INVESTIGATING LATENCY IN TWO WAYS COMMUNICATION OF VOIP IN WLAN

FAKRULRADZI BIN IDRIS
SHARIFAH HAFIZAH SYED ARIFFIN

APRIL 2009

ABSTRACT

Voice over Internet Protocol (VoIP) has stringent delay requirement. The quality of VoIP does not yet match the quality of a circuit-switched telephone network due to several challenges such as available bandwidth, delay or network latency, packet loss, jitter, echo, security and reliability. International Telecommunication Union – Telecommunications Standards Sector (ITU-T) recommends a certain maximum delay which tolerable to the user of VoIP service. To improve the performance of VoIP during handoff, handoff latency must be kept as low as possible. There are numerous method proposed to improve the performance of real time application such as VoIP during handoff. One of the methods is Mobile IP, which provides mobility management in the network layer of TCP/IP protocol. In this thesis, the operations and handoff procedure for standard Mobile IPv6 (MIPv6) are studied in detail. Fast Handovers for MIPv6 (FMIPv6) which was introduced as an extension of MIPv6 is used to improve the performance of VoIP application. In this project, the implementation of MIPv6 and FMIPv6 are modeled and simulated using Network Simulator 2 (NS2). Three different voice coding schemes namely G.711, G.723.1 and G.729 are studied to determine their effects on the handoff latency and other Quality of Service (QoS) parameters such as packet loss, throughput and delay.

ABSTRAK

‘Voice over Internet Protocol’ (VoIP) mempunyai keperluan untuk lengah masa yang sangat sensitif. Kualiti VoIP masih belum standing dengan kualiti rangkaian telefon ‘circuit-switched’ disebabkan oleh beberapa halangan seperti keadaan ‘bandwidth’, lengah masa atau kelambatan rangkaian, kehilangan packet ‘jitter’, ‘echo’, keselamatan dan ‘reliability’. International Telecommunication Union – Telecommunications Standards Sector (ITU-T) menggariskan nilai lengah masa yang tertentu yang dapat diterima oleh pengguna servis VoIP tanpa gangguan. Untuk meningkatakan prestasi VoIP ketika ‘handoff’ berlaku, lengah masa yang disebabkan oleh ‘handoff’ perlu dipastikan seminima yang boleh. Terdapat pelbagai kaedah yang dicadangkan untuk meningkatkan prestasi applikasi ‘real time’ seperti VoIP ketika ‘handoff’ berlaku. Salah satu kaedah ialah ‘Mobile IP’ yang memberikan pengurusan pergerakan di lapisan ‘network’ protocol TCP/IP. Dalam thesis ini, operasi dan prosedur ‘handoff’ bagi ‘Mobile IPv6’ (MIPv6) yang piawai dipelajari secara mendalam. ‘Fast Handovers for MIPv6’ (FMIPv6) yang diperkenalkan sebagai sambungan kepada MIPv6 digunakan untuk meningkatkan prestasi applikasi VoIP. Dalam projek ini, implementasi MIPv6 dan FMIPv6 adalah dengan cara dimodelkan dan disimulasi menggunakan Network Simulator 2 (NS2). Tiga skim ‘voice coding’ iaitu G.711, G.723.1 dan G.729 dianalisa untuk menentukan kesannya kepada ‘handoff latency’ dan parameter Kualiti Perkhidmatan (QoS) seperti kehilangan paket, ‘throughput’ dan lengah masa.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

3G	-	Third Generation
AAA	-	Authentication, Authorization and Accounting
ACELP	-	Algebraic Code Excited Linear Prediction
AP	-	Access Point
AR	-	Access Router
BA	-	Binding Acknowlegement
BU	-	Binding Update
BSS	-	Basic Service Set
CBR	-	Constant Bit Rate
CELP	-	Code Excited Linear Prediction coding
CN	-	Corresponding Node
CoA	-	Care of Address
CoT	-	Care-of Test
CoTI	-	Care-of Test Init
CS-ACELP	-	Conjugate Structure – ACELP
DAD	-	Duplicate Address Detection
DS	-	Distribution System
ESS	-	Extended Service Set
FBU	-	Fast Binding Update
FBAck	-	Fast Binding Acknowledgment

FHMIP	-	Fast Hierarchical Mobile IP
FMIPv6	-	Fast Handover for MIPv6
FNA	-	Fast Neighbor Advertisement
GHz	-	Giga Hertz
GPRS	-	General Packet Radio Service
HMIP	-	Hierarchical Mobile IP
HoT	-	Home Test
HoTI	-	Home Test Init
IP	-	Internet Protocol
IPv6	-	Internet Protocol version 6
ISP	-	Internet Service Provider
ITU	-	International Telecommunications Union
L2	-	Layer 2
L3	-	Layer 3
MAC	-	Media Access Control
MAP	-	Mobility Anchor Point
MGCP	-	Media Gateway Control Protocol
MIPv6	-	Mobile IP version 6
MN	-	Mobile Node
MOS	-	Mean Opinion Score
MTU	-	Maximum Transmission Unit
NAM	-	Network Animator
NA	-	Neighbor Advertisement
NAR	-	New Access Router
NCoA	-	New Care of Address
NS	-	Neighbor Solicitation

NS2	-	Network Simulator 2
OTcl	-	Object-oriented Tcl
PAR	-	Previous Access Router
PCM	-	Pulse Code Modulation
PCoA	-	Previous Care of Address
PrRtAdv	-	Proxy Router Advertisement
PSTN	-	Public Switched Telephone Networks
QoS	-	Quality of Service
RA	-	Router Advertisement
RFC	-	Request For Comments
RR	-	Return Routability
RS	-	Router Solicitation
RTP	-	Real-Time Protocol
RTCP	-	Real-Time Control Protocol
RtSolPr	-	Router Solicitation for Proxy Advertisement
SCCP	-	Skinny Client Control Protocol
SCTP	-	Stream Control Transport Protocol
SIP	-	Session Initiation Protocol
TC	-	Traffic Class
TCP	-	Transmission Control Protocol
UDP	-	User Datagram Protocol
UMTS	-	Universal Mobile Telecommunication System
VoIP	-	Voice over Internet Protocol
WLAN	-	Wireless Local Area Network

LIST OF SYMBOLS

P_r	-	Received power
P_t	-	Transmitted power
G_t	-	Gain of transmitting antenna
G_r	-	Gain of receiving antenna
h_t	-	Height of transmitter
h_r	-	Height of receiver
d	-	Distance from transmitter
L	-	System loss

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Voice over Internet Protocol (VoIP) is also known as IP Telephony which enables the transport of voice over data networks such as the Internet. VoIP became a workable alternative to the public switched telephone networks (PSTN) and increasingly deployed on corporate environment and campuses. A number of protocols are used to ensure that voice communication is appropriately established between users and that voice is transmitted with a quality close to as in PSTN. VoIP involves digitization of voice streams and transmitting the digital voice as packets over conventional IP-based packet networks like the Internet. The quality of VoIP does not yet match the quality of a circuit-switched telephone network due to several challenges such as available bandwidth, delay or network latency, packet loss, jitter, echo, security and reliability.

In a mobile Internet environment, when a mobile node (MN) intends to move and attach to another network, it needs to obtain a new IP address to continue communications with its correspondents. The IP routing mechanism relies on the information found in IP headers so that they can deliver data to the proper nodes, thus a movement from one location to another requires the old IP connections to be tear down and new connections to be reconstructed. Mobile IP (versions 4 and 6) provides a solution to overcome this problem without major modifications to the routers or the nodes in a network.

In Mobile IPv6 protocol, each mobile node is identified by a set of IP addresses. In the home network, a Home Agent (HA) assigns a local address to the mobile node and it is always reachable via its HA. However, when the node is away from its home, it obtains a Care of Address (CoA) from the foreign router and registers this CoA with its HA. The function of the HA is to intercept any packets destined for the mobile node while it is roaming in a foreign network and tunnel it to the mobile node. The inherent problem in this scenario is that, a timely configuration of CoA is required for continuous communication.

Handoff latency is defined for a receiving MN as the time that elapses between the last packet received via the old route and the arrival of the first packet along the new route after a handoff [1]. Latency is an important parameter for delay-sensitive applications like VoIP that could suffer from a period with a higher rate of packet drops due to a long latency time. This packet drop period would result in a noticeable disruption in the voice transmission

1.2 Problem Statement

Real time application such as VoIP is very sensitive to delay. The inefficiency of Mobile IPv4 (MIPv4) is due to triangle routing and the limited address spaces. Route optimization is introduced in Mobile IPv6 (MIPv6) to avoid triangle routing but the handoff latency in Mobile IPv6 is still so long that can result in packet loss and service disruption. There are several enhancements introduce to reduce the handoff latency. Fast Handovers for Mobile IPv6 (FMIPv6) is one of the method. This thesis will focus on the link layer and network layer of TCP/IP protocol in reducing the handoff latency to further improve the performance of VoIP during handoff.

1.3 Objectives

Base on the problem statements, the objectives of this study are

(i) To study the performance of voice over Internet Protocol (VoIP) during handoff in MIPv6.

(ii) To reduce the handoff delay as well as packet loss occurred in MIPv6 during handoff process using FMIPv6.

(iii) To analyze and compare the performance of VoIP in MIPv6 and FMIPv6.

1.4 Scope of Project

This project revolves around the performance of VoIP in Mobile IPv6 when handoff occurs. Research is carried out to identify factors contribute to handoff latency when mobile node moves from one network to another network Investigation on handoff latency in link layer (L2) and network layer (L3) of TCP/IP model will be done. The handoff procedure in basic Mobile IPv6 will be examined in more detail. Analysis and improvement of VoIP performance will be carried out using Fast handovers for MIPv6 (FMIPv6) with predictive mode approach to reduce handoff latency.

In this project, intra domain handoff process or micro mobility model will be focused. The Mobile IP network scenario using IEEE 802.11b protocol that simulated VoIP application between mobile node and corresponding node will be modeled. Three different voice coding scheme which are G.711, G.723.1 and G.729 are used. Network Simulator 2 (NS2) is be use to simulate the handoff operation using different mobility scheme. The simulation of MIPv6 architecture model is compared to FMIPv6. The simulation results are observe on the changes, or improvement and then lead to the conclusion and discussion on both methods.

1.5 Thesis Structure

This thesis is divided into 5 chapters. The first chapter introduces the problem statement, objectives, and scope of the project. The second chapter includes the details of the background, and the concepts of Voice over Internet Protocol are also discussed. Brief concepts of User Datagram Protocol and Internet Protocol version 6 (IPv6) are also described. Moreover, this chapter explains the different types of handoff. The end of chapter two describes some related works primarily on researches concerning Mobile IP and handoff.

The project methodology is presented in chapter three. This chapter describes in the detail of Mobile IPv6 and handoff procedures. FMIPv6 operation which will be used to improve the performance of VoIP is illustrated. The network model and simulation parameters for MIPv6 and FMIPv6 framework are presented in this chapter.

Analyses of the results from the NS2 simulation are described in chapter four. The results for MIPv6 and FMIPv6 are discussed and comparisons are made in order to obtain any performance improvement by the using FMIPv6.

Finally, chapter five presents the conclusions of this thesis and also proposed future work. The rest of this thesis includes the list of references and the appendices used in completing this project.

Packet based networks have developed rapidly during last decades. Currently, people are using more and more data transfer as digital the world is developing. Data transfer traffic has increased in volume compared with traditional voice traffic. This progress has led to the convergence of telecommunications and data communications networks. This also means that telecommunications networks are replaced with modern data communications networks and this too has led to the need of transmitting voice calls over Internet Protocol [2].

VoIP means that calls are transmitted over an IP network such as the Internet instead of Public Switched Telephone Networks (PSTN). Since access to the Internet is available at more and more places in the world, it is possible to use VoIP in a higher degree. VoIP converts standard telephone voice signals into compressed data packets that can be sent over IP. Before transmitted over packet switched networks, the speech signal has to be digitized at the sender; the reverse process is performed at the receiver. The digitalization process is composed of sampling, quantization and encoding. There are many encoding techniques that have been developed and standardized by the ITU such as G.711, G.729 and G.723.1 [3]. The encoded speech is then packetized into packets of equal size. Each such packet includes the headers at the various protocol layers such RTP (12 bytes), UDP (8 bytes), IP (20 bytes), 802.11 (34 bytes) and the payload comprising the encoded speech for a certain duration depends on the codec deployed.

As the voice packets are sent over IP networks and wireless channel, they incur variable delay and possibly loss. In order to provide a smooth playout delay, at the receiver, a playout buffer is used to compensate the delay variations. Packets are held for a later playout time in order to ensure that there are enough packets buffered to be played out continuously.

2.1.1. Voice Quality Indicators

The quality of a telephone conversation depends on low latency, jitter and packet loss as shown in Table 2.1 [4]

Table 2.1 Voice Quality Indicators

Characteristics	Definition	Recommendation	Source
Latency/Delay	The time it takes for data to get from one point to another point on the network. VoIP is sensitive to delay because conversations occur in real time.	Below 150 ms (one way)	IT-T G.114
Jitter	Jitter is the variance in delay. When some voice packets arrive with little delay followed by additional voice packets with greater delay, parts of the conversation on the receiving end will become uneven.	Below 40 ms	National Institute of Standards and Technology
Packet loss	Dropped packets. Voice typically uses the Real-Time Protocol (RTP) running over User Datagram Protocol (UDP), which doesn’t retransmit lost packets; meaning portions of the conversation can be lost	Below 1 % and 3%	National Institute of Standards and Technology

In a network without enough bandwidth or with other hidden performance problems, voice calls can quickly become corrupted. The challenge is to ensure that voice and data services function efficiently and harmoniously in the same network, and ultimately, deliver a quality of service that will delight the end user.

2.1.2 VoIP Protocols

There are several VoIP products available today. These products are usually developed to conform to one or more of the following standards or protocols:

(1) H.323: The International Telecommunications Union (ITU) designed H.323 to define how multimedia such as video and audio travel over a packet-switched network.

(2) MGCP: Media Gateway Control Protocol (MGCP) runs in conjunction with other Internet Protocols (IP) such as H.323 or Session Initiation Protocol (SIP) to bridge circuit switched and packet switched networks. It enables IP endpoint such as analog phones to connect to an IP backbone and function with the same feature set as its IP phone counterpart.

(3) RTP/RTCP: The Real-Time Transport Protocol (RTP) and Real-Time Control Protocol (RTCP) provides end to end delivery services for data with real-time characteristics such as audio, voice or simulation data.

(4) SCCP: The Skinny Client Control Protocol (SCCP) is a Cisco proprietary protocol allowing Cisco-compatible devices to operate in a client-server mode with IP endpoints issuing proxies for services from the Cisco Call Manager.

(5) SIP**:** Session Initiation Protocol (SIP) defined in RFC 3261. Like SCCP, SIP is a low overhead protocol. It performs basic call setup functions such as establishment of user location (i.e. translating from a user's name to their current network address), feature negotiation, call management, and changing features of a session while in progress. SIP is a catalyst for the next phase of open communications using not only

IP telephony and VoIP but also the full suite of IP-related protocols [5]. SIP is an interoperable protocol designed to allow equipment from different vendors to communicate with each other. Details on SIP are in [6,7].

2.1.3 VoIP Protocols Stack

The basic IP network protocol stack used to implement VoIP is shown in Figure 2.1. In order for the internet to provide useful services, Internet telephony required a set of control protocols such as H.323 or SIP for connection establishment, capabilities exchange as well as conference control.

H.323 / SIP
RTP, RTCP, RSVP
UDP, TCP, SCTP
Network Layer (IPv4, IPv6)
Data Link Layer
Physical Layer

Figure 2.1 VoIP protocol stack

H.323 is a standard that specifies the components, protocols and procedures that provide multimedia communication services such as real-time audio, video, and data communications over packet networks, including Internet Protocol (IP) based networks. An alternative to H.323 emerged with the development of Session Initiation Protocol (SIP). SIP is a more streamlined protocol, developed specifically for VoIP applications. Smaller and more efficient that H.323, SIP takes advantage of existing protocols to handle certain parts of the process. Real-Time Transport (RTP) protocol provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data, over multicast or unicast networks.

The RTP control protocol (RTCP) is used to monitor the quality of real-time services and to convey information about participants in an on-going session. There are components called monitors, which receive RTCP packets sent by participants in a session. These packets contain reception reports, and estimate the current quality of service for distribution monitoring, fault diagnosis and long-term statistics. Both Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) enable the transmission of information between the correct processes or applications on host computers. Although TCP provides a reliable service and UDP does not, many applications including VoIP fit better in the communication system by using UDP due to some reasons such as faster delivery of the application object, support for a larger number of active clients and smaller segment header overhead.

IP is responsible for the delivery of packets or datagram between host computers. IP is a connectionless protocol and it does not establish a virtual connection through a network prior to commencing transmission because this is the task of higher level protocols. IP makes no guarantees concerning reliability, flow control, error detection or error correction. The result is that datagram could arrive at the destination computer out of sequence, with errors or not even arrive at all.

2.2 User Datagram Protocol

The user datagram protocol (UDP) is one of transport-layer protocol that is placed on top of the network layer. UDP is a connectionless protocol, as no handshaking between sending and receiving points occurs before sending a segment. UDP does not provide a reliable service. The enhancement provided by UDP over IP is its ability to check the integrity of flowing packets. IP is capable of delivering a packet to its destination but stops delivering them to an application. UDP fills this gap by providing a mechanism to differentiate among multiple applications and deliver a packet to the desired application. UDP can perform error detection to a certain extent but not to the level that TCP can.

The format of the UDP segment is shown in Figure 2.2. UDP transmits segments consisting of an 8-byte header followed by the payload. The segment starts with the source port, followed by the destination port. These port numbers are used to identify the ports of applications at the source or the destination, respectively. The source port identifies the application that is sending the data. The destination port helps UDP to demultiplex the packet and directs it to the right application. The UDP length field indicates the length of the UDP segment, including both the header and the data. UDP checksum specifies the computed checksum when transmitting the packet from the host. If no checksum is computed, this field contains all zeroes. When this segment is received at the destination, the checksum is computed; if there is an error, the packet is discarded.

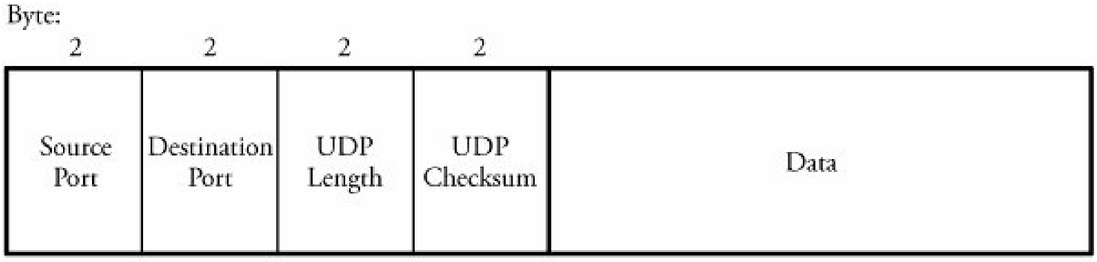

Figure 2.2 User datagram protocol segment

UDP takes messages from the application process, attaches source and destination port number fields and two other fields, and makes this segment available to the network layer. The network layer encapsulates the segment into an IP datagram (packet) and finds the best path to deliver the segment to the other end host.

2.3 Internet Protocol Version 6 (IPv6)

The use of IPv4 has resulted in the exhaustion of the 32-bit address space to the extent that IPv4 has run out of addressing spaces. Therefore, 128-bit address spacing was introduced with Internet Protocol version 6 (IPv6). It enjoys tremendous popularity because of its simplicity and flexibility in adapting to diverse network technologies. Compatible with IPv4, IPv6 also supports real-time applications, including those that require guaranteed Quality of Service (QoS). The IPv6 header is shown in Figure 2.3.

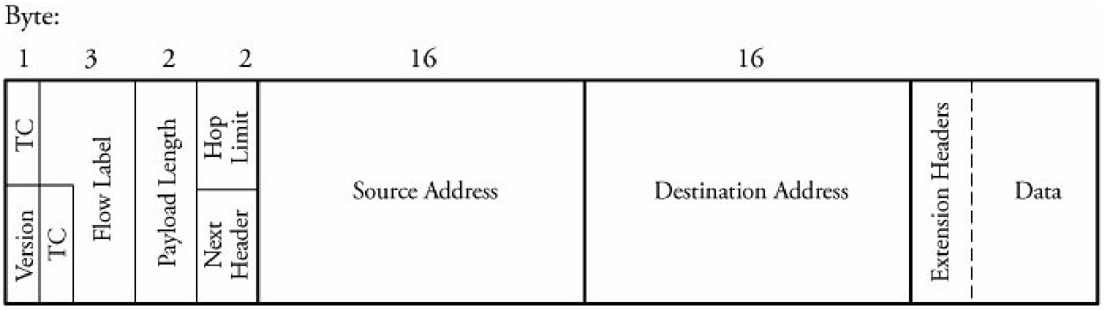

Figure 2.3 The IPv6 Header

Brief descriptions of the fields in the header are as follows:

- Version is the same as in IPv4, indicating the version number of the protocol.
- Traffic class specifies the priority level assigned to a packet.
- Flow label indicates the delay period within which application packets, such as real-time video, must be delivered.
- Payload length is the 16-bit specification of the length of the data, excluding the header.
- Next header specifies the type of extension header used. The functionality of the option field in IPv4 is specified in the extension header. In addition, the extension header is more flexible than the options field.
- Hop limit is the same as the time-to-live field in IPv4.
- Source address and destination address are each identified by a 128-bit field address.

The IPv4 and IPv6 header formats have some notable differences. First, IPv6 uses a 128-bit address field rather than the 32-bit field in IPv4. The 128-bit field can support a maximum of 3.4 x 10^{38} IP addresses. IPv6 has a simpler header format, eliminating the fragmentation, the checksum, and header length fields. The removal of the checksum field in IPv6 allows for faster processing at the routers without sacrificing functionality. In IPv6, error detection and correction are handled at the data link and the TCP layers. IPv6 can also accommodate the QoS requirements for some applications. Besides all these significant advantages, IPv6 can provide built-in security features such as confidentiality and authentication

Extension headers are positioned between the header and the payload. If multiple extension headers are used, they are concatenated, as shown in Figure 2.4. It is mandatory for them to be processed in the sequence in which they are listed. The sequence in which the extension headers are to be listed is also specified in the same figure.

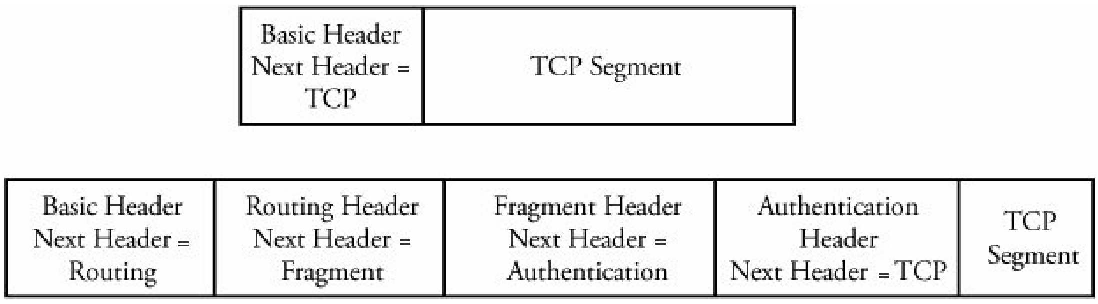
Basic Header Next Header = TCP
TCP Segment
Basic Header Next Header = Routing
Routing Header Next Header = Fragment
Fragment Header Next Header = Authentication
Authentication Header Next Header = TCP
TCP Segment

Figure 2.4 Concatenated IPv6 extension header

In IPv6, fragmentation is permitted only at the source. The result of this restriction is faster processing of packets at routers. Before transmitting a packet, a host performs a maximum transmission unit (MTU) discovery in the route of the packet. The minimum MTU obtained determines the packet size and thus requires the route from the host to the destination to remain steady. If this minimum value of the physical network is less than the packet size to be transmitted, the intermediate router discards the packet and sends an error message back to the source. In rare cases, the packet needs to be fragmented, and the extension header contains the fragmentation information.

2.4 Audio Codec

The primary functions of a voice codec are to perform analog/digital voice signal conversion and digital compression. Among three commonly used codec in Internet telephony are G.711, G.723.1, and G.729. These codecs differ in their coding rate (bps), frame rate (frames/s), algorithmic latency that will influence the speech quality or Mean Opinion Score (MOS) in a VoIP network [8].

2.4.1 G.711 codec

G.711 is primarily used in VoIP and applied for encoding telephone audio signal at a rate of 64 kbps with a sampling frequency of 8 kHz and 8 bits per sample. G.711 represents logarithmic pulse-code modulation (PCM). There are two main compression algorithms defined in ITU-T standard which are the μ-law and A-law. Typical algorithmic delay is 0.125ms with no look-ahead delay. In an IP network, voice is converted into packets with durations of 20ms of sampled voice, and these samples are encapsulated in a VoIP packet.

2.4.2 G.723.1 codec

G.723.1 is an audio codec for voice that compresses voice audio in 30 ms frames. An algorithmic look-ahead of 7.5 ms duration means that total algorithmic delay is 37.5ms. There are two bit rates at which G.723.1 can operate: 6.3Kbps (using 24byte frames) and 5.3Kbps (using 20byte frames) with Algebraic Code Excited Linear Prediction (ACELP) algorithm. The coder operates on speech frames of 30ms corresponding to 240 samples at a sampling rate of 8000 samples/s. G.723.1 is mostly used in VoIP applications due to its low bandwidth requirement

2.4.3 G.729 codec

G.729 codec belongs to the Code Excited Linear Prediction (CELP) model speech coders and uses Conjugate Structure - Algebraic Code Excited Linear Prediction (CS-ACELP). Standard G.729 operates at 8kbit/s, but there are extensions which provide rates of 6.4kbit/s and 11.8kbit/s. The coder compresses voice in packets of 10ms duration and required look-ahead delay of 5ms. The total algorithmic delay for the coder is 15ms.

2.5 Handoff

When a MN changes its point of attachment to the network, it moves from one network to another new network. This process is known as handoff or handover. During this process, the MN usually has disconnected from the old network before connecting to the new network (especially if using a single interface) and therefore there is a time when the MN has lost connectivity to the Internet. During this period it cannot send or receive IPv6 packets to the detriment of existing application sessions. While many TCP applications are designed to cope with intermittent loss of connectivity by retransmitting unacknowledged packets, UDP applications will not be able to recover such losses. Furthermore, both TCP and UDP applications (such as VoIP and audio or video streaming) that rely on timely packet delivery within certain acceptable thresholds will be sensitive to the length of time a MN loses connectivity while performing handover.

Such applications desire what is known as seamless handovers. Where seamless refers to handoffs that are both smooth (no or very little packet loss) and fast (low latency). Therefore, if the mobile Internet is to support these demanding applications, performing handoffs in MIPv6 must display these two qualities. If it does not, then additional optimizations and/or changes to the protocol will be deemed necessary.

2.5.1 Types of Handoff

The term handoff or handover refers to the process of a mobile node (MN) moving from one point of attachment to the Internet to a different point of attachment. There are different types of handoff according to which layers of the communication stack are affected [9]. In general, handoffs that only affect the link layer (L2) without resulting a change of IP in network layer (L3) state are known as horizontal handoffs. An example of this is when a MN moves between different Wireless LAN Access Points that are served by the same IP Access Router. In 802.11 terminology, both Access Points belong to the same Extended Service Set

(ESS). Handoffs that affect both L2 and L3 (a new IP address is obtained by the MN) are known as vertical handoffs.

Some literature makes a distinction between hard and soft handoffs. A hard handoff is when all the links (usually radio) in the MN are disconnected before the new links are established. Conversely, a soft handoff refers to the case where the MN is always connected to the network via at least one link. In this way, there is an overlap of different link usage during the handoff process. Of course, this implies either multiple interfaces or multiple radio modules on a single interface are available on the MN.

All the above types of handoff may be either inter-technology or intra-technology handovers. In inter-technology handoffs, the handoff is between different network technologies, which would usually mean separate interfaces on the MN. Intra-technology handoffs are handoffs of the same network technologies. Horizontal handoffs would usually be of the intra-technology type, although different network technologies could be used provided the IP layer sees no change its connectivity and associated state. Vertical handoffs can just as easily be inter- technology as intra-technology.

One could also categorize L1 handovers such as when a Wireless LAN station switches between different frequencies and/or coding schemes of its current link. However, these are not considered to be of much relevance in the scope of this project.

2.5.1.1 Horizontal Handoff

A simple case of a MN moving between Access Points (APs) and changing its point of attachment at the link layer is shown in Figure 2.5. In this case, the different APs are served by the same Access Router (AR) and will most likely belong to the same 802.11 Extended Service Set (ESS). The MN is still considered to be attached to the same link from the point of view of the IPv6 layer. Consequently, the MIPv6 handover procedure is not triggered because the MN can still use its current Care of Address (CoA). In fact, the IPv6 layer should be completely unaware that movement between APs has taken place provided that the inter-AP handoff does not disrupt any IPv6 communication.

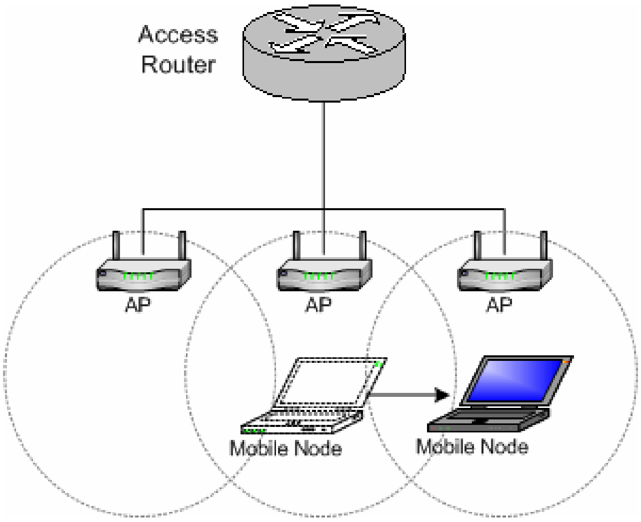

Figure 2.5 Mobile node moving within same ESS

As the MN roams between AP coverage areas, known as a Basic Service Set (BSS) in 802.11 terminology, the 802.11 client in the MN continually monitors the signal strength of all the different BSSs within the same ESS. It is this signal strength information that is used to decide if the MN should perform a handoff between APs.

2.5.1.2 Vertical Handoff

Some examples of a MN performing vertical handoffs are shown in Figure 2.6, Figure 2.7 and Figure 2.8. In the example of Figure 2.6, the MN moves between APs that belong to different ESSs and which are served by different ARs. This means that the MN would no longer be reachable from its previous AR and must now use its new AR.

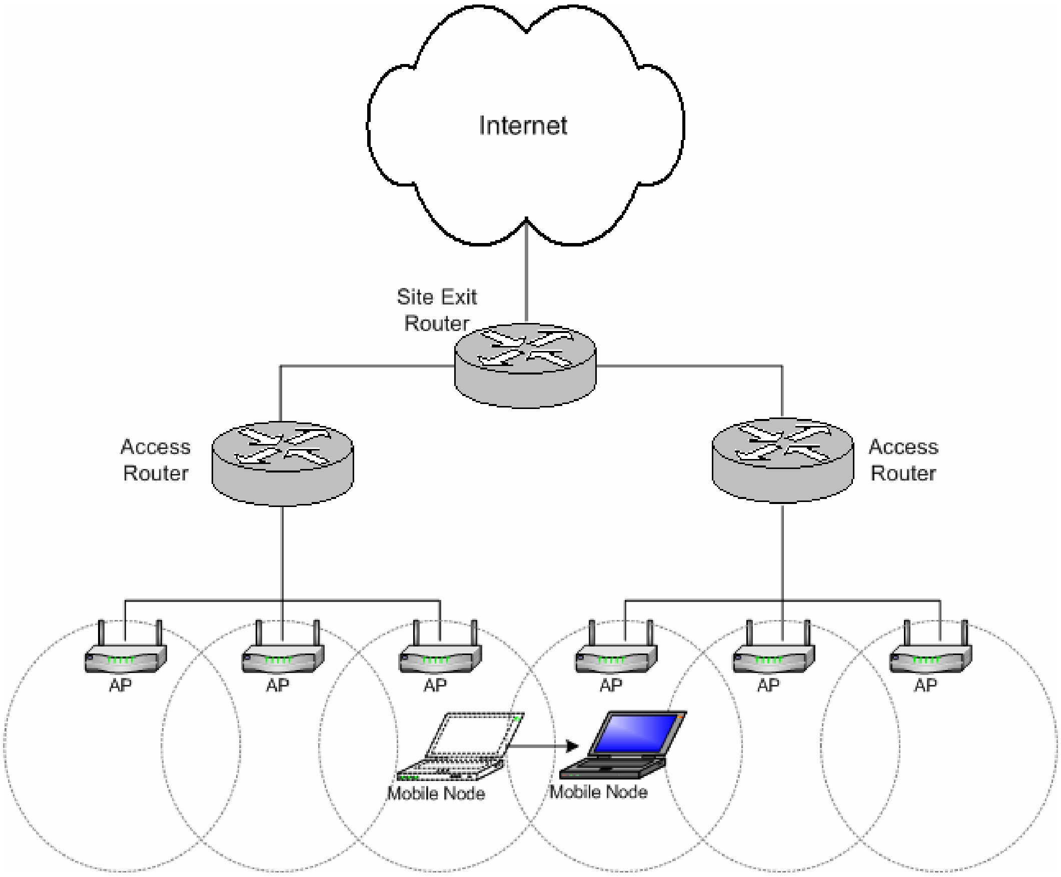

Figure 2.6 Mobile node moving between ARs of the same provider

However, in this example, both the Previous Access Router (PAR) and the New Access Router (NAR) belong to the same administrative domain (service provider). Although the MIPv6 handoff procedure must be activated, it is likely that any 'higher layer' state such as Authentication, Authorization and Accounting (AAA) or Quality of Service (QoS) information would not need to be re-negotiated within the same provider during handoff, thus lessening the overall handoff latency.

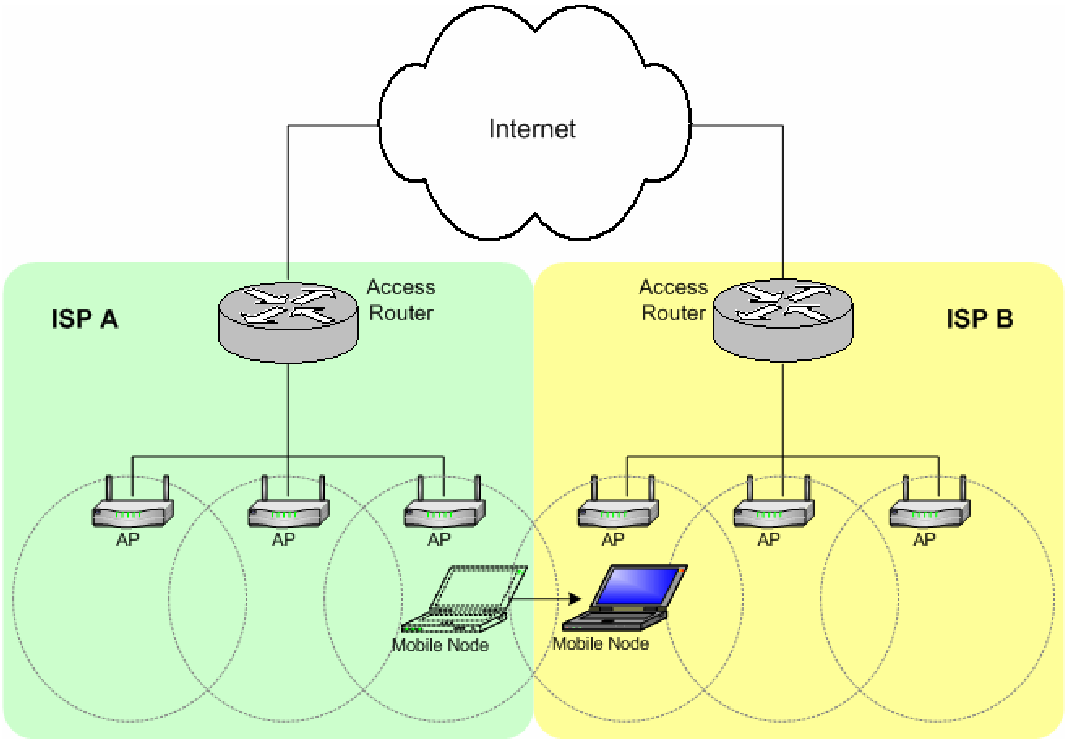

Figure 2.7 Mobile node moving between ARs of different providers

In the example of Figure 2.7, the MN has changed provider networks. Although geographically close, roaming to another AP owned by a different provider may well result in the MN moving a great distance topologically speaking. A good example of this is a MN moving between WLAN hotspots of different Internet Service Providers (ISPs) that are physically close to one another.

Another example would be a MN moving out of range of any WLAN coverage and thus 'falling back' to a third generation (3G) network employing General Packet Radio Service (GPRS) or Universal Mobile Telecommunication System (UMTS)[1]. Typically, the providers of the WLAN coverage and the 3G network coverage will be different. This scenario is illustrated in Figure 2.8.

[1] We do not have UMTS is Asia region. UMTS is deployed only in Europe.

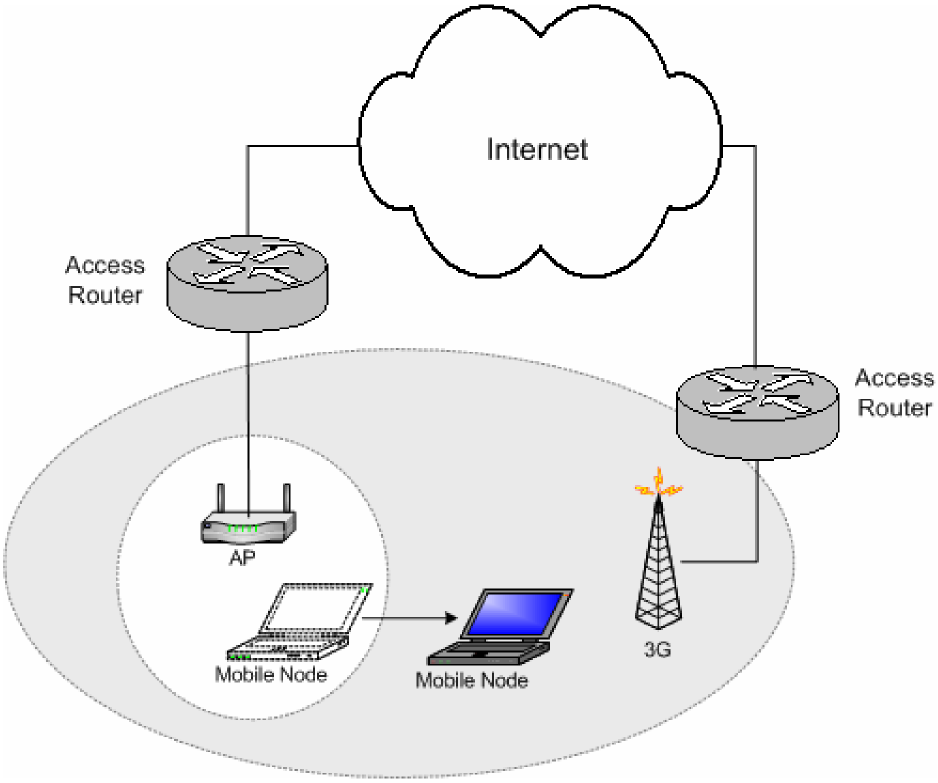

Figure 2.8 Mobile node moving between WLAN and 3G network of different providers

In all of the above examples the MIPv6 handoff procedure will be activated. Typically, the latency involved in the handoff procedure would be greater in the case of vertical handover between different provider networks. This is not only due to the fact of the topological distance between the PAR and NAR is generally greater, many ISPs usually block all outbound traffic from the MN until authentication and authorization have been satisfied. Thus, MIPv6 handoff success in this scenario depends on successful authentication and authorization on the new network by whichever methods are employed to accomplish this. The MIPv6 handoff procedure will be examined in detail in Chapter 3.

2.5.1.3 Administrative Domain Handoff

The term “Administrative Domain” was defined in [10] as “A collection of End Systems, Intermediate Systems, and Subnetworks, operated by a single organization of administrative authority. The components which make up the domain are assumed to interoperate with a significant degree of mutual trust among themselves, but interoperate with other Administrative Domains in a mutually suspicious manner.” Based on this definition of Administrative Domains, handoffs can be categorized as

i. Intra-Administrative Domain Handoff**:** a handoff process where the MN switches between base stations supporting the same or different technologies, managed by the same administrative domain.

ii. Inter-Administrative Domain Handoff: a handoff process where the MN switches between base stations supporting the same or different technologies, managed by different administrative domains.

2.5.1.4 Number of Connections Involved

This categorization mainly applies to the handoff process within cellular networks. These types of handoffs are hard handoff, soft handoff and softer handoff as shown in Figure 2.9

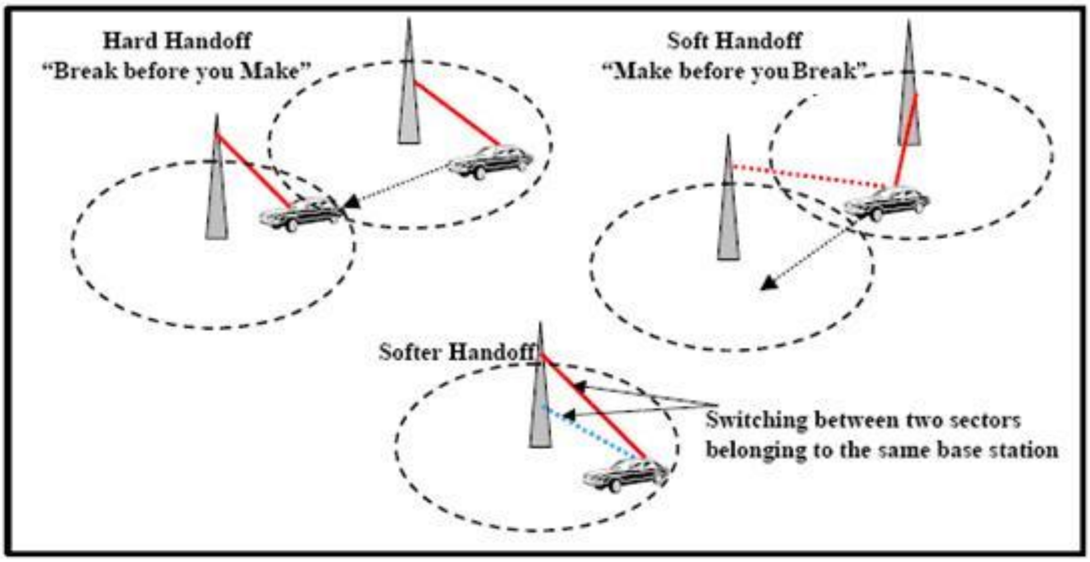

Figure 2.9 Handoffs category according to the number of connections involved

i. Hard Handoff**:** A term used to describe a handoff that involves the MN maintaining a connection with only one base station at any given time. This process is sometimes referred to as "Break before you make." Hard handoffs may be seamless or non-seamless depending on their severity and whether they are noticed by the user in the form of an interruption in service

ii. Soft Handoff**:** A term used to describe a handoff that involves the MN always being connected to at least one base station when moving between cells. This process is sometimes referred to as "Make before you break." Soft handoffs are possible in situations where the MN is moving between cells that operate on the same frequency. It has been realized as an option in 3G systems.

iii. Softer Handoff**:** A term used to describe a type of Soft Handoff that involves the MN switching connections over radio links that belong to the same base station. This type of handoff is possible in such networks where a base station serves several individual sectors of a cell.

2.5.1.5 Layers Involved

There are two types of handoff categorized as which layers are involved namely link layer (L2) handoff and network layer (L3) handoff. L2 handoff is a process which a mobile node changes its physical link-layer connection to another. When a mobile node moves to a new Access Point (AP), L2 handoff occurs. L3 handoff usually follows L2 handoff. In L3 handoff, a mobile node identifies that it moves to new link layer where new subnet prefix is used. This mobile node will change its primary CoA to new one. As mobile node moves, change of AP followed by the change of the subnet leads to L3 handoff. These handoffs process are shown in Figure 2.10

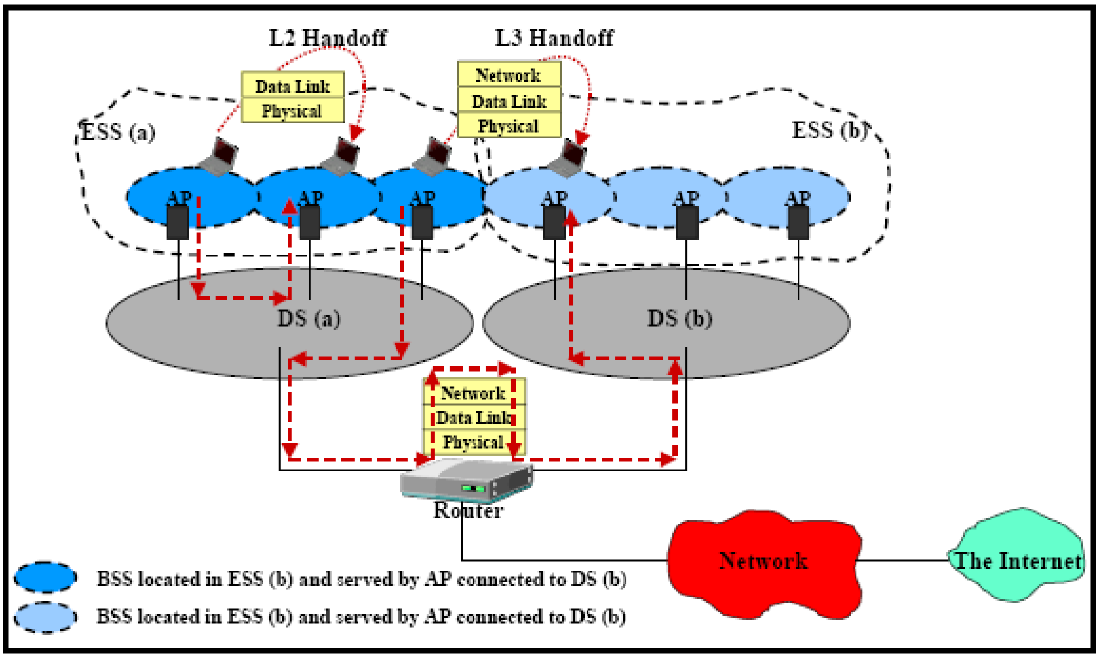

Figure 2.10 Layer 2 Handoff in ESS (a) and Layer 3 Handoff between ESSs (a) & (b)

L2 handoff occurs when the switching of the MN between two access points within the same ESS. On the other hand, L3 handoff would take place when the MN switches between two APs in different ESSs. In L3 handoff, signaling will go through an immediate router and network layer signaling is required to manage the routing of data to the MN's new location.

2.6 Related Work

Many alternatives have been proposed to allow mobility in wireless network. A number of well-known approaches based on micro- and macro-mobility management mechanism in IPv6 have been proposed in the literature. Cellular-IP, HAWAII, Hierarchical Mobile IPv6 [11], Fast Handover [12] and S-MIP [13] are some of these techniques.

In [14], the authors presented analytical results for inter-domain handoff latency under usage of the basic Mobile IP protocol with a Smooth Handover extension, which is basically a tunneling functionality between old foreign agent and new foreign agent. The authors state, that the handoff delay consists of two components: link layer establishment delay and signaling or disruption delay. They further assume that the link layer establishment delay is negligible compared to the signaling delay and, therefore, concentrate on the signaling delay.

In [15], the authors proposed a macro-mobility handoff scheme for Hierarchical Mobile IP (HMIP). It is based on bi-casting. During a handoff procedure the previous Mobility Anchor Point (MAP) of MN would bi-cast incoming packets to PAR in its own MAP domain and to NAR in the new MAP domain. NAR buffers the forwarded packets and delivers them to MN when it receives connectivity to it. The handoff delay in this paper is measured from the time the MN sends the FBU message to NAR until the time the first packet from CN, routed directly through the new MAP, reaches MN. Additionally, the authors also considered the rendezvous time which is the time needed for the MN to hear the beacon from NAR after roaming out of PAR's network. The sum of handoff delay and rendezvous time gives the complete handoff time.

In [16], the authors proposed a protocol for macro and micro mobility support in Mobile Broadband Wireless Access (MBWA) networks. Handoffs are mobile-initiated. The handoff decision is based on a comparison between the MN's received Signal-to-Noise-and-Interference-Ratio (SINR) from the serving AP to the neighboring APs. The proposed protocol is similar to Fast Hierarchical Mobile IP (FHMIP), although different terminology is used (domain AR instead of MAP).

Handoff latency is defined as the elapsed time from the point in time when a MN initiates a handover by sending a request to a NAR, until the MN is able to receive packets from the NAR. This occurs when the MN receives a handoff response from the new AP accepting its original request. Further, during inter-domain handoffs, a tunnel between previous and new MAP for forwarding of packets is created. An inter-domain handoff starts when the MN decides to switch ARs due to a SINR based decision (L2 mobile trigger). The MN sends a handover request to NAR from its CoA. The following procedure is equivalent to the reactive mobile-initiated FMIPv6. The handoff latency is measured in simulation experiments conducted with the OPNET simulation environment.

In [17], the authors propose a modified FHMIP protocol. The resulting mobility management protocol is called “two-way registration". The scheme aims to decrease the home registration latency and hence minimize the disruption caused by macro mobility handovers. The inter-domain handover procedure starts when the MN detects a NAR by receiving its agent advertisement on Layer 3. The MN sends a home registration request to the new MAP via NAR. The registration request contains addressing information about the old MAP. The new MAP bi-casts the registration request to HA and the old MAP. Both of them generate a registration reply and send it to the new MAP. The reply, which arrives first, is forwarded to the MN. The second one is discarded. It is assumed that old and new MAP are generally closer to each other than new MAP and HA. Therefore, the registration reply from the old MAP should arrive faster and thus accelerate the HMIP handover procedure. Upon reception of the handover request, the old MAP also starts forwarding of packets destined to Previous Care of Address (PCoA) to the new MAP. The handoff latency is measured from the MN's sending of the registration request until the reception of the registration reply (it regards any registration reply as if it comes from HA).The performance of two-way registration is compared to HMIP performance by using a C++ simulation program.

2.7 Conclusion

This chapter explains the concept of Voice over Internet Protocol and its protocols. In order to function properly, VoIP application has to perform according to protocols stack based on TCP/IP protocols. This includes the use of User Datagram Protocol and in transport layer and IPv6 in network layer. Both UDP and IPv6 are connectionless protocol.

Brief explanation of audio codecs is explained in this chapter. Codecs are used to perform analog/digital voice signal conversion and digital compression. The codecs used in the network can influence the speech quality of VoIP.

There are many challenges faced by VoIP application but this project focuses on delay due to handoff process. Therefore, different types of handoff are discussed in this chapter.

CHAPTER 3

METHODOLOGY

3.1 Introduction

The initial stage of this project includes the literature review on VoIP protocols, characteristics and performances. There are several handoff approaches and mobility managements being studied, however this project focuses on Mobile IPv6 and Fast Handovers for Mobile IPv6. The performance of MIPv6 and FMIPv6 is investigated in detail and performance analyses are also made. FMIPv6 is used to observe the improvement of the performance of VoIP during handoff. A suitable network scenario that simulates VoIP traffic in Internet will be developed. Simulation of the two different mobility schemes will be made base on the network scenario. Performance analysis and comparison between MIPv6 and FMIPv6 architectures will be made.

This chapter discusses the MIPv6 and FMIPv6 handoff procedures and the details on the network model and parameters that are being used in the NS2 simulation. The flowchart of this project methodology is shown in Figure 3.1.

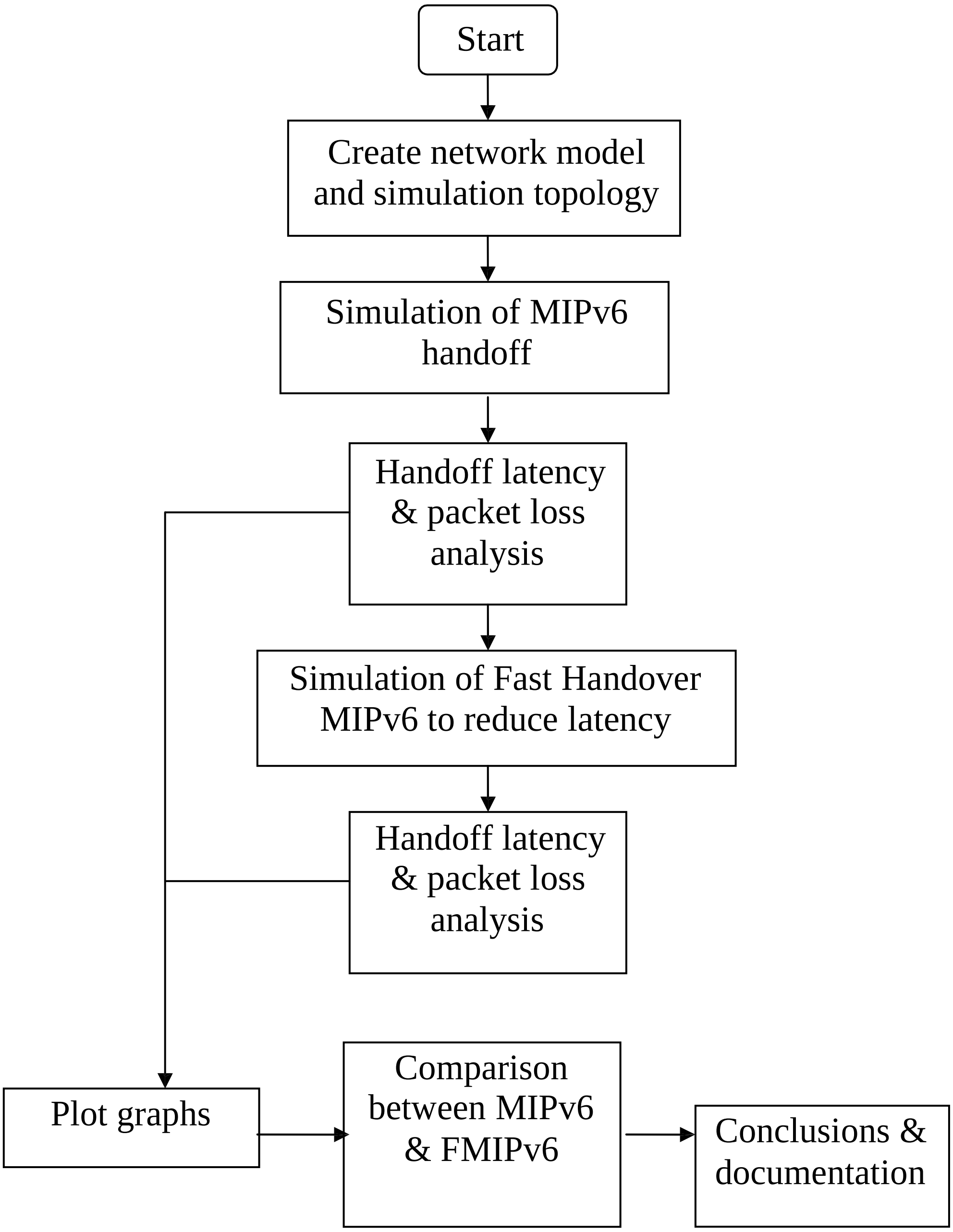
Start
Create network model and simulation topology
Simulation of MIPv6 handoff
Handoff latency & packet loss analysis
Simulation of Fast Handover MIPv6 to reduce latency
Handoff latency & packet loss analysis
Plot graphs
Comparison between MIPv6 & FMIPv6
Conclusions & documentation

Figure 3.1 Flowchart for research methodology of this project

3.2 Mobile IPv6

Mobile IP supports mobility of IP hosts by allowing them to make use of two IP addresses: a home address that represents the fixed address of the node and a care-of address (CoA) that changes with the IP subnet the mobile node is currently attached to. An entity is needed that maps a home address to the corresponding currently valid CoA. In Mobile IPv4 these mappings are exclusively handled by home agents (HA). A correspondent node (CN) that wants to send packets to a mobile node (MN) will send the packets to the MN’s home address. In the MN’s home network these packets will be intercepted by the home agent and tunneled, such as by IP-in-IP encapsulation, either directly to the MN or to a foreign agent to which the MN has a direct link. The outlined operation of MIPv6 is shown in Figure 3.2.

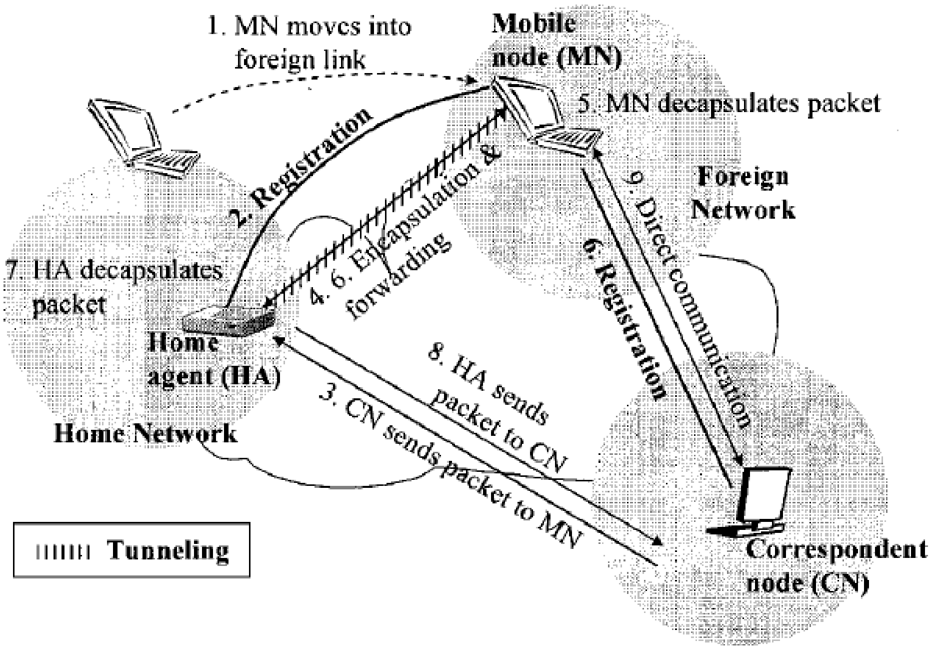

Figure 3.2 Outlined operation of MIPv6

In MIPv6 [18,19], home agents no longer exclusively deal with the address mapping, but each CN can have its own binding cache where home address plus care-of address pairs are stored. This enables route optimization compared to the triangle routing via the HA in MIPv4. In route optimization, a CN is able to send packets directly to a MN when the CN has a recent entry for the MN in its corresponding binding cache. When a CN sends a packet directly to a MN, it does not encapsulate the packet as the HA does when receiving a packet from the CN to be forwarded, but makes use of the IPv6 Routing Header Option. When the CN does not have a binding cache entry for the MN, it sends the packet to the MN’s home address. The MN’s home agent will then forward the packet. The MN, when receiving an encapsulated packet, will inform the corresponding CN about the current CoA.

In order to keep the home address to CoA mappings up-to-date, a mobile node has to signal corresponding changes to its home agent and/or correspondent nodes when performing a handoff to another IP subnet. Subnet is defined as a portion of the network's computers and devices that have a common, designated IP address routing prefix. For example, all devices with IP address that start with 100.100.100. would be part of the same subnet. Since in MIPv6 both, HA and CN, maintain binding caches, a common message format called binding updates (BU) is used to inform HA and CN about changes in the point of attachment. Additionally, since the BUs have associated a certain lifetime, even if the MN does not change its location a BU to it's HA and CNs is necessary before the lifetime expires to keep alive the entry in the binding caches. Binding updates can be acknowledged by Binding Acknowledgement (BA).

In contrast to MIPv4, where signaling is done using UDP, Mobile IPv6 signaling is done in extension headers that can also be piggybacked on regular packets. To acquire a CoA in Mobile IPv6, a mobile node can build on IPv6 stateless and stateful auto-configuration methods. The stateless autoconfiguration mechanism is not available in IPv4.

3.2.1 Handoff Procedure in Mobile IPv6

The Mobile IPv6 (MIPv6) specification is a proposed standard by the IETF to provide transparent host mobility within IPv6. The protocol enables a MN to move from one network to another without the need to change its IPv6 address. A MN is always addressable by its home address, which is the IPv6 address that is assigned to the node within its home network. When a MN is away from its home network, packets can still be routed to it using the MN's home address. In this way, the movement of a node between networks is completely invisible to transport and other higher-layer protocols.

When a MN changes its point of attachment to the Internet from one IPv6 network to another IPv6 network (also referred to as roaming), it will perform the MIPv6 handover procedure. The MIPv6 handover procedure is similar to the auto configuration procedure an IPv6 node booting up onto a network but has some important differences:

- the MN must detect that it has moved onto a new network.
- once configured, the MN must inform its home agent (HA) and each correspondent node (CN) of its new location.
- during the handoff procedure, upper layer connections will still be active so the handoff procedure should be performed as quick as possible to minimize disruption from lost and severely delayed packets.

The handoff or handover latency in term of seconds for MIPv6 is illustrated in Figure 3.3 [20]. It consists of two components, L2 handoff and L3 handoff. The term L2 handoff denotes its support for roaming at the link layer level while the L3 handoff occurs at the network layer level.

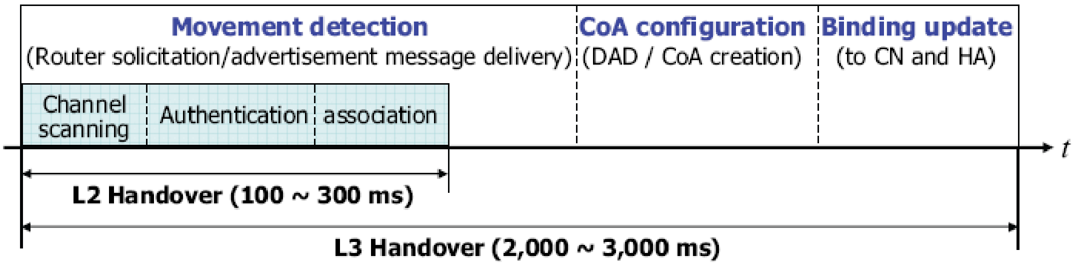

Figure 3.3 Handoff latency in MIPv6

The basic MIPv6 handoff procedure is composed of L2 handoff and L3 handoff as shown in Figure 3.4 [21]. The L2 handoff is classified into 3 levels which are channel scanning, authentication and association. The L2 handoff latency is about

100 to 300ms while L3 handoff latency is about 2 to 3 seconds and dependent on system configuration or network topology. This handoff latency is so long that mobile node suffer from packet loss and service disruption.

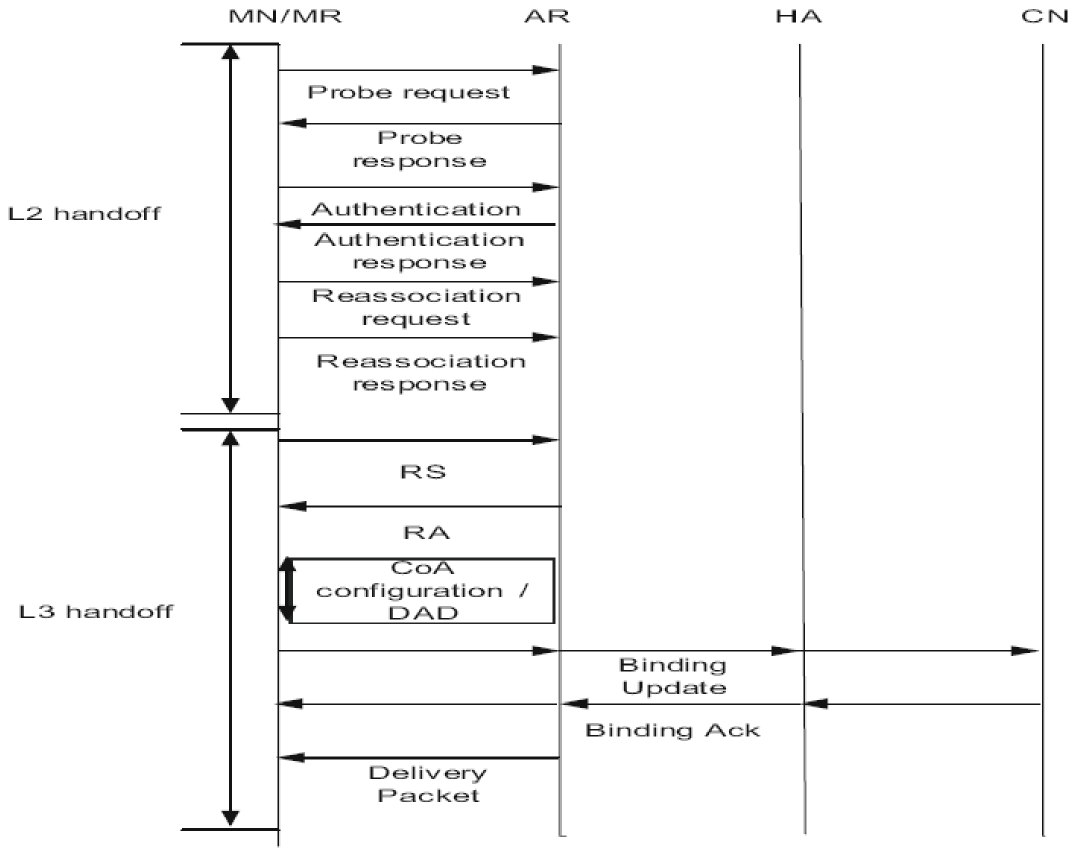

Figure 3.4 Basic MIPv6 handoff procedure

The MIPv6 handoff flow is illustrated in Figure 3.5 and is described in more detail in the following sections. In the IPv6 specifications, each stage of the procedure is mandatory with the exception of authentication and authorization [9], although this stage will be present, at least in some form, in most deployed networks.

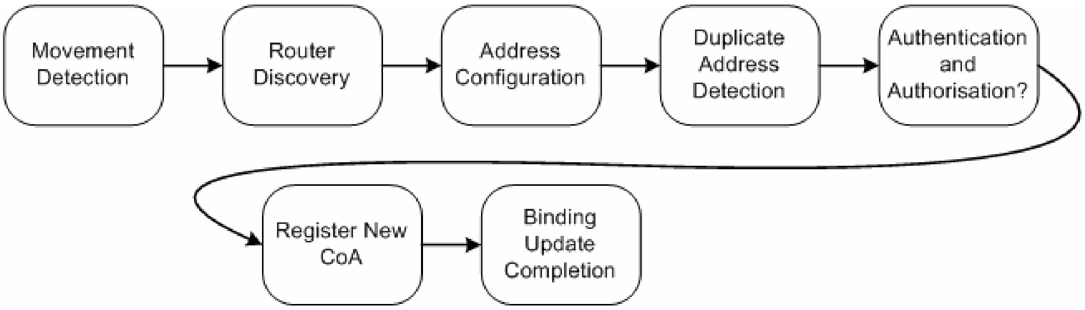
Movement Detection
Router Discovery
Address Configuration
Duplicate Address Detection
Authentication and Authorisation?
Register New CoA
Binding Update Completion

Figure 3.5 The MIPv6 handoff flow

The detail of layer 3 handoff procedure in MIPv6 without any Authentication, Authorization and Accounting (AAA) process is shown in Figure 3.6.

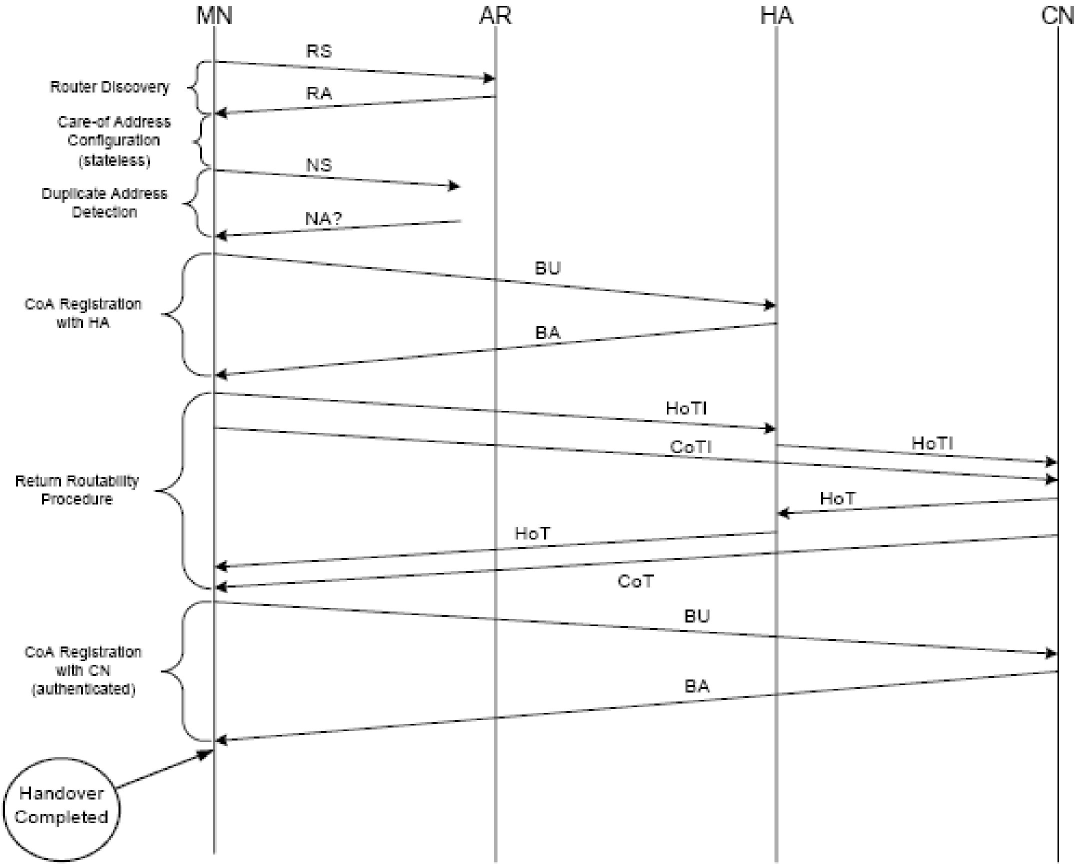
MN
AR
HA
CN
RS
RA
NS
NA?
BU
BA
HoTI
CoTI
HoTI
HoT
HoT
CoT
BU
BA
Router Discovery
Care-of Address Configuration (stateless)
Duplicate Address Detection
CoA Registration with HA
Return Routability Procedure
CoA Registration with CN (authenticated)
Handover Completed

Figure 3.6 Detail of layer 3 handoff procedure

3.2.1.1 Movement Detection

In Mobile IPv6, it is generally the responsibility of the MN to detect that it has moved between networks. Determining whether or not a MN has moved networks is not always a simple issue. However, the general rule of thumb that a MN has moved can be seen as:

i. the current access router is no longer reachable, and
ii. a new and different access router is available

In order to determine if its Present Access Router (PAR) is still bi-directionally reachable the MN performs Neighbour Unreachability Detection on a continual basis. Neighbour Unreachability Detection works in the following manner. When an IPv6 host has a packet to send, it checks the Neighbour Cache to determine the link layer address of the next hop node (either an on-link neighbour or a router). The Neighbour cache also has an associated state with each neighbour entry. A neighbour state of REACHABLE indicates that the neighbour is considered reachable.

In IPv6 a host considers a neighbour reachable if it has recently received confirmation that packets sent to the neighbour have been received. This is achieved in two ways: the receipt of a neighbour advertisement from the neighbour in response to a neighbour solicitation sent by the host, or a hint from upper layer protocols. The IPv6 stack utilizes the acknowledgements of upper layer protocols to register the fact that a packet has recently been received from a given destination address and so is considered reachable.

The IPv6 host will send a neighbour solicitation in the event that the neighbour cache entry not being set as REACHABLE when there is a packet to send. The neighbour unreachability detection only occurs when the MN has a packet to send. Therefore, in the worse case scenario when the MN is not any sending packets, it may not notice that it has moved networks until it receives an unsolicited router advertisement from the new on-link router (consistent with the normal router advertisement interval). Unfortunately, this may be the case when the MN is receiving real-time streams when an interruption in connectivity can cause packet losses and unacceptable latency while the new handover is taking place. In such a scenario, the MN may not actually be transmitting much data itself, perhaps occasional TCP or application layer acknowledgements, but nothing that will allow the unreachability of its PAR to be discovered in a timely fashion.

However, the MN noticing a new router advertisement only serves as a hint that the MN has moved networks and does not guarantee it. For example, one possibility could be that a new (additional) router has been activated on the existing link. Furthermore, as stated in [22] unsolicited router advertisements must not be used as confirmation of bi-directional router reachability since they only confirm reachability in the router to MN direction.

3.2.1.2 Router Discovery

Router Discovery is achieved through the receipt of a router advertisement sent from the New Access Router (NAR). This will either be in the form of a router advertisement sent periodically to the all nodes multicast address, or in response to a router solicitation sent by the MN. There is a potential race condition here. The MN will send a router solicitation if it discovers that its PAR is considered unreachable (its neighbour cache entry is not set to REACHABLE), and will receive a solicited router advertisement from the NAR, or it will receive an unsolicited router advertisement from the NAR as part of its periodic broadcasts.

There is no guarantee as to which method will occur first. It will depend on the exact circumstances at the time of handoff: the period or router advertisement transmissions by the NAR and the exact value of the various timers at that moment in time. One can predict that reducing the period of router advertisements will increase the likelihood of receiving an unsolicited router advertisement on the new link before realizing that the PAR is no longer reachable.

However, the receipt of a new unsolicited router advertisement is not necessarily a definite indication of having moved networks. Thus, the MN may also decide to confirm that its PAR is definitely unreachable before deciding to use the NAR. This would involve transmitting neighbour solicitations for a pre-determined time without receiving a corresponding neighbour advertisement.

3.2.1.3 Care of Address (CoA) Configuration

The MN must configure itself with an IPv6 address to be used on the new network. This will be the MN's New Care-of Address *(*NCoA). Address configuration can be performed in a stateful or a stateless manner. An IPv6 host may use both stateless and stateful address configuration completely independently from one another. The precise method to be used can be signalled with the setting of various flags in router advertisement messages.

A host may also be configured manually. However, any address configuration requiring manual input from the user would be a catastrophe for an appropriate MIPv6 handover. On the other hand, most network operators would not allow or do not allow its users to manually configure addresses. It is possible that state other than IPv6 addresses may be left to the user to configure manually.

There are two ways in which an IPv6 node can configure its address in a stateless style. The first way is using automatic address configuration with prefix discovery and the other method is using stateless DHCPv6. Automatic address configuration utilizing prefix discovery is specified in [23]. If the 'autonomous' flag of a Prefix Information Option contained in a router advertisement is set, the IPv6 host may automatically generate its global IPv6 address by appending its 64-bit interface identifier to the prefix contained in the router advertisement. There are different ways in which the host may choose how to generate its interface identifier. Stateless DHCPv6 is not mentioned as an option given in router advertisements. If DHCPv6 (stateless or stateful) is to be used by the host foraddress configuration it incurs an extra overheadthat is detrimental to expedient handovers. DHCPv6 requires an extra request/response exchange on the new network in addition to normal router discovery mechanism.

As far as the handoff is concerned, using stateful DHCPv6 is no different then using stateless DHCPv6 as the observed request/response times should be the same in most cases. However, it is possible that the extra overhead of reading and writing state to memory inside the DHCPv6 server may lead to a small increase in latency when compared to its stateless equivalent. From the perspective of a MIPv6 handoff, using stateless address configuration with prefix discovery is probably the preferred option. This is simply because it incurs less latency that using DHCPv6 or any other stateful mechanism.

In the NS2 simulation, the mobile node forms a new address using IPv6 stateless address autoconfiguration method.

3.2.1.4 Duplicate Address Detection (DAD)

Just as a node must perform Duplicate Address Detection (DAD) when it boots up onto an IPv6 network to ensure that its configured addresses are likely to be unique on the link, a MN that moves onto a new network must perform DAD on the CoA that it obtains from the CoA configuration phase. This holds true regardless of whether the CoA address has been obtained by stateless, stateful or manual means.

In IPv6, the DAD procedure is defined in [23], and uses the neighbour discovery procedures defined in [22]. A MN cannot begin to use a new CoA until the DAD procedure has been successfully executed. Until DAD has succeeded, the MN’s new CoA is seen as tentative, in which it can only be used for neighbour discovery purposes (of which the DAD procedure is part of). If a MN was to use its new CoA before successful DAD and another node was using the same address on the link, the MN would erroneously process packets intended for the other node.

To perform DAD, the MN sends out a neighbour solicitation message with its own new CoA address as the target address of the solicitation message. The destination address in the IPv6 header of the neighbour solicitation is set to the solicited-node multicast address of the target address with the source address being the unspecified address. If there is another node on the link that is using the same address as the MN's new CoA, one of two things will happen:

i. The duplicate node will receive the MN's neighbour solicitation message and reply with a neighbour advertisement (sent to the all-nodes multicast address) thus exposing the duplicated address to the MN.

ii. The MN will receive a neighbour solicitation with its new CoA as the target address from a duplicate node that is also in the process of performing DAD.

Therefore, the DAD procedure will give an explicit indication to the MN should there be another node on the network that is using its new CoA. In order to speed up the autoconfiguration process, a MN may choose to initiate DAD in parallel to router discovery. Since the value of the node's link-layer identifier is known in advance, the MN can perform DAD on its link local address before receiving a router advertisement. If the router advertisement instructs the node to use stateless address configuration, the MN need not perform DAD on its resultant global unicast address if it has already verified the uniqueness of it's link-local address.

As a router may delay responding to a router solicitation by a few seconds, a MN that performs DAD only after receiving a valid router advertisement may experience significantly longer autoconfiguration latency than performing the steps in parallel when stateless addressing is used. However, a MN may only detect that it has moved onto a new network as a result of receiving a new router advertisement; in which case the potential speed up of performing DAD in parallel to router discovery is lost.

In the NS2 simulation, no DAD procedure performed because the network address for PAR is already defined initially and it is not replicated.

3.2.1.5 Registration of New Care of Address

Once the MN has detected that it has moved networks, obtained a new CoA and has been granted access to the network, it must inform its Home Agent (HA) of its new location. During the time from when the MN lost connectivity with its PAR until it informs its HA of its new location, all packets that have been sent to it will have been lost and it will not have been able to send packets to CN. The MN registers its NCoA with its HA by sending it a binding update (BU). The HA acknowledges this by replying with a binding acknowledgement (BA) and is then able to tunnel packets bound to the MN's home address (HoA) to the MN's new location (the MN's NCoA).

3.2.1.6 Binding Update Completion

This stage refers to the MN informing its CN as to its new location and that it is reachable at its NCoA. As with registering its new location with its HA, the MN sends a BU to CN to inform its new location. However, an additional procedure is followed for BU that is sent to CN. The procedure is known as a Return Routability (RR) test and is used as a way of satisfying the CN that the BU it receives is authentic and not from a malicious third party.

In brief, RR uses a Home Test (HoT) and a Care-of Test (CoT). The CN issues the two tests to the MN via the HA and the route optimised path (direct to the NCoA) respectively. The MN replies with the answer to the two tests in the BU that it sends to the CN. If the tests are answered correctly, the CN acknowledges the BU. Once the MN has received BA from its CN, the handoff process can be considered completed. It is arguable that the handover can be considered as completed once the NCoA has been registered with the HA. However, the optimum handoff will see all optimised CN sessions restored to their optimised state.

In this project, no route optimization is used in the NS2 simulation. This means that the MN only registers its NCoA with its HA by sending a BU. The MN does not send BU to CN to inform its new address.

3.3 Fast Handover Mobile IPv6

The seamless communication using the layer 3 mobility exposes a serious drawback for the realtime applications such as VoIP, realtime broadcasting and the interactive gaming. To reduce delay and packet loss, a Fast Handovers for Mobile IPv6 (FMIPv6) [12,24] is introduced into MIPv6. It tries to reduce the handoff latency by shortening the time to get new CoA when the mobile node changes its subnet. In the fast handover, several portions of the layer 3 handover are performed in advance prior to the handover, such as new care of address (CoA) configuration and movement detection to reduce the handover latency. A tunnel is established between a currently attached access router and an anticipated access router not to lose packets from correspondent nodes during the handover. The basic process in FMIPv6 is shown in Figure 3.7. The fast handover enables the mobile node to quickly detect that it has moved to a new subnet by providing the new access point and the associated subnet prefix information when the mobile node is still connected to its current subnet.

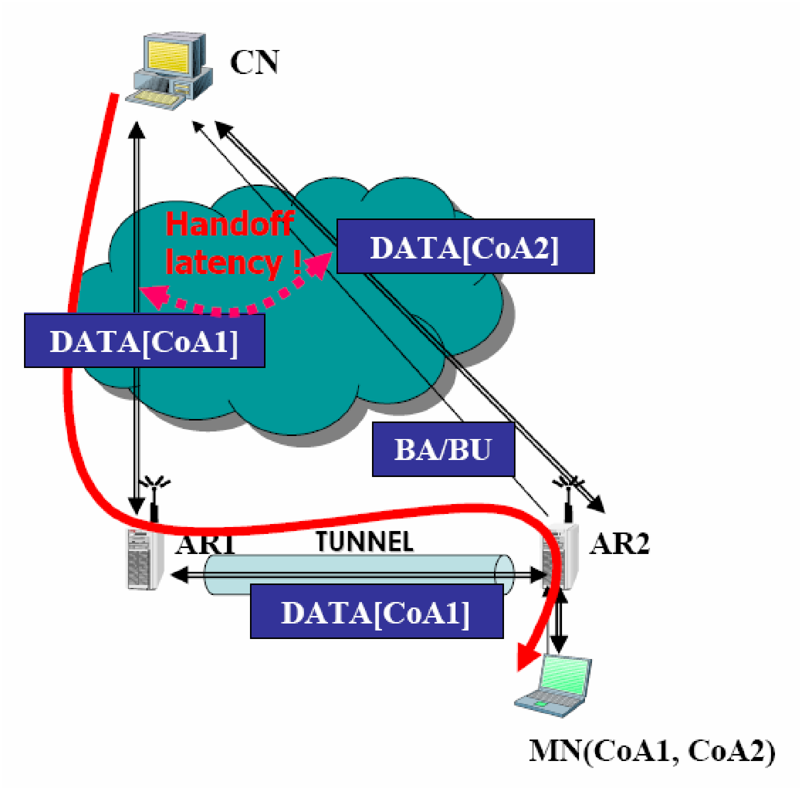

Figure 3.7 Basic operations in FMIPv6

Handoff latency comparison in MIPv6 and FMIPv6 is shown in Figure 3.8. Compared to normal MIPv6 operation, the FMIPv6 protocol claims to be more efficient in two respects. Firstly, it eliminates IPv6 configuration delay introduced by Router Discovery, Address Configuration and Duplicate Address Detection. Secondly, it removes the delay introduced by the MN performing BU procedures with is HA and CN.

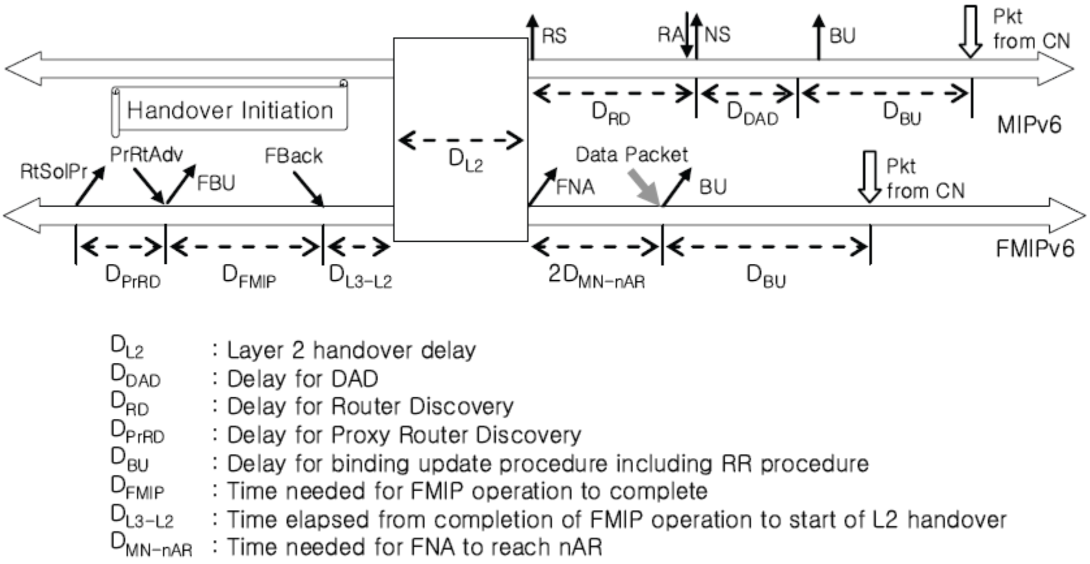
RS
RA
NS
BU
Pkt from CN
Handover Initiation
D_{RD}
D_{DAD}
D_{BU}
MIPv6
D_{L2}
RtSolPr
PrRtAdv
FBU
FBack
Data Packet
FNA
BU
Pkt from CN
D_{PrRD}
D_{FMIP}
D_{L3-L2}
$2D_{MN-nAR}$
D_{BU}
FMIPv6
D_{L2} : Layer 2 handover delay
D_{DAD} : Delay for DAD
D_{RD} : Delay for Router Discovery
D_{PrRD} : Delay for Proxy Router Discovery
D_{BU} : Delay for binding update procedure including RR procedure
D_{FMIP} : Time needed for FMIP operation to complete
D_{L3-L2} : Time elapsed from completion of FMIP operation to start of L2 handover
D_{MN-nAR} : Time needed for FNA to reach nAR

Figure 3.8 Handoff latency comparison in MIPv6 and FMIPv6

3.3.1 Mobile Node Initiated Handoff

For a Mobile Node (MN) initiated handoff, it is the MN that takes the decision to move links. The detail of FMIPv6 handoff procedure is shown in Figure 3.9.

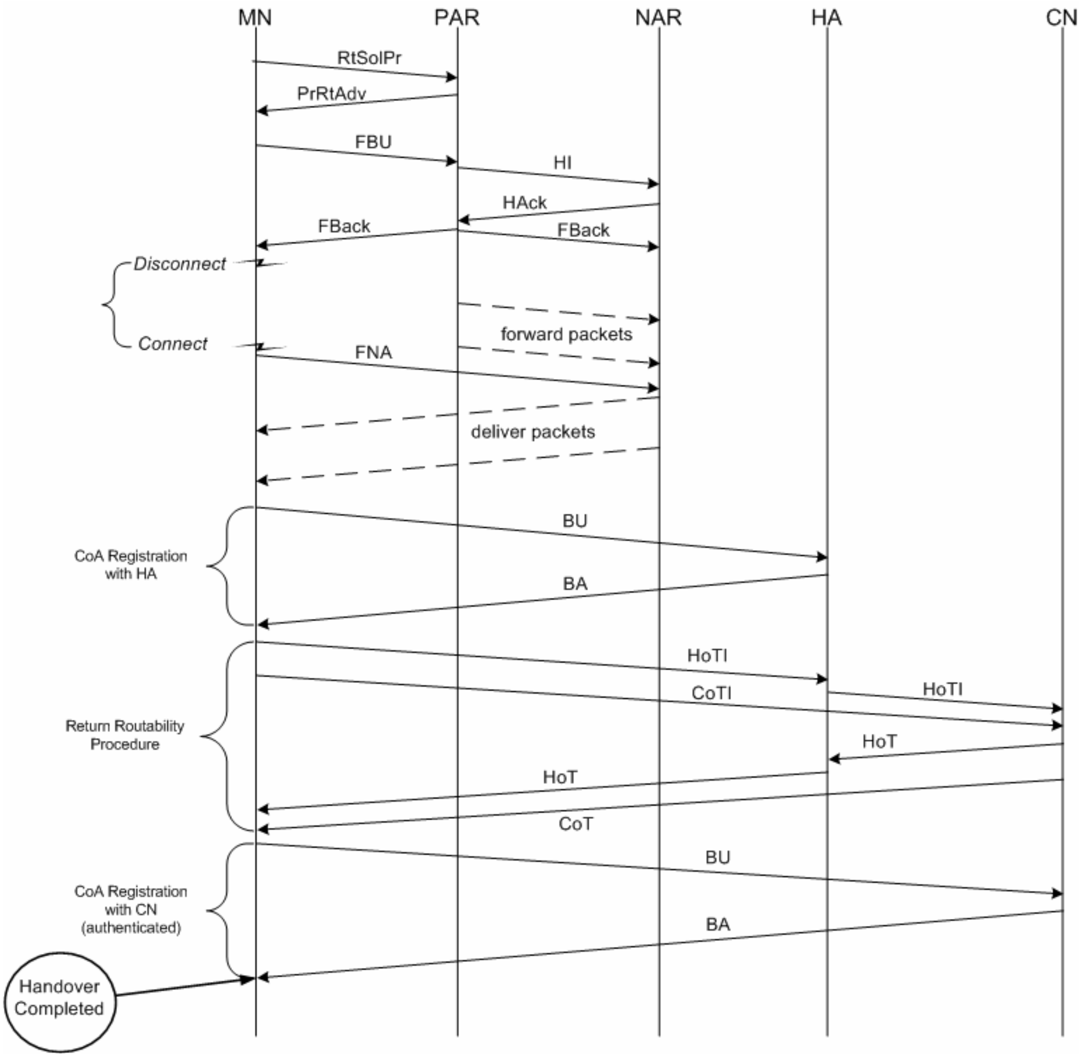

Figure 3.9 The FMIPv6 handoff procedure

The mobile node initiates the fast handover when a layer 2 trigger takes places. Then, the mobile node sends a Router Solicitation for Proxy Advertisement (RtSolPr) message to its access router to resolve one or more access point identifiers to subnet-specific information. In response, the access router (e.g. previous access router) sends a Proxy Router Advertisement (PrRtAdv) message. With the information provided in the Proxy Router Advertisement message, the mobile node forms a prospective new care-of address and sends a Fast Binding Update (FBU) message.

The purpose of the FBU update is to make the previous router to bind the previous care-of address (PCoA) to the new care-of address (NCoA) and establish tunnel between the previous access router (PAR) and the new access router (NAR), so that packets arrived from correspondent nodes can be tunneled to the new location of the mobile node. The FBU message should be sent from the mobile node at the previous access router’s link if possible. When the mobile node could not send the FBU message at the previous access router’s link, the FBU message is sent from the new link. It is encapsulated within a Fast Neighbor Advertisement (FNA) message to ensure that the NCoA does not conflict with an address already in use by some other node on link.

When the previous access router receives the FBU message, it sends Handover Initiate (HI) message to the new access router (NAR) to determine whether the NCoA is acceptable at the NAR. When the NAR verifies the NCoA, duplicate address detection (DAD) is performed to avoid duplication on links when stateless address autoconfiguration is used. Confirmed NCoA must be returned in the Handover Acknowledge (HAck) message from the NAR. Then, the PAR must in turn provide the NCoA in a Fast Binding Acknowledgment (FBAck). Thus, new care of address is determined by the exchange of HI and HAck messages.

The probability of interface identifier duplication on the same subnet is very low. In the fast handover, certain precautions are necessary to minimize the effects of duplicate address occurrences. In some cases, the NAR may already have the knowledge required to assess whether the mobile node's address is a duplicate or not before the mobile node moves to the new subnet. The result of this search is sent back to the PAR in the HAck message. The NAR can also rely on its trust relationship with the PAR before providing forwarding support for the mobile node. That is, it may create a forwarding entry for the new care-of address subject to approval from the PAR which it trusts. For preventing packet loss, this protocol provides an option to indicate request for buffering at the NAR in the HI message. When the PAR requests this feature for the mobile node, it should also provide its own support for buffering. Such buffering can be useful when the mobile node leaves without sending the FBU message from the previous access router's link. The PAR should stop buffering after processing the FBU message.

The usual MIPv6 handoff procedure for performing CoA registration with the HA and CN occurs after the FMIPv6 procedure. Latency effects on real time traffic will still exist but are reduced only to the time it takes to actually move, such as disconnect from the PAR and connect to the NAR.

3.3.2 Predictive Mode FMIPv6

Operations of the fast handover are composed of predictive mode and reactive mode. In this work, only predictive mode for FMIPv6 is considered. The predictive mode of operation is shown in Figure 3.10. In this mode of operation, the mobile node (MN) receives the FBAck message on the previous link. This means that packet tunneling would already be in progress by the time when the mobile node handovers to the new access router (NAR). As soon as the mobile node establishes link connectivity with the new access router, it should send a FNA message immediately, so that buffered packets can be forwarded to the mobile node right away.

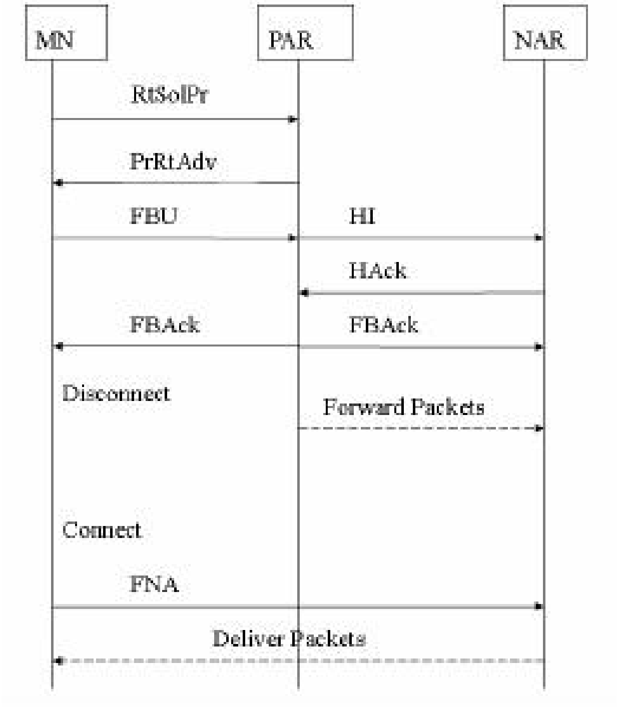

Figure 3.10 The predictive mode FMIPv6.

3.4 Network Simulator 2 (NS2)

The Network Simulator 2 (NS2) [25-27] is a discrete event driven simulator for computer networks and networks protocols. It is widely used for simulating local and wide area networks in networking research community as a large number of basic network components are available. The development of more complex network architecture can be facilitated by combining and building on top on these elements. Some of supported technologies and network protocols are

- Point-to-point connections, LANs, wireless links, satellite links
- Router queuing mechanisms (DropTail, RED, CBQ, etc)
- IP, Mobile IPv4
- Routing algorithms (Dijkstra etc)
- Multicasting (DVMRP, PIM, etc)
- Transport Protocol (TCP, UDP, RTP/RTCP, SRM, etc)
- Qos schemes (InterServ, DiffServ)
- Applications (Telnet, FTP, HTTP, etc)
- Mathematic support (random number generation, integrals, etc)
- Network emulation (i.e. interaction of the network simulator with a real operating mode)

NS2 was originally developed in 1989 at UC Berkeley as REAL network simulator for studying flow and congestion control schemes in packet-switched data networks. In 1995, several universities and research groups such as UCB, LBNL, ISI/USC, Sun, Xeroc PARC etc jointly developed various modules in NS2 for simulating variety of IP networks and network protocols. The NS2 project is now part of the VINT project that develops tools for display of simulation results, analysis and converters to integrate other network topology generators with NS2.

Simulation process in NS2 is shown in Figure 3.11. It is an Object-oriented Tcl (OTcl script) interpreter that has a simulation event scheduler, network component object libraries and network setup (plumbing) module libraries. An OTcl script is written to setup and run a simulation network. It first invokes the Tcl interpreter for basic network configuration and the program run using various modules in the simulator libraries. The simulator library initiates an event scheduler, sets up the network topology using the network objects and plumbing functions. It also notifies the traffic sources when to start and stop transmitting packets through the even scheduler. A new network object can be constructed by making a compound object from the object library and plumbing the data through the object.

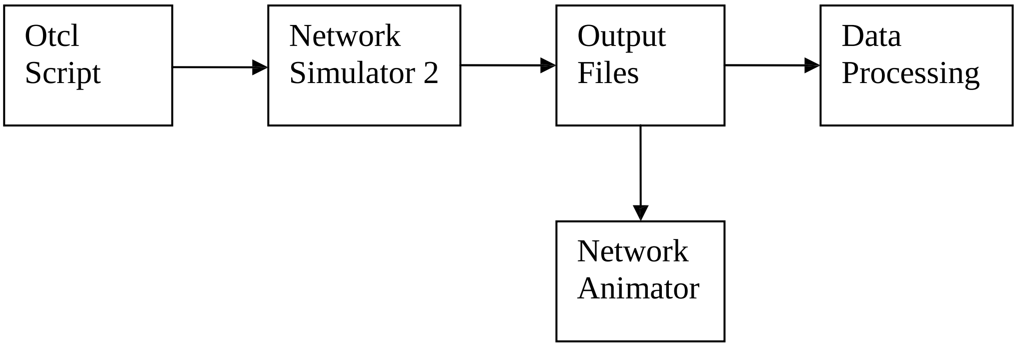

Figure 3.11 Simulation process in Network Simulator 2

The simulator framework uses a split-language programming approach. OTcl, an object-oriented language of Tcl, is used for the control structure and description of simulation scenarios. The event scheduling tasks and the dynamic configuration of network components during the simulation are also written in OTcl. The core processing unit of the simulator such as low-level event processing, packet processing and forwarding is written in C++ language. C++ is the object-oriented version of C, a very low-level programming language for control and structure. Therefore, using C++ in NS2 allows fast simulations for construction of large scenario. The compiled objects are made available to the OTcl interpreter through an OTcl linkage. In creates a matching OTcl object for each of the C++ objects, makes the control functions and the configurable variables specified by the C++ objects available to the corresponding OTcl object. The C++ and OTcl linkage and duality gives flexibility to network configuration but also adds complexity to the simulator. Particularly, error debugging in both languages simultaneously is a difficult task. To use and build the simulator, it is necessary to have knowledge and be proficient in both OTcl and C++ programming languages.

The network topology consists of network nodes connected by link with certain queuing model, delay and throughput. Agents are attached to the nodes to exchange packets between them. Traffic sources such as applications use these agents to communicate with traffic sinks at other nodes. There are several types of agent in NS2. A routing agent decides which link to forward a packet and a transport agent such as TCP or UDP sends and receives IP packets. Scenarios are usually constructed and written in the OTcl script by hand. When the simulation is finished, NS2 produces one or more text-based output files containing detailed simulation data. The data can be used for simulation analysis or visualized using a graphical simulation display toll called the Network Animator (NAM). For accurate simulation analysis such as calculating data throughput and network delay, these results can be graphically plotted from NS2 output trace files using plotting utilities such as GNUPlot or XGraph.

3.5 Network Model and Simulation Parameters

In this section, simulation topology and parameters are presented to compare the handoff latency in MIPv6 and FMIPv6. Previous simulation model based on ns-allinone-2.1b7a as in [28-30], is ported to ns-allinone-2.28 according to [31]. The network scenario for the simulation is shown in Figure 3.12.

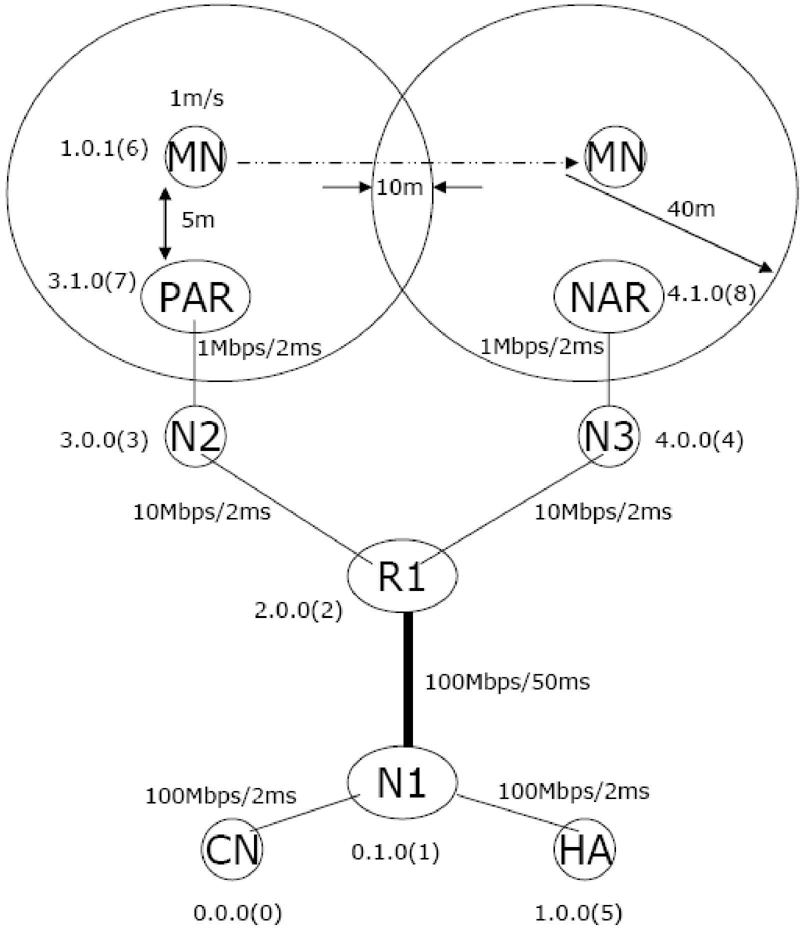

Figure 3.12 Network Model

The simulation model consists of a corresponding node (CN), a streaming VoIP traffic over UDP medium setup to a mobile node (MN), home agent (HA), gateway router N1, common router R1, routers N2 and N3, also previous access router (PAR) and new access router (NAR). The IEEE 802.11b is used as access technology and each access router has coverage area of 40 meters in radius with the overlapping region between PAR and NAR is 10 meters. The bandwidth and link delay between two intermediate wired nodes is set as shown in Figure 3.12. The L2 handoff delay is set to 20ms.

The CN produce a constant bit rate (CBR) traffic source, transmitting packets in an RTP over UDP medium. The MN acts as a sink, by receiving the packets from the CN at a constant inter-arrival rate. Loss monitor agent is attached to the MN to record the packet losses and throughput of the receiving packets.

In the beginning of the simulation, MN is situated near the HA. The CN start producing the CBR traffic 5s after the simulation started. One second later, the MN moves toward the transmission range of PAR (5m distance from PAR) at a very high speed of 100m/s. At 10 seconds after simulation started, the MN starts to move toward the NAR at a speed of 1m/s. The handoff process that is being considered, when MN moves from PAR toward the NAR.

3.5.1 VoIP Models

A one-way VoIP connection is modeled as a stream of packets with a fixed packet size and transmission rate [32]. In [33,34], a VoIP models is implemented based on [35] and included in Appendix A. In this project, VoIP models based on CBR traffic are used. The CN produces payload according to the voice coding payload size. The standard method of transporting voice packets through wireless local area network (WLAN) requires the addition of three headers. There are IP, UDP and RTP. An IPv6 header is 20 octets, a UDP header is 8 octets and RTP header is 12 octets. A total of 40 octets are therefore sent each time a packet containing voice payload is transmitted.

The main characteristics of the codec used in the simulation are summarized in Table 3.1.

Table 3.1 Audio/Voice codec parameters

Parameters	**G.711**	**G.723.1**	**G.729**
Bit rate (Kbps)	64	6.3	8
Framing interval (ms)	20	30	10
Payload (Bytes)	160	24	10
Packets/s, N_p	50	33	100

3.5.2 Wireless LAN Parameters

The network simulator will be used to form an appropriate network topology under the Media Access Control (MAC) layer of the IEEE 802.11b. According to the IEEE 802.11b protocol specifications, the parameters for the WLAN are shown in Table 3.2.

Table 3.2 Values of the IEEE 802.11b parameters

Parameter	**Value**
SLOT	20µsec
SIFS	10µsec
DIFS	50µsec
PHY header	192µsec
CW_{min}	31
CW_{max}	1023
Data Rate	Fixed at 11 Mbps
Basic Rate	1 Mbps
RTS	20
CTS	14
ACK	14

The two-ray ground reflection model is used in the simulation. This model considers both the direct path and a ground reflection path [36]. The received power P_r at a distance d from the transmitter for the two-ray ground reflection model can be expressed as:

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L} \qquad (3.1)$$

where G_t is the transmitter antenna gain, G_r is the receiver antenna gain, h_t is the height of the transmitter h_r is the height of the receiver and L is the system loss

The power level at which the packet was received at MAC layer is compared with the receiving threshold (RTX) and the carrier-sense threshold (CTX). If the power level falls below the carrier sense threshold, the packet is discarded as noise. If the received power level is above the carrier sense threshold but below the receive threshold, the packet is marked as a packet in error before being passed to the MAC layer. Otherwise, the packet is simply handed up to the MAC layer.

In NS-2, the transmitting power P_t is set to 0.03162 Watt, correspond to 15dbm. To accurately model the attenuation of communication radius between antennas close to the ground, in this simulation model, the RTX is set 40 meters, and the CTX with respect to the transmitting station, is set to 90 meters. Thus the RTX and the CTX can calculated as $1.90448e^{-9}$ and $3.76193e^{-10}$, respectively. The height of the omni-directional antennas is 1.5m above the ground plane operating in the 2.432 GHz frequency.

3.6 Conclusion

In this chapter, the network model and parameters that used for analysis and simulation are presented. The operation of MIPv6 and FMIPv6 are simulated according to the model but with some assumptions.

For both network layer mobility schemes, no route optimization is employed which mean that all packets from CN will be sent to HA first and then forwarded to MN. Stateless address autoconfiguration is used during CoA configuration and there is no delay due to DAD.

Simulation model uses the Network Simulator to simulate the network topology. The simulator works as a tool to analyze VoIP performance during handoff with different voice coding schemes and able to provide results such as handoff latency, throughput and delay.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

The results for MIPv6 and FMIPv6 framework such as handoff latency, packet losses, average throughput and end to end delay are obtained by analyzing the output files produced by NS2 simulations. Simulations software tool used in this project is NS2 version ns-allinone-2.28 on top of Red Hat Linux 9.0 environment. The example of NAM simulation is shown in Figure 4.1.

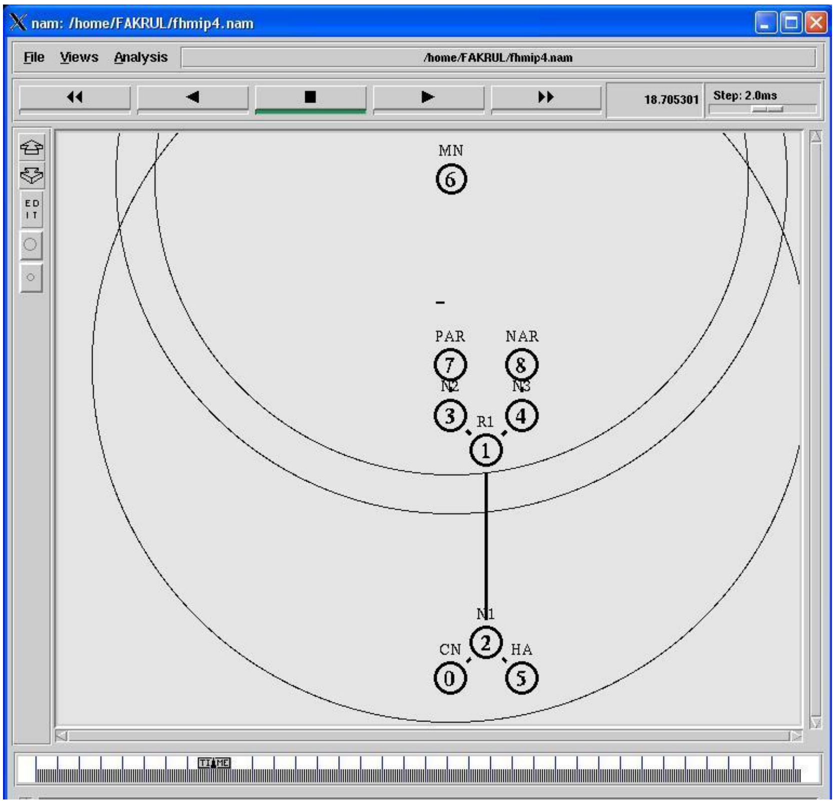

Figure 4.1 NAM simulation result

4.2 Simulation Results

The results obtained are as an average of 10 simulation times for each different voice coding scheme. For MIPv6 simulation, handoff latency and packet losses are observed during the movement of MN from PAR to NAR in all 10 independent simulation events. The packet number received by MN during the simulation time, sieved from one chosen simulation even (the maximum handoff latency obtained) is shown in Figure 4.2. The middle line is for G.711 codec while the upper line is for G.729 codec.

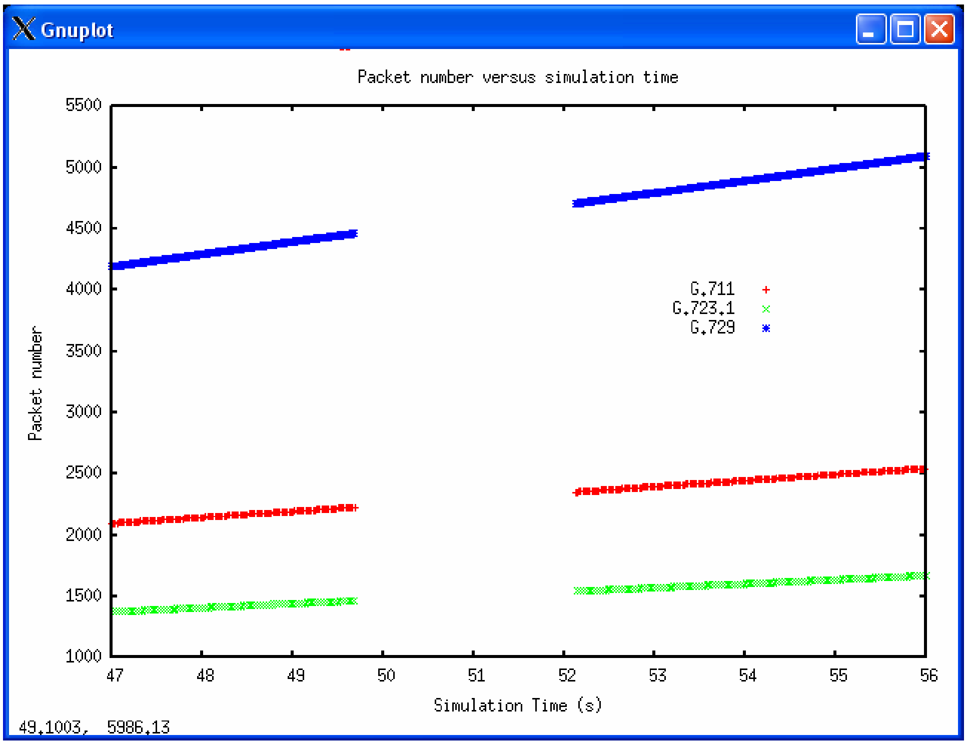

Figure 4.2 Packet number received versus simulation time for MIPv6

For MIPv6 scheme, packets will be dropped during handoff time and the UDP transport protocol does not react to this condition. Handoff latency is calculated as the time between the last packet receive by MN from PAR and the first packet MN receives from NAR. The handoff latency will result in service disruption for all three different codecs. The handoff latency for MIPv6 is shown in Figure 4.3. The average handoff latency calculated for all three codecs are almost same. The highest handoff latency occurred in G.729 codecs while the lowest handoff latency occurred in G.711 codec. The difference between these two codecs is just 24ms. This shows that the use of different codecs do not have significant effect on handoff latency.

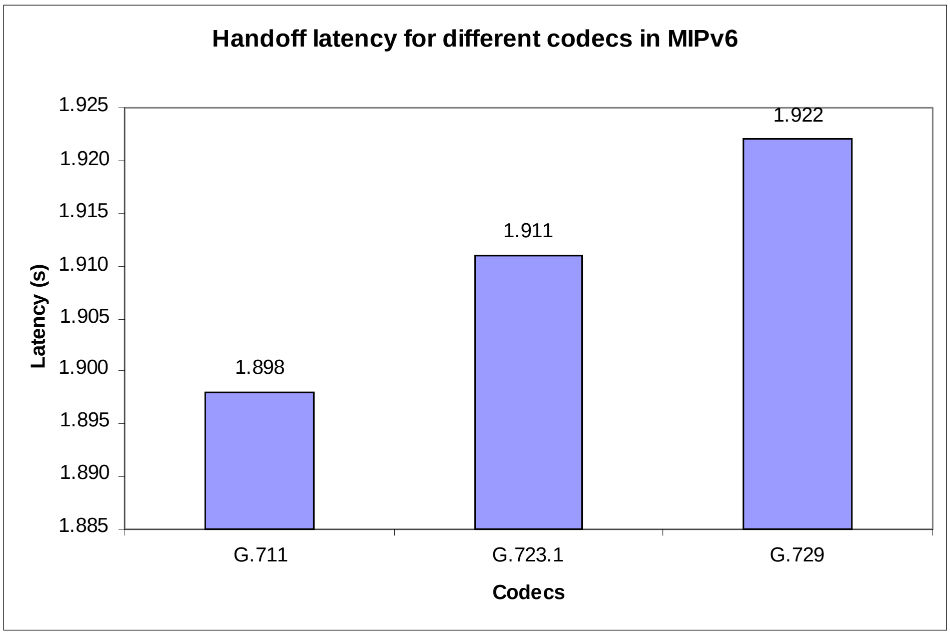

Figure 4.3 Handoff latency in MIPv6

Average packet losses for MIPv6 are shown in Figure 4.4. G.723.1 has the least number of packet losses (63 packets, correspond to packet loss ratio of 2.69%) followed by G.711 with 95 packets (packet loss ratio of 2.66%) and G.729 with 193 packets (packet loss ration of 2.7%). The number of packet losses are proportional to the number of packet sent every seconds or the framing interval.

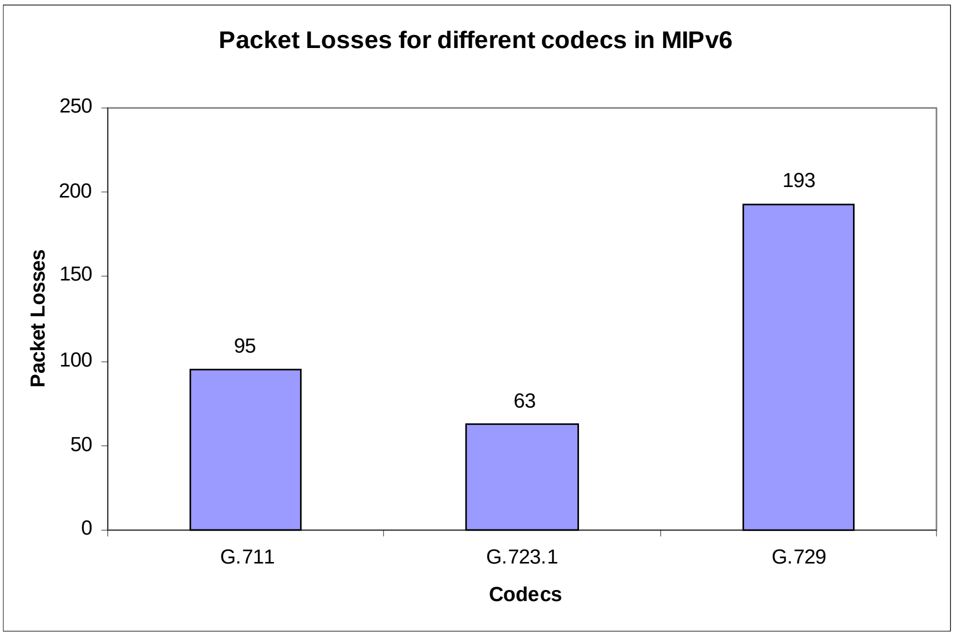

Figure 4.4 Packet losses in MIPv6

Throughput comparison for different codecs in MIPv6 sieve from one chosen simulation even is shown in Figure 4.5. For all codecs, the throughput will drop during handoff since no packet receives by MN at that time.

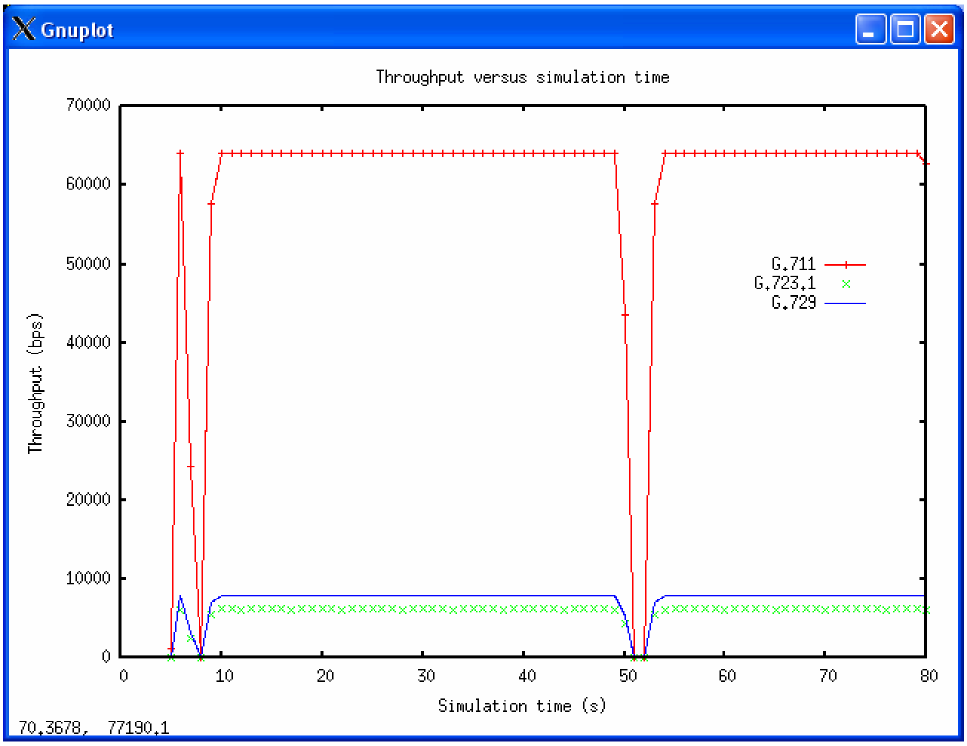

Figure 4.5 Throughput for different codecs in MIPv6

For FMIPv6 framework, for all different voice coding schemes, no packet loss is observed during the handoff time. The handoff latency for FMIPv6 is due to the routing when forwarding packet from the PAR to NAR. The average time calculate from the time PAR receives the HACK message from NAR until the NAR receives FNA message for all different codecs is approximately 140ms. The details view of packet number received by MN for FMIPv6 with G.711 coding scheme being considered is shown in Figure 4.6. In between 40.9s and 41.1s of simulation time, the total distraction of time when packet received MN is due to the packet tunneling form PAR to NAR.

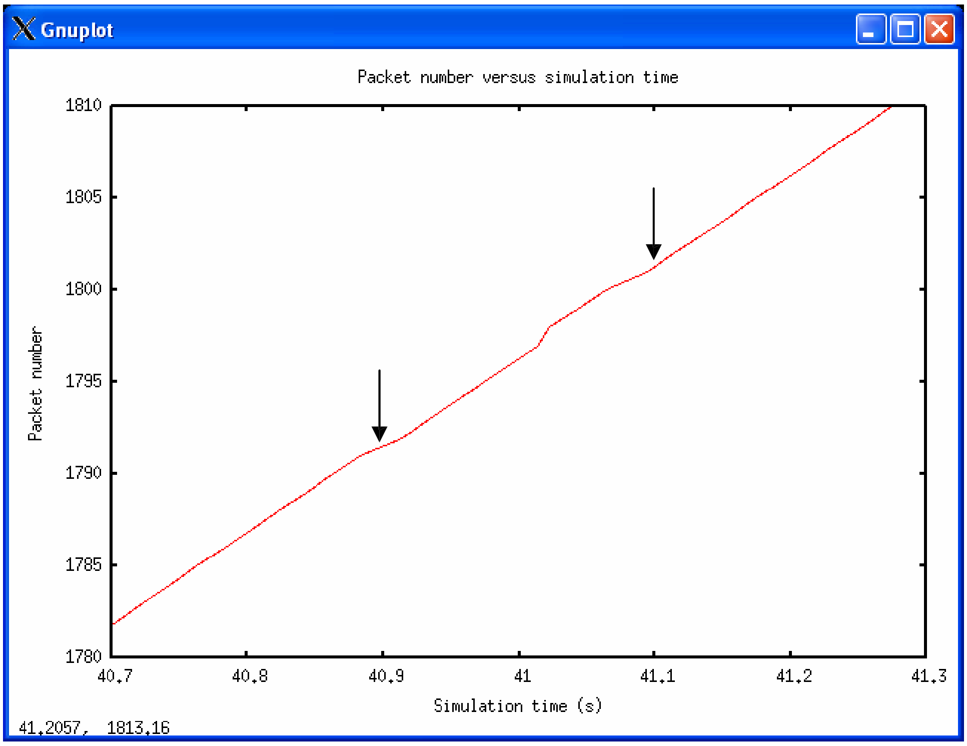

Figure 4.6 Packet number received versus simulation time in FMIPv6

Throughput comparison between MIPv6 and FMIPv6 is shown in Figure 4.7. Again, only G.711 coding scheme is considered here. For MIPv6, when handoff occurs, packets will be dropped at PAR, so MN does not receive any packets from CN. In comparison, simulation using FMIPv6 shows that during handoff, MN will always receive packets sent from CN but with additional delay. This means that there is no significant drop in throughput for FMIPv6, as experienced using MIPv6.

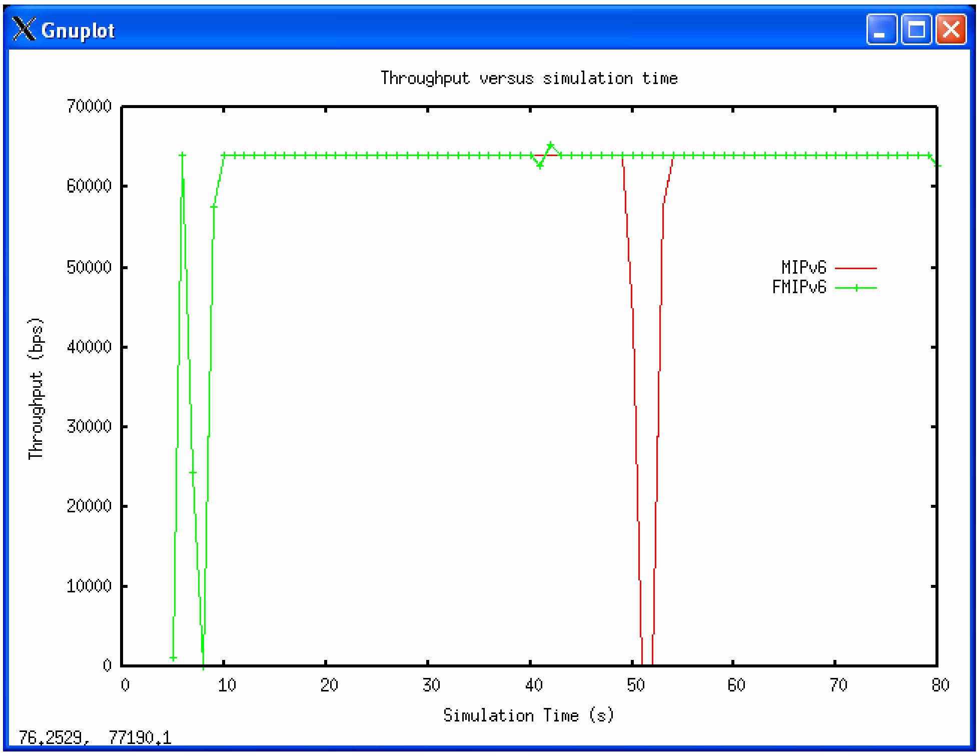

Figure 4.7 Throughput comparisons between MIPv6 and FMIPv6

Average throughput comparison between MIPv6 and FMIPv6 for three different codecs is shown in Figure 4.8. In terms of average throughput, the FMIPv6 scheme achieves higher system performance due to the fact that there is no packet loss observed in during handoff time

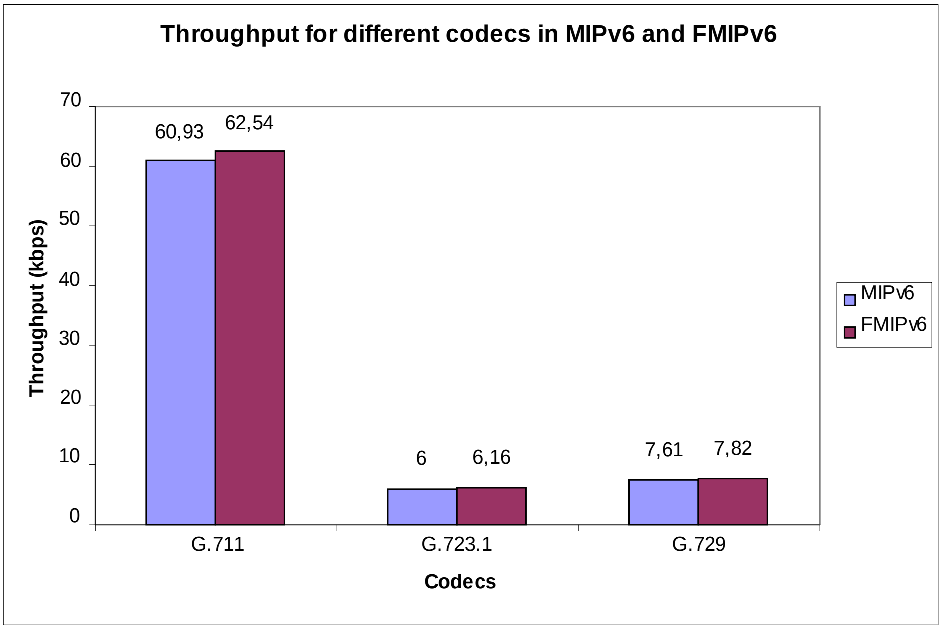

Figure 4.8 Throughput comparisons in MIPv6 and FMIPv6

The end to end delay comparison between MIPv6 and FMIPv6 is shown in Figure 4.9. This delay is the time taken for a packet to route from corresponding node to mobile node. For all three different codecs, the end to end delays for FMIPv6 scheme are slightly higher than in MIPv6. This could be due to the fact that there is bidirectional tunnel exists between PAR and NAR during handoff time. This mean than some packets may arrive at the mobile node at a slightly delayed time. Furthermore, there are additional messages need to be exchange by nodes in order for FMIPv6 to function. These extra messages make the signaling cost in FMIPv6 higher than in MIPv6.

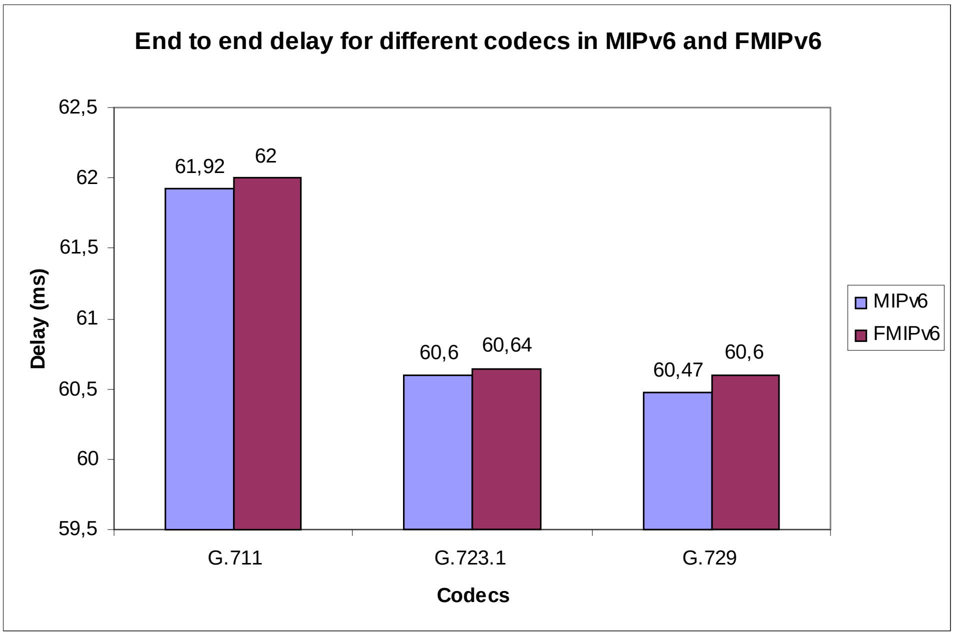

Figure 4.9 End to end delay in MIPv6 and FMIPv6

4.3 Conclusion

In this chapter, the MIPv6 and FMIPv6 schemes have been analyzed and compared. The simulation results show that FMIPv6 reduces the delay during handoff. FMIPv6 produces higher throughput compared to MIPv6 and allow VoIP application on wireless medium without any packet loss but with slight additional end to end delay due to its more complex procedure.

The main advantage of FMIPv6 is that it manages to reduce the handoff latency significantly. This helps to provide seamless communication and improve the performance of VoIP application.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

Voice over Internet Protocol (VoIP) faced many challenges in order to be deployed successfully. One of the obstacles is the delay due to handoff process. During handoff, a mobile node cannot receive or send any packets. This period of time is called the handoff latency which can degrade the performance of real time applications. Three different types of voice coding schemes to model the VoIP application namely G.711, G.723.1 and G.729 are analyzed to study their effect on handoff latency and other Quality of Service parameters.

Mobile IPv6 is a network layer solution to the mobile internet. MIPv6 allows a mobile node to move from one link to another without changing the mobile node home address. The movement of a mobile node away from its home link is transparent to transport layer and higher layer protocols and applications. One problem of MIPv6 is that the latency due to the movement of mobile node from one network to another is still large, which can result in service disruption. This handoff latency needs to be reduced in order to improve the performance of VoIP.

FMIPv6 is one of the enhanced handoff schemes introduced to reduce the handoff latency in MIPv6. There are four additional messages introduced in handoff procedure for FMIPv6. In FMIPv6, several portions of the L3 handoff are performed in advance prior to the handoff process to reduce the handoff latency. In the NS2 simulation, no route optimization and Duplicate Address Detection are employed for both MIPv6 and FMIPv6 frameworks.

Handoff latency for MIPv6 obtained in the simulation is approximately 1.9 seconds and packet losses occurred for all three different types of voice coding schemes. The simulation results show that FMIPv6 only experience transmission delay due to routing when forwarding packet from the PAR to NAR. The duration for tunneling process is approximately 140ms and therefore reduces the latency during handoff. FMIPv6 produces higher throughput compared to MIPv6 and allow VoIP application on wireless medium without any packet loss but with slight additional end to end delay due to its more complex procedure.

In conclusion, FMIPv6 manages to reduce the handoff latency significantly. This helps to provide seamless communication and improve the performance of VoIP application.

5.2 Proposed Future Work

Further works should be carried out to in order to improve the overall performance of real time traffic during handoff as suggested below:

i. Extend the simulation to support multiple users and real time video transmission based on MPEG 4 or H.263 coding schemes.

ii. Handoff latency simulation between two different systems such as 3G and Wireless LAN or vertical handoff.

iii. Use combination of different layers in TCP/IP protocol or cross layer design concept to reduce handoff latency and improve overall performance of VoIP

REFERENCES

1. Xavier, P. C and Hannes, H. “A simulation study on the performance of Mobile IPv6 in a WLAN-based cellular network.” Elseiver Science B. V. 2002. 191-204
2. Marko, L. “Voice Over IP” Seminar on Internetworking. HUT TML, 2001.
3. Wei, W., Soung C. L. and Victor O. K. L. “ Solution to Performance Problem in VoIP Over a 802.11 Wireless LAN”, IEEE Transactions on vehicular technology, vol. 54. no. 1, January 2005.
4. NetScout Systems Inc.*Voice Over IP (VoIP) Implementation Guide for Network Performance Management.* USA, 2005. http://www.netscout.com
5. Timothy, V. K. *VoIP for Dummies*. Indianapolis, Indiana: Wiley Publising Inc, 2005.
6. Henry, S. and Alan, B. J. *Internet Communications Using SIP*. 2nd Edition. Indianapolis, Indiana: Wiley Publishing Inc, 2006.
7. Alan, B. J. *SIP: Understanding the Session Initiation Protocol*. 2nd Edition. Norwood, MA: Artech House Inc, 2004.
8. Alias, M. and Ong, L. L. “Performance of Voice over IP (VoIP) over a wireless LAN (WLAN) for different audio/voice codecs”, Jurnal Technologi, UTM, 2007.
9. M. Dunmore, and T. Pagtzis, “Mobile IPv6 Handovers: Performance Analysis and Evaluation”, 6net, June 2005.
10. ISO, “OSI Routing Framework” ISO/TR 9575, 1989.
11. Soliman, H., Castelluccia, C., El-Maliki, K., Bellier, L, “Hierarchical Mobile IPv6 IPv6 Mobility Management (HMIPv6)” IETF, RFC 4140, August 2005.
12. R. Koodli, “Mobile IPv6 Fast Handovers,” IETF, RFC 5268, June 2008.
13. Hsieh, R., Zhou, Z. G., and Seneviratne, A. “S-MIP: A Seamless Handoff Architecture for Mobile IP”. In *Proceedings of IEEE INFOCOM*, vol. 3, March 2003, pp. 1774-1784.
14. T.T. Kwon, M. Gerla, and S. Das. "Mobility management for VoIP service: Mobile IP vs. SIP". IEEE Wireless Communications, Volume 9, Issue 5:66-75, Oct. 2002.

15. I. Vivaldi, M.H. Habaebi, B.M. Ali, and V. Prakesh. "Fast Handover Algorithm for Hierarchical Mobile IPv6 Macro-Mobility Management". In The 9th Asia-Pacific Conference on Communications. APCC 2003., volume 2, pages 630-634, 21-24 Sept 2003.
16. J. Chow and G. Garcia. "Macro- and Micro-Mobility Handoffs in Mobile IP based MBWA Networks". In IEEE Global Telecommunications Conference, 2004. GLOBECOM '04., volume 6, pages 3921-3925, 29 Nov.-3 Dec. 2004.
17. J. Zhang, D.A.J. Pearce, and T.C. Tozer. "Two-way Registration: a Fast Handoff Scheme for IPv4 Macro-Mobility Management". In 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, volume 2, pages 1252-1256, 5-8 Sept 2004.
18. D. Johnson, C. Perkins, and J. Arkko, “Mobility Support in IPv6,” RFC 3775, Jun 2004.
19. X. P. Costa, M. T. Moreno and H. Hartenstein, “A Performance Comparison of Mobile IPv6, Hierarchical Mobile IPv6, Fast Handovers for Mobile IPv6 and their Combination,” Mobile Computing and Communication Review, 2003. vol. 2, no. 4.
20. Y. A. Yoon, H. Y. Byung, W. L. Kang, Z. C. You, and Y. J. Woo, “Reduction of Handover Latency Using MIH Services in MIPv6,” IEEE Computer Society, 2006.
21. J. J. Woo, J. K. Hyung, J. L. Tae, C. Hyunseung and Y. C. Min, “Cross-Layer Design for Reducing Handoff Latency in Mobile Network” Springer-Verlag, ICCSA 2007, pp. 216-225.
22. T. Narten, E. Nordmark, W. Simpson, “Neighbor Discovery for IP Version 6 (IPv6)”, IETF RFC2461, December 1998.
23. S. Thomson, T. Narten, “IPv6 Stateless Address Autoconfiguration”, IETF RFC 2562, December 1999.
24. Y. Mun and H. K. Lee, *Understanding IPv6.* Spring Street, NY: Springer, 2005.
25. http://www.isi.edu/nsnam/ns/
26. E. Altman and T. Jimenez, “NS Simulator for beginners” Lecture notes, December 4, 2003.
27. Jae, C and Mark, C. “NS by Example” Worcester Polytechnic Institute.

28. R. Hsieh, A. Seneviratne, H. Soliman, and K. El-Malki, “Performance Analysis on Hierarchical Mobile IPv6 with Fast-Handoff over End-to-End TCP” Globecom, Taipei, Taiwan, 2002.
29. S. Haseeb and A. F. Ismail, “Handoff latency analysis of mobile IPv6 protocol variations,” Elsevier, 2006, pp. 849–855
30. J. Widmwr, “Extension to the ns Network Simulator,”http://www.informatik.uni-mannheim.de/~widmer.
31. S. Yankov and S. Wiethoelter, “ Handover Blackout Duration of Layer 3 Mobility Management Schemes”, TKN Technical Report TKN-06-002, Berlin, May 2006.
32. D. S. Nursimloo, G. K. Kalebaila, and H. A. Chan, “ A Two-Layered Mobility Architecture Using Fast Mobile IPv6 and Session Initiation Protocol”, Hindawi Publishing Corporation, 2008.
33. Marc T. M., Xavier, P. C and Sebastia, S. R. “A Performance Study of Fast Handover for Mobile IPv6,” Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks, 2003.
34. C-N.Chuah et al., “QoS Provisioning Using a Clearing House Architecture,” International Workshop on Quality of Service (IWQoS), June 2000.
35. ITU-T Recommendation P.59, *Artificial Conversational Speech*, March, 1993.
36. K. Fall and K. Varadhan, “The ns Manual” The VINT Project, March 14, 2008.

APPENDIX A

NS2 MODEL FOR VOIP SOURCE AND EXAMPLE

voice.tcl

```
#
# 10/21/1998  C-N. Chuah
# This file contains: on-off Markov model of voice sources
#                     random variables (uniforms & exponentials)
#                     CBR source


proc randomize v {
    return [expr 0.5 * $v + double([ns-random] % 10000000) / 1e7 *
$v]
}

proc uniform01 {} {
    return [expr double([ns-random] % 10000000) / 1e7]
}

proc uniform { a b } {
    return [expr ($b - $a) * [uniform01] + $a]
}

proc exponential lambda {
    return [expr - $lambda * log([uniform01])]
}

proc exponential2 lambdaInv {
    return [expr - $lambdaInv * log([uniform01])]
}

proc trunc_exponential lambda {
    while 1 {
        set u [exponential $lambda]
        if { $u < [expr 4 * $lambda] } {
            return $u
        }
    }
}

# create a single CBR audio source
proc create_audio_src { nodeSrc audioDest pktSize} {
    global ns framelength
#    set audioSource [new Agent/CBR]
    set udp0 [new Agent/UDP]
    $ns attach-agent $nodeSrc $udp0
    set cbr [new Application/Traffic/CBR]
    $cbr set interval_ ${framelength}
    $cbr set packetSize_ $pktSize
    $cbr attach-agent $udp0

    set null1 [new Agent/Null]
    $ns attach-agent $audioDest $null1

    $ns connect $udp0 $null1
    return $cbr
}

# create all of the audio sources
proc create_audio_model { numSources nodeSrc nodeDest pktSize
startInSteadyState } {
    global ns audioLambdaInv audioMuInv audioStartTime

    set audioDest [new Agent/Null]
    $ns attach-agent $nodeDest $audioDest

    set onProb [expr $audioMuInv / ($audioMuInv + $audioLambdaInv)]
    for {set base $numSources} {$base > 0} {incr base -1} {
        set audioSource [create_audio_src $nodeSrc $nodeDest
$pktSize]

# the audio source's class will be either 2*$base or 2*$base + 1
#        $audioSource set class_ [expr (2 * $base) + 1]

      $audioSource set class_ $base
        if {$startInSteadyState} {
           if {[uniform01] <= $onProb} {
                $ns at $audioStartTime "start_audio_source
$audioSource"
            } else {
                set time [exponential2 $audioLambdaInv]
                $ns at [expr $audioStartTime + $time]
"start_audio_source $audioSource"
            }
        } else {
            $ns at $audioStartTime "start_audio_source $audioSource"
        }
    }
}

proc start_audio_source { audioSource } {
    global ns audioMuInv
    $audioSource start
    set time [exponential2 $audioMuInv]
    $ns at [expr [$ns now] + $time] "stop_audio_source $audioSource"

}

proc stop_audio_source { audioSource} {
    global ns audioLambdaInv
    $audioSource stop
    set time [exponential2 $audioLambdaInv]
    $ns at [expr [$ns now] + $time] "start_audio_source
$audioSource"

    # now toggle class from even to odd (or v.v.)
    set class [$audioSource set class_]
}

# }}}
# {{{ Timer Class
# A simple timer class.  You can derive a subclass of Timer
# to provide a simple mechanism for scheduling events:
#
#       $self sched $delay -- causes "$self timeout" to be called
#                               $delay seconds in the future
#       $self cancel       -- cancels any pending scheduled callback
#

Class Timer

Timer instproc sched delay {
    global ns
    $self instvar id_
    $self cancel
    set id_ [$ns at [expr [$ns now] + $delay] "$self timeout"]
}

Timer instproc randsched lambdaInv {
    global ns
    $self instvar id_
    $self cancel
    set time [exponential2 $lambdaInv]
    set id_ [$ns at [expr [$ns now] + $time] "$self timeout"]
}

Timer instproc destroy {} {
    $self cancel
}

Timer instproc cancel {} {
    global ns
    $self instvar id_
    if [info exists id_] {
        $ns cancel $id_
        unset id_
    }
}


# note also that we account for the average percentage of time an
# audio source is "on" to determine how many will fit within th
# desired bandwidth...
# (if each source stays on with average time 1/mu, and off w/ avg
# time 1/lambda, then the fraction of time it's on is
# (1/mu)/(1/lambda + 1/mu)
# the fraction of time audio sources are really on:

set fractionOn [expr $audioMuInv / ($audioLambdaInv + $audioMuInv)]

if {$intBW != 0} {
    # compute the average inter-packet rate:
    set intPktRate [expr (8.0 * $intPktSize) / ($intBW * 1000)]
    # make the interfering traffic agent, set the class and packet
size, attach
    # it to a null sink destination, and start it at the appropriate
time:
    set intTraffic [new Agent/Message/Poisson $intPktRate]
    $intTraffic set class_ 0
    $intTraffic set packetSize_ $intPktSize
    $ns attach-agent $sourceNode $intTraffic
    set intDest [new Agent/Null]
    $ns attach-agent $destNode $intDest
    $ns connect $intTraffic $intDest
    $ns at $intStartTime "$intTraffic start"
}
```

test1.tcl

```
# Part of Clearing House test suite: test1.tcl 1/22/2000  C-N. Chuah
# Objective: experiment different path selection algorithms
#            add CH nodes

set ns [new Simulator]
set simTime 2
$ns color 0 blue
$ns color 1 red
$ns color 2 white

# }}}
# {{{ Source Constants
# Audio:
set audioMuInv 0.4
set audioLambdaInv 0.6
set intBW 0
set audSrcBitrate 6.3
# in kbps

set Fs 8000
set sourceBitPerSample 8
# length (in time) that audio frames cover

set framelength 0.030
set audioStartTime 0.0
set audioPktSize [expr round([expr $framelength * $audSrcBitrate *
1000 / 8.0])]
source voice.tcl

proc build_topology { ns which } {
        global bw delay
        foreach i "0 1 2 3 4" {
                global n$i
                set n$i [$ns node]
        }
        if { $which == "FIFO" } {
            $ns duplex-link $n0 $n1 3Mb 10ms DropTail
            $ns duplex-link $n0 $n2 1.5Mb 10ms DropTail
            $ns duplex-link $n1 $n2 1.5Mb 10ms DropTail
            $ns duplex-link $n1 $n3 1.5Mb 10ms DropTail
            $ns duplex-link $n1 $n4 1.5Mb 10ms DropTail
        } elseif { $which == "RED" } {
            $ns duplex-link $n0 $n1 $bw $delay  RED
        } else {
            $ns duplex-link $n0 $n1 $bw $delay FQ
        }
}

Queue set limit_ 5

set f [open out.tr w]
$ns trace-all $f

build_topology $ns FIFO
create_audio_model 2 $n0 $n3 $audioPktSize 1
create_audio_model 2 $n0 $n1 $audioPktSize 1
create_audio_model 2 $n0 $n4 $audioPktSize 1
create_audio_model 2 $n0 $n2 $audioPktSize 1
```

```
set tcp [$ns create-connection TCP $n0 TCPSink $n3 0]
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 0.2 "$ftp start"
# $ns at 1.2 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n3
$sink"

$ns at [expr $audioStartTime + $simTime] "exit"
$ns run
```

APPENDIX B

NS2 SIMULATION SCRIPTS

```
#
# This script was written for the sole purpose of showing the fhmip
ns-2 extnesion.
#
# July 2003 - Robert Hsieh
#
# upgraded to NS2.28 by Fakrulradzi Bin Idris (UTM/Skudai), August
2008  (FHMIP+NOAH)

# Unity gain, omni-directional antennas
# Set up the antennas to be centered in the node and 1.5 meters
above it
Antenna/OmniAntenna set X_ 0
Antenna/OmniAntenna set Y_ 0
Antenna/OmniAntenna set Z_ 1.5
Antenna/OmniAntenna set Gt_ 1.0
Antenna/OmniAntenna set Gr_ 1.0

Phy/WirelessPhy set L_ 1.0                      ;# System Loss
Factor
Phy/WirelessPhy set freq_ 2.432e9               ;# channel-5 >>
2.432GHz
Phy/WirelessPhy set bandwidth_ 11e6             ;# (11Mb)
Phy/WirelessPhy set Pt_ 0.031622777             ;# Transmit Power
(Watt) or 15dbm
Phy/WirelessPhy set CPThresh_ 10.0              ;# Collision
Threshold
Phy/WirelessPhy set CSThresh_ 3.76193e-10       ;# Carrier Sense
Power at 90m
Phy/WirelessPhy set RXThresh_ 1.90448e-9        ;# Receive Power
Threshold at 40m range

#DSSS (IEEE802.11b)
Mac/802_11 set SlotTime_ 0.000020               ;# 20us
Mac/802_11 set SIFS_ 0.000010                   ;# 10us
Mac/802_11 set PreambleLength_ 144              ;# 144 bit
Mac/802_11 set PLCPHeaderLength_ 48             ;# 48 bits
Mac/802_11 set PLCPDataRate_ 1.0e6              ;# 1Mbps
Mac/802_11 set dataRate_ 11.0e6                 ;# 11Mbps
Mac/802_11 set basicRate_ 1.0e6                 ;# 1Mbps

#Mac/802_11 set basicRate_ 1Mb
#Mac/802_11 set dataRate_  11Mb

Mac/802_11 set CWMin_ 31
Mac/802_11 set CWMax_ 1023
Mac/802_11 set RTSThreshold_ 0                  ;# bytes: with
RTS/CTS, replace with 3000 to turn RTS/CTS off
Mac/802_11 set ShortRetryLimit_ 7               ;# retransmittions
Mac/802_11 set LongRetryLimit_ 4                ;# retransmissions

set rng [new RNG]
```

```
$rng seed 0
set tmp [new RandomVariable/Uniform]
$tmp set min_ 65250
$tmp set max_ 65750
$tmp use-rng $rng
$rng seed [expr int([$tmp value])]
set opt(seed) [$rng seed]
if {$opt(seed) > 0} {
puts stderr "Seeding Random number generator with $opt(seed)\n"
ns-random $opt(seed)
}

set ns_ [new Simulator]
$ns_ node-config -addressType hierarchical

AddrParams set domain_num_ 5
 lappend cluster_num 2 1 1 2 2
AddrParams set cluster_num_ $cluster_num
lappend eilastlevel 1 1 2 1 1 1 1 1
AddrParams set nodes_num_ $eilastlevel


# to show ack number, header flags, header length
# Note: only useful though if using tcpfull
#Trace set show_tcphdr_ 1



set tracefd [open fhmip4.tr w]
$ns_ use-newtrace
$ns_ trace-all $tracefd


set namtracefd [open fhmip4.nam w]
$ns_ namtrace-all $namtracefd


set topo [new Topography]
$topo load_flatgrid 1000 1000
set god_ [create-god 1]


##############
# NODE SETUP #
##############

# Wired nodes => CH, R1, N1, N2, N3
#

#CH - 0
set CN [$ns_ node 0.0.0]

#R1 - 1
set R1 [$ns_ node 2.0.0]

#N1 - 2
set N1 [$ns_ node 0.1.0]

#N2 - 3
set N2 [$ns_ node 3.0.0]
```

```
#N3 - 4
set N3 [$ns_ node 4.0.0]

# NOAH nodes (wireless+wired) => HA, PAR, NAR
# MN is a special node (i.e. a NOAH node with wiredrouting turned
off)

set chan_ [new Channel/WirelessChannel]
$ns_ node-config -mobileIP ON \
                 -adhocRouting NOAH \
                 -llType LL \
                 -macType Mac/802_11 \
                 -ifqType Queue/DropTail/PriQueue \
                 -ifqLen 50 \
                 -antType Antenna/OmniAntenna \
                 -propType Propagation/TwoRayGround \
                 -phyType Phy/WirelessPhy \
                 -channel $chan_ \
                 -topoInstance $topo \
                 -wiredRouting ON \
                 -agentTrace ON \
                 -routerTrace OFF \
                 -macTrace ON


#HA - 5
set HA [$ns_ node 1.0.0]
[$HA set regagent_] priority 3

#MN - 6
$ns_ node-config -wiredRouting OFF
set MN [$ns_ node 1.0.1]
[$MN set regagent_] set home_agent_ [AddrParams addr2id [$HA node-
addr]]
$ns_ node-config -wiredRouting ON

#PAR - 7
set PAR [$ns_ node 3.1.0 2.0.0]
[$PAR set regagent_] priority 3

#NAR - 8
set NAR [$ns_ node 4.1.0 2.0.0]
[$NAR set regagent_] priority 4



#####################
# PLACEMENT of NODE #
#####################

$CN set X_ 80.0
$CN set Y_ 5.0

$N1 set X_ 120.0
$N1 set Y_ 10.0

$HA set X_ 160.0
$HA set Y_ 5.0

$MN set X_ 160.0
$MN set Y_ 5.1
```

```
$R1 set X_ 120.0
$R1 set Y_ 15.0

$N2 set X_ 85.0
$N2 set Y_ 60.0

$N3 set X_ 155.0
$N3 set Y_ 60.0

$PAR set X_ 85.0
$PAR set Y_ 135.0

$NAR set X_ 155.0
$NAR set Y_ 135.0

$ns_ at 0.0 "$CN label CN"
$ns_ at 0.0 "$N1 label N1"
$ns_ at 0.0 "$HA label HA"
$ns_ at 0.0 "$MN label MN"
$ns_ at 0.0 "$R1 label R1"
$ns_ at 0.0 "$N2 label N2"
$ns_ at 0.0 "$N3 label N3"
$ns_ at 0.0 "$PAR label PAR"
$ns_ at 0.0 "$NAR label NAR"

##############
# LINK SETUP #
##############

# droptail = (FIFO), RED = Random Early Detection
$ns_ duplex-link $CN $N1 100Mb 2ms RED         ;# Since consitiute a
domain, so we simplify it by just use 100M and keep the delay of 2ms
constant
$ns_ duplex-link $HA $N1 100Mb 2ms RED         ;# same as above
$ns_ duplex-link $R1 $N1 100Mb 50ms RED       ;# We increase the
dealy to 50ms to show the advantange of R1
$ns_ duplex-link $N2 $R1 10Mb 2ms RED         ;# All nodes below R1
belongs to a single domain, therefore we keep the delay at constant
2ms and vary the
$ns_ duplex-link $N3 $R1 10Mb 2ms RED         ;#  bandwidth in a
decreasing order, i.e. from 100M to 10M to 1M.
$ns_ duplex-link $PAR $N2 1000Kb 2ms DropTail
$ns_ duplex-link $NAR $N3 1000Kb 2ms DropTail

$ns_ duplex-link-op $CN $N1 orient right-up
$ns_ duplex-link-op $HA $N1 orient left-up
$ns_ duplex-link-op $N1 $R1 orient up
$ns_ duplex-link-op $R1 $N2 orient left-up
$ns_ duplex-link-op $R1 $N3 orient right-up
$ns_ duplex-link-op $PAR $N2 orient down
$ns_ duplex-link-op $NAR $N3 orient down
```

```
#####################
# APPLICATION SETUP #
#####################


# Setup a UDP connection
set udp0 [new Agent/UDP]
$udp0 set fid_ 2
$ns_ attach-agent $CN $udp0

$ns_ color 2 Red

# Setup a CBR over UDP connection
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 160     ;# according to codec used
$cbr0 set interval_ 0.02      ;# optional
$cbr0 set rate_ 64kb          ;# according to codec used
$cbr0 attach-agent $udp0

set sink [new Agent/LossMonitor]
$ns_ attach-agent $MN $sink
$ns_ connect $udp0 $sink

set fthrput [open throughput4.txt w]
set fpcklost [open packet_loss4.txt w]
set fpktrcv [open packet_receive4.txt w]
set flstpktt [open last_pkt_time4.txt w]
set fexpt [open expected_seq_pkt4.txt w]

proc record {} {
          global fthrput fpcklost sink fpktrcv flstpktt fexpt
          #Get an instance of the simulator
          set ns [Simulator instance]
          #Set the time for polling
          set time 0.02
          #How many bytes have been received by the traffic
            set bw1 [$sink set bytes_]
            set lpkts [$sink set nlost_]
            set pktrcv [$sink set npkts_]
            set lstpktt [$sink set lastPktTime_]
            set expt [$sink set expected_]
            set now [$ns now]
                  puts $fthrput "$now [expr $bw1/$time*8]"
                  puts $fpcklost "$now $lpkts"
                  puts $fpktrcv "$now $pktrcv"
                  puts $flstpktt "$now $lstpktt"
                  puts $fexpt "$now $now $expt"
                  #Reset the bytes_ values on the traffic sinks
                  $sink set bytes_ 0
                  $sink set nlost_ 0
                  $sink set npkts_ 0
                  $sink set lastPktTime_ 0
                  $sink set expected_ 0
          #Re-schedule the record function
          $ns at [expr $now+$time] "record"
  }

$ns_ at 0.0 "record"

$ns_ at 5.0 "$cbr0 start"
$ns_ at 80.0 "$cbr0 stop"
```

```
############
# SCENARIO #
############

$ns_ at 6.0 "$MN setdest 85.0 140 100"               ;# pmsrve:
move MN to PAR at a really high speed


$ns_ at 10.0 "$MN setdest 155.0 140 1"


for {set t 10} {$t < 80} {incr t 10} {
    $ns_ at $t "puts stderr \"completed through $t/80 secs...\""
}

$ns_ at 0.0 "puts stderr \"Simulation started...\""


set opt(stop) 80


$ns_ at [expr $opt(stop) + 0.0001] "puts stderr \"Simulation
finished\""
$ns_ at [expr $opt(stop) + 0.0002] "close $tracefd"
$ns_ at [expr $opt(stop) + 0.0002] "close $namtracefd"
$ns_ at [expr $opt(stop) + 0.0003] "$ns_ halt"

$ns_ run
```