as explained in Section 3.3, is a set of confident rules based on the frequent mobility sequences the user has taken. This latest output set is also written to file. A function then opens and scans this file, and using string parsing and manipulation functions copies the rules matching the node's current position to a new list. The predictions (or values succeeding the "=>" characters) are checked for their validity against a list of current neighbouring ARs (N), such that a valid prediction is contained in N.

Recall that a variable M was previously defined to represent the number of predictions made by the prediction algorithm at each hop; i.e., the number of target ARs that are chosen as possible next-hops. This user-defined value should read from some common memory space on the MN or read in as an input from the user. For the experiments, M is set at the beginning of each simulation. From the file containing the final list of mobility rules, the most confident M predictions are read and passed to the network layer.

If the value of M exceeds the maximum number of predictions that the algorithm can output, for instance if the database is relatively empty, the algorithm randomly chooses the remaining AR*id*s from the list of neighbouring ARs (N).

Once the predictions have been passed to the network layer, the duties of the prediction algorithm are complete. It must wait until the network layer has completed a successful handover, the new AR*id* is appended to D, and `PredictionAlgorithm()` is executed again.

As a feedback method, the algorithm has a means of monitoring its prediction accuracy (and method to automatically set M). It does this by comparing its previous prediction(s) to the new AR*id* following a handover.

4.7 Simulation Environment Considerations

4.7.1 Wireless Access

Each AR requires wireless connectivity. The choice of wireless standard is limited to those available in ns-2. New standards in the IEEE 802.11 family like "g" and "n" are available but with limited support. Even 802.16e (WiMax) and GPRS modules are available for certain releases of ns-2. The focus of this work is not specific to any link-layer technology, therefore 802.11b was chosen for this evaluation due to its widespread use and availability in ns-2. In fact, the exact link-layer and physical-layer properties of 802.11 are not of much interest here. The only properties that affect the evaluation are data rate and radio range, both of which are easily defined in the code to suit the simulation topology. For simplicity, each access router is co-located with a single 802.11b access point. As discussed in Chapter 3, the AR-AP mapping makes no difference to the prediction

algorithm. And in ns-2, there is no differentiation between ARs and APs. For this setup, each (co-located) AR is given 802.11b AP wireless properties.

The WaveLan 802.11 module for ns-2 was conceived through the CMU's Monarch project but was developed mainly for the use in wireless ad-hoc networks (broadcast mode). This presents a problem as the module is not actually a full implementation of 802.11.

In infrastructure mode, the 802.11 module uses only a single (broadcast) channel and does not actually support a complete link-layer handover like one that would typically involve disassociation, scanning and re-association procedures. Besides the single channel issue, the fact that these handover operations are left out results in a fairly unnoticeable link-layer latency. Hsieh et al. [15] acknowledges this issue and instead simply includes a constant link-layer delay parameter in the code to emulate a typical link-layer handover. Of course a complete 802.11 implementation would be ideal, however this simple emulative solution is sufficient for this evaluation. This link-layer delay parameter is in fact very useful in evaluating the effect of the duration of link-layer latencies on system performance. It may even be used to emulate hypothetical or experimental wireless technology or the effect of lengthy operations such as AP authentication. Furthermore, the nature of wireless mediums make the duration of link-layer handovers inconsistent. A constant delay would simplify the analysis of results, and place more emphasis on the network-layer handover procedures that influence its overall performance.

An important feature of 802.11 handovers is that they are hard / break-before-make handovers. A soft link-layer handover, in the case of CDMA/3G cellular technologies, would defeat the purpose of this research.

4.7.2 Topology Considerations

To create the 36 AR network of topology 1 in ns-2 required the use of a topology generator called Topoman. Topoman is an ns-2 library that is able to create, configure and manipulate large network topologies in an easy manner. It is used here to ensure that all nodes are routable, properly configured and positioned. The technical details of how this was completed are included in Appendix D.

The ARs and MN have equal transmission power equating to a range of 140m each. Placing the ARs at 200m apart means that the MN should never lose wireless connectivity. The derivation of these values is presented in Appendix D.1.

The data rate of the 802.11 MN is set at 1Mbps. Since the mobility of a single MN is considered here, it seems pointless to over-provision the available wireless bandwidth. In the next chapter, this scenario is subjected to a number of application and data-rate variations. The role of available bandwidth will be explored in more detail then.

The topology in Figure 4.1 is essentially to be used to evaluate the prediction part of the PA-FMIP proposal. To test the handover performance of the proposal, a simpler, more repeatable scenario is needed. Following from Figure 4.1, the result of mobility between two adjacent ARs of topology 1 is explored.

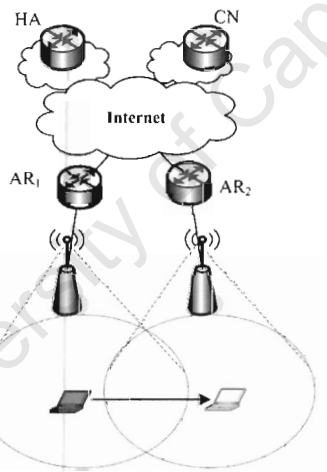

Figure 4.2: Typical handover scenario showing the MN's mobility between two co-located AR/AP pairs.

The overall performance of a handover between two subnets is affected by the the end-to-end routing topology. Thus the IP signalling between the visited network and the home network must be modelled. The scenario in Figure 4.2 resembles the mobility between two ARs on the city section of topology 1. This is realised into a second ns-2 simulation topology called Topology 2.

Here, the Internet cloud is replaced with two intermediate routers (N1 and N2).

Both the Home Agent (HA) and the Corresponding Node (CN) are connected to N1 with 100Mbps links. The link latency values give a representation of the physical distance between the nodes, and the effect of congestion on the link due to any background traffic.

The movement velocity of the MN between the two points ($5ms^{-1}$) matches the average velocity as specified by the RRWP mobility model. With the topology in Figure 4.3 the mobility of the node can be controlled and the experiments repeated with consistency.

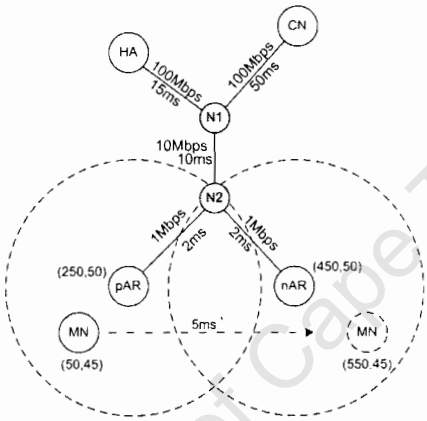

Figure 4.3: Representation of the ns-2 simulation topology (referred to as **Topology 2** for future reference) used to resemble the mobility between two access routers in figure 4.1(b). It is used to evaluate the performance of the proposal and other schemes.

4.8 Implementation of PA-FMIP in ns-2

4.8.1 FHMIPv6 Extension

A mobility management protocol is needed to maintain IP connectivity as it traverses the AR network of topology 1. Since a basic ns-2 distribution is only bundled with a standard Mobile IPv4 suite, an ns-2 extension published by Hsieh [47] is used to implement Mobile IPv6 functionality. This extension completely modifies the Mobile IPv4 suite files with new code that implements Fast Hierarchical Mobile IPv6 (*fhmipv6*). The code is structured in such a way that it allows

the author to simulate any combination of MIPv6, FMIPv6[2] or HMIPv6.

The goals of Hsieh are similar to some of those in this research, to evaluate Mobile IPv6 and current fast handover protocols. To achieve this in ns-2, Hsieh did not need a complete set of IPv6 features. The key differentiator between Mobile IPv4 and Mobile IPv6 in simulation is the Binding Cache Management. This includes the IP Destination Option, which is necessary to support the Home Address Option. These were implemented on top of existing registration, packet encapsulation and decapsulation mechanisms [15]. Essentially, Hsieh modified the Mobile IPv4 suite code enough to obtain suitable Mobile IPv6 protocol functionality.

A new routing agent (NOAH wireless extension module by Widmer [48], a prerequisite for *fhmipv6*) is needed to handle the routing of packets specifically for handover scenarios. NOAH is a wireless routing agent that (in contrast to DSR and DSDV) only supports direct communication between wireless nodes, or between base stations and mobile nodes. It also provides new peer-peer encapsulation and decapsulation functionality, an important feature for fast handover protocols. For the fast handover protocols themselves, the ns Node entity is modified to facilitate the use of these encapsulator / decapsulator functions. This allows IP in IP encapsulation / tunnelling to be setup between any type of node (wired / wireless) in ns-2.

For the FMIPv6 protocol operation, the following signalling messages are added: PrRtAdv, RtSolPr, HI, HAck, F-BU, FBAck and FNA. The MN and BaseStation Node in ns-2 are further modified to handle these messages.

The *fhmipv6* extension, as the name says, also includes HMIPv6. The workings of this protocol are not relevant in this project and therefore not discussed here. Further details about fhmipv6 installation are presented in Appendix B.3.

4.8.2 PA-FMIP

For the purposes of this research, the C++ code of *fhmipv6* was extended to implement PA-FMIP, reactive FMIPv6 and Simultaneous Bindings for FMIPv6 [4]. The first step in this implementation once *fhmipv6* was successful installed, was to implement reactive FMIPv6, the protocol that PA-FMIP is based on. Most

[2]proactive FMIPv6

of the modifications were done in the file that handled the signalling sequence of the fast handover (*mip-reg.cc*). Then the AR Notice and AR Notice Ack message types were added and the MN and BaseStation Node in ns-2 were modified to support the message exchanges.

The AR Notice messages use the output of the prediction algorithm to fill the address destination field. Practically, the ARids may be resolved to IP addresses by DNS or other means. The IP addresses of the ARs could in fact also be used as identification if they are stored as a string type within the database. The Notice/Ack messages of PA-FMIP also setup the required tunnelling between pAR and nARs using specific C++/Tcl functions.

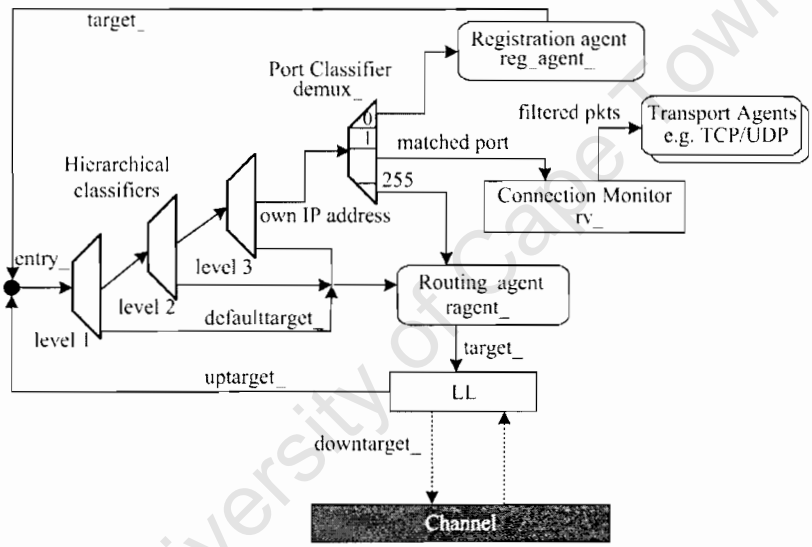

Figure 4.4: Schematic of a MN in ns-2 [15].

As explained in section 4.7.1, the WaveLan 802.11b module for ns-2 does not support a full link-layer handover due to the lack of a full implementation. Instead, a forced link-layer delay is executed in the handover code with the use of a timer (`MIPMAAgent::timeout()`) . The execution and completion of this timer within ns-2 indicate the link-down and link-up triggers.

An innovative way of allowing the MN to experience packet loss or non-connectivity during a handover was proposed by Hsieh. This was done by placing a new entity called a Connection Monitor between the port classifier/decapsulator and the receiving agents. This entity essentially traps or drops specific packets going

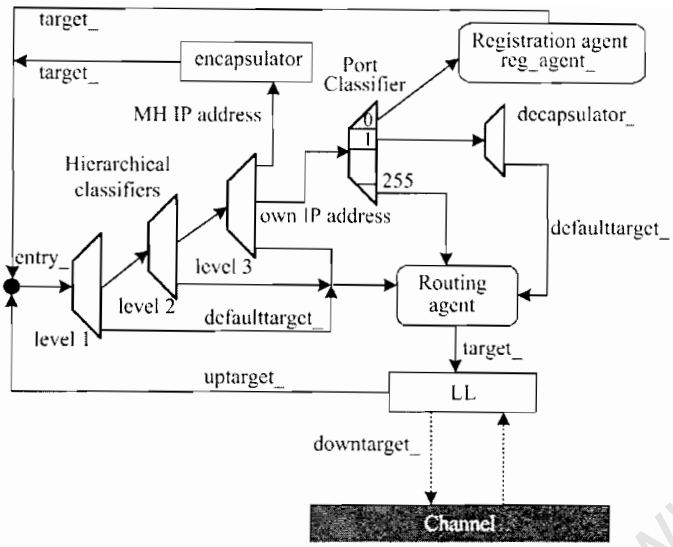

Figure 4.5: Schematic of BS in ns-2 [43].

to the transport agents (e.g. TCP/UDP) during a link change. By doing this, it emulates channel change in a 802.11 network that uses only a single channel.

A parameter in the Connection Monitor class called the `ReceiveVarifier` or `rv_` is a flag that determines the when the MN may send / receive packets depending of course on the handover protocol. Reactive FMIPv6 for example can only regain IP connectivity after a link-layer handover and only once the FBU/FBack signalling has completed, upon which time any buffered or tunnelled packets are delivered to the MN. PA-FMIP on the other hand, may begin to receive its tunnelled packets at the same point, except the MN's packets are forwarded by the pAR from the start of the link-layer handover (see Figure 3.2 for reference). The result here is a shorter disruption in the packet stream and hence fewer packet drops. The experiments in the following chapter will corroborate this statement.

The Connection Monitor is also able to treat the reception of control messages (periodic beacons from the ARs which arrive at the registration agent `regagent_`) as if operating in periodic channel scanning mode. Figure 4.4 illustrates the functional blocks of a MN in ns-2, including the Connection Monitor. Figure 4.5 shows the functional block layout of an AR (BaseStation Node) in ns-2.

4.8.3 Simultaneous Bindings

Simultaneous Bindings was relatively easy to implement. The MN was simply allowed (using the Connection Monitor) to send and receive data packets during a fast (proactive FMIPv6) handover up until the moment the link changed. This removes any possibility of packet loss due to the link-change timing ambiguity. In other words, its packet stream is bi-cast onto its current link, and future link.

The full technical details of the above implementations are included in Appendix B.3.

4.8.4 CARD Protocol

All the ARs in the network discover neighbouring ARs through the use of the Candidate Access Router Discovery protocol [42]. It allows MNs to report their previous attachments to its current AR, or the ARs themselves can solicit other ARs. Currently, there are no available implementations of this protocol, although a number of authors in the literature have proposed its use in a number of schemes. The static structure of Topology 1 means that all ARs / MN can have a list of neighbouring ARs hardcoded for each simulation. This is merely a simple alternative to coding CARD into ns-2. CARD also only requires the exchange of two messages prior to a handover. This is a trivial contribution to network bandwidth usage. And since the messages (in a real implementation of CARD) are exchanged outside of the handover period, they do not contribute to any type of handover delay.

4.8.5 Development Notes

The *fhmipv6* extension by Hsieh was released in 2003 and specifically for ns version 2.1b7a. This required substantial amount of work to compile in ns version 2.27. The process of implementing various components of this simulation architecture was very tedious. A disadvantage of using ns-2 is that it does not provide user-friendly debugging. When any modification is made to the root files or libraries (C++ or Tcl), ns-2 needs to be re-compiled. C++ compile errors are relatively simple to fix compared to simulation run-time errors. Ns-2 does not even give

descriptions of memory access errors or protocol faults. Modifying the original code, as Hsieh has done, is a far simpler task than creating a completely new protocol with new files.

The only other Mobile IPv6 implementation for ns-2 is Mobiwan2. Mobiwan2, once installed, completely changes the interoperability of ns modules. It is also extremely complex which prevents any type of major modification such as implementing FMIPv6 beside it.

The simple Mobile IPv4 patch approach by Hsieh provides all the necessary functionality of Mobile IPv6 except Route Optimisation and Return Routability. The result is a simple, reliable and accurate base for evaluating or modifying Mobile IPv6 related functionality and performance.

4.9 Discussion

The preceding sections detailed how the concept of PA-FMIP was implemented into simulation. The design of topology 1 and 2 resemble real-world mobility scenarios. Topology 1 was contrived specifically to evaluate the mobility prediction capabilities of PA-FMIP. Topology 2 is an enlarged version of the mobility of a MN between two access routers of topology 1. This topology is necessary to evaluate the handover performance of PA-FMIP and other schemes in a consistent and repeatable manner. A number of difficulties in the ns-2 implementation of PA-FMIP were overcome. Specifically, the incomplete ns-2 802.11 implementation required a work-around for infrastructure mode.

The coded implementation of the prediction algorithm has a number of parameters which could effect the overall prediction accuracy. The mobility database D, $\min_{supp}$, $\min_{conf}$, and the number of prediction at each hop (M) need to be investigated in the next chapter.

By implementing other fast handover variations like FMIPv6 and Simultaneous Bindings for the same simulation environment, the overall performance of PA-FMIP may be gauged. Simplifications to the architecture, such as using a static CARD implementation and co-located AR/APs, should make no difference to the experiments that will be conducted in the following chapter.

Chapter 5

Evaluation Results and Analysis

5.1 Introduction

This chapter presents the performance evaluation of the proposed mobility prediction assisted fast handover protocol (PA-FMIP) within the simulation architecture of Chapter 4. Each aspect of this work is evaluated individually and is presented in separate sections. The first section investigates the capabilities of the proposed mobility prediction algorithm. The second section evaluates the handover performance of PA-FMIP. The impact of the various prediction parameters are explored quantitatively. The handover experiments are performed on Topology 2 where the mobility of the node can be controlled and all experiments repeated with consistency. The results of the proposed scheme are obtained using a number of performance metrics and compared to the following four mobility management protocols:

- Mobile IPv6
- Reactive FMIPv6
- Proactive FMIPv6
- Simultaneous Bindings for FMIPv6

5.2 Performance of the Mobility Prediction Algorithm

Human movement is generally made up of regular and random components [23]. The performance of mobility pattern matching approaches are primarily restricted by the randomness of a user's mobility path. One cannot know exactly when a mobile user will choose a new or random path, nor is it possible to predict irregular radio effects such as cell bouncing. It is up to the prediction algorithm to compensate for such irregularities.

As previously explained, the proposed prediction algorithm uses the record of a user's mobility history to infer its future location. This is based on the fact that the user's mobility over a period of time has some regularity. The data-mining processes in the algorithm compensate for random entries in the history through support-counting mechanisms and a minimum support threshold parameter (min_{supp}). A new path taken by the mobile user would most likely cause a incorrect prediction. The algorithm compensates for this by predicting more than one future location or access router at each hop, increasing the prediction-hit probability. Recall that the parameter M was defined for this value.

The experiments in this section use the Restricted Random Waypoint model (RRWP) to emulate human mobility. As explained in the previous chapter, the model generates purely random node mobility except that the random mobility is restricted to a defined topology, thus creating a certain degree of regularity. Ideally, one would want to vary or control the overall randomness of the mobility through the model and record the impact it has on the overall accuracy. Unfortunately, the RRWP model does not support this, nor is it possible to quantitatively measure the resulting regular-random mobility ratio. This means that the evaluation procedure must take this into account, especially when analysing and discussing the results.

The three main aspects of the prediction algorithm that need assessment can be summarised as follows:

1. The number of predictions made at each hop;
2. The data-mining threshold parameters; and

3. The mobility history (or mobility database).

The objective of these experiments is to determine the maximum accuracy that the proposed mobility prediction algorithm can achieve. This accuracy result will indicate whether the simple data mining approach is effective for the purpose of mobility prediction, and how the prediction parameters influence the overall result. Analysis of mobility scenarios and the overall results will allow useful conclusions to be reached. Results will also uncover any practical limitations of this approach, and determine the best type of environment for it to be implemented.

The three criteria listed above form the three experiments that will evaluate the prediction capabilities of this algorithm. Two performance measures are used in the quantitative evaluation of the algorithm, viz:

Accuracy: defined as the number of correct predictions divided by the number of inter-subnet handovers.

Precision: defined as the number of correct predictions divided by the total number of predictions made at each time. Precision counts the M number of predictions (prediction hit and misses) made at each hop.

5.2.1 Simulation configuration and details

All simulation experiments in this section are performed using the city section topology (Topology 1) described in the previous chapter. Below is a summary of the initial ns-2 simulation and prediction algorithm parameters:

Simulation Parameters	Value
Simulation time	2000s
Total number of simulations	80
min_{supp}	10%
min_{conf}	90%
Session time (T)	170s
Size of initial mobility database (D)	30000s
Number of predictions per hop (M)	1 to 8

The initial mobility database D is formed by allowing the mobile node to traverse the city topology for a certain period of time. This time period is an input to the RRWP model and determines the percentage of area that the mobile node will eventually cover. The size of D therefore determines how much of the topology the node is able to "learn". This period is similar to a training set in learning algorithms such as neural networks or similar artificial intelligence techniques. Without an initial D, the data-mining prediction algorithm would continue to make incorrect predictions until the node traverses an area of the city it has previously seen. Creating a reasonably large enough database had to be done through preliminary (trial) simulations to ensure that the algorithm was given a fair chance to succeed in such a large topology. Results showed that a database equivalent to 30000 seconds of initial mobility covered a sufficient portion of city section. The maximum duration of each session is restricted to T=170s. This restricts the number of items in each transaction. 30000s corresponds to a database of 154 mobility transactions with an average transaction length of 4 items (ARs).

With this initial database in memory, each simulation was run for 2000s. A mobile node traverses the city with the same mobility characteristics as in D. These RRWP characteristics are stated below:

RRWP Parameters	Value
Average speed	$5ms^{-1}$
Max. speed deviation	$2ms^{-1}$
Mean pause time	1s
Max. pause deviation	0s
Number of nodes	1

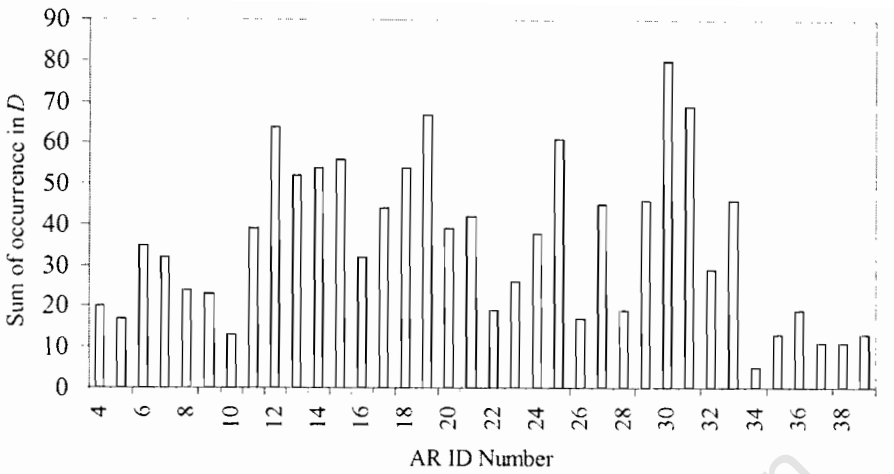

Figure 5.1: Histogram illustrating the number of times each AR occurs in the initial mobility database.

Figure 5.1 illustrates the distribution of ARs in the initial mobility database. As can be seen, the ARs frequency is not uniform. This histogram does not show any insight into the regularity of the mobility patterns, it merely shows that there are no initially untraveled networks.

5.2.2 Impact of the number of predictions made

This experiment varies the number of allowable predictions made at each hop and monitors the overall accuracy and precision results. These results are recorded through 10 simulation sets per value of M for $1 \leq M \leq 8$. During each simulation period an average of 25 handovers occurs. The results are calculated at run time, and averaged for each interval of M.

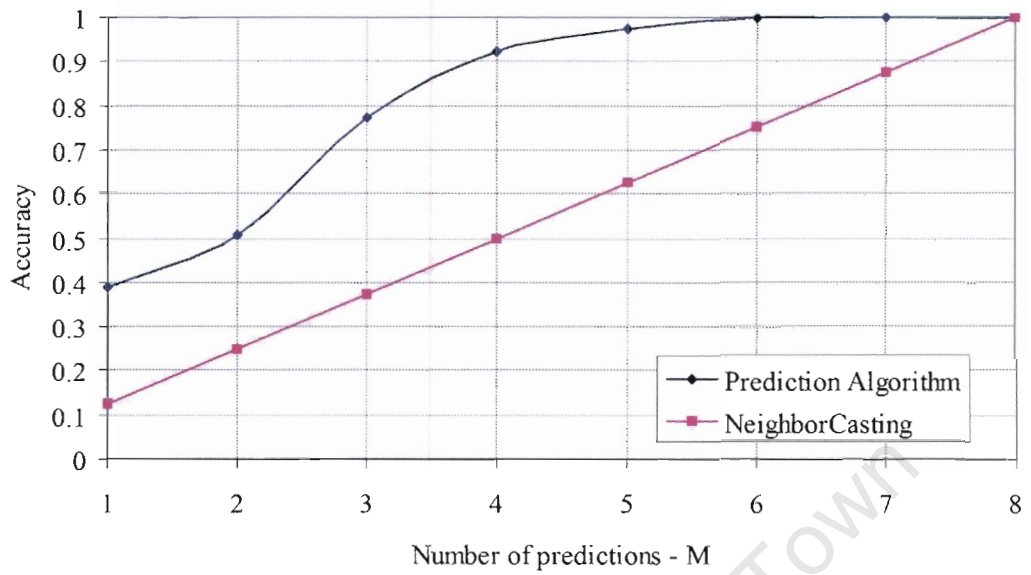

Figure 5.2: Prediction Accuracy

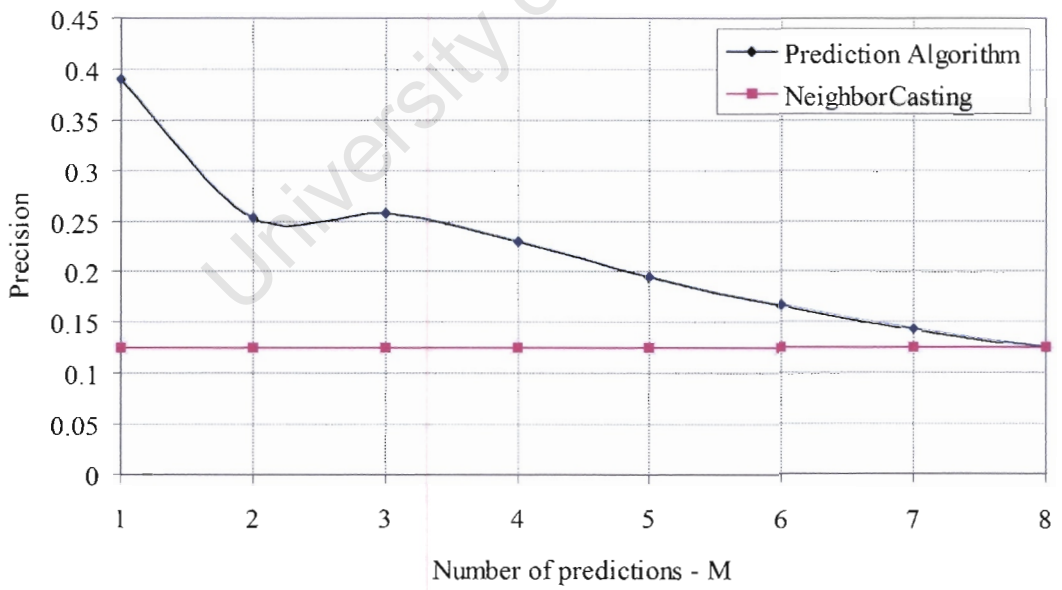

Figure 5.3: Prediction precision results as a function of the maximum number of predictions

The prediction accuracy results are shown in Figure 5.2. Accuracy for this experiment shows an expected increase in accuracy as M increases, with a minimum of 39.05% for M=1. 100% accuracy is only obtained when 6 predictions are made. The results for low values of M are very dependent on the regularity within the initial database, and if the routes taken during the simulation are in fact similar to those already traversed. Higher values of M achieve a higher prediction hit probability simply because they over-predict at each hop.

The objective of this algorithm is to predict the next location as accurately as possible. Strictly speaking, the algorithm should predict with the highest precision as number of selected target ARs affects the surrounding network bandwidth overhead. There is a clear trade-off here: a high enough accuracy to ensure a prediction hit (handover performance) versus the number of predictions (bandwidth overhead). The precision of the prediction algorithm is therefore plotted against M. The precision metric counts the number of predictions (M) as well as the prediction misses. One would therefore expect that the more prediction attempts are made, the lower the precision would be. This trend is evident in Figure 5.3 with a maximum value of 39.5%.

The precision results are relatively poor, and not what one would expect from a prediction algorithm. The highest precision results correspond to the lowest prediction accuracy, and get progressively worse as the number of predictions is increased. Again, the mobility model and the initial database are the restricting factors here, and are explored in more detail further on.

NeighborCasting

No evaluation is complete without a comparative analysis with a similar scheme, especially one that would provide a suitable benchmark. Unfortunately the code for other prediction algorithms in the literature is not publicly available. And no compiled algorithms or modules have been published for ns-2 (or any other simulator). The results presented in Figure 5.2 and 5.3 are compared to the analytical results of NeighborCasting, the scheme which is probably the most similar scheme to PA-FMIP in the literature. Upon a link-down trigger, the mobile node's packet streams are tunnelled to *all* of its neighbouring subnets. NeighborCasting thus compensates for the fact that the node does not know *a priori* to which sub-

net it is going to handover. It makes an uninformed selection (prediction) of its next location. The result of this over-provisioning packet forwarding approach is a reduced handover latency and reduced packet loss, at the expense of excess bandwidth usage.

Figure 5.2 shows the prediction accuracy of NeighborCasting as a function of the number of neighbours N. From Topology 1, each AR has an average number of $\overline{N}$= 8 neighbours (even though the ARs on the perimeter only have between three and five neighbours). The maximum number of ARs that NeighborCasting scheme chooses at each handover is limited, and the result is equivalent to a linear gradient of $\frac{n}{N}$ where n is the specified maximum number of ARs. In this comparison n is equivalent to M, such that it follows a linear increase in accuracy as M increases, with a minimum of $\frac{M}{N} = \frac{1}{8} = 12.5\%$. The constant number of neighbours also means that NeighborCasting achieves a constant precision value of 12.5%.

Discussion

It is important that an optimal value for M is found here. Considering the accuracy-overhead trade-off, there is no clear equilibrium. From Figure 5.2, choosing $M \geq 4$ would ensure an adequate prediction hit probability. Anything less would defeat the purpose of having a prediction algorithm. Figure 5.4 shows the improvement in accuracy that the prediction algorithm has over the NeighborCasting scheme. It also indicates that the maximum utility of this scheme is achieved when M=4. The impact that the choice of M has on the bandwidth overhead is illustrated and discussed in Section 5.5.

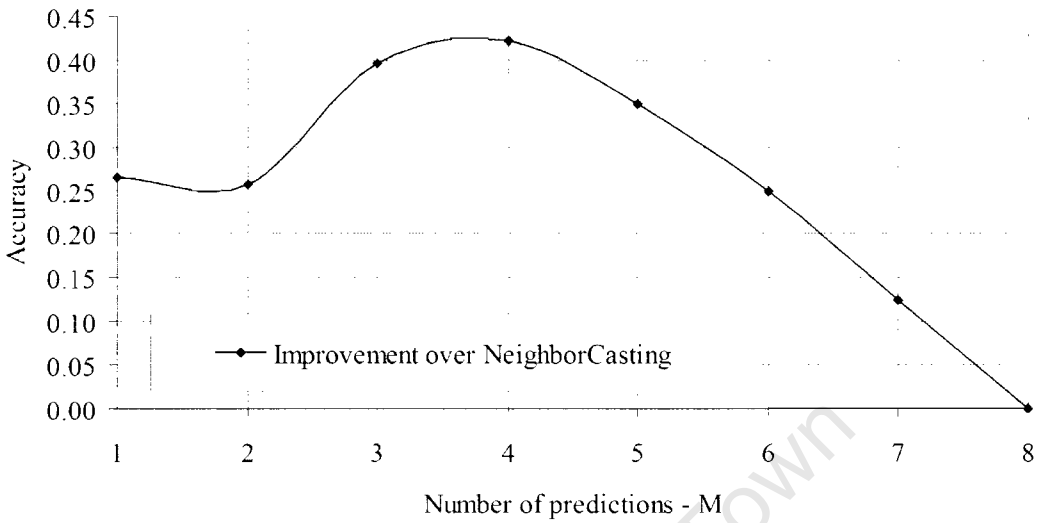

Figure 5.4: The improvement in accuracy of the proposed prediction algorithm over NeighborCasting.

5.2.3 Impact of prediction parameters

Data mining threshold values

Further simulations were performed varying min_{supp} and min_{conf} to determine the effect they had on the overall prediction accuracy. It was found that for this mobility scenario, no significant change was observed in the accuracy (and precision) so long as the two pruning parameters stayed within the following ranges:

$$4\% < min_{supp} \leq 10\%,$$
$$min_{conf} \leq 80\%$$

Setting min_{supp} and min_{conf} outside these ranges caused the algorithm to output irregular and inaccurate predictions. A minimum confidence value that was too high eliminated the chance for a less regular AR to be chosen. A minimum support threshold that is higher than 10% significantly reduced the output of the

frequent itemset mining stage. Below 4% and the frequent itemset mining stage outputs far too many patterns, including the irregular or random paths that it should eliminate. This has a ripple effect on the number of mobility rules that are generated from this large pattern set. This then affects the quality of the final predictions that are made.

Processing time

The processing time associated with the Apriori data-mining algorithms has also been measured. These time measurements provides a crude assessment of the complexities involved in the data mining processes. A comparison of processing times indicates the impact that particular parameters have on the system performance. Table 5.1 illustrates these results.

Size of D	Session time (T)	Av. length	Processing time
$<$ 2000s	170s	4	$<$ 100ms
2000 to 20000s	170s	4	100-300ms
$>$ 20000s	170s	4	$>$ 300ms
2000 to 20000s	500s	13	$>$ 1s
$>$ 20000s	500s	13	$>$ 1.5s

Table 5.1: Approximate CPU processing times for Apriori data mining processes.

It is clear that the main contributing factor to the overall processing time is the length of each transaction in the database. This explains the purpose of limiting the session time (T) to 170s. It can be seen that datasets equating to less than 2000 seconds of mobility take a negligible amount of time for a Pentium 4 processor. Datasets equivalent to 20000 to 30000 seconds or more take approximately 300-400ms to process.

Increasing the session time (T) to 500s creates transactions with approximately 10-15 entries. This results in a processing time of over 1.5 seconds for large amounts of recorded mobility. This is a very long period of time, but fortunately it occurs prior to a handover and does not contribute to the handover latency at all.

It is difficult to gauge here how this processing time would affect the performance of battery operated devices. Technically, the full amount of resources used by the

prediction algorithm should be considered, not just the processing time however, this is outside the scope of this research. Future work may investigate it in a more thorough manner.

Size of the database

It was found that when the size of the initial mobility database was increased to over 30000s, there was no major improvement in accuracy. This is again attributed to the mobility model being used. Once the initial database has "covered" a large enough percentage of the city, subsequent additions to the database (using RRWP) are not regular ones.

Improving the precision of the algorithm would require more regular mobility. The following experiment investigates this issue by replacing the mobility model with a single set mobility path.

5.2.4 Impact of the initial mobility history

To assess the relationship between D and the prediction accuracy, a simple mobility scenario was created using multiple runs of a hard-coded mobility path instead of the RRWP model. In this experiment, the initial database is initially empty and the node is forced to traverse a short set path across the city. The objective is to investigate how long it would take the algorithm to achieve 100% accuracy. This would indicate the responsiveness and learning potential of the prediction algorithm when presented with purely regular mobility.

The accuracy is plotted against the number of "runs" or number of times the node traversed the set path. A mobile node is made to traverse the path shown in Figure 5.5 with an initially empty mobility history. The route corresponds to the the logical network layer mobility path:

21, 20, 19, 18, 17, 23, 29

When the node is activated, it connects to the available AR which is AR 21. From there it begins its path towards AR 29. The results in Figure 5.6 show that the prediction algorithm fares poorly when faced with a completely new path.

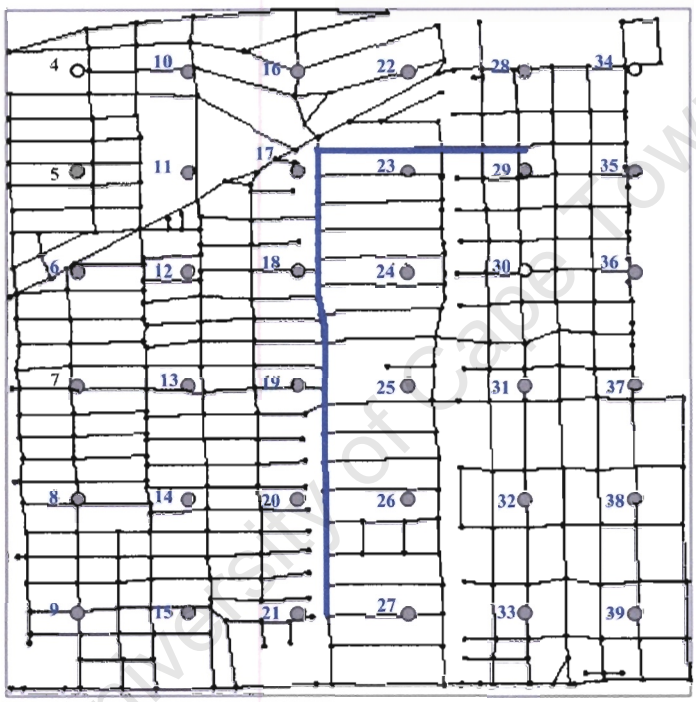

Figure 5.5: Topology 1 with a set mobility path or "run" instead of RRWP mobility model. The 36 Access Router's identification numbers are shown.

When M is increased, it compensates for the empty database by making random predictions based on the set of neighbouring ARs. On the second run, i.e. with 21, 20, 19, 18, 17, 23, 29 stored in D for the first time, the prediction algorithm does well. For M>2 the algorithm correctly predicts all 6 ARs on the path.

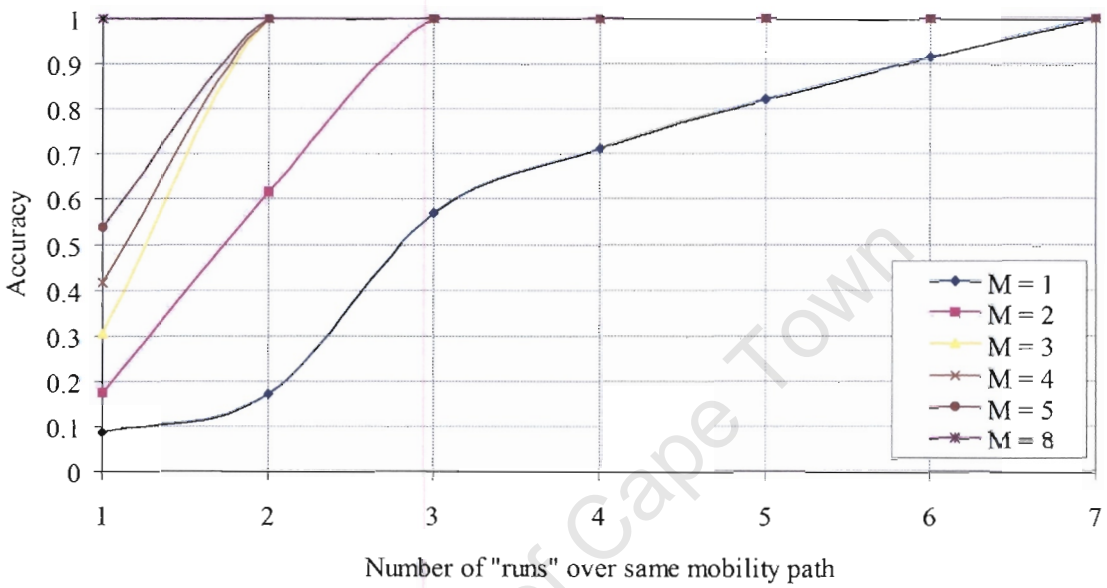

Figure 5.6: Impact of the number of times the node traverses a single set path on the prediction accuracy.

Discussion

These accuracy results indeed show that the prediction algorithm has potential. With an initially empty mobility database, on average it takes at least one run over the path to achieve 100% accuracy. The results achieved on the first run for each value of M are random selections of the AR neighbours, hence the low but acceptable accuracies. It also means that for a large number of neighbours, the initial prediction hit probabilities (for low number of M) will be smaller.

This has the following implications when considering a practical implementation:

- The size of the topology affects the overall accuracy;

- A large number of neighbouring ARs reduces the initial predictions for an empty database;
- A large topology would take longer to cover, and therefore longer to reach 100% prediction accuracy; and
- A real-life mobile user is expected to have more regular mobility characteristics than those achieved with the RRWP model. An increase in regular mobility would increase the prediction accuracy significantly.

These points suggest that this type of mobility prediction algorithm is perhaps best suited to finite mobility environments, i.e., a PCS network or vehicular network. The RRWP model used in these simulations emulated vehicular mobility but in a random manner. Although the algorithm managed to compensate for this random mobility, it performed significantly better in the fixed route scenario.

5.3 Handover Performance Evaluation with Real-time Applications

Real-time applications require their QoS parameters to be strictly adhered to. Examples of real-time applications are VoIP, video conferencing or live Internet broadcasts. Such applications are often grouped into classes according to their QoS requirements. Instant messaging or chat programs can be considered real-time, although users are willing to tolerate far higher message latencies than in a VoIP conversation. On the other hand, new services provided by network operators such as IPtv, offer Triple-Play, i.e. the simultaneous (real-time) transmission of voice, video and data content over a single IP connection. This type of broadband media has a very low delay tolerance.

The following experiments investigate the case where a mobile user is forced to perform a subnet change during a real-time broadband application. The handover performance of the PA-FMIP proposal is evaluated using three primary metrics: latency, packet loss and jitter.

5.3.1 Handover Latency

The application that closely resembles a real-time application is a Constant Bit Rate (CBR) application. In this experiment, a CBR application is configured with a constant data rate of 300kbps and a UDP packet size of 210 bytes. A VoIP application typically requires a bandwidth of 64kbps with the same packet size. The seemingly high data rate is actually far lower than the bandwidth requirements of IPtv or other Triple-Play services. Given the definition of handover latency metric below, the 300kbps data rate also provides a suitably high resolution for measuring time intervals between sequential packets. Also, VoIP / Video applications use specific codecs to encode and decode data streams. This CBR application is an emulated approach and the effect of different codecs is not explored.

Handover latency in this case is defined as the maximum period of disruption in the CBR packet stream during a handover.

The simulations are performed on Topology 2 for a duration of 80 seconds each. A handover occurs approximately midway between the pAR and nAR. The results of the five handover protocols are compared in Figure 5.7.

PA-FMIP shows an improved performance over both proactive FMIPv6 and reactive FMIPv6. The significantly longer handover latency of proactive FMIPv6 is due to the timing ambiguity problem. Referring to Figure 3.2, the pre-trigger at $t_{\text{proactive}}$ occurs 202ms before the link-layer handover. This is the time required by the MN to complete the CoA negotiation and home registration. Once the registration process has begun, the MN does not change links for another 62ms (t_{reg}) and it cannot receive packets from the pAR. Simultaneous Bindings for FMIPv6 achieves the lowest latency since the mobile node is able to receive packets on its own link up until the link change, mitigating any trigger timing issues.

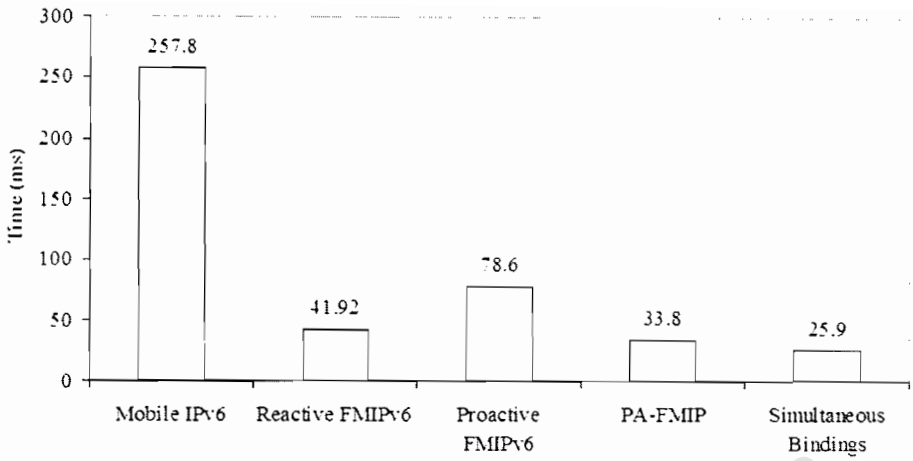

Figure 5.7: Handover latency results for a CBR application.

A pause or delay during a VoIP conversation that is greater than 150ms is considered unacceptable [49]. One can see from the latency results that Mobile IPv6 alone does not perform adequately enough. The other four protocols achieve relatively low handover latencies. PA-FMIP however is the only one that would not suffer from additional delays when forced to perform a context transfer during a handover.

5.3.2 Packet Loss

Packets that are lost during a VoIP conversation for example are never heard by the end user. Typically, packet loss is measured as an average (percentage) over the duration of the call. And a VoIP call with 1% packet loss will always sound better than one with 10% packet loss. The type of packet loss can also be categorised: *Random* packet loss describes a call that loses say 100 packets over the duration of the call, while *bursty* packet loss describes how 100 packets may get lost over a very short period of time. However these losses are specific to network congestion and routing issues. The focus of this experiment is on the effect of a fast handover on a high data rate application. Only the packets that are dropped or lost during the handover period are counted.

Packet loss is defined as the sum of all lost or dropped packets during the

handover period.

With this CBR application, all incoming packets have a uniform arrival rate. Therefore the number of packets lost during a handover should be proportional to the handover latency of the respective scheme. Figure 5.8 illustrates and compares the recorded packet loss results. In PA-FMIP, the node's packet stream is forwarded to it next (predicted) AR as soon as the link-layer handover occurs. The result is a 50% improvement in the number of lost packets over reactive FMIPv6. This shows that PA-FMIP can provide a faster and smoother handover than reactive FMIPv6.

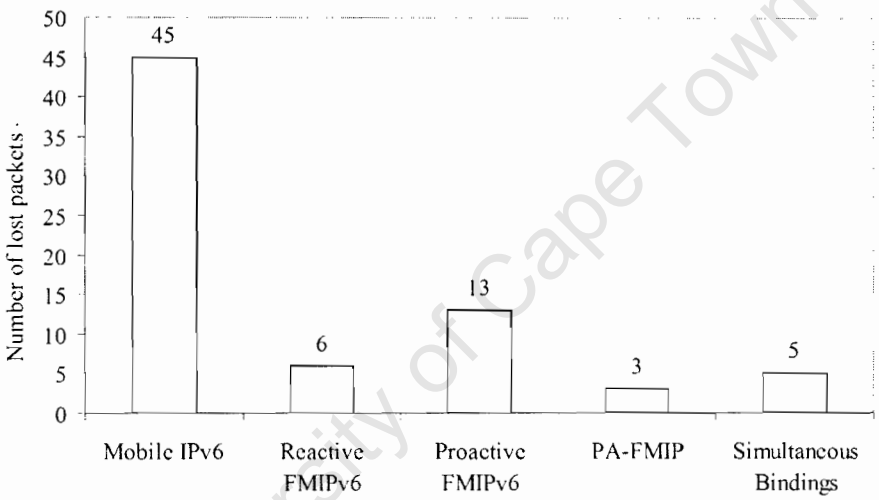

Figure 5.8: Packet loss results for CBR application.

5.3.3 Jitter

At the source, packets are sent in a continuous stream with the packets spaced evenly apart. Due to network congestion, improper queueing, or configuration errors, this steady stream can become lumpy, or the delay between each packet can vary instead of remaining constant. It is a common problem for real-time applications (such as Skype) that rely on the Internet to provide a suitable QoS level. End nodes compensate for jitter through the use of jitter buffers. Although a jitter buffer can remove the jitter from arriving packets, it does so by increasing overall delay. A small buffer will introduce less delay than a large buffer, but it

can overflow and lose real-time packets when jitter is high, with a resulting loss in voice or video quality. A larger buffer, on the other hand, will lose fewer packets but can introduce unacceptable delay.

Jitter is defined as the variance in arrival time of the received packet stream.

Typical QoS restrictions for VoIP require a maximum of 50ms of jitter to maintain an intelligible two way conversation [49]. This value is based on the arrival rate of a VoIP - typically between 20ms and 30ms. The arrival rate in this 300kbps experiment is 5.6ms per packet, which is approximately five time less. The maximum permissible jitter ranges for streaming video are unclear in the literature. For best effort networks, jitter usually causes high data-rate video packets to be dropped as a result of scheduling collisions.

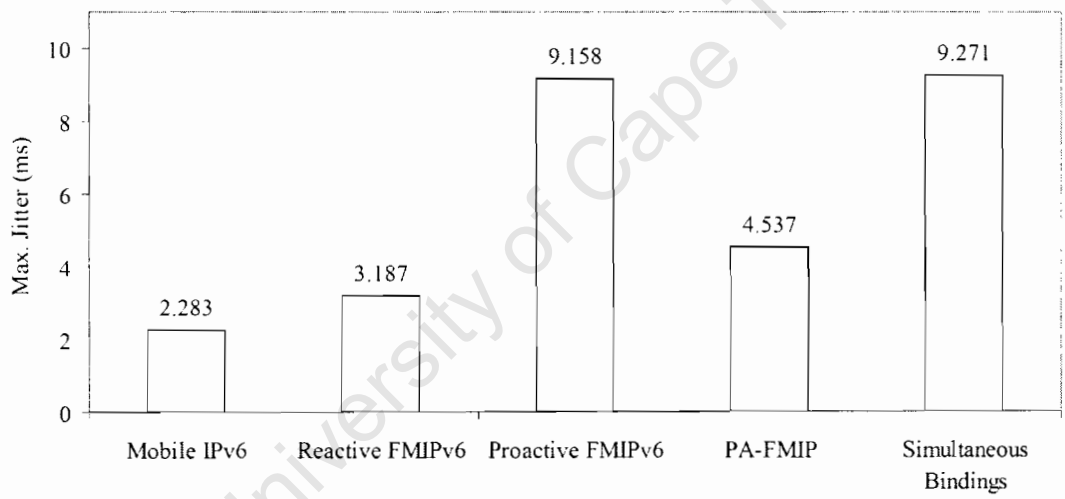

Figure 5.9: Recorded jitter values for the 5 handover protocols.

A comparison of maximum recorded jitter values are shown in Figure 5.9. As expected, PA-FMIP has a higher recorded jitter than reactive FMIPv6. Proactive FMIPv6 and Simultaneous Bindings however achieved the highest (max) jitter by far. These two schemes forwarded data from pAR to nAR for the longest period of time because of the pre-handover signalling period. In this simulation no additional buffering mechanisms were used, therefore the embedded link buffers and queues filled up more during the proactive FMIPv6 and Simultaneous Bindings handovers.

5.3.4 Discussion

These experiments were designed to represent a simple network architecture, one that did not complicate the final analysis by introducing extra variables such as background traffic or buffering mechanisms.

The handover latency and packet loss figures achieved by PA-FMIP are indeed better than reactive and proactive FMIPv6. This means that a real-time application would experience less of an interruption / artifact. The jitter experienced during a PA-FMIP was almost half of a proactive FMIPv6 handover. The implications of these results is that an end user would ultimately receive a better quality audio or video stream.

The mobility prediction approach taken in this work is one step towards perfectly seamless IP mobility. The only restricting factor is the prediction accuracy; a prediction miss would force the MN to fall back to a reactive FMIPv6 handover. In scenarios where the MN requires additional security or QoS context transfers, the difference in handover performance between PA-FMIP and reactive FMIPv6 could be far more severe.

5.4 Handover Performance Evaluation with File Transfer Applications

Virtually all of the Internet based connection-oriented services make use of the Transport Control Protocol (TCP). It provides end-to-end flow control services that manage the transfer of information between network nodes. This flow control is crucial in dealing with network congestion, transmission timeouts and lossy links. The experiments in this section focus on the effect of fast handovers on TCP behaviour, this time using handover latency, packet loss and throughput as the three primary performance measures.

The File Transfer Protocol (FTP) is an application that runs exclusively over TCP. It allows an FTP client to manipulate or download files from an FTP server in a reliable and efficient manner. In this experiment, an FTP download scenario is created where the MN (client) downloads a 3MB file from the CN (server) as it moves between subnets. TCP-Tahoe is used with a packet size of 512 bytes

and a window size of 32 packets. The download session begins 10 seconds after it begins to move. The impact that the handover has on the end-to-end application is monitored.

TCP-Tahoe is a common and well researched type of TCP. Its particular characteristics are expected to influence the overall results. To this end, a number of different types of TCP are explored in some detail in later sections. However the underlying mechanisms of the TCP variant are not the focus of this study.

5.4.1 Handover Latency

Handover latency is defined for this TCP experiment as the period of time between the first retransmitted packet and the last time this packet was sent [15].

The TCP sequence numbers of the streaming packets are plotted against time for the period of the handover (Figure 5.10). This shows the response of the TCP congestion control algorithm of TCP-Tahoe when faced with a fast handover.

The latencies of each handover protocol are measured from the TCP sequence number plot of figure 5.10. The handover latency of reactive FMIPv6 and PA-FMIP is represented by d_r and d_{pa} respectively. The measured values are compared in Figure 5.11.

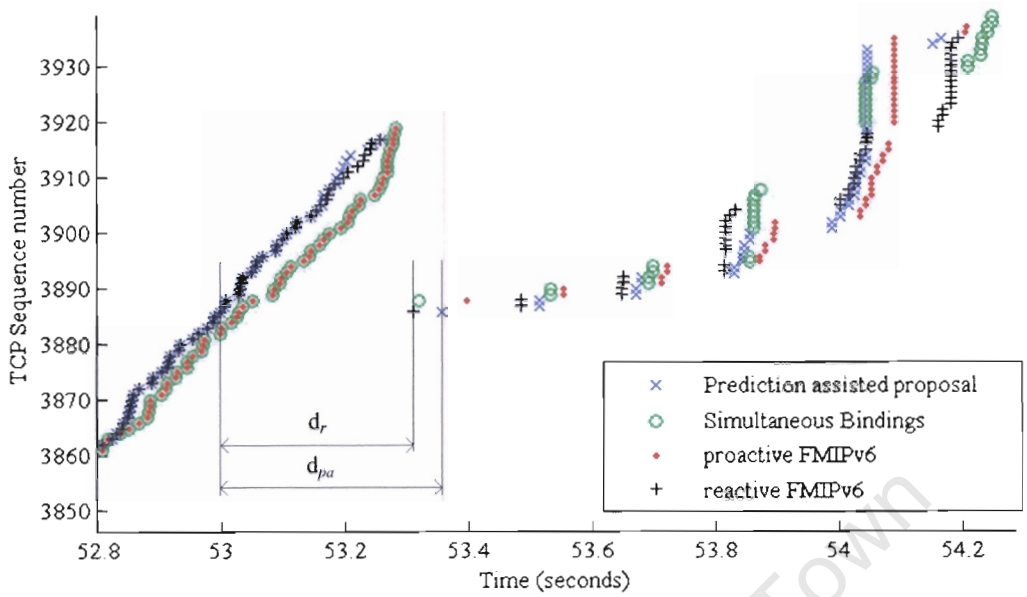

Figure 5.10: TCP sequence numbers plotted against time. Handover latency values are measured between sent and re-transmitted packets.

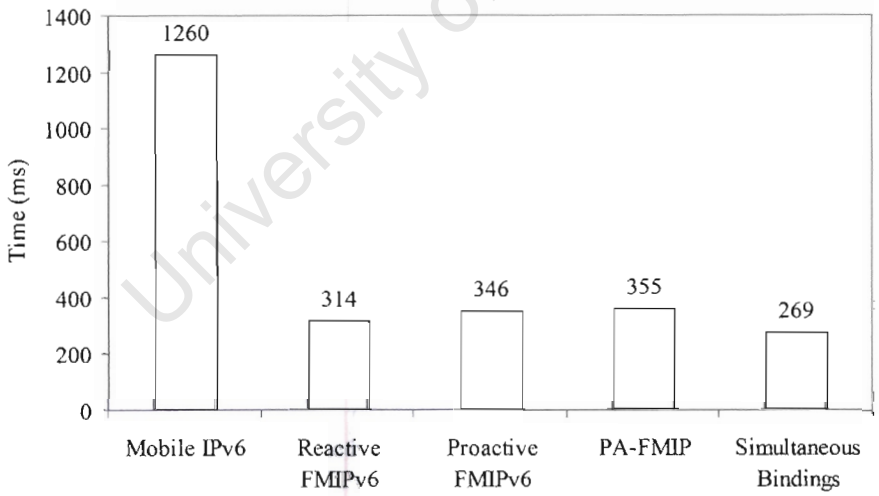

Figure 5.11: Handover latency results for an FTP download application.

An interesting result is immediately visible. PA-FMIP (d_{pa}) has a 41ms longer handover than reactive FMIPv6 (d_r), and 9ms longer than proactive FMIPv6. This is a direct result of the congestion effect of the large packet size and the

packet forwarding between the pAR and the nAR. The arrival rate of the FMIPv6 control messages (e.g. FBU, FBACK, etc) are therefore decreased. Bi-casting (simultaneous bindings) has a similar effect on the arrival rate of signalling messages [15] like PA-FMIP, however it is not as noticeable.

Further analysis of this issue points to the ns-2 buffering mechanism.

Buffer usage

The ns-2 link buffering mechanisms normally handle link congestion of this nature. It can be seen in the following comparison of peak buffer usage (Figure 5.12) that PA-FMIP and proactive FMIPv6 have similar results. Unfortunately there is no way to reduce these buffer statistics except by not tunnelling the node's data. The buffers used in the simulation are merely link buffers (FIFO queues) set with an unlimited maximum size. Packets enter the queues only when the transmission bandwidth exceeds the capacity of the wireless link. No active buffering is done at the ARs to temporarily store packets while the node transitions between networks. This type of buffering is extremely complex and usually tuned to one particular application type.

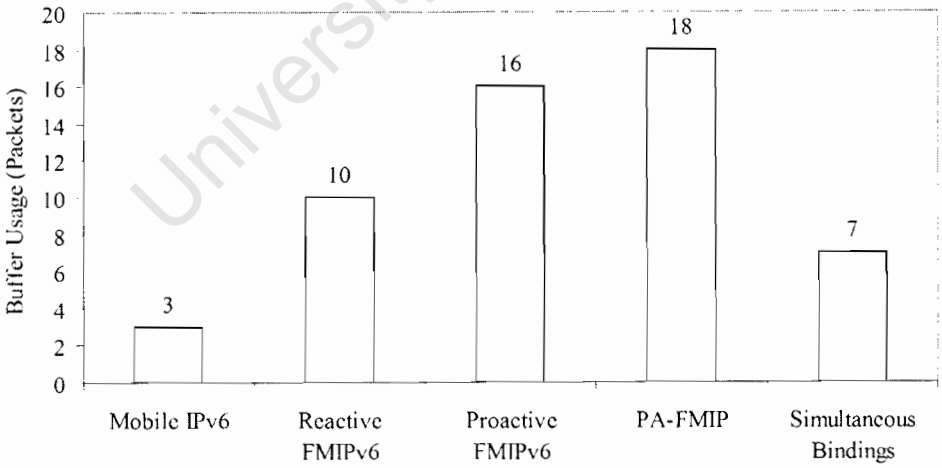

Figure 5.12: Comparison of buffer usage statistics.

5.4.2 Packet Loss

Presented below are the packet loss results for this experiment. It can be seen that the results are influenced by the packet forwarding mechanisms of each scheme.

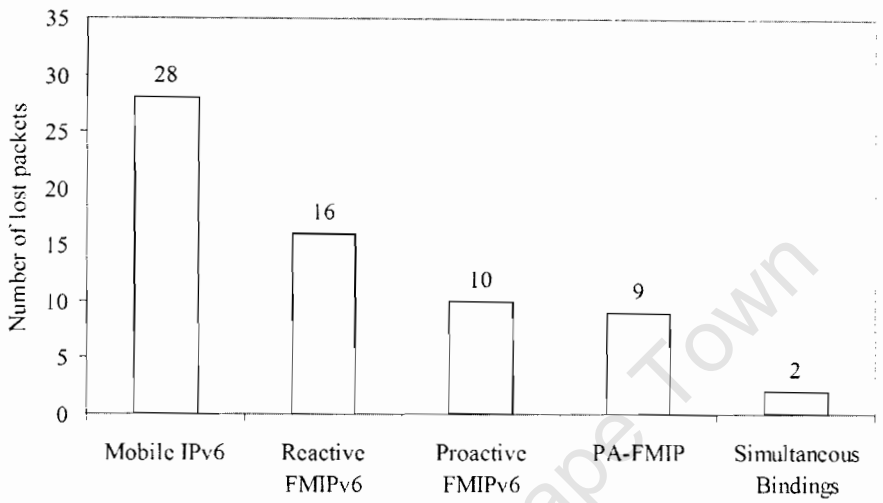

Figure 5.13: Comparison of TCP packet loss results.

A positive spin-off from the packet forwarding is a significant decrease in the number of lost packets by PA-FMIP. This is shown in Figure 5.13. As echoed in the previous CBR experiment, PA-FMIP shows a 43% decrease in packet loss compared to reactive FMIPv6. This illustrates the main contribution of the mobility prediction towards improving the seamlessness of reactive FMIPv6. It is on par with proactive FMIPv6, the difference is that the mobility prediction algorithm has replaced the need for a pre-trigger.

Another advantage is that PA-FMIP is not limited to one type of access technology. The improved seamless mobility can be maintained over heterogeneous wireless networks, while proactive FMIPv6 is limited by trigger restrictions.

5.4.3 Throughput

The average throughput results of Figure 5.14 (displayed in kB/s) illustrate the effect that handover latency and packet loss have on a user's download. The five

protocols are compared to the throughput achieved when no handover occurs.

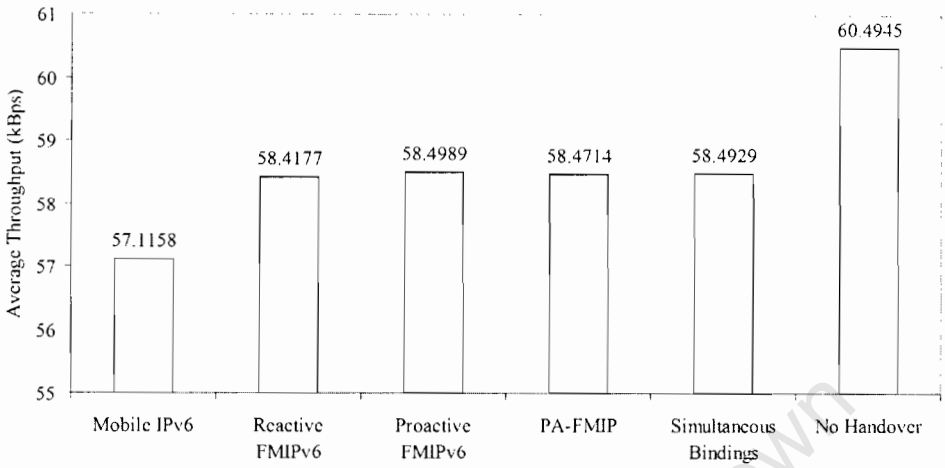

Figure 5.14: Average throughput values for the 3MB file download.

The download results indicate that throughput performance is influenced more by packet loss than latency. In TCP, the number of lost packets generally determines the number of retransmitted packets.

Besides the slightly slower TCP handover and buffer usage due to the forwarding, PA-FMIP shows competitive throughput and packet loss figures next to proactive FMIPv6, and a definite improvement over reactive FMIPv6. Although the differences in throughput are marginal, one can still gauge the effect that the fast handover protocol's performance has on a user's download. It is expected that this would be compounded for frequent handovers, causing significant decrease in throughput for schemes with higher packet loss.

As mentioned before, there are a number of different types of TCP available. Each has the potential to perform differently in a handover scenario, thus warranting further investigation.

TCP variants

The TCP variant (or type of TCP) also plays a part in the overall throughput results. This is because each variant deals with congestion and packet loss in

different ways. A comprehensive TCP evaluation would require a specific evaluation topology and focus on the bandwidth sharing and friendliness properties on each implementation.

Without trying to perform a comprehensive TCP evaluation, seven common TCP implementations are compared for a PA-FMIP and reactive FMIPv6 handover. The objective is to gauge the effect that mobility would have on a user's download, and find a general trend when considering the different TCPs. This experiment does not explore the effect of data loss due to spurious timeouts [50] of lossy links.

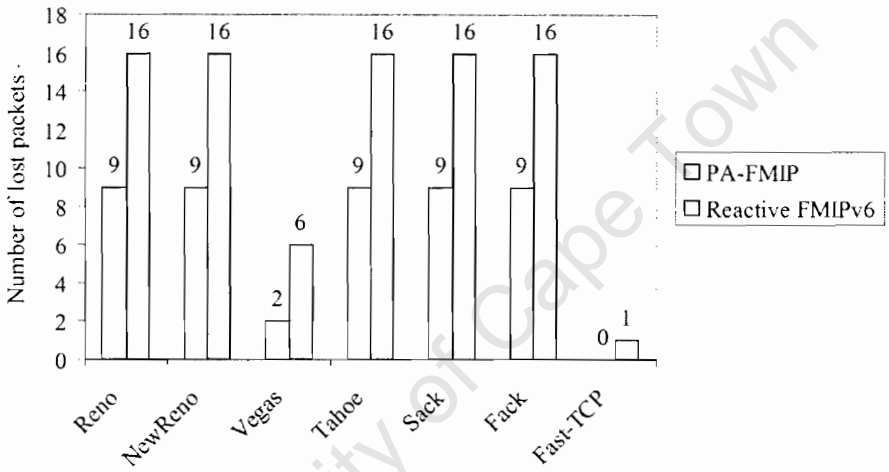

Figure 5.15: Comparison of packet loss results for different TCP variants.

It is observed that the number of lost packets is generally consistent for each type of TCP. Looking at the throughput results of Figure 5.16, it is clear that the different TCP implementations respond to packet loss in different ways. Fast-TCP [51] is based on TCP-Vegas which uses queueing delay instead of loss probability as a congestion signal [52]. This is the main difference that TCP-Vegas and Fast-TCP have over the other TCP variants, explaining the reduced number of lost packets. TCP-Sack and TCP-Fack rely on selective acknowledgements, reducing the number of acknowledgements it has to wait for. More information regarding the different mechanisms in TCP can be found in [53, 50, 54].

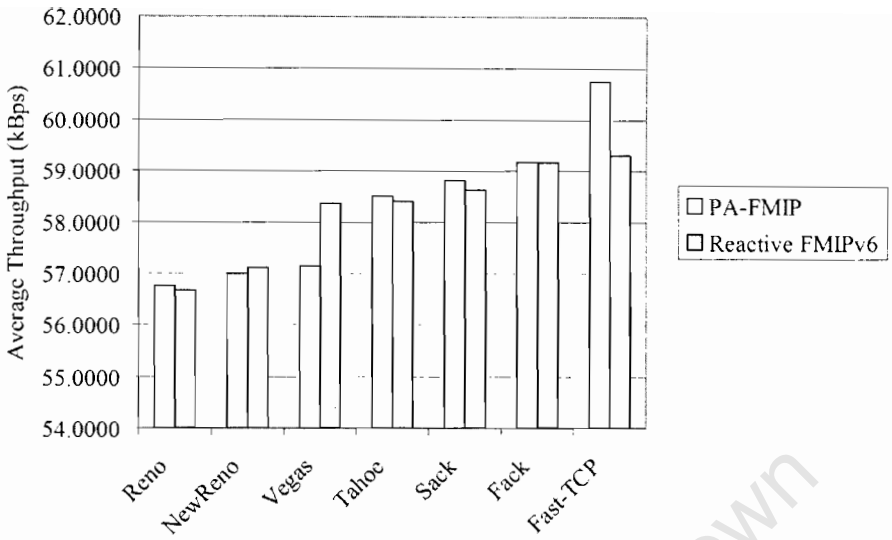

Figure 5.16: Comparison of throughput values for PA-FMIP and Reactive FMIPv6 for different TCP variants.

Discussion

Observing the throughput values of Figure 5.16, one can see that PA-FMIP outperforms reactive FMIPv6 based on the number of packets it loses during a handover. The implication is that PA-FMIP can improve a users download times regardless of which TCP is being used. A mobile user does not normally have access to the embedded TCP on its system, however these results do show that the correct TCP choice can further improve download speeds.

Besides the overhead of the scheme, which is investigated in the next section, the proactivity introduced into PA-FMIP by the prediction algorithm is definitely justified here.

5.5 Overhead Evaluation

To determine the overhead of the PA-FMIP scheme, the simulation topology (Topology 2) is modified to include a total of eight nARs so that it resembles the grid topology of Topology 2. The CBR application from Section 5.3 is used here.

The number of tunnelled packets were recorded for ranging values of M. This was done instead of measuring the rate at which data was forwarded, since the rates are essentially the same for PA-FMIP and proactive FMIPv6. A per-packet measurement would show a more accurate reflection of the overhead. The overhead results are compared to the NeighborCasting scheme assuming a handover of the same duration. It must be noted that the overhead is only imposed on the wired links and not on the wireless bandwidth. The authors of NeighborCasting argue that today's wired bandwidth is getting cheaper and more abundant, and that to provide the lowest possible handover latency and QoS to the user, the inefficient use of wired bandwidth is a "sound engineering trade-off" [16].

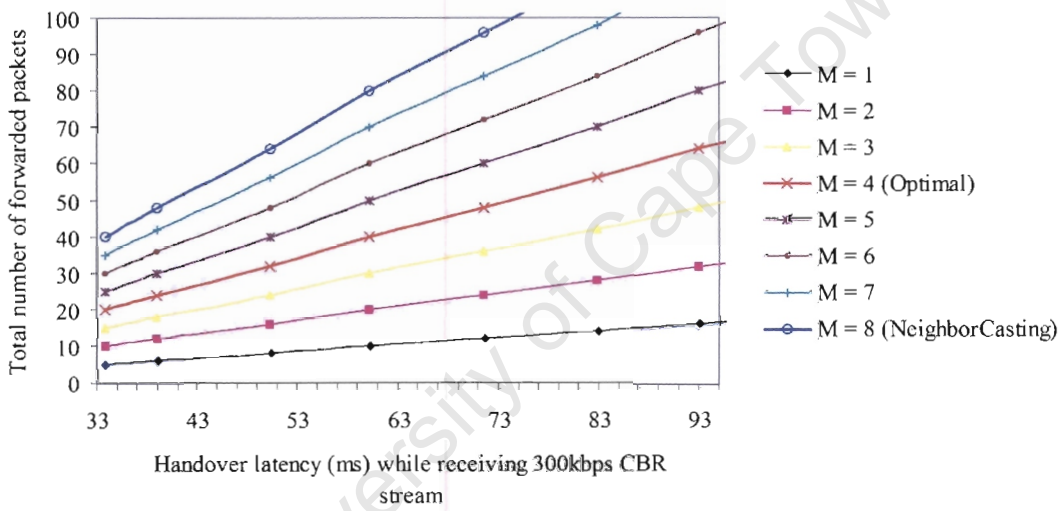

Figure 5.17: Packet overhead plotted against handover duration.

Figure 5.17 shows the number of tunnelled packets plotted against the handover latency. It is observed that for a PA-FMIP (UDP) handover of 33.8ms, M=4 (prediction hit probability 92.5%), a total of 20 packets are forwarded with only 15 of these being redundant. These 15 redundant (wasted) packets are spread over 3 separate ARs, totalling of 5 x 210bytes = 1,05kBytes of overhead per AR. With a similar latency, NeighborCasting achieves the results equivalent to M=8: a total of 40 forwarded packets, 35 of these being redundant. Therefore PA-FMIP running with optimal prediction parameters achieves a 50% improvement

in forwarding overhead over NeighborCasting.

To compare PA-FMIP to proactive FMIPv6 in the same experiment, the number of packets that each forwards during their respective handover is measured. From these figures, the *cost* of the one scheme is compared to the other. The minimum cost of a 33.8ms PA-FMIP handover (with M=4) can be approximated as $\frac{20}{f}$ times the cost of one proactive FMIPv6 handover, where f is the number of packets proactive FMIPv6 forwards during its handover. In this experiment f=7 packets, therefore PA-FMIP costs at least 2.86 times more than proactive FMIPv6. This overhead is however only imposed on the wired bandwidth. Today, Ethernet links run at speeds of at least 100Mbps, generally increasing towards the core network. Routing these extra packets to their target subnets is relatively inexpensive.

Recall that the number of lost packets, and hence the number of forwarded packets, is proportional to the CBR data rate and the handover latency. It follows that the simulation results in Figure 5.17 are also confirmed analytically by the following equation:

$$P = \frac{\lambda}{pktsize \times 8} \times M \times t_h \tag{5.1}$$

Here the number of forwarded packet (P) is a function of the total streaming bit rate (λ) and packet size, the number of target nARs (M), and the duration of the handover t_h. This equation does not take into account the variable effects of congestion and link loss, nevertheless, it closely models the dynamics involved in this experiment as shown. This is shown in Figure 5.18.

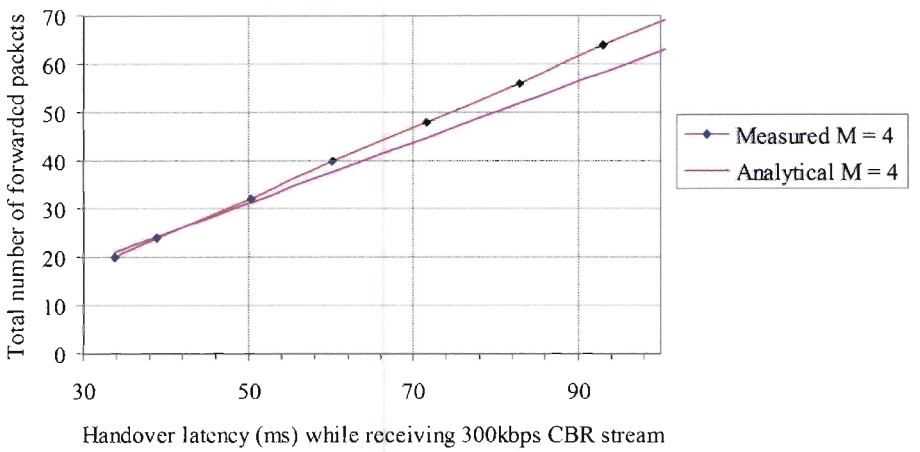

Figure 5.18: Comparison of simulated results to analytical results for M=4.

Discussion

Because of the mobility prediction algorithm, PA-FMIP is 50% more efficient than NeighborCasting, and this is a significant improvement. Compared to the forwarding characteristics of proactive FMIPv6, PA-FMIP becomes more and more expensive (for the wired links only) for higher handover latencies. This has several implications:

- Given the improvement in handover performance, there is an overhead trade-off which could impact the network. Reducing the overhead requires particularly accurate mobility prediction, as well as a relatively short handover latency.
- The data rate of the application is a scaling factor in the overhead projection (Equation 5.1). Thus the type of application used by the mobile node plays a part in where it would be best deployed. Whether it is UDP or TCP based, PA-FMIP would provide a performance advantage compared to FMIPv6.
- The handover latency is affected by the length of the link-layer handover. This has implications on the type of networks that would suit PA-FMIP. For instance, vertical handovers between heterogeneous wireless networks

often take longer to perform, ultimately increasing the total number of forwarded packets. The issue of pre-trigger implementation for proactive FMIPv6 seriously limits its deployment in heterogeneous networks. This is perhaps where the cost of PA-FMIP pays off.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Wireless edge networks have provided users with ubiquitous connectivity to the Internet, and given rise to a number of IP mobility management issues. This study began by introducing these issues. It was identified that one of the principal factors affecting QoS is the service disruption inherent to handovers between networks. A number of fast handover protocols presented in the literature have addressed this issue. Cross-layer communication in the form of link-layer triggers is a common requirement for fast handover protocols to perform adequately. A thorough review of the literature revealed that the practical implementation of pre-triggers is difficult. It was found that a number of schemes in the literature have proposed a variety of methods to minimise the negative effects of handovers. Mobility prediction is one technique that authors have used as a means for streamlining a users transition between networks.

Previous chapters have presented the design and implementation of the PA-FMIP proposal. The mobility prediction algorithm followed a simple data mining approach to analyse the user's previous mobility history and determine the users next location. The objective of this proposal was to determine if in fact mobility prediction can improve the quality of the handover procedure.

A software simulation framework was successfully implemented using the ns-2 simulator. The framework was used to quantitatively evaluate the individual

aspects of PA-FMIP. A number of experiments were performed on topologies resembling real world environments. Results were compared to similar schemes to gauge the achieved performance improvement . Based on findings in preceding chapters, the following conclusions are drawn:

- Given the QoS requirement of real-time applications, the standard Mobile IPv6 protocol cannot support an artifact-free handover. This is in-line with the original claims stated in the literature. Fast handover protocols (PA-FMIP, FMIPv6) achieved latency and packet loss values of up to 3 times less than Mobile IPv6.

- Pure advertisement based movement detection is unsuitable for supporting real-time applications. The periodic intervals between broadcasts are simply too long and affect the performance and stability of the handover. Timeous link-layer triggers are definitely a requirement for future mobile networking devices. It is also an area that requires much research and standardisation given the number of technologies and mobility related protocols.

- Proactivity in mobile communication systems has advantages, especially for performing pre-handover operations such as context transfer. Pre-trigger based FMIPv6 has the potential to support seamless handovers. However where a pre-trigger is available, Simultaneous Bindings should be used instead of proactive FMIPv6 since it supports a smoother handover and mitigates the pre-trigger timing ambiguity problem.

- Data mining is a useful and simple means for mobility prediction. The achievable accuracy is affected by a user's new or random mobility paths. The algorithm was shown to effectively learn a user's mobility patterns, and achieve 100% accuracy for regular mobility or with a trade-off in precision. CPU Processing overheads are a limitation but only for long mobility sessions. This can be managed by correctly configuring the mining parameters. From the the evaluation of the prediction algorithm, it can be concluded that the concept of mobility prediction is a suitable alternative for a pre-trigger.

- The evaluations have demonstrated that the proactive forwarding of a node's data to future location can improve the seamlessness of a reactive

handover. Compared to other schemes, the effect of the forwarding is an improvement in user-perceived real-time quality and an improvement in throughput for file transfers. This is the main contribution of this work.

- The overhead of PA-FMIP is 50% less than the NeighborCasting scheme. The cost of the overhead is limited to wired network bandwidth. We follow the maxim that wired bandwidth is relatively cheap and in abundance. The overhead would only impact the quality of the data stream if the available wired bandwidth is small and bottlenecked, or if the size of the data packet is very large. Real-time media is usually encoded and packetised into small segments for this very reason.

- TCP is a very common protocol and an essential part of the Internet. The effect of mobility on the TCP stream is influenced by the underlying TCP congestion control mechanisms. A comparison of seven TCP variants showed that the choice of TCP impacts on the end users' throughput. A user should make an intelligent selection of TCP in order to maximise download performance. For all the TCP variants in this test it was shown that PA-FMIP outperformed reactive FMIPv6.

- Given the accuracy and performance results of the prediction algorithm, it is advisable that it be deployed in specialised environments such as PCS networks or vehicular networks, i.e., a network environment with regularity and finite mobility boundaries. When considering additional handover packet exchanges such as security association negotiation and AAA[1] context transfer, PA-FMIP is more tolerant than reactive FMIPv6 and is more applicable for mobile users.

6.2 Recommendations and Future Work

This study encompasses a broad spectrum of networking technologies. These range from mobility management protocols and real-time applications, to mobility prediction in wireless networks. While conducting this work, a number of issues

[1] Authentication, Authorisation and Accounting.

and avenues for further research became evident. Listed below is a brief outline of some important recommendations to be considered.

- The concept of mobility prediction was the focus of this work rather than data mining itself. The Apriori Algorithm was chosen here because it is known to be an effective and efficient data mining algorithm. It would however be interesting to determine how it fares against other algorithms in its class. Also, Apriori algorithms used here were all pre-compiled and executed at run time, it is expected that directly customising the Apriori code to the topology / environment itself (and re-compiling) would improve the quality of the predictions.

- The processing and resource consumption of PA-FMIP should be investigated more comprehensively to determine the impact it would have on low power battery operated devices.

- The ping-pong (random) effects of wireless networks were addressed by the prediction algorithm. The handover performance of PA-FMIP however was not subjected to ping-pong cell bouncing. Further tests should investigate this issue since PA-FMIP (and FMIPv6) utilise tunnelling to smooth handover performance. Ping pong movement can be related to extreme movement velocities which is also an important issue since it limits the maximum amount of time the prediction algorithm is allowed to complete its mining operations.

- The accuracy and handover performance testing was done separately for simplicity reasons. Perhaps one could better gauge the performance of PA-FMIP as a whole if the handover performance was recorded over the city section of Topology 1 instead of Topology 2. The use of the RRWP mobility model on this topology would subject PA-FMIP to frequent handovers and also ping-pong mobility. Analysing the handover results would very difficult, but would allow any reliability and scalability issues to be answered.

- The 802.11 link layer handover was emulated in ns-2 by using constant delay period in the code. This served its purpose for the evaluation but a more comprehensive result could have been found if a full implementation

was available. One of the advantages of PA-FMIP over FMIPv6 is that it is not limited to one type of wireless technology, allowing it to perform fast vertical handovers with less trigger complexities. It is recommended that the evaluation environment be expanded to include overlapping heterogeneous wireless networks such as WLAN and 3G, or WLAN and WiMax. This would allow more conclusive results to be found regarding vertical handover performance.

- Security considerations in new protocols are always a great concern and have largely been ignored in this study. Deploying or implementing PA-FMIP as a software kernel patch would first require a detailed security analysis. The forwarding of data streams to foreign networks has Denial of Service and flooding implications if used maliciously. Setting up security associations using *AR Notice* and *AR Ack* messages is a possible solution. Also, migrating through wireless networks is often restricted by authorisation and authentication issues. Cross domain vertical handovers are especially complicated. AAA mechanisms are often lengthy and can affect handover performance if not done proactively.

- The real-time application experiments performed in this work used UDP to emulate encoded video streams. Real-time protocols usually use the Real Time Protocol (RTP). It involves some pre-session signalling to negotiate data rates and QoS but is essentially similar to streaming UDP. To extend the results, future evaluations should consider using RTP.

- It is the author's belief that context transfer is an important subject for handover schemes. Further and more comprehensive testing in this area would uncover its actual impact on handover performance.

- Wireless (lossy) effects on link quality and error rates impact on the end user. A number of types of TCP (called Wireless-TCP) have been proposed specifically for this problem. The focus of this work was on the impact of mobility and handovers. The effects of lossy wireless links however will continue to be an issue in the future, especially for mobile users. Further research should explore this in more detail.

Bibliography

[1] C. Perkins, "Mobility Support for IPv6," *IETF RFC 3775*.

[2] R. Koodli, "Fast Handovers for Mobile IPv6," *IETF RFC 4068*, July 2005.

[3] P. McCann, "Mobile IPv6 Fast Handovers for 802.11 Networks," *IEFT Internet Draft*, November 2005.

[4] K. Malki and H. Soliman, "Simultaneous Bindings for Mobile IPv6 Fast Handovers," *IETF Internet Draft*, May 2003.

[5] A. Kopsel and A. Wolisz, "Voice Transmission in an IEEE 802.11 WLAN Based Access Network," *International Workshop on Wireless Mobile Multimedia*, 2002.

[6] GSMAssociation, "GSM usage statistics http://www.gsmworld.com."

[7] IEEE/ANSI, "IEEE Wireless LAN Standard," tech. rep.

[8] IEEE802, "IEEE 802.16 WiMax Specifications," tech. rep., http://www.ieee802.org/16/.

[9] IEEE802.21, "IEEE 802.21 Generalised Trigger Model for 802 Networks," tech. rep.

[10] A. Yegin, "Link-layer Triggers Protocol ," *IETF Internet Draft*, June 2002.

[11] H. Jang, J. Jee, and Y. Han, "Mobile IPv6 Fast Handovers over IEEE 802.16e Networks," *IETF Internet Draft*, April 2006.

[12] H. Yokota and G. Dommety, "Mobile IPv6 Fast Handovers for 3G CDMA Networks," *IETF Internet Draft*, May 2006.

[13] V. Gupta, "IEEE 802.21 Media Independent Handover Service Draft Technical Requirements," tech. rep., Intel Corporation, July 2004.

[14] D. M. (Editor), "Generic Open Link-Layer API for Unified Media Access - GOLLUM," tech. rep., December 2004.

[15] R. Hsieh and A. Seneviratne, "A Comparison of Mechanisms for Improving Mobile IP Handoff Latency for End-to-End TCP," *MobiCom'03*, vol. San Diego, USA, pp. 14–19, Sept 2003.

[16] E. Shim, H. Wei, and R. D. Gitlin, "Low Latency for Wireless IP QoS with NeighborCasting," *in Proc. ICC 2002*, April 2002.

[17] T. Schmidt and M. Wahlisch, "Performance Analysis of Multicast Mobility in a Hierarchical Mobile IP Proxy Environment," *TERENA Networking Conference '04*, 2004.

[18] X. P. Costa, R. Schmitz, H. Hartenstein, and M. Liebsch, "A MIPv6, FMIPv6, HMIPv6 Handover Latency Study: Analytical Approach," *Proc. of IST Mobile and Wireless Telecommunications*, June 2002.

[19] C. Perkins and R. Koodli, "Fast Handovers and Context Transfers in Mobile Networks," *ACM Computer Communication Review*, October 2001.

[20] T. Patgzis and P. Kirsten, "Proactive Mobility for Future Wireless Access Networks," *Proc. of 6th IASTED/IEEE WOC'02, Banf, Canada*, July 2002.

[21] R. Ludwig and R. H. Katz, "The Eifel Algorithm: Making TCP Robust Against Spurious Retransmissions ," *ACM Computer Communications Review*, vol. 30, January 2000.

[22] K. Xu, Y. Tian, and N. Ansari, "TCP-Jersey for Wireless IP Communications," *IEEE Journal on Selected Areas in Communications*, vol. 22, pp. 747–756, May 2004.

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

[24] L. Song, D. Kotz, R. Jain, and X. He, "Evaluating Location Predictors with Extensive Wi-Fi Mobility Data," *In Proc. InfoCom'04*, vol. 2, pp. 1414–1424, March 2004.

[25] G. Yavas, "A data mining approach for location prediction in mobile environments," *Data and Knowledge Engineering*, vol. 54, pp. 121–146, 2005.

[26] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," *Proc. 20th. Conf. Very Large Data Bases (VLDB'94)*, pp. 487–499, Sept 1994.

[27] R. Agrawal and R. Srikant, "Mining Sequential Patterns," *Proc. IEEE Conf. Data Eng. (IEEE ICDE'95)*, pp. 3–14, March 1995.

[28] A. Nanopoulos, D. Katsaros, and Y. Manolopoulos, "Effective Prediction of Web-user Access: A Data Mining Approach," *in Proc. of the WebKDD Workshop (WebKDD'01)*, 2001.

[29] A. Nanopoulos, D. Katsaros, and Y. Manolopoulos, "A data mining algorithm for generalized web prefetching," *IEEE Trans. Knowl. Data Eng.*, vol. 15, pp. 1155–1169, 2005.

[30] D. Levine, I. Akyildiz, and M. Naghshineh, "A resource estimation and call admission algorithm for wireless multimedia networks using the shadow cluster concept," *IEEE/ACM Transactions on Networking*, vol. 5, pp. 1–12, Feb 1997.

[31] A. Aljadhai and T. Znati, "Predictive mobility support for QoS provisioning in mobile wireless environments," *IEEE JSAC*, vol. 19, pp. 1915–1930, Oct 2001.

[32] X. Shen, J. Mark, and J. Ye, "User mobility profile prediction: An adaptive fuzzy interference approach," *Wireless Networks*, vol. 6, pp. 363–374, 2000.

[33] W. Soh and H. Kim, "Dynamic bandwidth reservation in cellular networks using road topology based mobility prediction," *in Proc. IEEE INFOCOM 2004, Hong Kong*, March 2004.

[34] A. Mishra, M. Shin, and W. Arbaugh, "Context Caching using Neighbor Graphs for Fast Handoffs in a Wireless Network," *IEEE Infocom in Hong Kong*, March 2004.

[35] S. Pack, H. Jung, T. Kwon, and Y. Choi, "A Selective Neighbor Caching Scheme for Fast Handoff in IEEE 802.11 Wireless Networks,"

[36] F. Feng and D. S. Reeves, "Explicit Proactive Handoff with Motion Prediction for Mobile IP," *Proc. of IEEE WCNC'04*, vol. 2, pp. 855–860, March 2004.

[37] N. V. den Wijngaert and C. Blondia, "A Predictive Low Latency Handover Scheme for Mobile IP," *ICMU'05, Osaka, Japan*, April 2005.

[38] A. K. Saha and D. B. Johnson, "Modeling Mobility for Vehicular Ad Hoc Networks," *ACM VANET'04, Philadelphia, USA*, Oct 2004.

[39] B. Goethals, "Survey on Frequent Pattern Mining," tech. rep., HIIT Basic Research Unit, University of Helsinki, http://www.adrem.ua.ac.be/ goethals/software/, 2003.

[40] F. Bodon, "Trie-based Apriori Implementation for Mining Frequent Item-sequences," *OSDM'05*, vol. Bart Goethals and Siegfried Nijssen and Mohammed J. Zaki editors, Chicago, IL, USA, 2005.

[41] Q. Zhao, *Mining Deltas of Web Structure: Issues, Challenges and Solutions.* PhD thesis, Nanyang Technological University, http://www.cais.ntu.edu.sg/ qkzhao/pdf/FYR.pdf, June 2003.

[42] A. S. (Editor), "Candidate Access Router Discovery Protocol," *IETF RFC 4066*, July 2005.

[43] S. McCanne and S. Floyd, "ns2 - Network Simulator." http://www.isi.edu/nsnam/ns/, April 2006.

[44] S. PalChaudhuri, J. L. Boudec, and M. Vojnovic, "Perfect Simulations for Random Trip Mobility Models," *Published at 38th Annual Simulation Symposium, San Diego, California*, April 2005.

[45] USCensus-Bureau, "Tiger/line page." http://www.census.gov/geo/www/tiger/, April 2006.

[46] B. Goethals, "Association Rule Mining Implementation." http://www.adrem.ua.ac.be/ goethals/software/, 2006.

[47] R. Hsieh, "FHMIPv6 extension for ns2." http://mobqos.ee.unsw.edu.au/ robert/nsinstall.php, 2003.

[48] J. Widmer, "Noah extension for ns-2," *http://icapeople.epfl.ch/widmer/uwb/ns-2/noah/index.html.*

[49] D. Miras, "A Survey of Network QoS Needs of Advanced Internet Applications," tech. rep., Computer Science Department, University College London, 2002.

[50] A. Gurtov and R. Ludwig, "Responding to Spurious Timeouts in TCP," *IEEE INFOCOM'03*, 2003.

[51] D. X. Wei, C. Jin, S. H. Low, and S. Hegde, "Fast TCP: Motivation, Architecture, Algorithms, Performance," *IEEE/ACM Trans. on Networking*, to appear in 2007.

[52] Wikipedia, "Online Encyclopedia FAST TCP," *http://en.wikipedia.org/wiki/FAST TCP.*

[53] Y. Tian, K. Xu, and N. Ansari, "TCP in Wireless Environments: Problems and Solutions," *IEEE Radio Communications*, pp. 27–32, March 2005.

[54] M. Allman and A. Falk, "On the Effective Evaluation of TCP," *ACM Computer Communication Review*, October 1999.

[55] A. Yegin, E. Njedjou, S. Veerpalli, N. Montavont, and T. Noel, "Linklayer Event Notifications for Detecting Network Attachments," *IETF Draft*, Feb 2005.

Appendix A

Link Layer Triggers

This appendix describes the extra details relevant to link-layer triggers, their design and implementation in typical mobile networks.

The authors of [55] have discussed how the triggers can be formed from GPRS, CDMA2000 and IEEE 802.11 link layers.

- In GPRS they can be formed from successful activation / deactivation of a PDP Context,
- in CDMA2000 IPV6CP opened / closed state,
- and in IEEE 802.11 a successful association / disassociation with an AP.

The IEEE 802.21 generic trigger model defines a number of generic triggers, these include: Link Up, Link Down, Link Going Down, Link Going Up, Link Quality crossing threshold, Trigger Rollback, Better Signal Quality AP Available. Each of these triggers are described by the 3 types of information: The event that causes the L2 trigger to fire, the IP entity that receives the trigger, and the parameters delivered with the trigger.

The events are usually driven by a particular metric crossing a threshold value. Different models use different metrics, each having to be tuned to the type of access technology. For example, an 802.11 WLAN network may use received signal strength (RSS) as an indicator, while a 3G network may use a bandwidth probe metric or a pre-defined SNR metric with hysteresis threshold.

The following are practical examples of how link-layer triggers are physically implemented in network entities or embedded systems:

- The link-layer driver may allow the IP stack to register an API callback function that is called when the trigger fires.
- The device's operating system may allow a thread to call into a system call for the appropriate trigger.
- The trigger may consist of a protocol for transferring trigger notification and parameter information at between L2 and L3 devices. This allows the IP stack on a separate machine to react to the trigger. The Inter Access Point Protocol (IAPP) protocol and the Link-layer Triggers Protocol [10] are examples of such a protocol.

Appendix B

PA-FMIP Implementation Issues and Procedures for ns-2

B.1 Apriori Implementation

The Apriori frequent sequential itemset mining (Apriori_Seq) by Ferenc Bodon was published as source code as well as a compiled executable - `fsm`. Bodon is an active researcher in Apriori related data mining areas. This frequent sequential itemset mining is available as part of the *fim_ env.tar.bz2* package.

It is possible to edit and recompile the code in Linux. This would be useful to vary the internal components of the algorithm for some custom operations. For this study, using the compiled `fsm` was sufficient.

The Apriori Rule mining algorithm used in this work was published by Bart Goethals in *rules.tar.gz*.

B.2 RRWP in ns-2

The restricted random waypoint model takes TIGER street map info along with desired mobility data as inputs. In this work, the TIGER data pertaining to West University Area is used. The *WestUnivPlace.dat* file is obtainable from the TIGER/line website [45]. The RRWP model generate the mobility patterns for

one or more mobile nodes. These patterns are in the form of `setdest` coordinates and speeds that correspond to the streets of the (TIGER captured) city section. The model is presented as a compiled Linux application called Palm. It may generate a variety of types of mobility patterns (such as random walk mobility), the city section mobility in this study is handled by "Palm/Graph".

The input parameters mould the nodes mobility behaviour. Besides the simulation time, all the other input parameters were constant. An example of the Palm/Graph (RRWP) mobility pattern generator is as follows:

```
# Bash Usage:
#./graph <input file> <number of lines in input file>
          <pause time mean> <pause time delta>
# <number of nodes> <simulation time (sec)> <output file>
input_file=westUnivPlace.dat
num_lines_in_file=594
pause_mean=1
pause_delta=0
num_nodes=1
sim_time=1000
output_file=<path>/"$input_file"_"$num_nodes"_MN_"$sim_time".txt
```

The resultant output file consists of hundreds of ns `setdest` commands. This command sets the destination (coordinates on ns-2 topology) and speed (in m/s) much like a random waypoint model would, except sequential coordinates all fall within the confines of the street map:

```
$ns_ at 260.000000 "$node_(0) setdest 356.587006 167.162003 4.839399"
```

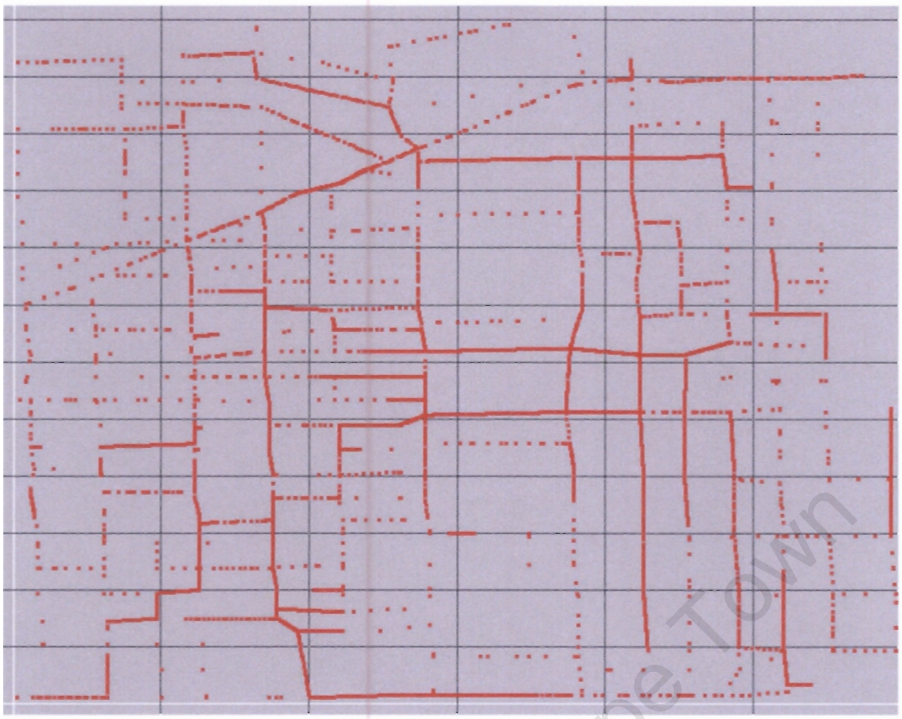

Figure B.1: Example of recorded mobility of a single MN over the simulation topology. Graph drawn using Xgraph.

B.3 Fast Handover code

The installation of *fhmipv6* in `ns-allinone-2.27` requires the manual patching of the following files in ns-2:

mip.cc mip.h mip-reg.cc mip-reg.h ns-default.tcl ns-agent.tcl ns-mip.tcl ns-lib.tcl ns-node.tcl ns-packet.tcl packet.h

This includes two new files to be added to the `/ns-2.27/mobile` directory as well as the Makefile:

fasthandover.cc and fasthandover.h

The NOAH routing agent by Widmer is also needed to be included and compiled as part of *fhmipv6*:

noah.cc, noah.h and noah.tcl

The *fhmipv6* package was originally coded for installation using `ns-allinone-2.1.7b`. Using a newer version of ns-2 was necessary because the older versions were plagued with bugs and compile errors for certain application modules.

Most of the fast handover functionality is contained in *mip-reg.cc*. This file contains definitions for `MIPHeader`, `MIPBSAgent` and `MIPMHAgent`. The `MIPHeader` defines the new MIP packet. The `MIPMHAgent` represents the mobile node. The `MIPBSAgent` defines how an access router and home agent behave depending on what type of message is received and from whom it was sent.

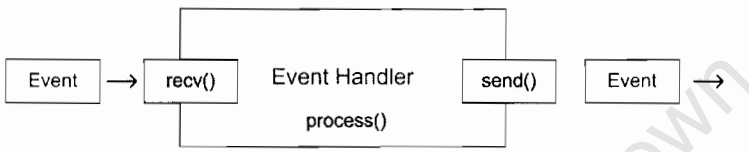

Figure B.2: Event processing in ns-2.

Recall that ns-2 is a discrete event simulator. This means that a change of state only happens at a discrete point in time. Events are the basic components that change the system state. Event handlers process events in a structured manner; Agents, Nodes, and links are examples of ns-2 event handlers.

For the fast handover code, we deal with the MIP agents. Each of which will have a `recv()` function. In ns-2, packets are handled from entity to entity. This means that packets are not 'sent' down a link, but rather 'called' by the link's `recv()` function. This function basically consists of a large switch statement operating on the type of message that is received. The message type is a field in the received packet. So based on the type of received message, the agent performs tasks relating to the reception of the message. This structure is illustrated as follows:

```
void MIPBSAgent::recv(Packet* p, Handler *)
{
...
switch (miph->type_)
{
...
case FAST_RTSOLPR:
 {...
  iph->daddr() = MN; //destination address of MN
  iph->dport() = 0;
  iph->saddr() = addr();
  iph->sport() = port();
  miph->type_ = FAST_PRRTADV;
  ch->num_forwards() = 0;
  ragent_->recv(p, (Handler*)0); //Pass packet to MN's recv().
 }
...
}
```

This is an intelligent code structure. Code segments are separated by `#ifdef` and `#ifndef` operators that only permit certain parts of code to be executed if a particular string is defined at the beginning of the code (e.g. `#define FAST_REACTIVE` for reactive FMIPv6 only). This allows the MIPv6, Proactive FMIPv6, Reactive FMIPv6, PA-FMIP and Simultaneous Bindings to be written in the same file but only compiled independently of one another.

B.4 Tunnel management in ns-2

Setting up tunnels in C requires the use of oTcl functions to access Tcl functions at runtime. The following sets up a tunnel between the MN's current AR and a new AR at the address `m_coa_`.

```
tcl.evalf("%s decap-route-encap %d %s %lf", name_, miph->haddr_,
        objname, miph->lifetime_);
tcl.evalf("%s tunnel-exit %d %d %lf", name_, miph->haddr_, m_coa_,
        miph->lifetime_);
```

The decapsulation is implemented by the following:

```
tcl.evalf("%s decap-route %d %s %lf", name_, miph->haddr_,
        objname, miph->lifetime_);
```

Appendix C

802.11b Configuration

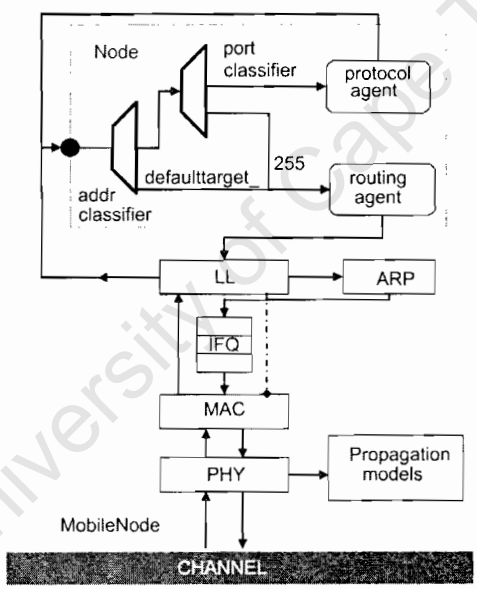

Figure C.1: General schematic of a mobile node in ns-2 [43].

The network stack for an 802.11 enabled mobile node consists of the components defined below.

Link Layer The ARP module is connected to LL resolves all IP to hardware (Mac) address conversions. Normally for all outgoing (into the channel) packets, the packets are handed down to the LL by the Routing Agent. The LL hands down packets to the interface queue. For all incoming packets (out of the chan-

nel), the mac layer hands up packets to the LL which is then handed off at the node_entry_ point.

ARP The Address Resolution Protocol (implemented in BSD style) module receives queries from Link layer. If ARP has the hardware address for destination, it writes it into the mac header of the packet. Otherwise it broadcasts an ARP query, and caches the packet temporarily. For each unknown destination hardware address, there is a buffer for a single packet. Incase additional packets to the same destination is sent to ARP, the earlier buffered packet is dropped.

Interface Queue (IFQ) The class `PriQueue` is implemented as a priority queue which gives priority to routing protocol packets, inserting them at the head of the queue. It supports running a filter over all packets in the queue and removes those with a specified destination address.

Mac Layer The IEEE 802.11 distributed coordination function (DCF) Mac protocol has been implemented by CMU. It uses a RTS/CTS/DATA/ACK pattern for all unicast packets and simply sends out DATA for all broadcast packets. The implementation uses both physical and virtual carrier sense.

Network Interfaces (PHY) This layer serves as a hardware interface which is used by the mobile node to access the channel. The wireless shared media interface is implemented as class `Phy/WirelessPhy`. This interface subject to collisions and the radio propagation model receives packets transmitted by other node interfaces to the channel. The interface stamps each transmitted packet with the meta-data related to the transmitting interface like the transmission power, wavelength etc. This meta-data in `pkt` header is used by the propagation model in receiving network interface to determine if the packet has minimum power to be received and/or captured and/or detected (carrier sense) by the receiving node. The model approximates the DSSS radio interface (LucentWaveLan direct-sequence spread-spectrum).

Radio Propagation Model It uses Friss-space attenuation ($l=r^2$) at near distances and an approximation to Two ray Ground ($1=r^4$) at far distances. The approximation assumes specular reflection off a flat ground plane.

Antenna An omni-directional antenna having unity gain is used by mobilenodes.

The resulting node configuration for each MN and AR in the simulations are configured as follows using Tcl:

```
set opt(chan) Channel/WirelessChannel ;# channel type
set opt(prop) Propagation/TwoRayGround ;# radio-propagation model
set opt(netif) Phy/WirelessPhy ;# network interface type
set opt(mac) Mac/802_11 ;# MAC type
set opt(ifq) Queue/DropTail/PriQueue ;# interface queue type
set opt(ll) LL    ;# link layer type
set opt(ant) Antenna/OmniAntenna ;# antenna model
set opt(ifqlen) 50 ;# max packet in ifq
set opt(nn) 1     ;# number of mobilenodes
set opt(adhocRouting) NOAH ;# routing protocol
set opt(cp) ""    ;# connection pattern file not used
set opt(sc) ""    ;# node movement set manually.
set opt(x) 1000   ;# x coordinate of topology
set opt(y) 1000   ;# y coordinate of topology
set opt(seed) 0.0 ;# random seed
set opt(stop) 100 ;# time to stop simulation
```

C.0.1 Configuring Radio Range

How to determine the desired propagation threshold parameters for 802.11b, the following line is run from "~/ns-2/ns-allinone-2.27/ns-2.27/indep-utils/propagation":

```
>> ./threshold -m TwoRayGround -fr 2.4ghz 140m
```

Values for Pt_ (Power) and RXThresh_ (Receive threshold) are outputted:

```
Pt_ = 0.281838
RXThresh_ = 3.71409e-09
```

These values are used in the main Tcl simulation script as follows:

```
# Power and RxThresh set for Range =140m
Phy/WirelessPhy set Pt_ 0.281838 ;

# Freq set at 2.4Ghz default
Phy/WirelessPhy set RXThresh_ 3.71409e-09 ;
```

Appendix D

Topology Design Considerations and Procedures

The topology of Topology 1 was created using Topoman. Topoman is bundled with the Mobiwan package, however it can be used separately. As shown in D.1, the 36 subnet are bundled into 4 sites, connected in a ring formation through intermediate routers. Base Station 0 also functions as the corresponding node.

Topoman (tm) is executed in the following manner:

```
$tm_all_nodes(0) label [$TOPOM get_addr_by_id 0] ;#"Node 0 (CN)"
$tm_all_nodes(0) color red
$tm_all_nodes(0) shape "box"
$tm_all_nodes(1) label [$TOPOM get_addr_by_id 1]
$tm_all_nodes(1) color cyan
$tm_all_nodes(1) shape "hexagon"
...
```

This creates the actual network nodes. Each node however requires a hierarchical ns-2 address as well as a physical network link. With the help of topoman, the assignment of addresses for Topology 1 is relatively simple. The hierarchical addresses for Topology 2 were assigned manually. The exact configuration of Topology 1can be found in *topofile.tcl.*

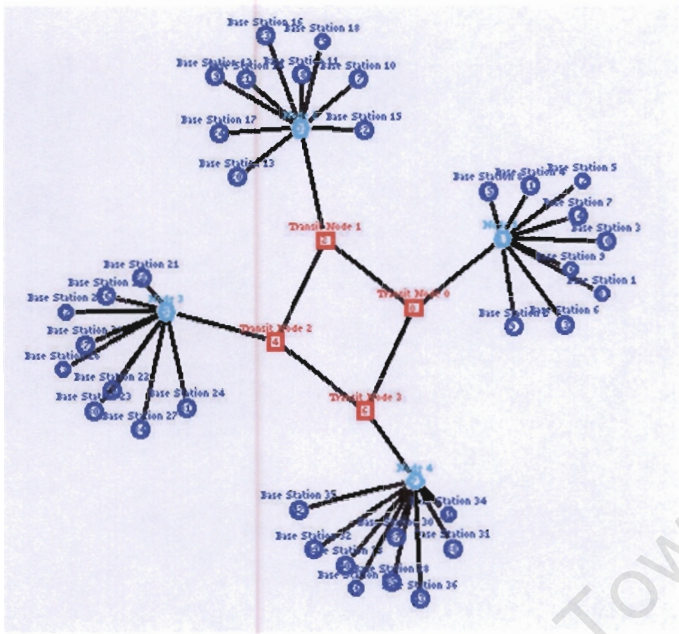

Figure D.1: A large IPv6 network of 36 access routers forming four domains with a total of 36 subnets. Image captured from the Network Animator (NAM) display.

The duplex wired links between the base stations, as well as their positions on the square topology are also configured in *topofile.tcl*.

The 1200x1200m area is split up into four (Cartesian) quadrants for simplicity. Here, 1 unit on the simulation topology is equivalent to 1m. One must understand the layout in D.1 is not spatial (because of the limitations of the NAM animator). The spatial position of the nodes is managed in the code with X and Y coordinates.

```
###########(1200,1200)
####################
###### Q3 | Q2 ######
###### -------#######
###### Q0 | Q1 ######
####################
#(x=0,y=0)###########
```

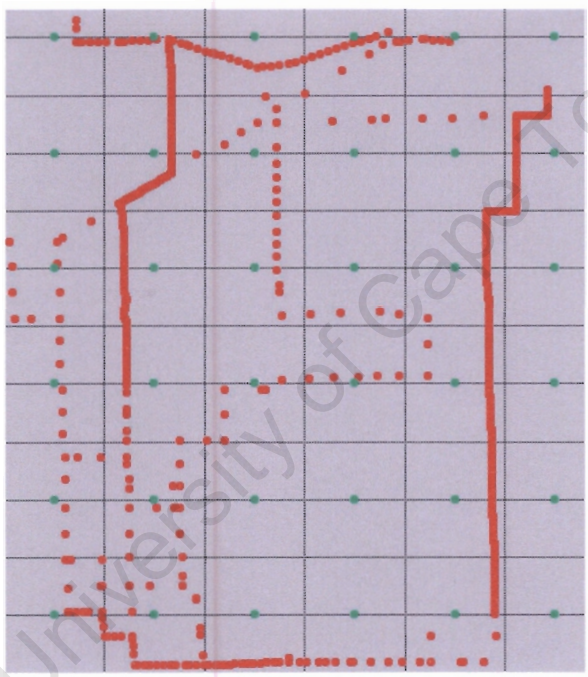

Figure D.2: Layout of 36 ARs on Topology 1. The mobility (red) of a single node is shown over the AR grid (green). Screen captured from NAM output.

D.1 Node Position and Range Values

Topology 1 is 1200x1200m covered by 36 access routers. Simple planning of the layout lead to their 200m separation. In order for the MN to not lose wireless connectivity as it traverses the topology, the circular radio ranges of the ARs need

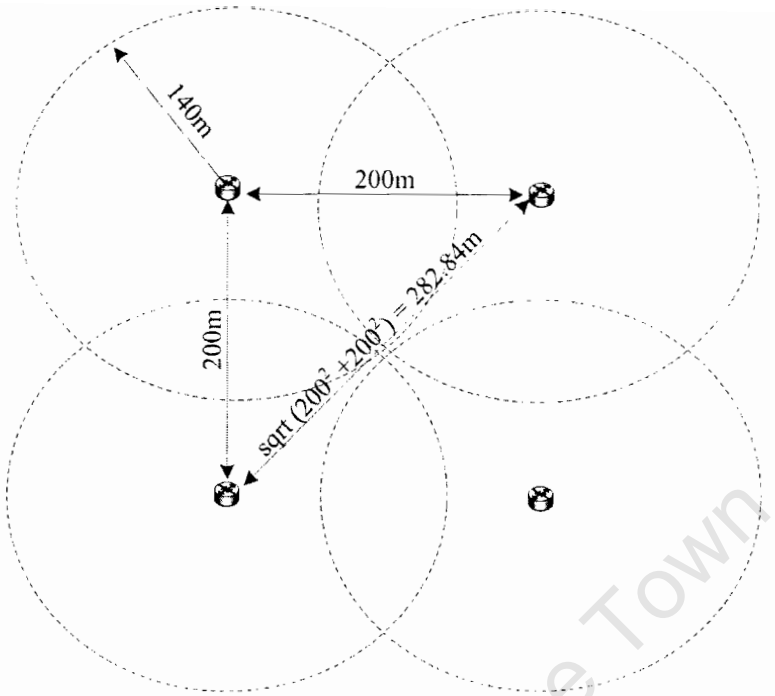

Figure D.3: Derivation of a suitable radio range for the ARs.

to overlap slightly. As shown in Figure D.3, an AR range of 140m is sufficient to achieve this as long as the MN also has a range of at least 140m.

Appendix E

Recording and Analysing ns-2 Trace Data

Ns-2 simulations are written in a script format using the Tcl language. The network topology, nodes, simulation settings, protocol applications and file outputs are specified in a particular order. Ns-2 outputs all simulation events to a trace file. Extra simulation tools such as NAM use the trace to animate the simulation scenario. To collect specific data or results from the trace log AWK scripts are used. The AWK language is simple and provides a useful means to process and filter large amounts of data. These AWK files (and other files such as Perl, Bash or grep scripts) can be executed from the Tcl script at the end of the simulation.

See *fsimple.tcl* for the simulation configuration and details.

The events in the trace file have a specific format. Timestamps, packet ID numbers, and sequence numbers are the most useful parameters used when analysing ns-2 traces. Results are calculated according to the a particular metric (such as latency / throughput). An example of the trace format is:

```
r -t 53.740974752 -Hs 5 -Hd -2 -Ni 5 -Nx 313.70 -Ny 45.00 -Nz 0.00
   -Ne -1.000000 -Nl MAC -Nw --- -Ma 95e -Md 3 -Ms 1 -Mt 0
```

Additional text information is outputted to the command prompt during the simulation. This is usually ns-2 status information or error notifications. C++

commands such as printf or putchar also print to the screen. This data can be piped to a separate file and analysed using more AWK files.

Each ns-2 simulation is executed from the Linux shell prompt. For example, the handover experiment tcl file is executed as follows:

```
ns fsimple.tcl > info
```

At the end of the simulation, a number of AWK scripts are automatically executed, and results are collected and displayed.

Appendix F

Ns-2 Applications

F.1 TCP for ns-2

The TCP variants used in this study were all part of the `ns-allinone-2.27` distribution except Fast-TCP. Fast-TCP, published by the California Institute of Technology (Caltech), was relatively simple to install by following step-by-step instructions as set by the authors. For more information on Fast-TCP see the comprehensive documentation in [52] and [51].

Setting up the FTP application in the ns-2 simulation script (fsimple.tcl) was done simply as follows:

```
proc set-tcp { } {
   global ns_ MN N opt ftp1-start
set tcp_(1) [$ns_ create-connection TCP/Newreno $N(0)
                                    TCPSink/Sack1 $MN 1]
$tcp_(1) set window_ 32
$tcp_(1) set packetSize_ 512
 # RCH Setting connection monitor - to compensate for 802.11.
$ns_ connection-monitor 1 $MN
set ftp_(1) [new Application/FTP]; #Create new application
$ftp_(1) attach-agent $tcp_(1);    #Attach FTP to TCP agent
$tcp_(1) set close_on_empty_ true
 # Start sending 3Mb file
$ns_ at 25.0 "$ftp_(1) send $send_size"
#$ns_ at 1.0 "$ftp_(1) start"
#$ns_ at 99.0 "$ftp_(1) stop"
}
```

As shown, the FTP application is between node 0 (CN) and the MN. The different versions of TCP are easily alternated between simulations. The source (`TCP/..`) and sink types (`TCPSink/..`) are specified here.

The connection monitor function is used to make sure the TCP agent does not receive packets during particular points in a link handover. As explained in Chapter 4, this is a work-around (introduced by Hsieh) for the incomplete implementation of the ns-2 802.11 module.

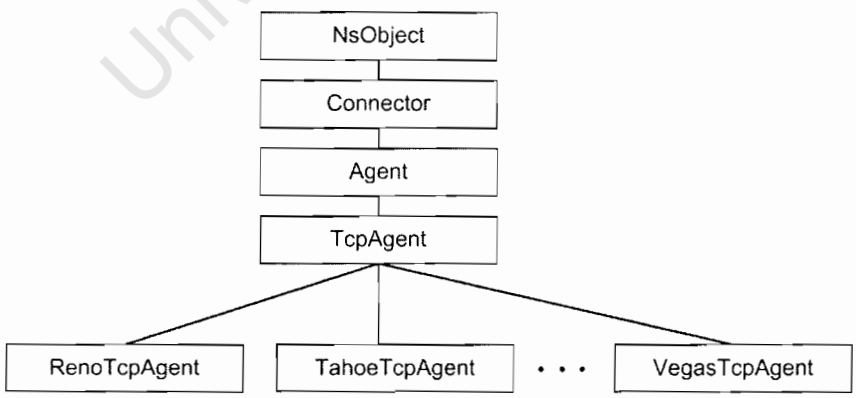

Figure F.1: TCP agents in ns-2.

F.2 UDP in ns-2

The setup of the real-time emulation through UDP was done as follows:

```
proc set-cbr { } {
  global ns_ N MN
set udp_(1) [$ns_ create-connection UDP $N(0) Null $MN 1]
# RCH  Setting connection monitor - to compensate for 802.11.
$ns_ connection-monitor 1 $MN
set src [new Application/Traffic/CBR]
$src set packetSize_ 210 ;#bytes
$src set rate_ 300k      ;#bps
$src set random_ 0       ;#no random noise
$src attach-agent $udp_(1)
$ns_ at 20.0 "$src start"
}
```

Again, the connection monitor was required to drop packets while the node performs a link change or according to the characteristics of the mobility management protocol. This simulation setup is meant to emulate a high data rate application such as IPTV or similar. For a strict VoIP application, the UDP setup would look as follows:

```
proc set-VoIP { } {
   global ns_ N MN
#load =  Mean burst time / (Mean inter-arrival time)
#     =  Mean burst time / (Mean burst time  +  mean idle time)
   set load 0.6
   set udp_(1) [$ns_ create-connection UDP $N(0) Null $MN 1]
# RCH  Setting connection monitor - to compensate for 802.11.
$ns_ connection-monitor 1 $MN
set expoo [new Application/Traffic/Exponential]
$expoo attach-agent $udp_(1)
$expoo set packetSize_ [expr 160+40]; #data + header
$expoo set burst_time_ 180
$expoo set idle_time_ [format %.1f
                      [expr [$expoo set burst_time_]*(1/$load-1)]]
$expoo set rate_ 64000
$ns_ at 20.0 "$expoo start"
}
```

This code models the exponential distribution of inter-arrival times for a typical (packetised) VoIP data stream. The burst and idle times of the application can be adjusted, as well as the 64kbps transmission rate. The 300kbps CBR approach taken in our evaluation provides more flexibility in terms of the different types of streaming multimedia applications, and is in-line with the vision of next generation networks and applications.

Appendix G

Source Code for Simulation Experiments

G.1 Mobility Prediction Tests

G.1.1 Topology

File name: topofile.tcl

G.1.2 Accuracy and Precision

File names: working-sim.tcl, prediction.cc, prediction.h

G.2 Handover Performance Tests

G.2.1 Fast handover protocol implementations

File name: mip-reg.cc, mip.cc, mip.h

G.2.2 Tcl scripts

File name: fsimple.tcl

G.2.3 Result collection

TCP

File names: plot_seq.pl,

tcp_duplicatedpkts_newtrace_extract.awk,

tcp_seqnum_newtrace_extract.awk,

total_dropped_pkts_awk.awk,

total_dropped_pkts_during_1_HO_awk.awk,

calc_tcp_goodput_from_perl.awk,

calc_tcp_segments_sent_from_0.awk

UDP

File names: plot_udp_seq.pl, cbr-jitter-awk.txt,

udp_seqnum_newtrace_extract_from_recv5.awk,

total_number_of_fwded_pkts_awk.awk,

total_dropped_pkts_awk.awk,

total_dropped_pkts_during_1_HO_awk.awk

Appendix H

Publications

The work in this thesis has been accepted for publication in the following conferences:

A. E. Bergh and N. Ventura, "Prediction Assisted Fast Handovers for Mobile IPv6", *Proceedings of SATNAC' 06*, Sept. 2006.

A. E. Bergh and N. Ventura, "PA-FMIP: A Mobility Prediction Assisted Fast Handover Protocol", *To Appear in Proceedings of IEEE MILCOM' 06*, Washington D.C., USA, Oct. 2006.

A. E. Bergh and N. Ventura, "Prediction Assisted Fast Handovers for Seamless IP Mobility", *To Appear in Proceedings of IEEE CCNC'07*, Las Vegas, USA, Jan. 2007.

Appendix I

Accompanying CDROM

The accompanying CDROM is located on the inside back cover of this document. The contents of the CDROM are as follows:

- A soft copy of this thesis in PDF format
- The source code files of the evaluation framework
- Relevant publications used during the research of this thesis.