uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Wenjing Zhang

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.Sc. (Systems Science)

GRADE / DEGREE

Systems Science

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Implementation of Micro-cell Mobile MPLS Based Test Networks

TITRE DE LA THÈSE / TITLE OF THESIS

Dimitrios Makrakis

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Ahmed Karmouch

Ali Miri

Gary W. Slater

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES /
DEAN OF THE FACULTY OF GRADUATE AND POSTDOCORAL STUDIES

IMPLEMENTATION OF MICRO-CELL MOBILE MPLS BASED TEST NETWORKS

Wenjing Zhang

The thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements
for the M.Sc. degree in Systems Science

Systems Science Program

University of Ottawa

Your file Votre référence
ISBN: 0-494-14979-5
Our file Notre référence
ISBN: 0-494-14979-5

TABLE OF CONTENTS

ABSTRACT

The increasing presence of ubiquitous mobile computing devices requiring uninterrupted Internet access has prompted a higher need to support efficient and seamless roaming. The introduction of micro mobility is an approach towards a more flexible, customizable and scalable mobility architecture that also reduces signaling load and handover latency. Micro-cell Mobile MPLS (MM-MPLS) is such a protocol. It extends MPLS, a successful routing/forwarding technology of the core of the Internet, to the micro-mobility environment. MM-MPLS reduces the delay and delay jitter of a roaming mobile host.

This thesis describes the implementation of experimental MM-MPLS based network. The developed testbed is composed of seven desktop-PC running on the Linux operation system.

Performance comparisons of MM-MPLS and the hierarchical mobile IP have confirmed the proper operation of the testbed and the superiority of the protocol.

ACKNOWLEDGEMENTS

At the moment of completing my thesis, I dearly extend my best thanks to everyone who helped me, supported me and worked with me during the time of my study at the University of Ottawa – their memory will last forever.

My sincere thanks and deepest gratitude goes to my supervisors, Professor Dimitrios Makrakis and Professor Nicolas D. Georganas. Without their guide, encouragement and help, I could not have finished this thesis.

Also my faithful thanks to my best friend, Dr. Weijie Liu. He always gave me a helpful hand whenever I was in a difficult situation.

My special thanks are expressed to my son, who is eleven years old this summer. But his assistance for me is far beyond his age.

Finally, I would like to express my heart-felt thanks to my parents, my brothers, sisters, and my friend Cindy. I am so thankful to their support and encouragement; without them I could not overcome all difficulties that I met to reach to this point.

I also wish to express my sincere thanks to professor Ahmed Karmouth and professor Ali Miri, who have had the kindness to take an interest in my work and to accept to be in the examining board of my thesis.

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AF	Assured Forwarding
AI	Advertisement Interval
BER	Bit Error Ratio
CBQ	Class-Based Queuing
CBR	Constraint-Based Routing
CLS	Controlled Load Services
CN	Corresponding Node
COA	Care Of Address
CoS	Class of Service
CPU	Central Processing Unit
CR-LDP	Constraint-based Label Distribution Protocol
CSPF	Constrained Shortest Path First
DiffServ	Differentiate Services
DSCP	Differentiate Services Code Point
EF	Expedited Forwarding
ER-LSP	Edge Router –Label Switching Path
FA	Foreign Agent
FDA	Foreign Domain Agent
FEC	Forwarding Equivalent Class
FTN	FEC to NHLFE
GS	Guaranteed Services

GSM	Global System for Mobile communication
GWR	Gateway Router
HA	Home Agent
HAWAII	Handoff-Aware Wireless Access Internet Infrastructure
H-MIP	Hierarchical Mobile Internet Protocol
H-MPLS	Hierarchical Mobile Multi-Protocol Label Switching
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
ILM	Incoming Label Mapping
IntServ	Integrated Services
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISP	Internet Service Provider
LAN	Local Area Network
LDP	Label Distributed Path
LER	Label Edge Router
LSP	Label Switching Path
LSR	Label Switching Router
MA	Mobility Agent
MIP	Mobile Internet Protocol
MM-MPLS	Micro-cell Mobile Multi-Protocol Label Switching
MN	Mobile Node
MPLS	Multi-Protocol Label Switching

MPLS-TE	Multi-Protocol Label Switching Traffic Engineering
NHLFE	Next Hop Label Forwarding Entry
NIC	Network Interface Card
OSPF	Open Shortest Path First
PHB	Per-Hop Behavior
PSB	PATH State Block
QoS	Quality of Service
RAM	Random Access Memory
RFC	Request for Comments
RSB	RESV State Block
RSVP	Resource Reservation Protocol
RSVP-TE	Resource Reservation Protocol – Traffic Engineering
SFA	Switching Foreign Agent
SLA	Service Level Agreement
TCP	Transmission Control Protocol
TCSB	Traffic Control State Block
TE	Taffic Engineering
TOS	Type of Service
TTL	Time-To-Live
UDP	User Datagram Protocol
VoIP	Voice over IP
VPN	Virtual Private Network
WFQ	Weighted Fair Queuing
WIPPOA	Wireless IP Point Of Attachment

Chapter 1 Introduction

1.1 Motivation

The increasing demand of ubiquitous mobile computing devices requiring uninterrupted Internet access has prompted a higher need to support efficient and seamless roaming.

Traditionally, the Internet employs Internet Protocol version 4 (IPv4) for packet delivery in fixed networks. In order to overcome the need for mobility support in IP networks, the Mobile IP (MIP) was designed. MIP provides transparent mobility to higher layers. Unfortunately, this protocol suffers from many weaknesses, especially when the point of attachment of the mobile device changes frequently [28]. Thus, the mobility problem is divided into two sub problems: macro-mobility and micro-mobility [11]. The distinction between them depends on the range of the stations movements[1]. Micro-mobility covers the management of users' movements within a local scope, inside a given wireless network. Many solutions have been proposed to manage this type of mobility within IP networks, such as Hierarchical Mobile IP [58], Fast Handoff [57], Cellular IP [59] and HAWAII [60]. The extensions of Mobile IP for micro-mobility management and the interaction between different foreign agents[2] (FAs) in the network are the basic mechanism of the mobility management for these proposals.

Multi Protocol Label Switching (MPLS) is a technology that substitutes conventional packet forwarding within a network, with a faster operation of label lookup and switching. It is a protocol between the OSI layer 2 and 3 [6]. Some of the new services that can be deployed with MPLS are traffic engineering, Class of service based forwarding, and Virtual Private

[1] The properties of Mobile IP allow its utilization as a macro-mobility management protocol.
[2] In Mobile Internet Protocol (Mobile IP), a foreign agent is a router serving as a mobility agent for a mobile node. As specified in IETF RFC 2002, a foreign agent works in conjunction with another type of mobility agent, known as a home agent, to support Internet traffic forwarding for a device connecting to the Internet from any location other than its home network.

Networks (VPNs). MPLS defines signaling mechanisms to support both Class of Service (CoS) and Quality of Service (QoS). It provides the means to relate this to the IP DiffeServ markings of the originating IP traffic. MPLS is emerging as the technology of choice to facilitate traffic engineering and internetworking.

In recent years, there has been an interest to extend the MPLS capability to the wireless access networks. Recently there is a growing interest in applying MPLS in IP mobility management. An architecture integrating MPLS and Mobile IP thus obviating the needs for IP-in-IP tunneling is presented [3]. It proposes a MPLS based mobility management scheme, which improves the system efficiency and facilities QoS provisioning in the wireless IP access network.

Micro cell Mobile MPLS (MM-MPLS) is a protocol that enables the extension of MPLS to the micro mobility environment, thus allowing the mobile user to benefit from the fast switching, small state maintenance, and high scalability features of MPLS. As was shown in [1] [2], as well as will be demonstrated in this thesis, MM-MPLS makes possible to reduce the traffic delay and delay jitter experienced by a roaming mobile host.

1.2 Methodology

1.2.1 QoS over Internet

As real-time multimedia applications and wireless communications become popular, there is a growing demand on supporting QoS on Internet. To address this issue, the Internet Engineering Task Force (IETF) developed several standards for QoS support in Internet. The most well known and extendible works are the Multi-Protocol Label Switching (MPLS) [6], Integrated Services (IntServ) [48], and Differentiated Services (DiffServ) [49]. All of them are defined for the wired Internet. However, the number of mobile users grows faster than the number of Internet users. Such a trend generates the need for defining and standardizing

QoS supporting technologies for wireless networks. In the following sections, QoS mechanisms proposed for the Internet and the wireless network in the literature will be reviewed.

1.2.1.1 Multi-Protocol Label Switching and Traffic Engineering

Multi-Protocol Label Switching (MPLS) uses a label, which is a short fixed-length value carried in the packet's header to identify a Forwarding Equivalence Class (FEC), to handle the packets. An FEC identifies a set of packet streams that can be forwarded over the same path through a network. When a packet enters an MPLS-enabled network domain, it obtains an added MPLS header, which is encapsulated between the link layer's header and the network layer's header. An MPLS capable router is called Label Switching Router (LSR). Such a router decides where a certain packet should be forwarded by using only the input/output port and label. An MPLS node may also be capable of forwarding native L3 packets [6].

There are three types of router in a MPLS domain, i.e., ingress router, core router and egress router. All the three kinds of routers are involved in the LSP establishment process. Their function is to identify the new label and outgoing port of an incoming MPLS packet, using the packet's incoming label and incoming port, replaces the incoming label with the outgoing label and sends the packet out using the identified outgoing port. Ingress router and egress router are label edge routers (LERs). Core routers are the remaining routers within the path. An ingress router accepts IP packets, transports them to the MPLS layer, inserts in each packet the appropriate label and sends it out from the appropriate outgoing port. An egress router performs the opposite function. It receives MPLS packets, strips off the label, transports them to the network (IP) layer and routes them using IP routing.

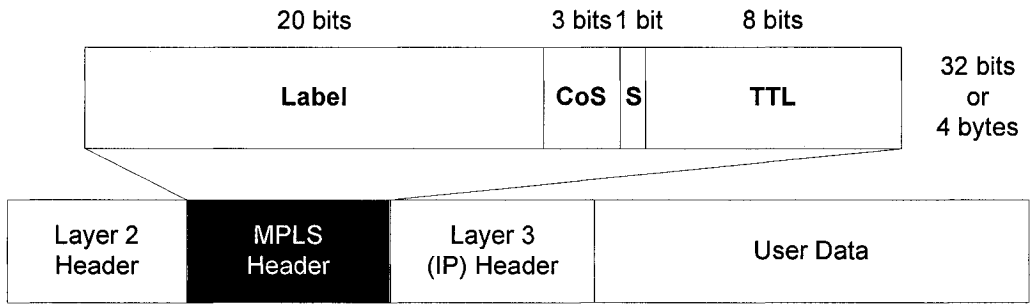

Figure 1.1 MPLS header

As shown in Figure 1.1, the 32-bit MPLS header contains the following fields:

- The label field (20 bits), which carries the value of the MPLS label.
- The CoS field (3 bits) is used for defining the class of service.
- The Stack (S) field (1 bit), which indicates the label stack[3].
- The TTL (time-to-live) field (8 bits) is used to specify conventional IP TTL.

MPLS uses protocols to distribute labels within the domain by establishing label switching paths (LSPs), which are the packet routing paths between the ingress and egress LSRs. To set up an LSP, MPLS uses a variety of path discovering, signaling, resource reservation and label distribution protocols, such as Open Shortest Path First (OSPF) [65], Resource reservation protocol (RSVP) [66] or Label Distribution Protocol (LDP) [67]. Each MPLS-enabled router has a routing table for the labels, which is managed by the LDP. When an LSR receives a labeled packet, it will use the label and input port as an index to look up the forwarding table. The packet is then processed according to the label table entry. Each packet gets a MPLS label at the entrance of a MPLS domain, which is used by the internal routers for routing and for traffic control. Before a packet leaves the MPLS domain, the egress router removes its MPLS label.

Detailed description of the process performed at each type of MPLS enabled node is described below.

[3] A label stack is an ordered set of labels. Usually, a labeled packet carries a number of labels, organized as last-in, first-out stack. An unlabeled packet can be thought of as a packet whose label stack is empty (i.e., whose label stack has depth 0). If a packet's label stack is of depth *m*, we refer to the label at the bottom of the stack as the level 1 label, to the label above it (if such exists) as the level 2 label, and to the label at the top of the stack as the level *m* label.

(1) **Process at ingress LER**

The label-swapping forwarding algorithm requires packet classification at the ingress edge of the network, where the initial label is assigned to each packet. The label switch performs a longest-match routing table lookup and maps the packet to an FEC. The ingress label switch then assigns a label to the packet and forwards it to the next hop of the label-switching path .

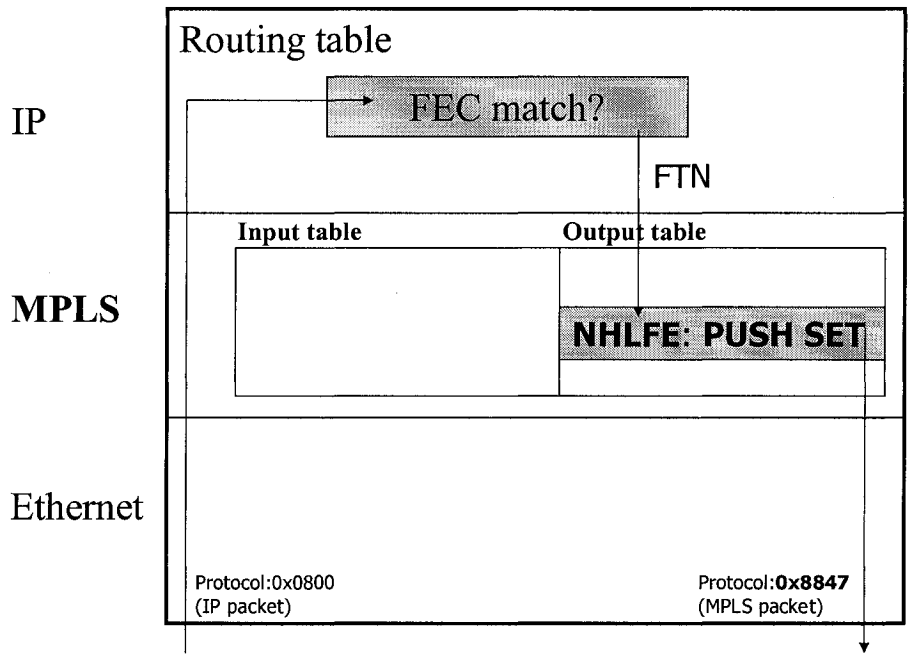

Figure 1.2 Process performed by the ingress LER

As shown on figure 1.2, the process at ingress router includes:

- The Ethernet layer of the LER receives a frame with a protocol field in Ethernet header set to 0x0800, which is the protocol code for IPv4
- The Ethernet layer passes the incoming frame to the IP layer
- The MPLS router searches for an entry in the IP routing table to make the routing decision. But since this entry matches a FEC it has been modified so that the packet is passed to the MPLS layer instead of being routed by the IP layer
- The additional information contained in the IP routing table is a FEC to Next hop label forwarding entry (FTN), that is, a point to an MPLS label entry. This output

label entry is a Next Hop Label Forwarding Entry (NHLFE) that contains two instructions (**PUSH**: defines the label number of the packet; **SET**: defines on which interface the packet should be sent to)

- The MPLS layer adds at the beginning of the packet an MPLS header which contains the label found in the NHLFE, and passes the packet to the Ethernet layer
- The Ethernet layer generates a frame with the protocol field set to the code assigned to MPLS unicast packet (0x8847) and sends the frame over the wire

(2) **Process at the core LSR**

In the core of the network, label switches ignore the packet's network layer header and simply forward the packet using the label-swapping algorithm. When a labeled packet arrives to such a switch, the forwarding component uses the input port number and label to perform an extract match search of its forwarding table. When a match is found, the forwarding component retrieves the outgoing label, the outgoing interface, and the next-hop address from the forwarding table. The forwarding component then swaps (or replaces) the incoming label with the outgoing label and directs the packet to the outbound interface for transmission to the next hop in the LSP.

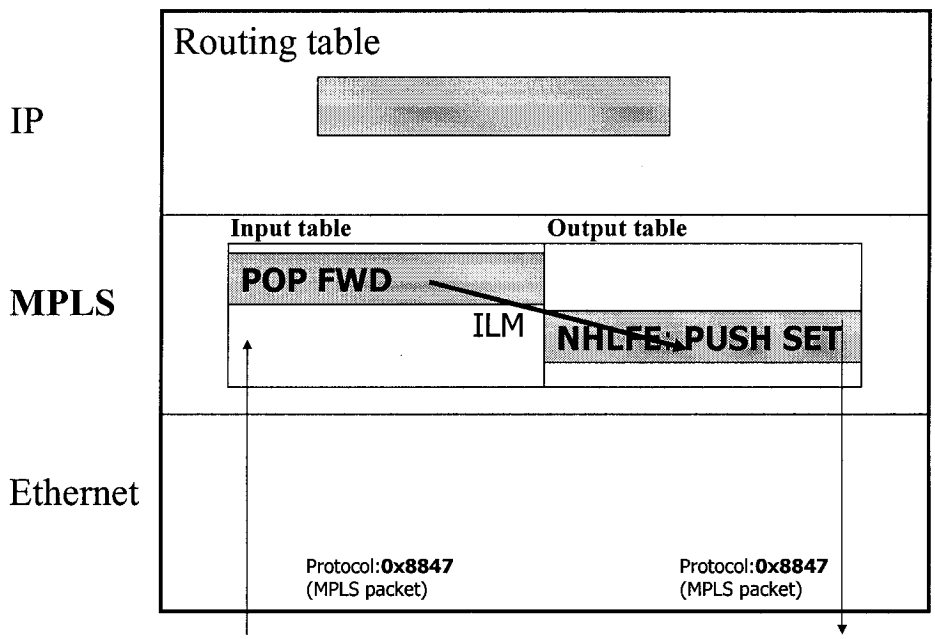

Figure 1.3 Process performed by the LSR

As shown in figure 1.3, the process at the core router includes:

- The Ethernet layer of the LSR receives a frame with a protocol field in the Ethernet header set to 0x8847. Since 0x8847 is the code assigned to MPLS unicast packets encapsulated in the Ethernet frames, the Ethernet layer passes the frame to the MPLS layer of the LSR
- The MPLS layer searches in the MPLS input table for the entry that matches the label embedded in the shim header of the packet. The input table implements the ILM (Incoming Label Mapping) and tells the MPLS layer what to do with the packet. The input table entry contains two instructions. (**POP**: tells the LSR to remove the MPLS header; **FWD**: points to an entry of the MPLS output table.)
- The entry in the MPLS output table in turn contains two instructions (**PUSH**: contains the new label for the packet and tells the LSR to add a shim header on the packet with this new label; **SET**: tells the LSR on which Ethernet interface the packet should be sent)
- The Ethernet layer then builds a frame with a protocol field of 0x8847 and send it over the wire

Note: In MPLS_LINUX the SWAP operation is implemented by successively popping and pushing a shim header. The instructions required to pop and push a label are located in each of the MPLS tables. In this case, the NHLFE is contained at the same time in the input table and output table.

(3) Process on egress LSR

When the labeled packet arrives at the egress label switch, the forwarding component searches its forwarding table. If the next hop is not a label switch, the egress switch discards the label and forwards the packet using conventional longest-match IP forwarding.

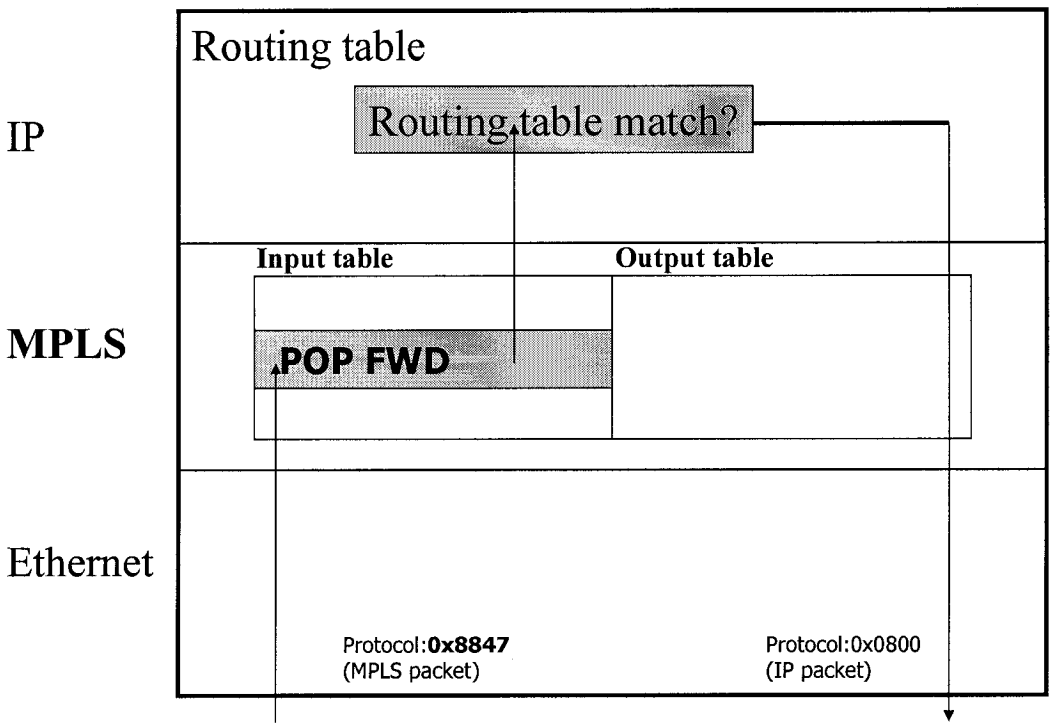

Figure 1.4 Process performed by the egress LER

As shown in figure 1.4, the process at egress router includes:

- The Ethernet layer of the LER receives a frame with a protocol field in the Ethernet header, set to 0x8847, and therefore passes the frame to the MPLS layer.
- The MPLS input table entry that matches the label of the packet contains two instructions (**POP**: tells the LER to remove the shim header from the packet; **DLV**: tells the LER to pass the packet to the IP layer where it will be processed like any other IP packet.

Note: In this case, the NHLFE is fully contained in the input table entry and tells the packet to pop the shim header.

Table 1.1 gives a list of the description of MPLS instructions we mentioned in the above process.

Field	Description
POP	Remove the top label from the label stack. (in/out)
PEEK	Look at the label on top of the label stack, look the label up in the list of incoming label for this interface, and start executing the instructions associated with it. If there is not a label, execute a DLV. (in)
PUSH	Push another label on the label stack. (in/out)
DLV	Send this packet to the layer 3 protocol stored with this in label. (in)
FWD	Send a packet to an outgoing label structure to be processed. (in/out)

Table 1.1 Description of MPLS instructions [75]

Traffic Engineering (TE) [68] is defined by the IETF as the aspect of network engineering dealing with the issue of performance evaluation and performance optimization of IP networks. With MPLS, traffic engineering will be applied easier and more efficiently allowing optimization of the IP traffic routing strategy. MPLS traffic engineering (MPLS-TE) [69] employs "constraint-based routing", in which the path for a traffic flow is the shortest path that meets the resource requirements of the traffic flow, as well the bandwidth requirements, media requirements, a priority versus other flows, and so on.

Overall, main advantages of MPLS are:

- Fast forwarding;
- Efficient tunneling of packets;
- Easy application of Traffic Engineering.

1.2.1.2 Integrated Services

Integrated Services architecture, in short called IntServ, is defined by IETF in RFC 1633 [48]. The main idea behind this proposal is support of real-time services in the Internet. Integrated Services introduces a fundamentally new concept for the Internet. This protocol assumes that resources are reserved for every flow requiring QoS at every router hop in the path between the sender and the receiver. To be able to support per-flow traffic management, the network needs to establish an end-to-end path by using signaling, which is provided by RSVP. This is in contrast to the traditional approach used in Internet in the past, where intermediate routers

do not store routing information for each flow. Besides the best-effort traffic class, Integrated Services provides two additional QoS classes:

1. ***Guaranteed services*** **(GS)** [51] for applications requiring bounded end-to-end queuing delay of packets and bandwidth guarantees. The delay has two parts: fixed delay and queuing delay. Fixed delay is a property of the chosen path by the setup scheme. Hence, only the queuing delay is determined by the guaranteed service. In this concept a flow is described using a token bucket and given this description of the flow, a service element (e.g., a router) computes various parameters describing how the service element will handle the flow's data. However, a setup mechanism, e.g. Resource reservation Protocol (RSVP) must be used for guaranteed reservations. To achieve bounded delay requires that every node in the path supports the guaranteed service, although one may benefit also with its partial deployment.

2. ***Controlled load services*** **(CLS)** [52] for applications requiring reliable and enhanced best effort service. This service uses admission control to assure that the expected performance is received even when the network is overloaded. In other words, the controlled load does not accept or provide specific target values for delay and loss, but it provides a commitment by the network element to provide service equivalent to that provided by uncontrolled (best-effort) traffic under lightly loaded conditions. A possible implementation of this service is by using a two-priority queuing mechanism. A high priority for controlled load traffic and a lower priority for best-effort traffic.

To be able to provide such QoS classes, network nodes must maintain a per-flow soft state (i.e., flow-specific state). A soft state is a temporary state governed by the periodic expiration of resource reservations. Soft states are refreshed by periodical RSVP messages called PATH messages. Usually, a PATH message is sent every 30 seconds [66] in order to maintain the reservation. It is routed through the Internet as an ordinary IP packet. PATH messages contain the traffic characteristics of the source. After reception of the PATH message, the receiver sends a so called RESV message back to the sender. When this packet passes through the intermediate routers on the path between the sender

and the receiver, it performs reservation of resources. Each router may accept or reject such a reservation request (if some router rejects the reservation request, it sends a notification packet to the source). If all intermediate routers accept the reservation request, then each of them allocates resources, i.e. link bandwidth and buffer space at the router, for the flow.

1.2.1.3 Differentiated Services

The Differentiated Services architecture [49] is proposed as a solution to the scalability problem of Integrated Services. The Differentiated Services architecture reduces the state of information stored in the network by providing QoS to limited number of classes. DiffServ is based on class identification by using the DS header field, which replaced the definition of IPv4 Type of Service (TOS) octet and IPv6 traffic class octet [55]. Figure 1.5 shows the IP header with DS field included. In the DS field, 6 bits out of 8 bits are used as a DS code point (DSCP) and specify the QoS requirements, while 2 remaining bits are currently unused.

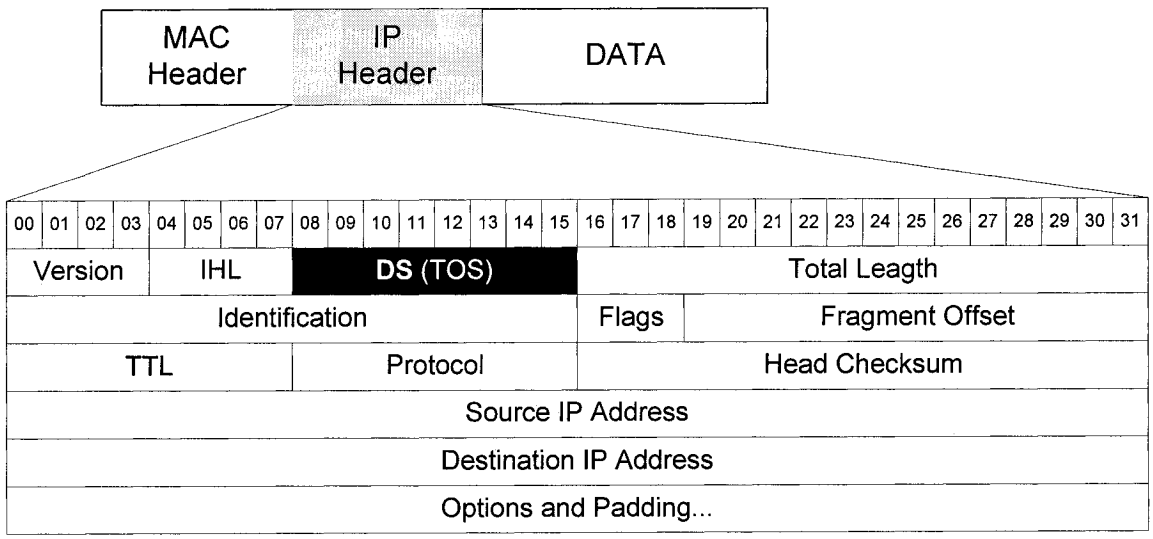

Figure 1.5 Differentiated Services field in IP header

DSCP is used to differentiate aggregate flows from different traffic classes. It is incompatible with IPv4 TOS, where the first 3 bits are used to specify the precedence, and the next 4 bits

are used to specify the requirements on delay, throughput, reliability and cost. The presumption is that DS domains protect themselves by deploying demarking boundary nodes.

The basic principle of DS is the use of class-based packet-forwarding treatment, which is defined by the per-hop behavior (PHB). Basic service in DS is the best-effort service (all DSCP bits are zero). By marking the DiffServ field differently and handling packets based on their DiffServ fields (e.g., by traffic conditioners), we may create several differentiated service classes. Therefore, one may refer to DiffServ as a relative priority scheme.

The network under control of one ISP is usually called a domain. With the aim to provide DS, edge routers of the DS domain should classify, police, and shape the traffic entering the network domain. When a certain packet enters one domain from another, its DS field may be re-marked according to the Service Level Agreement (SLA) between the two domains. A classifier selects the packet based on the DSCP value in the packet header. Using the QoS mechanisms, such as classification, policing, shaping and scheduling different service classes can be provided.

DiffServ conceptually differs from IntServ. The number of classes is limited within DiffServ due to the limited size of the DiffServ (or TOS) field in the IP headers. Furthermore, DiffServ does not have scalability problems as IntServ does. The amount of information stored at a network node is proportional to the number of classes rather than to the number of flows. Another advantage of DiffServ is in that classification, policing, shaping, and admission control are performed only at the boundary routers of an Internet Service Provider (ISP) domain. This way, core routers easily perform fast forwarding of packets.

So far, IETF has proposed two PHB classes: expedited forwarding (EF) and assured forwarding (AF).

1.2.1.3.1 Assured Forwarding Service

The Assured Forwarding (AF) service is to be used by non real-time, throughput sensitive applications. It provides protection and some form of throughput guarantees. AF is subdivided into four different subclasses. Each DS node allocates a certain amount of resources (e.g., buffer space and, in many cases, bandwidth) for each AF class.

Classification and policing are performed at the ingress routers of the ISP network. All packets that do not exceed the negotiated QoS profile are considered as in-profile, while the excess packets are considered as out-of-profile. In most developed schemes, in a case of network congestion, out-of-profile packets are discarded first.

An AF mechanism must detect and respond to long term congestion. Short bursts may be handled by buffering the packets. But long-term congestion should be dealt with by dropping packets. One way to perform such queue management is random packet dropping.

Each AF subclass provides three different subclasses. Thus provides generating 4×3=12 service differentiation policies. Each node in a DS domain should have separate queues for each AF traffic class. Network nodes with DS capability perform class differentiation by matching the DSCP field to a particular packet handling mechanism. Packets received with an unrecognized code point are forwarded as if they were marked for the default behavior (e.g. best effort service).

class	subclass	class name	DSCP
class 1	low	AF11	001010 (10)
	medium	AF12	001100 (12)
	high	AF13	001110 (14)
class 2	low	AF21	010010 (18)
	medium	AF22	010100 (20)
	high	AF23	010110 (22)
class 3	low	AF31	011010 (26)
	medium	AF32	011100 (28)
	high	AF33	011110 (30)
class 4	low	AF41	100010 (34)
	medium	AF42	100100 (36)
	high	AF43	100110 (38)

Table 1.2 AF classes with DiffServ

1.2.1.3.2 Expedited Forwarding Services

Expedited Forwarding (EF) service is used by applications that have stringent requirements on packet delay, jitter, as well as assured bandwidth, such as Internet telephony, video conferencing, etc. The delay and delay jitter occur due to the variability of queuing delay packets experience at the network nodes. Increase in the traffic queue occurs when the departure rate is close or slower than the arrival. A popular method of processing EF packets is by applying priority queuing, with the EF class having the highest priority [37]. Also, in order to maintain short delays, the buffers used to store EF packet are of small size. Due to the "privileged" treatment EF packets receive, it is important that EF traffic streams are tightly policed at the network boundaries; else, unexpected surges of EF traffic due to inefficient policing could force the lower classes to starvation, resulting in denial of service [49].

1.2.1.4 QoS in Wireless Networks

Wireless networks differ from wired networks in terms of access technology and in the characteristics of the transmission medium. User mobility and cellular topology are the reasons that handovers are necessary. Also, a mobile node frequently changes its location within a single cell, thus resulting in time-varying bit error ratio (BER) and interference, which affects the QoS for that connection.

Handover schemes experience the so-called handover latency. This is the duration of the mobile node (being in the process of moving from one cell to another) unable to send or receive IP packets. In certain scenarios, the handover latency resulting from Mobile IP handover procedures may be greater than that is acceptable for real-time services. Also, handovers may cause packet losses. Such losses may disrupt both real-time and nonreal-time services, and hence are undesirable.

The growth of the wireless Internet is similar to that of cellular mobile networks. The idea for Mobile Internet is already widely accepted by the Internet service providers and cellular operators. In order to design a cost-effective wireless IP network, however, we need to create many small network domains that should be interconnected as well as connected to the commercial cellular networks, e.g. Global System for Mobile communication (GSM).

As was mentioned earlier, IETF has defined the Mobile IP protocol, which is the standard for macro-mobility in the Internet. There is also a significant research effort towards QoS support and efficient micro-mobility management in mobile IP networks. So far, we have several proposals on mechanisms for QoS provisioning in the Internet, such as: MPLS, Integrated Services, and Differentiated Services. These QoS mechanisms were initially created for wired IP networks. After minor modifications, however, we may also apply them in wireless access networks. The Differentiated Services scheme is foreseen as the most suitable QoS mechanism for the wireless access networks because of its limited processing and space requirements at the network nodes.

The introduction of mobility to the Internet requires the creation of mechanisms that can deal with handovers, location management, and location-dependent bit errors. Over the past several years a number of IP micromobility protocols have been proposed, such as Cellular IP, HAWAII, and multicast-based intra-handover management. Most of them are created for Mobile IPv4, but they may be applied to Mobile IPv6 networks.

1.2.2 Resource Reservation Protocol – Traffic Engineering

The Resource Reservation Protocol – Traffic Engineering (RSVP-TE) [5] protocol is an extension to the Resource reservation Protocol (RSVP) [66]. RSVP is a signaling protocol developed initially to implement the signaling process that supports the resource reservation for IntServ. RSVP-TE extends two particular functions for TE purpose in MPLS network. One is the possibility to setup LSPs and Edge Router –Label Switching Paths (ER_LSPs), the other is the function for traffic engineering.

The six main RSVP signaling messages are: PATH, RESV, PATHERR, RESVERR, PATHTEAR, and RESVTEAR. Table 1.3 shows the functions of each message. RSVP allows the use of source routing, where the ingress router determines the complete path through the network. The ingress router can use a Constrained Shortest Path First (CSPF) [65] approach to determine a path to the destination, ensuring that any QoS requirements are met. The resulting path is then used to establish the LSP.

Message	Functions
PATH	PATH message is send from the Sender along the downstream direction to the receiver. It is a soft state message.
RESV	RESV message is send from the Receiver along the upstream direction to the Sender. It is a soft state message.
PATH ERROR	PATH ERROR message is used to report the error in installing the "path state". It is send from the error node to the Sender along the reverse path of PATH message.
RESV ERROR	RESV ERROR message is used to report the error in installing the "reservation state". It is send from the error node to the Receiver along the reverse path of RESV message.
PATH TEAR	PATH TEAR message is used to remove "path states". It is send from the Sender to the Receiver along the nodes with the "path state" installed.
RESV TEAR	RESV TEAR message is used to remove "reservation states". It is send from the Receiver to the Sender along the nodes with the "reservation state" installed.

Table 1.3 RSVP messages

With the RSVP mechanism, the upper LSR sends PATH messages to the downstream nodes in order to create or refresh local path states. In reverse, the lower LSR sends RESV messages to upstream nodes in response to PATH messages, thus creating or refreshing local RESV states. A session is ended if the state machine is not refreshed within the RSVP tunnel timeout period, which is determined as follows:

$$\textbf{RSVP tunnel timeout period in seconds} = \left\{\left[\left(\textit{cleanup timeout factor} + 0.5\right)\times 1.5\right]\times\left(\frac{\textit{refresh period}}{1000}\right)\right\}$$ [4]

For example, for the default values,

[4] Clearnup timeout factor is used to specify the number of refresh messages that can be lost before the PATH or RESV state is ended, together with the refresh period. It defines the RSVP tunnel timeout period.

$$\text{RSVP tunnel timeout period in seconds} = \left\{\left[(3+0.5)\times 1.5\right]\times\left(\frac{30{,}000}{1000}\right)\right\} = 157.5$$

An upstream LSR sends a PATHTEAR message when its path state times out as a result of not being refreshed. The PATHTEAR message removes the PATH and RESV states in each LSR as it proceeds in the downstream. Downstream LSRs similarly send the RESVTEAR message when their RESV state times out, in orderto instruct the upstream LSRs to remove the corresponding RESV states.

If a downstream LSR determines that it received an erroneous path message, it sends a PATHERR message to the sender. If a reservation (label) request fails, the request initiator sends a REESVERR message to the downstream LSRs.

RSVP-TE is used to establish MPLS LSPs when there are traffic engineering requirements. It is mainly used to provide QoS and load balancing across the network core. RSVP-TE is a *soft-state* protocol, meaning that much of the session information is embedded in a state machine on each LSR. The state machine must be refreshed periodically to avoid session termination.

1.2.3 Mobility in IP Networks

Broadband wireless networks are quickly evolving towards all-IP networks. Mobile IP is probably the most widely known mobility management proposal. A mobile device will change its network point of attachment each time it moves to a new IP subnet. Mobile IP is the current standard protocol developed by the Mobile IP Internet Engineering Task Force (IETF) working group, which is able to inform the network about such a change in network attachment, so that the packets can be delivered seamlessly to the new point of attachment [20]. Mobile IP is intended to service mobile nodes moving from one IP subnet to another. Mobility now can be classified into two main categories: Macro-Mobility and Micro-Mobility. In the following section, mobile IP and its extended micro-mobility protocols, such as Hierarchical mobile IP, Cellular IP and HAWAII, are introduced.

1.2.3.1 Mobile IP

Macro-Mobility is the management of IP nodes at a global scale. We think of Mobile IP as solving the "macro" mobility management problem, which is the management of the movements of a mobile node between distant wireless domains and across the Internet (macro-mobility).

Three functions are defined for Mobile IPv4:

- Agent Discovery
- Registration
- Routing Consideration (data forwarding)

The agent discovery procedure used in Mobile IP is based on the ICMP router advertisements. The router advertisements are extended and also contain the required care-of-address (COA). These extended router advertisements are known as agent advertisements. Home agents (HA) and foreign agents (FA) typically broadcast at regular intervals and in random fashion. A mobile node can also send a solicitation on the link to learn if any prospective agents are present.

Depending on its method of attachment, the MN will register either directly with its home agent, or through a foreign agent, which forwards the registration to the home agent. After a mobile node gets the care-of-address, it will inform the home agent about it. In Mobile IP, this can be accomplished by using the registration procedure. The mobile node sends a registration request with the care-of-address information. This information is received by the home agent. As soon as the request is approved, the home agent will add the necessary information to its routing table and send a registration reply back to the mobile node.

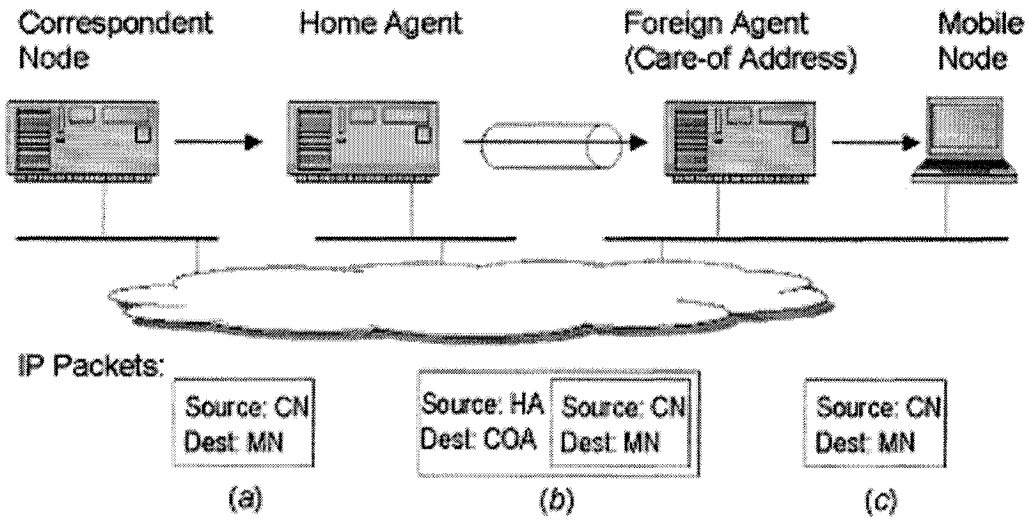

Figure 1.6 Describes the basic working flow of Mobile IPv4 [41]

Figure 1.6 shows a basic working flow of Mobile IPv4. Data forwarding is accomplished by using encapsulation mechanisms. All mobility agents (MAs), i.e. HAs and FAs, which are using mobile IPv4, must be able to use a default encapsulation mechanism included in the IP-in-IP protocol. By using this protocol, the HA inserts an IP tunnel header in front of the header of each original IP packet addressed to the mobile node's home address. The destination of this tunnel is the mobile node's care-of-address. In IP-in-IP protocol, there is a way to indicate that the next protocol header is again an IP header. We may therefore forward the packet hop-by-hop according to the routing table.

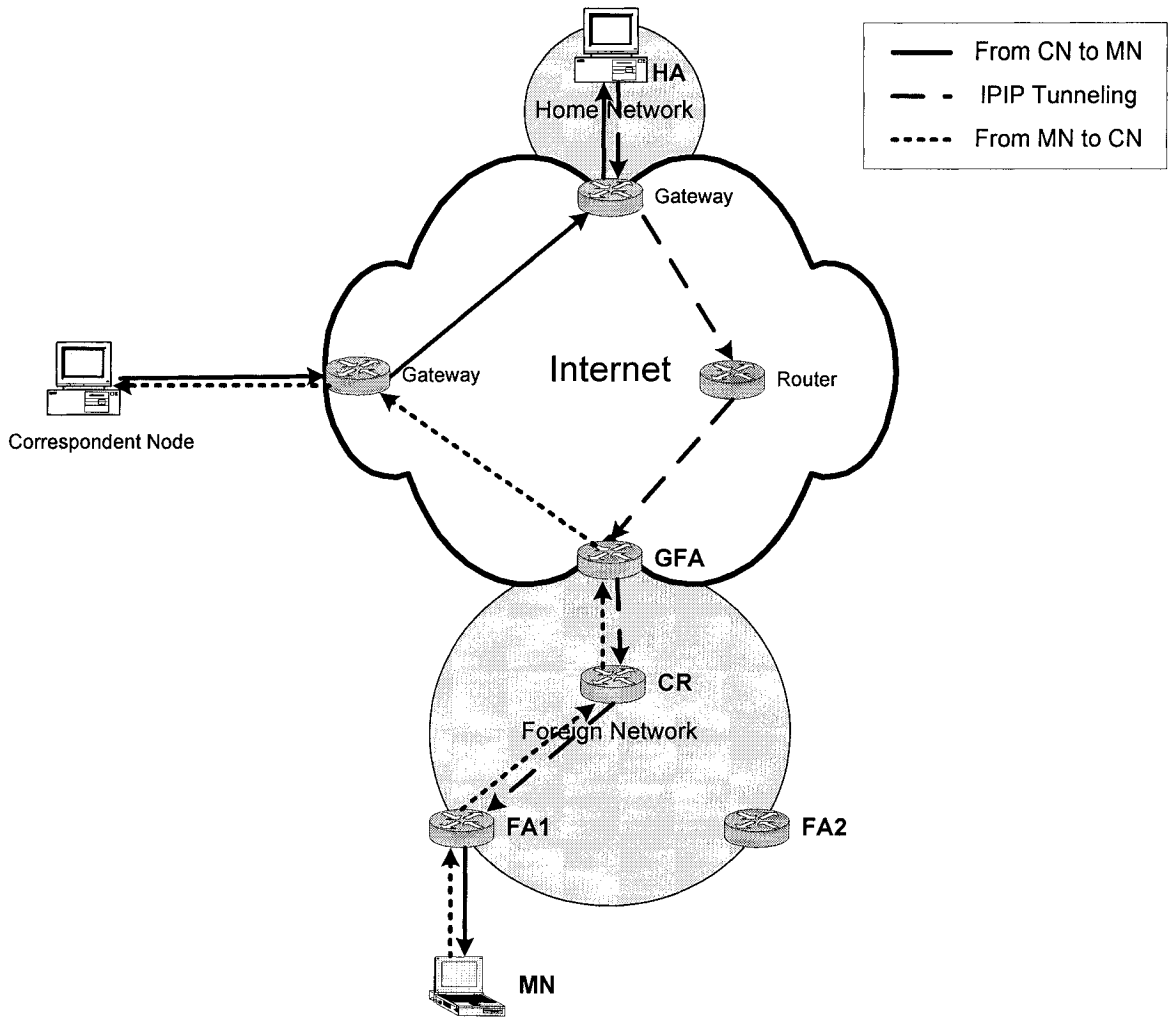

Figure 1.7 Basic traffic flow when using Mobile IP

In Mobile IP, the basic mobility management procedure is composed of two parts: the movement detection by the MN and the registration to the HA. Every time the mobile changes its Wireless IP Point of Attachment (WIPPOA), these two steps must be accomplished to allow the MN to continue to receive packets. However, it is the MN that initiates the process by sending a registration request once it has detected that it moved from one network to another and has obtained a new COA. This introduces two types of latency: move detection latency and registration latency. The move detection latency is the time required by the MN to detect that it has changed its WIPPOA. It can be large, since the move detection mechanisms in Mobile IP are based on either the expiration of the lifetime of FA agent advertisements or on the comparison of the address prefix of two different agent

advertisements. The registration latency is the required time to complete the registration with the HA. As this HA can be located anywhere in the Internet, this process can take a long time and sometimes be impossible to complete. Those latencies delay the redirection towards the new location of the MN. During this time, the mobile is already connected to its new WIPPOA but the packets that are sent to it are routed to its old WIPPOA.

1.2.3.2 Micro-Mobility Protocols

For a fast moving mobile, which changes network rapidly, the registration process according to Mobile IP becomes inefficient. Moreover, this mechanism produces a lot of control traffic inside the local domain and across the Internet. The micro-mobility approach tries to reduce the latency of the handover management. This approach does not always reduce the control traffic, but it allows reduction of the number of network stations that process the control packets by restricting the propagation of those packets to a smaller part of the network.

The micro-mobility approach integrates two different protocols in order to manage the mobility:

- Mobile IP, in order to manage the movements of the MN *between* distant wireless domains and across the Internet (macro-mobility),
- A micro-mobility protocol, in order to manage the movement of the MN within a wireless domain.

Several solutions have been proposed to solve the problem of micro-mobility in IP networks. Some of the most popular protocols are described below.

1.2.3.2.1 Hierarchical Mobile IP

Hierarchical Mobile IP (H-MIP) [58], proposed by Ericssion and Nokia, is an extension to Mobile IPv4, aiming to efficiently support the micro-mobility approach. In this solution, there are three types of foreign agents. The Gateway Foreign Agent (GFA) is the highest

foreign agent. The Lowest Foreign Agent (LFA) is the agent that can act as the point of attachment for a MN. The other agents located between the GFA and LFA are the Intermediate Foreign Agents (IFA). Figure 1.8 shows the architecture of Hierarchical Mobile IP.

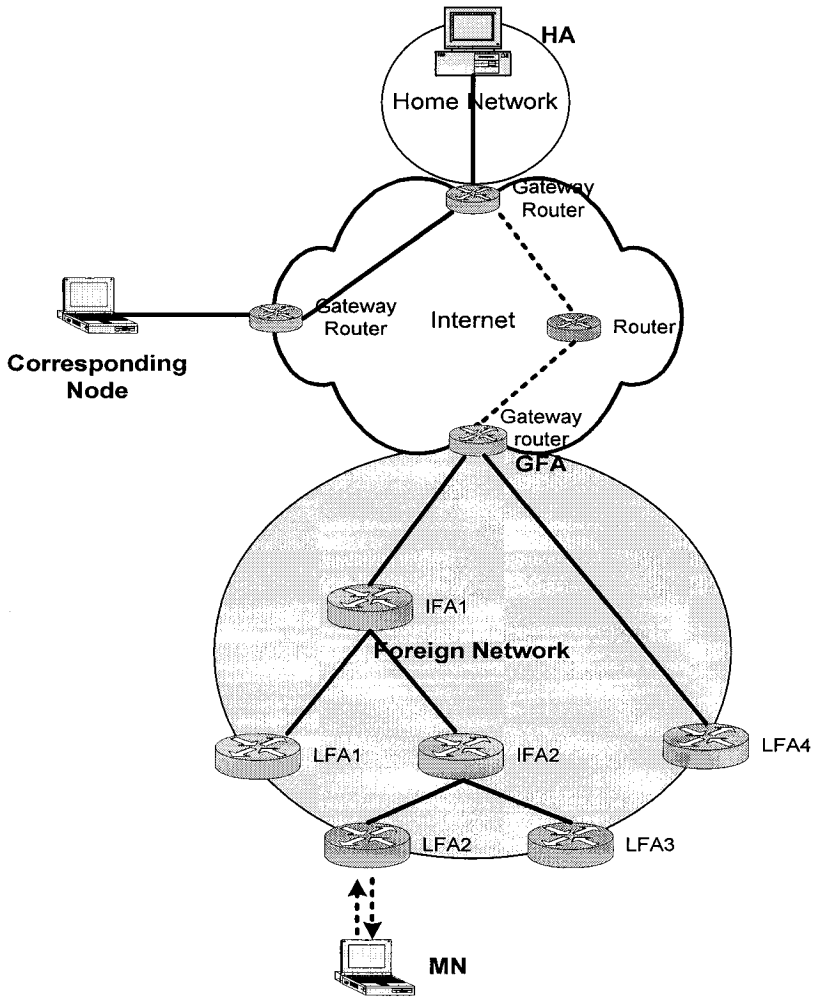

Figure 1.8 Architecture of Hierarchical Mobile IP

The procedure of registration is shown in Figure 1.9. When the MN moves for the first time into a foreign domain, Home Registration is required (Figure 1.9-a). This registration is made with the address of the GFA as COA. This COA won't be changed when the MN changes its FA within the same foreign domain. When the MN moves in a foreign network, each intermediate FA in the path from MN to the HA examines if it already has a binding for the specified MN. This allows them to perform local location updates. A regional registration (Figure 1.9-b) contains a new COA that it is the FA which MN is connected with.

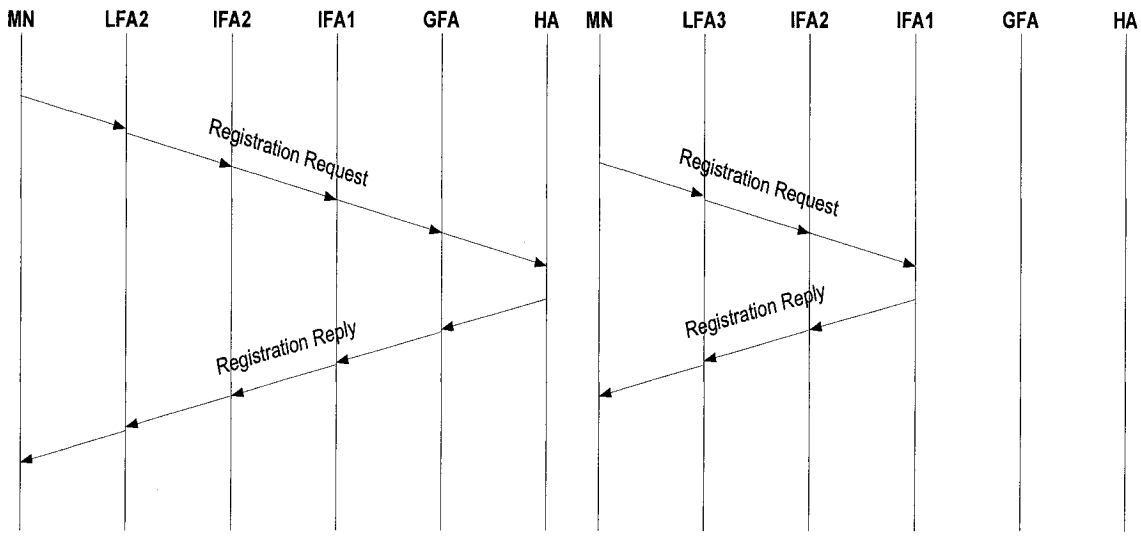

Figure 1.9-a. Home registration in Hierarchical Mobile IP

Figure 1.9-b. Regional registration in Hierarchical Mobile IP

Figure 1.9 Registration message in Hierarchical Mobile IP

1.2.3.2.2 Cellular IP

Cellular IP [72] is a micro-mobility protocol. It can be used to provide local mobility and handoff support [59]. A cellular IP network is made up of cellular nodes connected hierarchically. In a cellular IP network, there is a gateway at the boundary with the Internet backbone and a number of cellular nodes with an interface on a wireless link, known as base stations. Each cellular node always has only one uplink neighbor toward the gateway. The shortest path between a base station and the gateway is given by a chain of cellular nodes. Figure 1.10 shows a Cellular IP network.

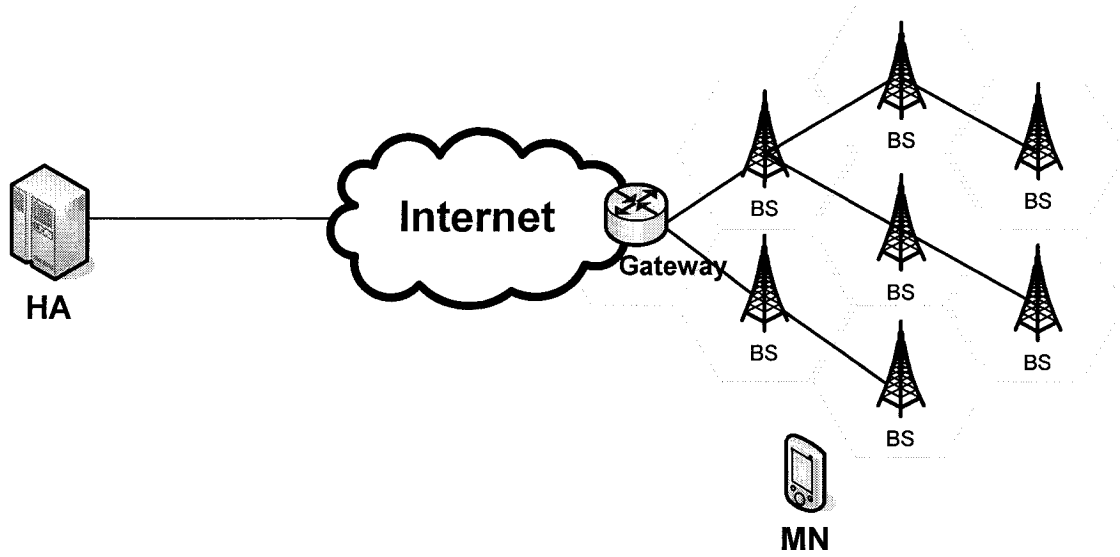

Figure 1.10 Cellular IP network

When a mobile host wants to attach itself to the cellular IP network, it sends a signaling packet to the serving base station. This is then relayed hop-by- hop along the chain of uplink neighbors, until it reaches the gateway. At each hop the packet creates a soft state[5] , which contains an association between the mobile host's home address and the downlink interface from which the message was received. The chain of these states corresponds to a path, along which it is possible to route packets from the gateway to the mobile host. As a consequence, a packet bound for a mobile host in the cellular IP network must first reach the gateway and then follow the complete downlink path toward the mobile host, independently of where it was send from, inside or outside the cellular IP network.

The states along the path must be regularly refreshed. This can be done by either signaling packets sent by the mobile host within the expiration time, or data packets that flow along the path in the uplink direction. State refreshing through data packets enables a considerable reduction of signaling load in the whole network, saving processing capabilities.

In cellular networks, two options for handoffs are available. They are hard handoff and semi-soft handoff. When performing a hard handoff, a mobile host switches to the new base

[5] A soft state is a state that needs regular refreshing by special signaling packets, or else it is cancelled within a definite period

station and sends a signaling packet to it. This is then relayed hop by hop until it reaches the gateway in the same way as before and creates a new path. Packets that are traveling along the old path after the switching instant will be lost. In the case of a semi-soft handoff, the mobile host waits for the new path to be created and continues receiving packets from the old base station before switching its receiver to the new one. The new path may partially overlap with the old one. In this case, the crossover node where the two paths join has a new state added for the new route created by a signaling packet coming from the mobile host; the old state pointing to the old route is left in place. As a result, there will be an interval of time in which the packets will flow through the new and the old path at the same time, to be finally relayed by both the old and new base stations. During this time, the mobile host can switch its receiver to the new base station and experience minimal packet loss. A subsequent signaling message will delete states that point to the old path.

1.2.3.2.3 Handoff-Aware Wireless Access Internet Infrastructure (HAWAII)

HAWAII is a proposal dealing with micro-mobility [60]. It has also been developed with mobile IP in mind and has many common points with Cellular IP. The approach is domain based, where a domain is defined by a hierarchy of gateways and base stations.

The edge gateway, connecting the access network to the Internet core, is called the domain root gateway. Location management is distributed, having many common points with cellular IP. Each gateway has a default route inside the domain, pointing towards the domain root gateway. For every mobile node establishing its path, an entry for its IP address is added and associated with the appropriate interface. Figure 1.11 shows a simple architecture of HAWAII network.

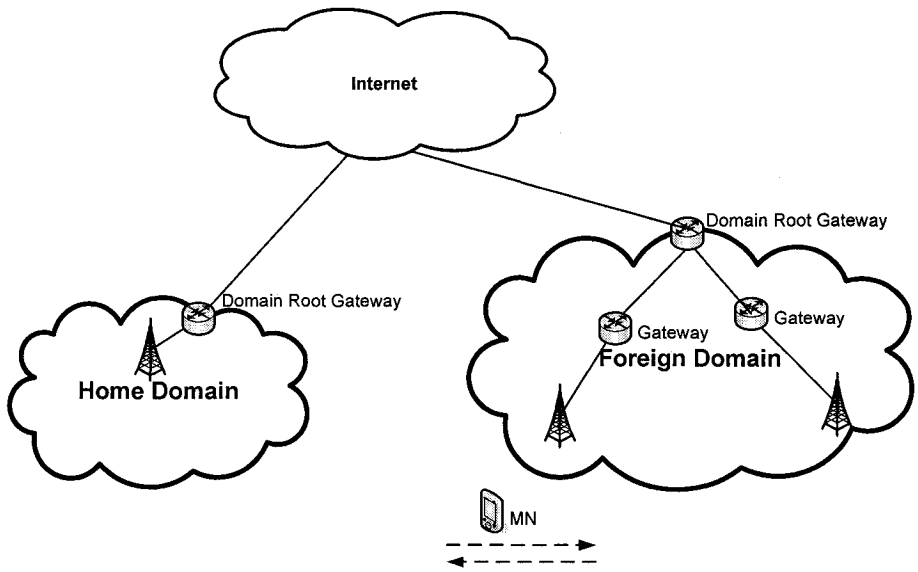

Figure 1.11 Architecture of HAWAII network

HAWAII is transparent to mobile IP. A mobile node moving in a HAWAII-administrated domain will not need to change its COA, and no communication with the home agent is required.

In the handover case, HAWAII defines two different handover mechanisms adapted to different radio access technologies. One is the forwarding scheme and the other is the non-forwarding scheme. The decision to choose which scheme is depended on whether the MN can communicate with more than one base station at the same time or not. Here, a crossover router is introduced, which is the router closest to the mobile host that is at the intersection of two paths, one between the domain root router and the old base station, and the second between the old base station and the new base station. With the forwarding scheme, the packets are forwarded from the old base station to the new, whereas in the non-forwarding scheme, they are diverted at the crossover router.

The signaling between the mobile node and the base station is performed by means of mobile IP signaling. The base station communicates in the fixed part of the HAWAII network with the other gateways and base stations, using some special path setup packets that travel only within the HAWAII network. The consequence is that while the gateway processes only

HAWAII messages, the base station must also implement mobile IP foreign agent functionality.

1.2.4 MPLS-enabled Wireless Networks

In "legacy" networks, data traffic and voice traffic is transferred separately, each with its own switching/routing architecture, network management plate form, and support staff. In the developing next generation wireless network, wireless service providers are attempting to converge the separate architectures into a single network over a common packet core. The recent emergence of Voice of IP (VoIP) technology makes it possible to integrate the voice and video traffic into a single network [71]. QoS support is highly demanded in this type of emerging networks. MPLS is a technology which provides the control mechanism to facilitate traffic engineering and internetworking. In recent years, there has been an interest to extend the MPLS capability to the wireless access networks. Recently there is growing interest in applying MPLS in IP mobility management. An architecture integrating MPLS and Mobile IP thus obviating the needs for IP-in-IP tunneling is presented [3]. It proposes a MPLS based micromobility management scheme, which improves the system efficiency and facilities QoS provisioning in the wireless IP access network. Furthermore, hierarchical mobile MPLS and micro-cell mobile MPLS networks are also introduced in order to improve the performance of mobile MPLS network.

1.2.4.1 Mobile Multi Protocol Label Switching network

Multi Protocol Label Switching (MPLS) is a technology that substitutes conventional packet forwarding within a network or a part of a network, with a faster operation of label lookup and switching. It is a protocol between the OSI layers 2 and 3. Conventional forwarding is based on the network layer information in the packet header, which is analyzed at each hop. In an MPLS based network segment, the network layer header is analyzed only at the entry and exit points (Ingress and Egress) of the segment. As a result of the analysis at the entry point, the packet is assigned to a specific forwarding equivalence class (FEC) and the FEC is

encoded into the packet's extended header as a short fixed-length label. At the subsequent hops within the segment, no further analysis is performed. Instead, the label is used as an index into a lookup table that specifies the next hop and the new label value. The appropriate mapping to a FEC, in order to reach the correct exit point, is either determined through the link state information of the segment or is statically engineered. The path taken by a packet belonging to a FEC, inside the MPLS segment, is referred to as the label switching path (LSP) for that FEC. The internal nodes that perform MPLS switching are called label switching routers (LSRs), while the routers located at the boundaries of the segments are usually referred to as label edge routers (LERs).

It is well known that in Mobile IP the data forwarding process involves IP tunneling from HA to MN. It means that the HA needs to search the IP routing table to tunnel the packet out. The amount of the processing required by HA in this data forwarding process depends on the number of mobile nodes belonging to the home network currently registered. The more the MNs registered, the longer the forwarding process will take.

To integrate mobile IP with MPLS is an efficient way to improve the scalability of the mobile IP data forwarding process since it can take the advantage of MPLS features such as fast switching, small state maintenance and high scalability. In mobile MPLS, the overhead of mobile IP data forwarding is reduced by using a label (4 bytes) to replace the longer IP header (>=20 bytes). Also, IP tunneling is replaced by LSP so that only the petite label rather than the lengthy routing tables is searched.

When a MN receives advertisement message from a FA in a foreign domain, it acquires a temporary COA from the FA and sends back a registration request to the FA. Since the FA is an edge LSR, it will analyze the incoming registration request and obtain a destination address. The FA then updates its routing table by adding a new row containing the home address of the MN. Based on the hop-by-hop IP routing table, the FA forwards the registration request message to the HA. When the HA receives the registration request message and the COA, it first searches its label table to locate the row with the MN's home address as FEC and then sends a label request using LDP to the FA with the COA as FEC.

The FA replies to the HA with a LDP label mapping message. When this message arrives at the HA, an LSP is established. And the row in HA's label table with the MN's home address is subsequently updated. The empty out label and outgoing port entries are also updated using values of out label and outgoing port of the LSP from the HA to the FA. In this way, the HA can relay the packets destined to the MN home address to its current location in the foreign network. After that, the HA sends a registration reply to the FA along the LSP from the HA to the FA. When the FA receives the registration reply, it records the incoming port number and the label value of the reply message, and then adds this value in a new row in the label table. Figure 1.12shows the registration process in mobile MPLS.

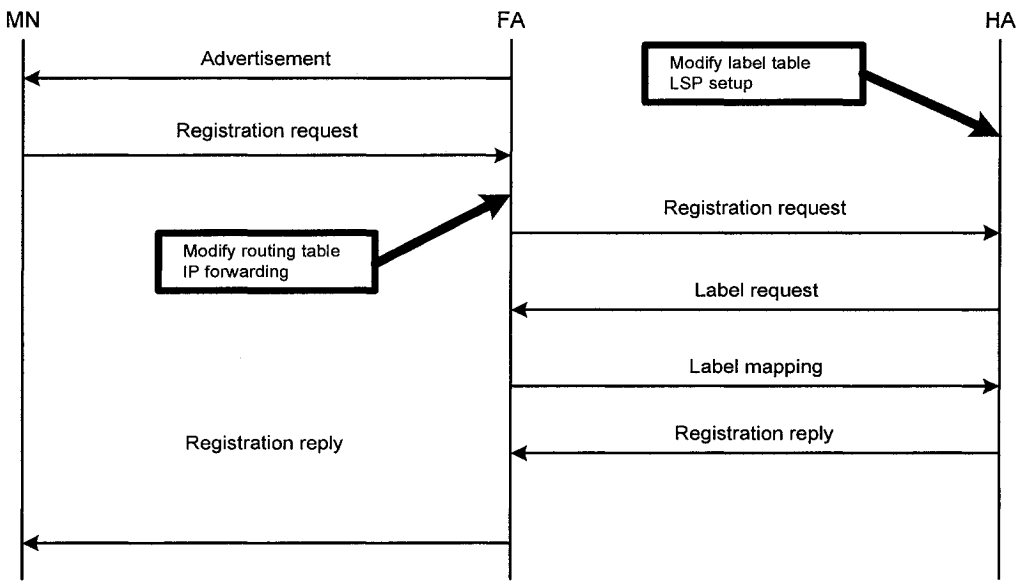

Figure 1.12 Registration Process in Mobile MPLS

Packets sent from CN to MN are addressed by the MN's home address and are intercepted by the HA. The HA uses the incoming label value as an index to lookup its label table, extracts the values indicated as "out" label and outgoing port for this index, replaces the old label with the new (outgoing) label, inserts the label value into the packet, and send it out through the port indicated as outgoing port. If the MN is still in its home network, "out" label and outgoing port are empty. In this case, the packet will be sent to the IP layer and is forwarded to the MN using IP outing. By contrast, if the MN is in a foreign domain, the packet is send from HA to FA along the LSP by label swapping. The FA receives the packet and examines its label table. Since it is the egress of the LSP from HA to FA, the out label and outgoing

port entries are empty. The FA strips off the label and sends the packet to the IP layer. Then, the FA forwards the packet to the MN using IP routing. Finally, the MN receives the packet from the CN. Figure 1.13 illustrates the procedure of datagram delivery followed by mobile MPLS.

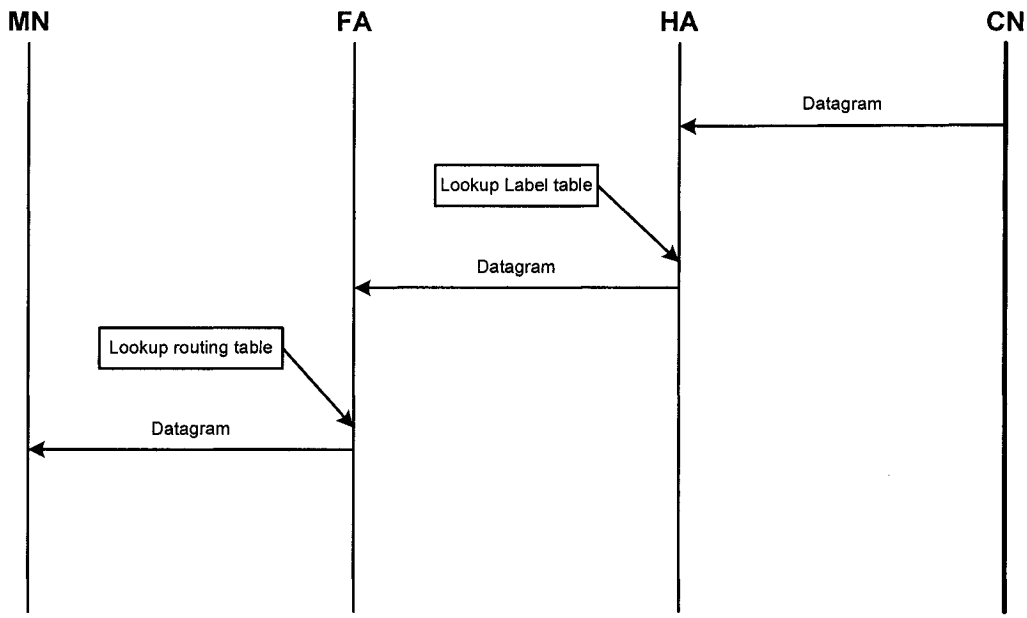

Figure 1.13 Datagram delivery procedures

1.2.4.2 Hierarchical Mobile MPLS

Hierarchical Mobile MPLS is a natural extension of Mobile MPLS in order to efficiently support micro mobility. FDA is a gateway foreign agent, which connects the foreign domain with Internetwork. After the first connection of a MN to a domain and its home registration with the address of the FDA as COA, the MN will perform regional registrations only. This COA will not be changed when the MN changes its FA within the same foreign domain. A regional registration contains a new COA, it been the address of the new FA, to which the MN is now connected.

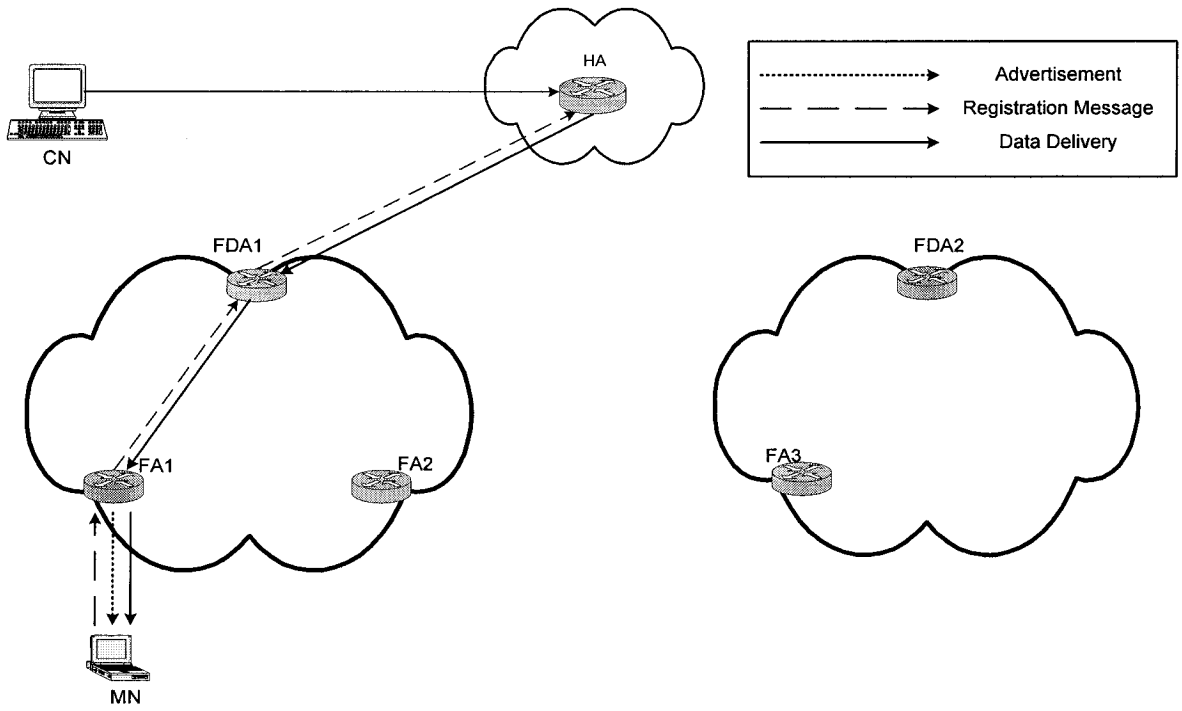

Figure 1.14 Architecture of Hierarchical Mobility Support

Figure 1.14 shows the architecture of hierarchical mobility support. In a hierarchical mobile MPLS architecture, a MN determines whether it is in its home networkor a foreign domain, when it receives an agent advertisement message, broadcasted by a FA. If it is in a foreign domain, the MN acquires a temporary COA from the FA and sends back a registration request. Instead of sending this Registration Message directly to the HA, the FA forwards it to the Foreign Domain Agent (*FDA*) using normal IP forwarding. The FDA further forwards it to the HA of the mobile host. The FDA replaces the FA's address with its own before forwarding. Thus, when the HA receives the Registration Message, it knows the IP address of the FDA. If an LSP does not exist between the HA and this FDA, the HA sends a label request to the FDA using the LDP protocol, with the IP address of the FDA as FEC. The FDA replies with an LDP label mapping message back to the HA and sends a label request to the FA of the subnetwork where the mobile host currently locates. When the label mapping message arrives to the HA, the LSP from the HA to the FDA is established. Similarly, the FA replies with an LDP label-mapping message back to the FDA leading to the establishment of the LSP from FDA to the FA. Thus, the HA searches the label table to locate the row with the MN's home address as the FEC and to update the outgoing port and out label according

to the LSP from HA to the FDA. Finally, the HA sends a registration reply to the FDA along the LSP from HA to FDA. The FDA will then forwards the registration reply to the FA along the LSP from the FDA to the FA. When the registration reply finally arrives at the FA, its label value and port number will be inserted into a newly added row in the FA's label table under the "In label field" and "incoming port field". Figure 1.15 shows the registration procedure for mobile hosts in Hierarchical Mobile MPLS.

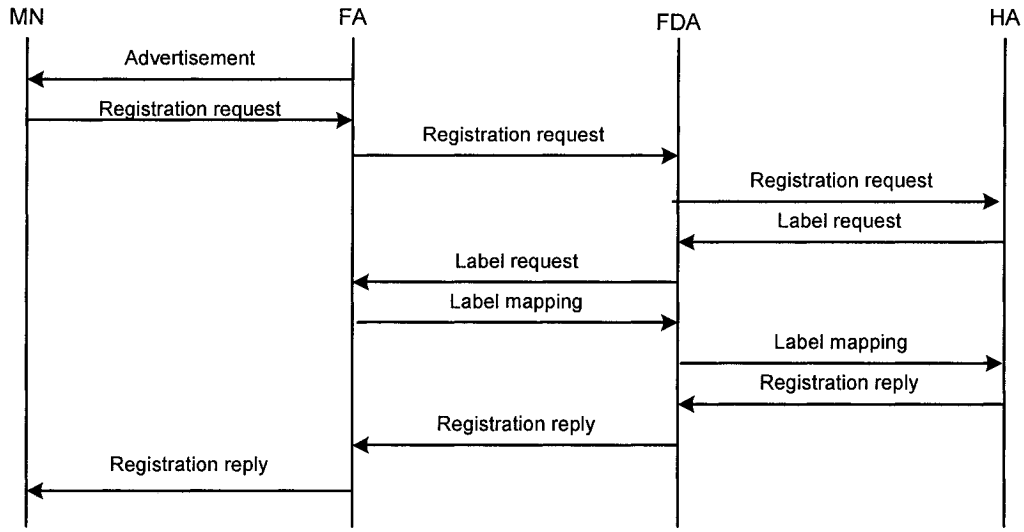

Figure 1.15 Registration at FDA and HA in Hierarchical Mobile MPLS

When a CN sends packets to a MN located in a foreign network, the HA of the MN intercepts these packets. In hierarchical mobile MPLS, the HA uses the incoming label value as an index to look up its label table and find the outgoing label and port for this packet. Then the packets are delivered from the HA to the FDA along the LSP by label swapping. The FDA receives the packets and subsequently forwards them through the LSP from FDA to the FA. After the FA receives the packets, it looks up first its label table. Since it is the egress of the LSP from the FDA to FA and the out label and outgoing port fields are empty, the FA strips off the label and sends the packets to the mobile host through the IP layer. Finally, the mobile host receives the packets sent by the CN.

During handoff, the mobile host may also notify the previous FA of its new COA, by sending a binding update message [7]. This allows the previous FA to cache the new binding of the

mobile host. If the FDA forwards later a packet to the mobile host using out-of-date cache entry, the previous FA will receive the packet, setup a LSP to the new FA, and send the packet to the new FA through the LSP. Figure 1.16 shows the registration procedure of the MN's handoff from one subnetwork to another within the same MPLS domain.

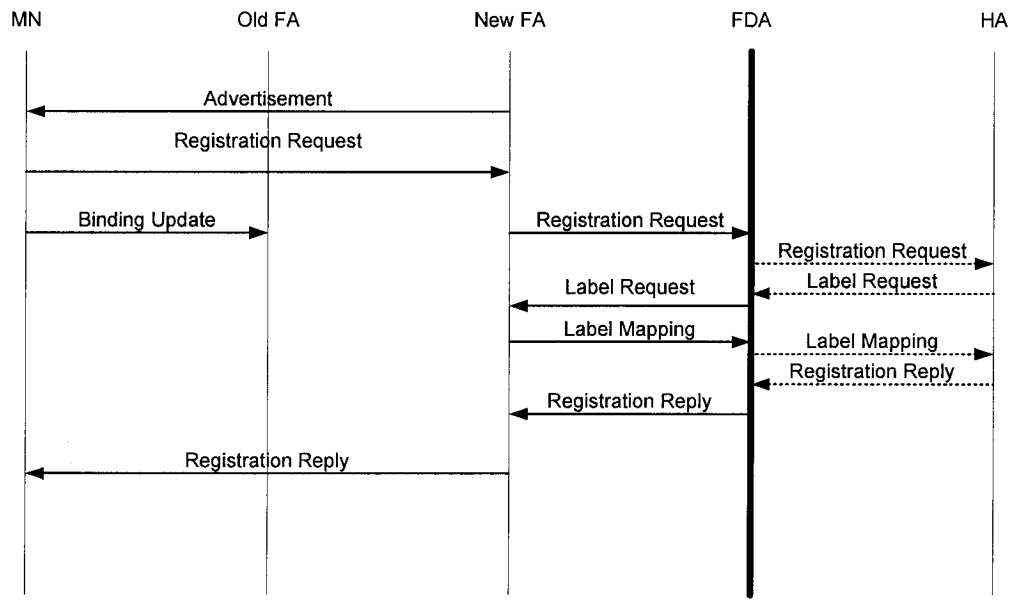

Figure 1.16 Regional Registrations at FDA in Hierarchical Mobile MPLS

Hierarchical Mobile MPLS also support multi-level hierarchy of FAs between leaf FA and FDA. Each FA in the hierarchy must maintain a binding in its visitor's list for each MN connected to a leaf FA in the hierarchy. These bindings are established and refreshed by regular registration requests and replies that the mobiles exchange in the network. The regional registrations sent by a MN are only forwarded to the first FA that already has a binding for this MN. The upper levels of the hierarchy are not aware of the details of the mobiles movements since they don't need to change their binding. This way, the message exchange for handoff is limited to a localized, involving only small number of nodes.

1.2.4.3 Micro-Cell Mobile MPLS

Similar to H-MPLS, MM-MPLS is also designed for the purpose of handling the movements of a MN within the same domain in a manner that is transparent to the MN's HA. Micro-Cell

Mobile MPLS (MM-MPLS), however, has some intermediate LSRs between the previous mentioned FDA and FA. Unlike the FDA defined in H-MPLS, these intermediate LSRs can provide functionality similar to the FDA in H-MPLS, i.e., they can recognize if a MN requests new path establishment is already served through them, and make the switch to the new location without tearing down the remaining path. Thus, the FDA may even not need to know the movement of a MN. This has the benefit of further reducing the delay and jitter during a handoff period. Figure 1.17 shows the architecture of MM-MPLS network.

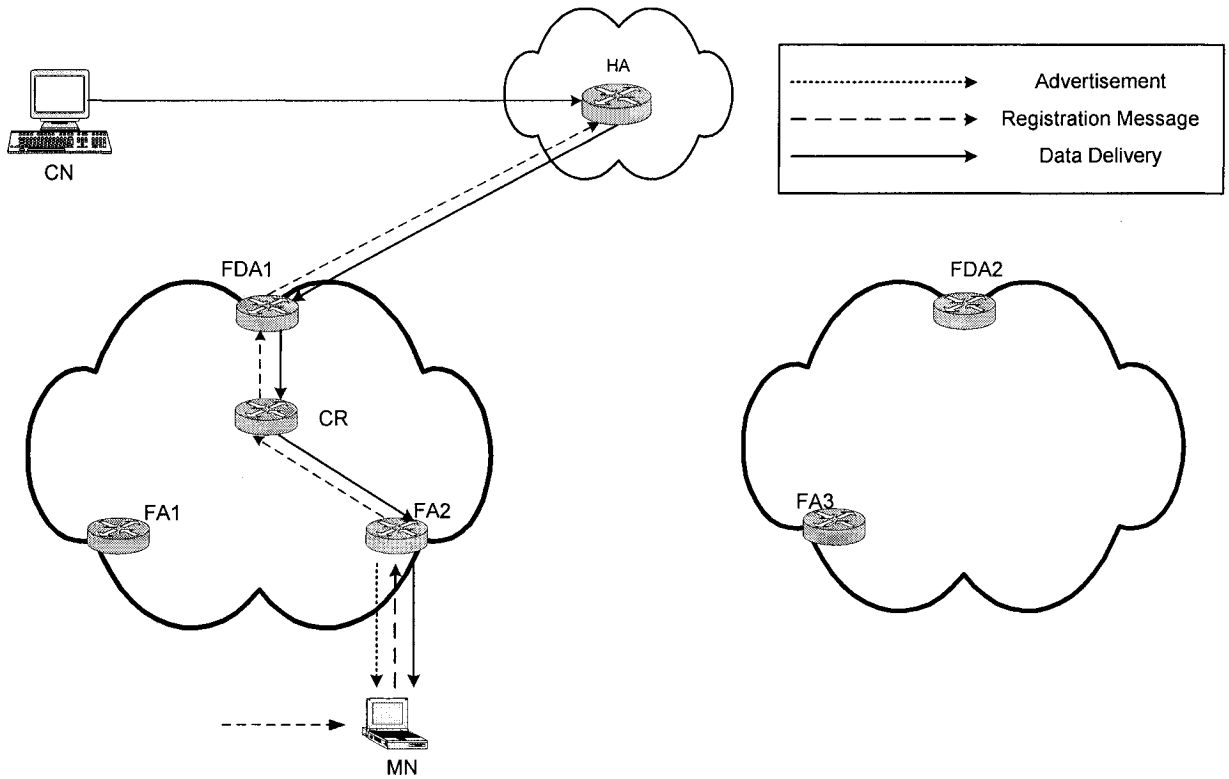

Figure 1.17 Architecture of Micro-cell MPLS network

When a MN handoffs from one access network to another within the same foreign domain, it sends a registration request to the new FA. Then the new FA will relay the registration message towards the FDA of the domain. Every intermediate and MM-MPLS enabled LSR in the path from the new FA to the FDA will check whether there is an entry for the MN. We define the crossover LSR is the MM-MPLS enabled LSR closest to the MN that is at the intersection of two paths, one between the FDA and the previous FA, and the other between FDA and the new FA. Upon receiving the registration request, the crossover LSR will setup

a new LSP for the new FA using the MN's home address as FEC. After the new LSP from the crossover LSR to the new FA is established, the crossover LSR updates its label table and redirects the LSP, with the MN's home address as FEC, to the new FA. Figure 1.18 shows the message activities when the MN handoffs within a foreign domain.

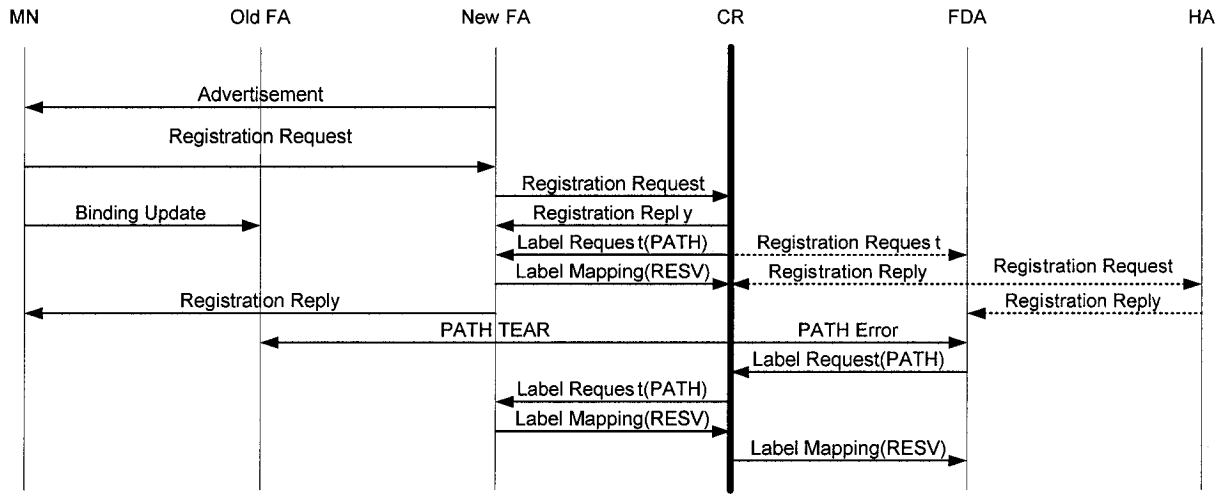

Figure 1.18 Messages process in MM-MPLS when the MN handoffs

1.3 Contribution of This Work

A test network implementing the MM-MPLS and H-MIP protocols has been established and its proper functionalities were confirmed through elaborate testing. This experimental network provides the opportunity for the testing of applications in order to assess their performance when are run by mobile users, serviced through MM-MPLS or H-MIP enabled networks.

Our experiments provided accurate statistical information, confirming the advantage of MM-MPLS over H-MIP in terms of packet forwarding delay.

In addition, issues coming from the operating system itself, obtaining the performance of these Linux based routing nodes were identified. Such knowledge could not gain through analysis or simulation based evaluations. It gives new material for the implementation of better functioning nodes.

1.4 Thesis Organization

The remainder of this thesis is organized as follows:

- In Chapter 2, an overview of the implementation environment, the Linux OS, is given and relevant features and applications are explained.
- Chapter 3 provides the implementation design of the MM-MPLS testbed. A detailed function and data structure reference is given in the appendix of this document.
- The testbed has been setup in order to emulate representative scenarios and demonstrate the key capabilities of the underlying protocols. This has been accomplished by conducting a number of tests and performance assessment experiments. They are provided in chapter 4.
- Chapter 5 concludes this work.
- In Appendix, additional experimental results and performance measurements are conducted.

Chapter 2 Linux Apparatus

2.1 Linux Operating System

The Linux operating system and its kernel is an evolutionary project started in 1991, and since then involved hundreds of developers around the world. Because of its nature, no complete design documentation exists that could be used as basis for planning the introduction of new functionality. Information covering the Linux OS and related features is available by means of books, documents, "howto"s, manuals, mailing lists and FAQs.

An alternative approach to understand the functionality of the Linux OS and related programs is of course given through the source code itself. In order to add new functionalities to an existing implementation, an intensive study of its source code is indispensable. Regarding the Linux kernel, analyzing its sources seems to be the only kind of available documentation for many parts of its implementation, especially that of the MPLS network layer.

The Linux kernel has the possibility to separate part of its functionality in modules that can be loaded when needed and unloaded when becoming obsolete. This provides the running operating system with the capability to potentially offer a lot of functionality and at the same time keep the amount of code and processes currently running in kernel space relative small. Another advantage of loadable kernel modules emerges when developing new kernel functionality. By way of that, the possibility is given to load, test, unload and modify a prototype implementation in kernel space even multiple times without the constraint to recompile the complete kernel or reboot the whole system. Regarding the RSVP-TE for DiffServ over MPLS implementation all the new kernel functionality required is either provided as an individual kernel module or applied as an extension to an already existing

one. Many of the features available are uses in the MM-MPLS implementation are provided as kernel modules.

The proc-filesystem is a special filesystem, which only exists in the memory of the Linux OS. The kernel and kernel modules can use this filesystem as an interface to the userspace[73]. The kernel module can create a special directory tree and files in the filesystem, which can be read with normal user tools like cat [74]. The other way around, it is possible to write a file to the proc-filesystem in order to pass parameters to the kernel. In the MM-MPLS implementation, the proc-filesystem is used to define tunnel information for an LSP and provide feedback (about whether the operation succeeded) to the user space. It is used to access available resource information on each node of the MM-MPLS testbed.

2.2 QoS in Linux

When the kernel has several packets to send out over a network device, it has to decide which ones to send first, which ones to delay, and which ones to drop. This is the job of the packet scheduler, and several different algorithms for how to do this "fairly" have been proposed. With the Linux QoS subsystem (which is constructed of the building blocks of the kernel and user space tools like ip and tc command line utilities) it is possible to apply flexible traffic control.

2.2.1 Linux Traffic Control

Traffic control capabilities have been available in the Linux kernel since its 2.2 series. Since then, it has matured considerably. Shaping configurations are implemented using a variety of available packet schedulers and shapers, which can be configured using the tc binary included in the iproute2 package of tools. qdisc is the term used to refer to these schedulers under Linux.

The Linux traffic shaping implementation allows us to build arbitrarily complicated configurations, using as building block the qdisc. We can choose from two kinds of qdiscs: classless and classful. By default, all Ethernet interfaces get a classless qdisc, which is essentially a FIFO. We replaced this with something more interesting. Classful qdiscs, on the other hand, can contain classes to arbitrary levels of depth. Leaf classes, those which are not parents of additional classes, hold a default FIFO style qdisc. Classless qdiscs are schedulers. Classful qdiscs are generally shapers, but some schedule as well.

Recent Linux kernels offer a wide variety of traffic control functions. The kernel parts for traffic control and several user-space programs to control them cover the mechanisms required for supporting IntServ and DiffServ.

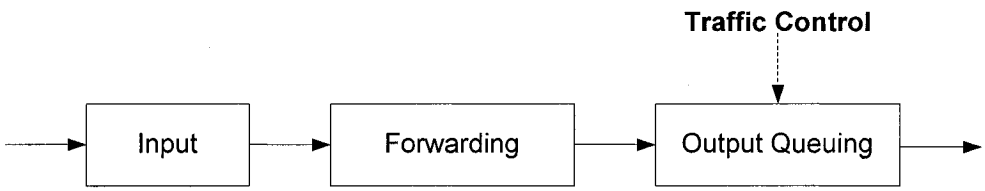

Figure 2.1 Processing of network data

Figure 2.1 shows in block diagram form how the kernel processes data received from the network, and how it generates new data to be sent to the network. "Forwarding" includes the selection of the output interface, the selection of the next hop, encapsulation, etc. Once all this is done, packets are queued on the respective output interfaces. Then traffic control decides which order the packets should be sent on the net, in certain cases delaying or dropping some portions of traffic. Once traffic control has released a packet for sending, the device driver picks it up and emits it on the network.

The traffic control code in the Linux kernel consists of the following major conceptual components:

- queuing disciplines
- classes (within a queuing discipline)
- filters
- policing

2.2.2 IntServ on Linux (RSVP)

In the integrated services framework the first function is provided by QoS control services. The second function can be provided in a number of ways, but is frequently implemented by a resource reservation setup protocol such as RSVP [53]. RSVP is a unicast and multicast signalling protocol, designed to install and maintain reservation state information at each router along the path of a stream of data. The state handled by RSVP is defined by services specified by the Integrated Services.

Figure 2.2 shows a simplified view of the functionality within the RSVP daemon from the project of RSVP in Linux [54]. This implementation supports:

- interfaces to application
- signaling
- functionality of reservation styles (GS, CLS) to link-layer-specific parameters
- interface to traffic control (in the kernel)

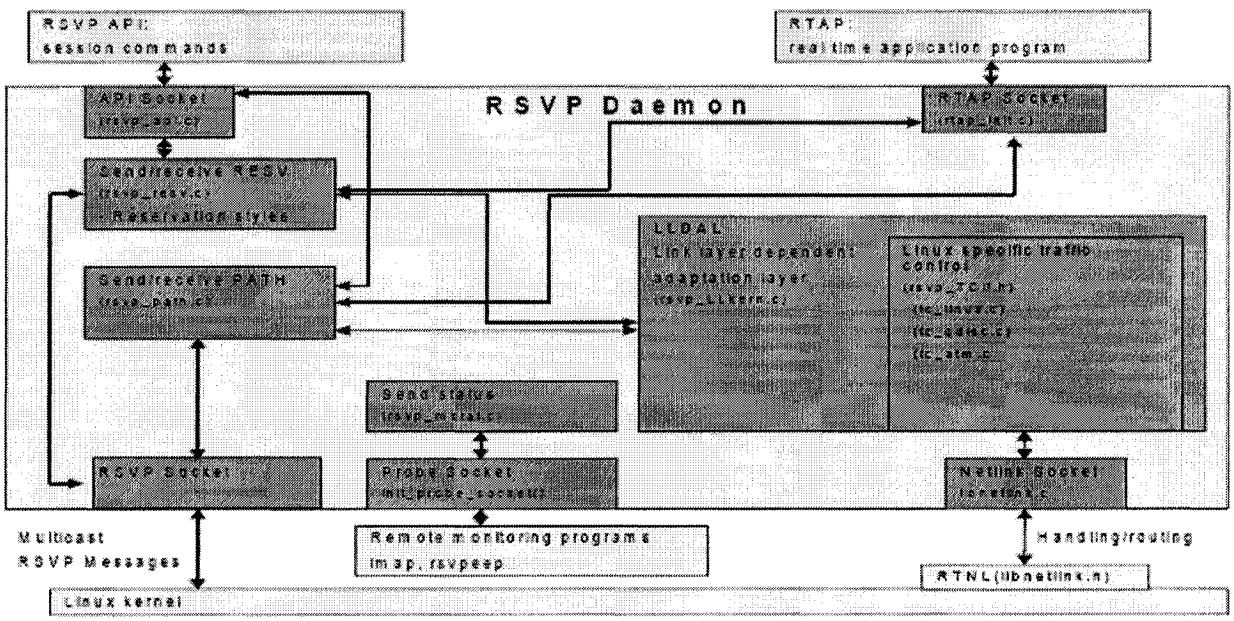

Figure 2.2 Block Diagram of the RSVP daemon [45]

2.2.3 DiffServ on Linux

The Linux implementation of DiffServ supports basic classification and DS field manipulation required by DiffServ nodes. The design enables configuration of the first PHBs that are being defined in the DiffServ. The implementation allows maximum flexibility for node configuration and for experiments with PHBs, while still maintaining a design that does not unnecessarily sacrifice performance.

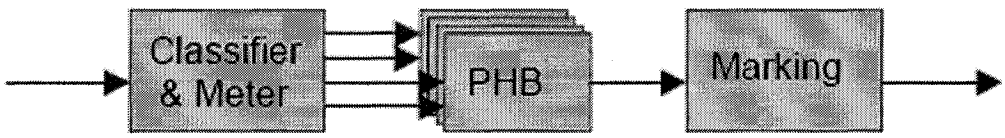

Figure 2.3 General DiffServ forwarding path [54]

Figure 2.3 shows the general structure of the forwarding path in a DiffServ node. The prototype implementation of DiffServ support requires the addition of three new traffic control elements to the kernel: (1) the queuing discipline "sch_dsmark" to extract and to set the DSCP1, (2) the classifier "cls_tcindex" that uses this information, and (3) the queuing discipline "sch_gred" that supports multiple drop priorities and buffer sharing. The current implementation supports:

- Classification
- Marking
- Cascaded meters
- PHBs Implementation (e.g. EF, AF, ...)
- Shaping
- End systems functionality

2.3 Justification of Using Linux

Our decision to use the UNIX like Linux OS as the platform for the implementation of the test network is because it is a free and well documented OS, already proved to be an excellent option for the development of prototype network functionalities in various cases. A reasonable amount of experience with the Linux OSalready existed in our lab and the authors of this manuscript were already reasonably familiar with Linux. After research, the decision was made to take use of some open-source projects under Linux. One is the Dynamics Mobile IP from Helsinki University of Technology (HUT), the other is the RSVP-TE daemon for DiffServ over MPLS under Linux from The Broadband Communication Networks research group (IBCN) of the Department of Information Technology (INTEC), Ghent University. Both systems are very stable, well maintained, and user mailing list indicated the wide usage, acceptance and opportunity for discussion of potential future problems. At last, a new MM-MPLS testbed has been setup to gather experience with the implementations and their configurations. The source code has been explored and possible hooks for the necessary extensions have been identified.

2.4 Dynamics Mobile IP for Linux

The HUT Dynamics Mobile IP system was developed at Helsinki University of Technology (HUT). It is a scalable and Hierarchical Mobile IP implementation using the Linux operating system [56].

As we early described in Section 1.2.3.2, Hierarchical Mobile IP is an extension to Mobile IP that supports a hierarchy of FAs between the MN and the HA to efficiently support the micro-mobility approach. Figure 2.4 shows an example of Hierarchical Mobile IP network.

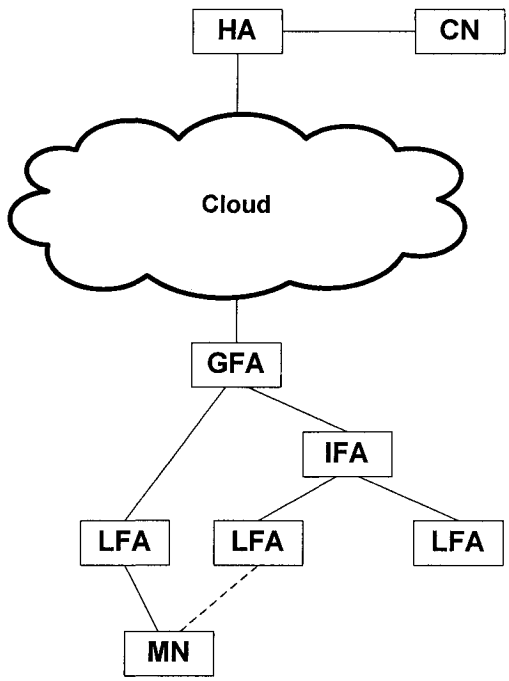

Figure 2.4 Hierarchical foreign agents

HUT Dynamics supports several types of FAs with hierarchy level. The tunnel is established between the HA and the registered location of the MN. Tunnel is build with segments between the HA to the root FA (GFA), between the GFA to the lower FA, which is either a Intermediate FA or a lowerest FA, and between each FA in the path down to the lowest FA (LFA). In FA decapsulation mode the LFA decapsulates the encapsulated packets and send them directly to the MN.

When MN moves within a foreign network, each FA on the path from MN to the HA examines if it already has a binding for this specific MN. Then a localized location update is performed in the FA hierarchy. A new tunnel is created to the new path location of the MN. Switching FA (SFA) is the FA that replies to the registration request of the MN. Localized location update is shows in figure 2.5.

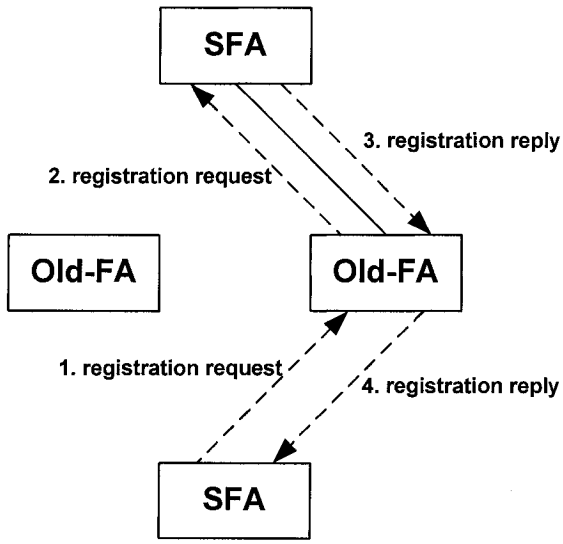

Figure 2.5 Local registration update

In HUT Dynamics Mobile IP implementation, the MN makes handoff decisions and the FA hierarchy assists the MN in the handoff management. Both HAs and FAs are called Mobility Agent (*MA*). The handoff decision is made by a policy-based MA selection and detection. The MN can gather information agent advertisement from the MAs and then use different policies and criteria to choose the communicating MA. The expiration of the agent advertisement in Mobile IP provides a method for MA detection. Mobility agent selection is based on priority comparison. Priorities are modified and analyzed in the signal quality analyzer. Node selector makes the final decision based on the priority and currently used MA.

Node selector is a component that compares the priority values. It communicates with the message handling module and decides whether the MN should change the MN. The node selector uses certain rules to make selection decision. The set of rules is called policy. Comparisons and selections are based on the node priority, which can be modified to achieve

the required results by the policies. The node selector picks up the node that has highest priority.

There are currently four different policies for the node selector: Newest-Advertisement, Newest FA, Eager switching and Early Expire. In addition to different policies, the monitoring system in MN is configurable with parameters include: *threshold, min-balance, expire-percent, old-FA-factor, worst-min-time, worst-max-time*, and *average-length*. Each value has a default value that can be changed by the *dynmn_tool* configuration tool. Figure 2.6 shows the relationship between the configuration parameters and policies.

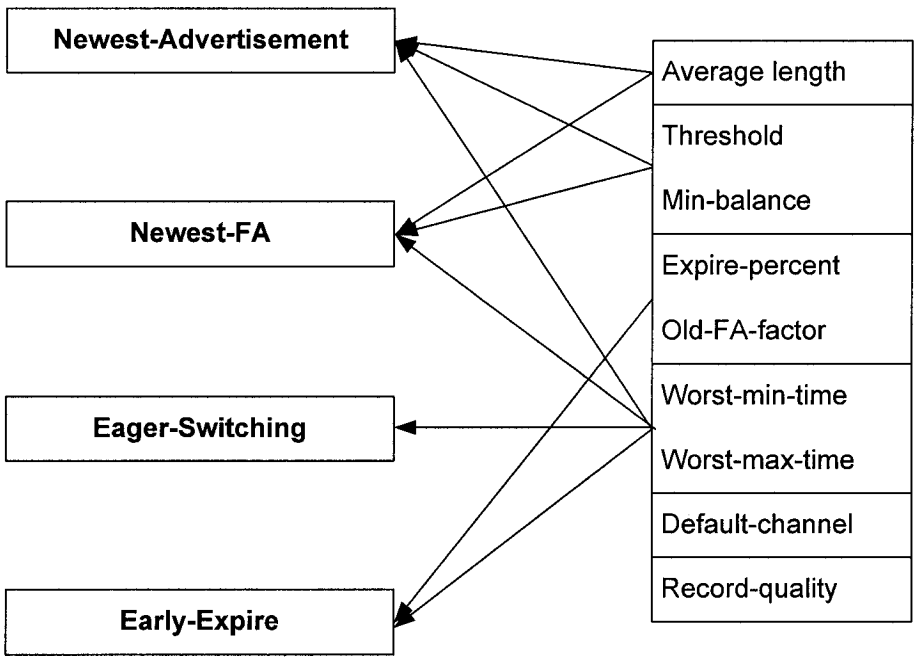

Figure 2.6 Policy and configuration parameters

2.5 RSVP-TE for DiffServ over MPLS in Linux

The RSVP-TE daemon for DiffServ over MPLS under Linux is a signaling daemon for Diffserv over MPLS, originally developed by the Internet Broadband Communication Networks research group (IBCN) of Information Technology (INTEC) of the Ghent University. The daemon is based upon the Nistswich version 2.0 daemon for Free BSD by USC and a part of an IntServ RSVP daemon for Linux developed by Alexey Kuznetsov. The MPLS Linux kernel code is based upon mpls-linux by James R. This effort combines the daemons so that the MPLS support found in the Nistswitch version is now available on Linux. Moreover support for DiffServ over MPLS (DS/MPLS) is also added.

The overall architecture (Figure 2.7) consists of a number of components located in the user space or kernel space. The important parts of the kernel that are used are netfiler to classify the packets, QoS and faire queuing to support service differentiation between flows and of course MPLS support.

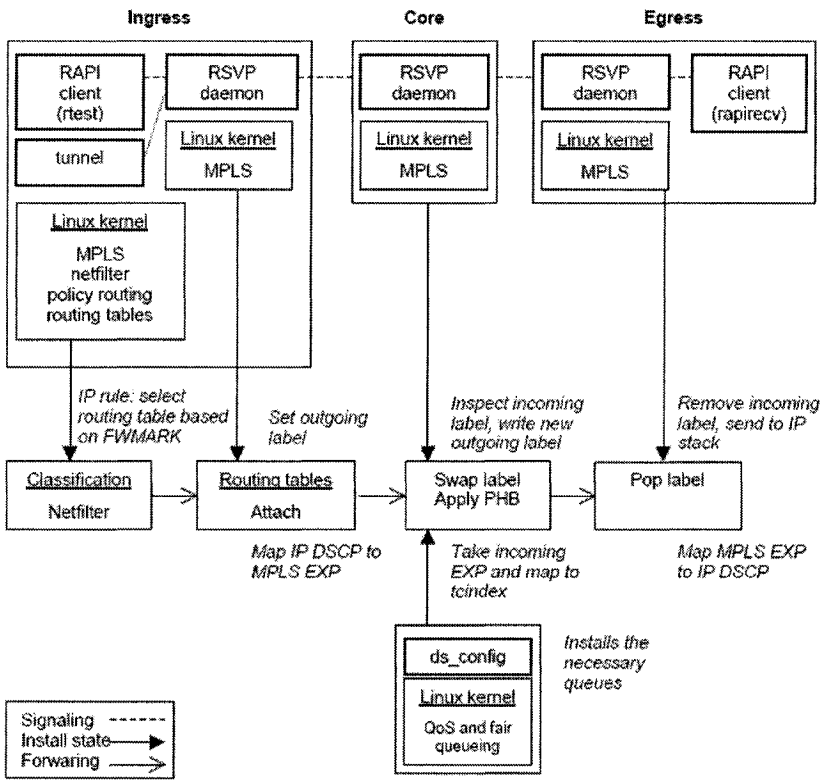

Figure 2.7 Architecture of DiffServ over MPLS using RSVP-TE under Linux [62]

The prime user space component is the RSVP daemon that is responsible for the RSVP signaling and the maintenance of the MPLS state. The daemon is responsible for the allocation and installation of the MPLS labels during LSP set up and for freeing and removing labels during LSP tear down. Two components use the RAPI: "*rtest*" and "*rapirecv*". "*rtest*" is an application that takes LSP requests and issues them to the daemon. "*rapirecv*" is an application that receives label requests at the egress and dictates the daemon to send a response back to the ingress.

"*Tunnel*" is an application that maps traffic on an existing LSP. Mapping can be based on destination address, protocol, destination ports and port ranges (TCP and UDP). Basically, anything that netfilter supports. Tunnel sends status packets requests to the RSVP daemon in order to receive information about the installed state in the daemon (existing sessions, labels, reservations etc.).

The packets are classified at the ingress router using netfilter. Packets are filtered on the OUTPUT and PREROUTING chain of the mangle table[6]. The mangle table is needed because the fwmark needs to be set. The OUTPUT and PREROUTING chains are used in order to filter on both locally generated and incoming traffic. Based on the value of the fwmark a routing table is selected (using policy routing). In the resulting routing table there is a MPLS tunnel interface acting as the default gateway. The tunnel interface encapsulates the packets on the LSP by attaching the correct outgoing label. The incoming DSCP is mapped to the EXP field in the MPLS header. A limitation of this architecture is that netfilter can only write (mark) on the mangle table and that only a single mark operation is possible. Thus while we do can map traffic on the LSP also setting the DSCP at the same node is impossible. The solution is to set the DSCP before the traffic enters the ingress LSP.

[6] The Linux kernel has the built-in ability to filter packets, allowing some of them to be received by or pass through the system while stopping others. The kernel's netfilter has three built-in tables or rules lists. They are filter table, NAT table and mangle table. The mangle table is used to alter certain fields in the headers of IP packets. It can be used to change the Time to Live (TTL), change the Type of Service (TOS) field, or mark packets for later filtering.

In the core node the MPLS stack inspects the incoming label and sets the new outgoing label and next hop. At the DiffServ level the current EXP value of the packet is inspected. There is a mapping from this EXP value to a tcindex. The tcindex in turn determines the correct outgoing queue so that the correct forwarding behavior (PHB) can be applied.

Finally in the egress the incoming EXP field is mapped to the DSCP field and then the MPLS header is stripped off and the packet is sent to the IP layer.

2.6 Extensions for Supporting Micro-cell Mobile MPLS Networks

Mobile IP daemon and RSVP daemon was developed from the open source project we downloaded from the internet. After some modifications, they have been the two daemons running in the MM-MPLS domain. The Basic extension of MM-MPLS is that it provides the RAPI and the API for path redirection from RSVP-TE daemon to the Dynamics Mobile IP daemon. "*rtest2*" on the HA and "*rapirecv_auto*" on the FAs are extended version of *rtest* and *rapirecv* (shown in Figure 2.7) respectively that support the automatic set up of a (large) number of LSPs. The tunnel from the HA to the FA is the LSP, which will lead the packets transferring on the MPLS layer. Thus, at the HA, the essential tunneling function is located at LSP tunnel setup instead of the original IP-to-IP tunnel that is used in Mobile IP. "*rapirecv_fda*" is also added to maintain the FDA to terminate the LSP from the HA to the FDA and to start the LSP from the FDA to the leaf FAs. On the Cross-over Router (CR), the IP-to-IP tunnel redirecting, which occurs during handoffs, is replaced by the LSP redirecting in RSVP-TE daemon.

Main functions running on each mobile agent are the following:

Home Agent

dynhad	home agent function on Mobile IP. (Note: rtest2, the PATH setup function for RSVP-TE is embedded in ha.c)
rsvpd –D	foreign agent module(FA_M) of RSVP-TE
tunnel	mapping traffic onto the label switching patch from the HA to the FDA

Foreign Domain Agent

dynfad	foreign agent function on Mobile IP
rsvpd –D	foreign agent module(FA_M) of RSVP-TE
rapirecv_fda	terminate the PATH from the HA, send RESV back to the HA, and start another PATH from the FDA to the FA
tunnel	mapping traffic onto the label switching patch from home agent to foreign domain and send back RESV message to the HA

Crossover Router (Foreign agent with two or lower foreign agent connections)

dynfad	foreign agent function on Mobile IP
rsvpd –D	Crossover Module (CR_M) of RSVP-TE

Foreign Agent

dynfad	foreign agent function on Mobile IP
rsvpd –D	foreign agent module(FA_M) of RSVP-TE
rapirecv_auto	terminate the PATH from the FDA to the FA and send back RESV message to the FAD

Mobile Node

dynmnd	mobile node function on Mobile IP

Chapter 3 Implementation of MM-MPLS

This chapter describes how the concepts discussed in Chapter 2 were applied to the implementation. In this chapter, a description of the MM-MPLS testbed is also provided.

3.1 Architecture of MM-MPLS Testbed

The initial specifications set for the testbed was to be capable of emulating micro and macro mobility movements as well as providing QoS support on MPLS network. Such a testbed provides a network environment for emulation of a variety of mobility scenarios and the opportunity to test applications in more detail under more realistic network conditions.

The following network properties were identified as significant for testing the MM-MPLS functionality and are integrated in the testbed.

- The testbed must include a HA, CN and MN.
- To emulate a macro movement, a FDA should be included. To emulate a micro movement, at least two FA should be available.
- To handle the local movement in a foreign domain, there should an intermediate route functioning as CR.
- To trigger the movement of the MN, it must be possible to enable and disable the connection of the MN to any mobile agent and its advertised links.
- The transmission of a message always causes a certain delay until it is received by its destination node. This transmission delay in real networks is typically much longer than that in a testbed, which is build up in a single room. To emulate the effects of the MM-MPLS scheme that can be expected from its deployment in a real network environment, some transmission delay must be considered seriously.

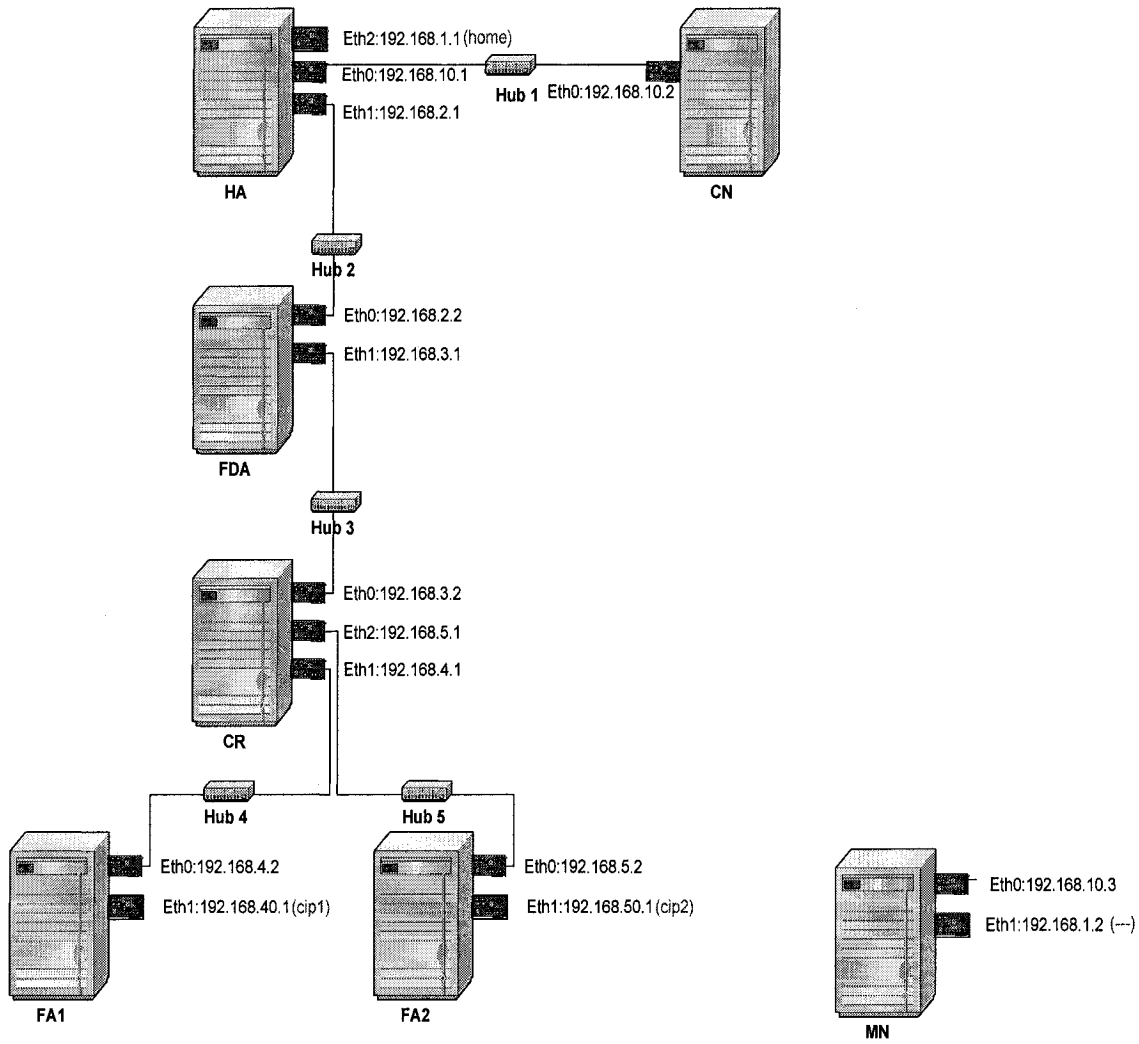

Figure 3.1 Architecture of MM-MPLS test network

Figure3.1 shows the network architecture of the developed MM-MPLS testbed. The prototype of MM-MPLS network uses 7 desktop-PCs. Orinoco11b PCI Adapters with Orinoco 802.11b Gold PC Cards are installed in the home agent, the two foreign agents and the mobile node, in order to simulate the roaming process from home domain to foreign domain and within the same foreign domain. All the wireless cards are configured to operate in the Ad-Hoc mode.

3.2 Design of Modules

There are two daemons running in the MM-MPLS domain. Mobile IP is adopted on all computers including the mobile node, which is based on the Dynamics Mobile IP system. Changes have been made in order to be able to use label switching tunnel instead of the IP-IP tunnel. The other daemon is the RSVP-TE daemon for DiffServ over MPLS under Linux. The registration signal from Mobile IP is being used as the activator to handle the LSP setup in RSVP-TE. RAPI and Self-defined Interface for the Unix Domain socket, which are the two kinds of interface between the RSVP daemon and D-MIP daemon, are deployed for integrating Mobile IP with RSVP-TE. Figure 3.2 shows the basic module of a node in the MM-MPLS domain.

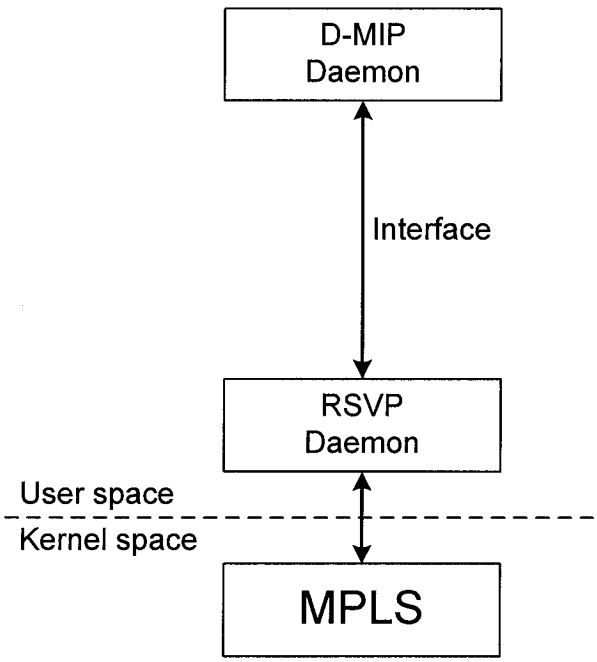

Figure 3.2 Basic modules for MM-MPLS nodes

3.2.1 Functionality of the Modules

According to the mechanism of MM-MPLS, each node has its own functionality when working in the MM-MPLS domain.

Home Agent (HA)

Home Agent is a router located in a mobile node's home network that is responsible for intercepting datagram destined to the MN, when the MN is in a foreign domain, and tunnels these datagram to the mobile node. It also maintains current location information for the mobile node. In MM-MPLS, the tunnel is a label switching path which is setup by the RSVP-TE. When the mobile node enters a foreign network, the whole tunnel is made of two PATHs, one is from the HA to the FDA and the other is from the FDA to currently registered FA. When Home Agent gets the Registration Request message from FDA and knows the IP address of the FDA, it will send a PATH message including a label request object to the FDA with the IP address of FDA as the FEC. The FDA will later reply a RESV message. It means that the LSP between the HA and the FDA is established. The HA then send a Registration Reply message back to the FDA. When the connection between the HA and the FDA is established, the Mobile IP registration messages and RSVP messages will refresh periodically to maintain the tunnel.

In summary, the HA module should be able to perform the following functionalities:

- Establishment of the LSP from HA to FDA.
- Maintenance of the tunnel for Mobile IP. Mobile IP requires that registration request messages are received regularly, in order to maintain the soft state of the connection. If the lifetime of the last registration message (usually placed at 300 seconds) expires, the HA will delete the tunnel from the HA to the MN in order to release the resources reserved for the MN.
- Refreshment of the PATH message for the LSP from the HA to the FDA.

Foreign Domain Agent (FDA)

The Foreign Domain Agent initially introduced in Hierarchical Mobile MPLS, has functionality similar to the gateway foreign agent of Hierarchical Mobile IP. The difference between the two of them is that the FDA is enabled to act as an MPLS Ingress/Egress router

and is responsible to setup and maintain the LSP from itself to the lower level FA that is acting as the Mobile node's point of attachment to the foreign domain.

While in H-MPLS, the FDA is always involved in the handoff process of a MN. It is not the case in MM-MPLS. As we mentioned earlier, several intermediate LSRs, named Crossover Routers (CR) are developed to be able to recognize if a MN was serviced through them while under the control of another FA or not. If the first case is true, the CR terminates the transfer of the registration request message send by the MN, redirects the LSP towards the new FA by establishes a path to the new FA, and let the connection to the old FA expire. A PATHERR message with the address of the new FA is included in the ERO to the FDA. After receiving the PATHERR message, the FAD will be sending the PATH refreshing message towards the new FA.

In summary, the main functions of the FDA module are:

- To terminate the LSP from the HA to the FDA
- To establish and maintain the LSP from FDA to the mobile node through an FA when mobile node moves in its domain.
- To maintain the registration message from the FA within this foreign domain to the HA.

Crossover Router (CR)

A crossover router is a foreign agent with at least two connections to the lower foreign agents. It is the intermediate LSR, located at the intersection of the old FA and the new FA. When the mobile node roams in the local domain, the CR will take charge and handle handoff process instead of passing the message upwards and have the FDA to deal with the handoff process. Once the CR gets a Registration Request message from a new FA, it sends a Registration Reply to that FA and then redirects the LSP PATH to the new FA by replacing the ERO with the address of the newly registered FA. When it gets the RESV message from the new FA, it tears down the old PATH from CR to the old FA by sending a PATHTEAR message to the old FA. CR will receive a PATH message from FDA to the old FA. Since the

ERO has been changed to the new FA, the CR will send a PATHERROR message with the new ERO up to FDA in order to inform the FDA to redirect the refreshing PATH message for LSP towards the new FA afterwards.

We may summarize the functions of the CR module as follows:

- To transfer the regular registration messages from FA to HA.
- To deliver the regular PATH/RESV messages for the LSP established between FDA and FA.
- To handle the local registration management when the mobile node moves within the same foreign domain and the CR is the intersection between the LSP from the FDA to the old FA and the LSP from the FDA to the new FA.
- To redirect the PATH to the new FA and to inform the FDA about the path changing when it works as a CR during the handoff.

Foreign Agent (FA)

The Foreign Agent is a router on a mobile node's visited network which provides routing services to a mobile node while registered with this FA. The foreign agent de-tunnels and delivers datagrams to the mobile node that were tunneled by the mobile node's home agent. For datagram sent by a mobile node, the foreign agent may serve as a default router for registered mobile nodes. In the MM-MPLS testbed, the FA acts as egress MPLS router. When a packet is transferred from HA to the MN along the LSP that terminates at the specific FA, it will be passed from MPLS layer to the IP layer at the FA. Then, it will be processed as IP packet.

Main functions of the FA module are:

- To broadcast the beacon signal regularly that is carrying the agent advertisement.
- To deliver the registration message from the MN to the HA when the mobile node attaches to this access point.
- To terminate the LSP from the FDA to the mobile node and maintain the RSVP messages refreshing regularly.

Mobile Node (MN)

A Mobile Node is a host or a router that changes its point of attachment from one network or subnetwork to another. A mobile node may change its location without changing its IP address; it may continue to communicate with other Internet nodes at any location using its permanent IP address, assuming that link-layer connectivity to a point of attachment is available. The Mobile Node works with the same functions as those in Mobile IP. There is no functional change for the mobile node when using MM-MPLS.

3.2.2 Interfaces between Mobile IP and RSVP daemons

As indicated in figure 3.3, there are two kinds of interfaces between the RSVP daemon and the D-MIP daemon. They are RAPI (an RSVP Application Programming Interface) and Self-defined Interface based on Unix Domain socket.

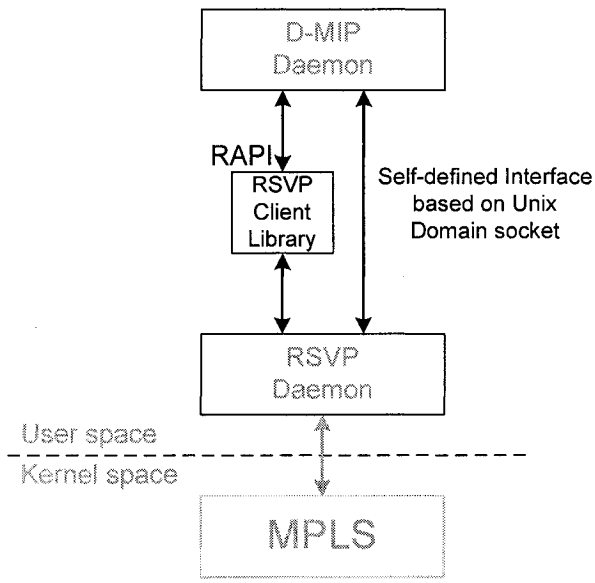

Figure 3.3 Interfaces of daemons in MM-MPLS

The RAPI is used by the D-MIP daemon to request the RSVP daemon to setup and to release the LSPs. In the module of HA, for example, the D-MIP daemon requests from the RSVP daemon to establish an LSP with the FEC of a mobile node from the HA to the FDA.

A self-defined Interface based on the UNIX domain socket is used to exchange other events between these two daemons. To request path re-direction and to reply to its functions occurring at a crossover router (CR) is an example of self-defined interface.

3.2.3 Components of MM-MPLS

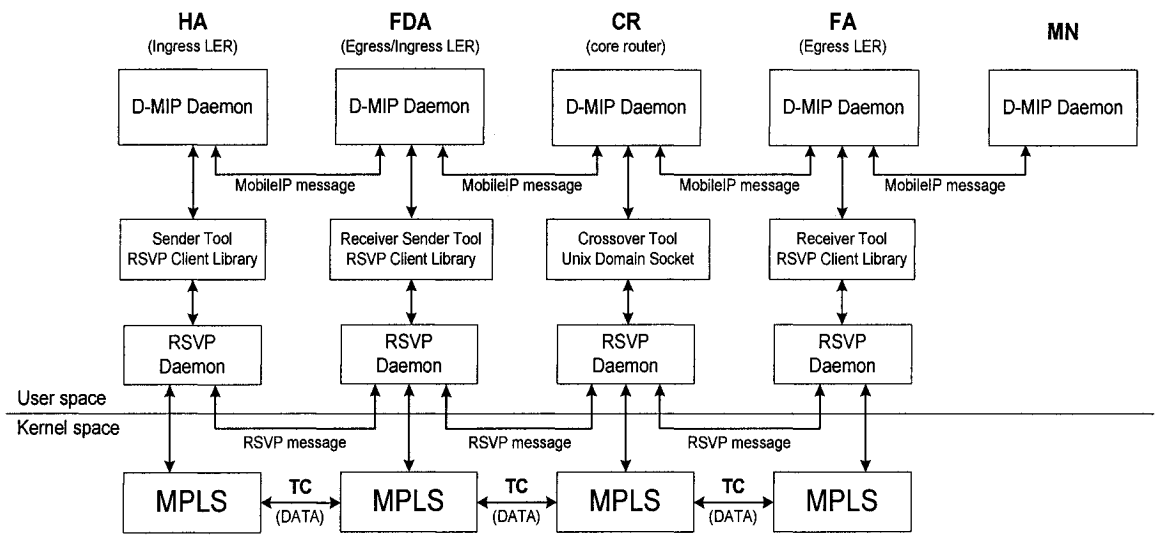

Figure 3.4 Components of MM-MPLS testbed

As shown in figure 3.4, each node in the MM-MPLS network is composed of the D-MIP daemon, the RSVP daemon and the interface worked with specific tools. In MM-MPLS, HA is the sender of the LSP from HA to FDA. The FDA is a receiver of the LSP from HA to FDA, and it is also a sender of the LSP from FDA to FA. At the CR node, the D-MIP daemon works as a CR tool to handle the path redirection. Then, FA is the receiver, with the function of terminating the LSP from the FDA to the FA.

3.3 Events Handling

Two types of events are mainly designed for the MM-MPLS test network. The Path-Setup process is in charge of setting up LSP paths from HA to an FA when it is away from its home

network. The path-redirection process is responsible for redirecting the LSP when the mobile node under the control of attached FA, which is however belongs to the same foreign domain.

3.3.1 LSP Setup

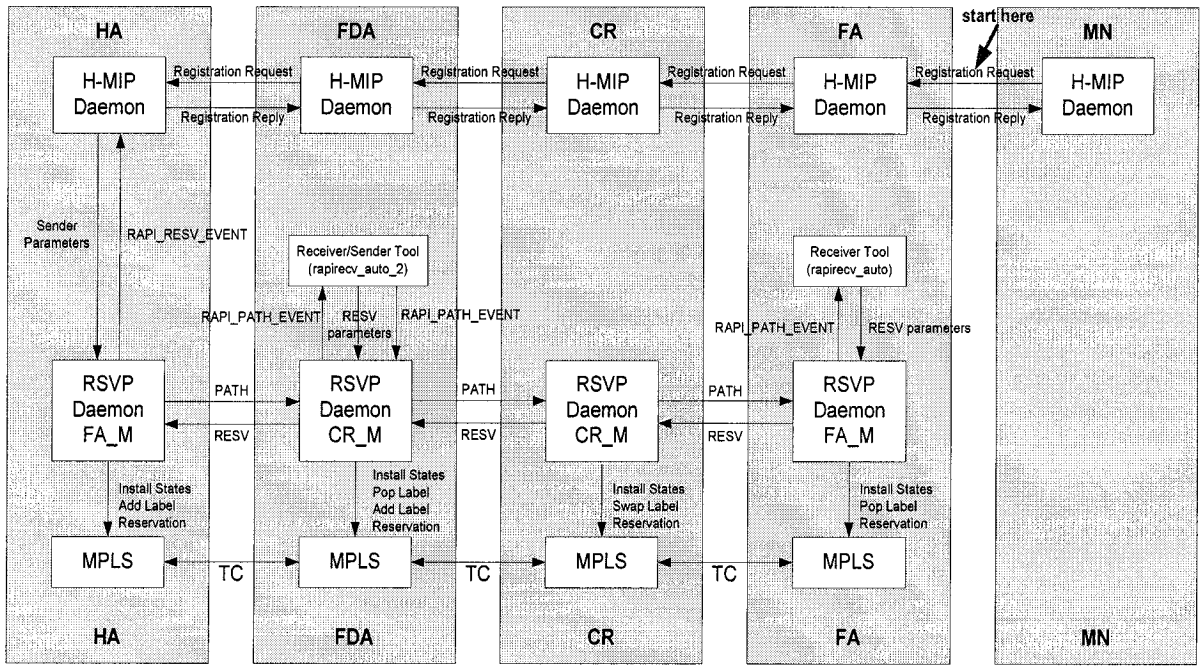

Figure 3.5 LSP setup process

The entire message exchange occurring between the various network nodes for establishing the LSPs from HA to FDA and from FDA to FA is described in figure 3.5.

In Figure 3.6, an example of the sequence in time on the by which the messages are generated is shown.

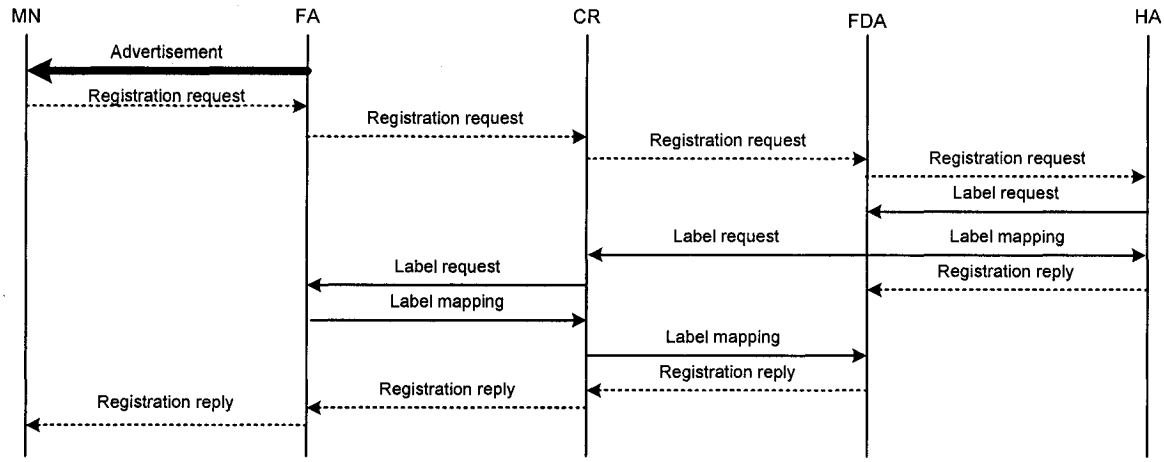

Figure 3.6 Messages in LSP setup by time sequence

Figure 3.7 shows the examples of messages on some nodes of MM-MPLS when LSP setup.

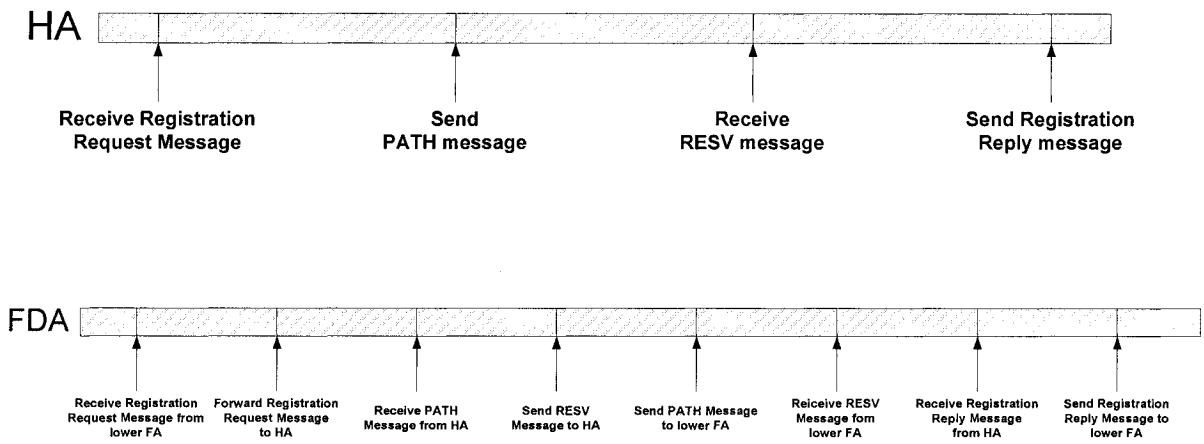

Figure 3.7 Messages on HA and FDA during the LSP setup

3.3.2 Path Redirection (CR_M Module Description)

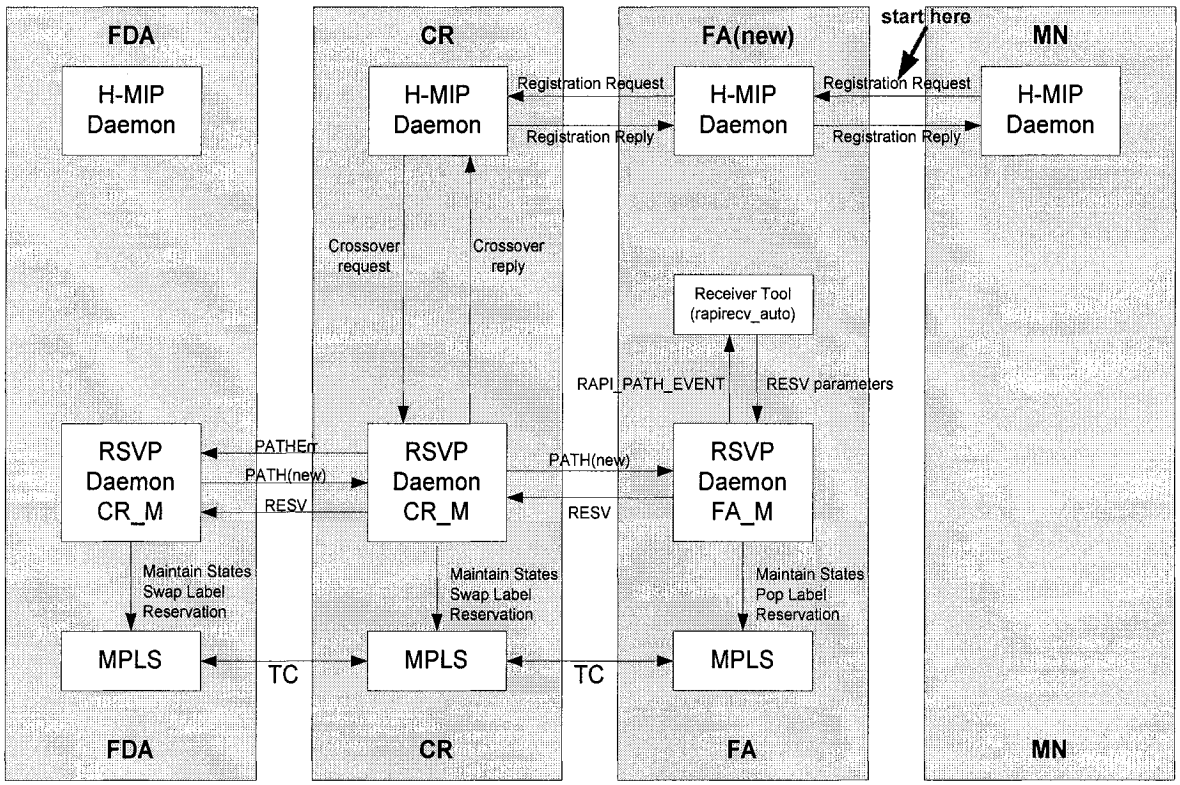

Figure 3.8 PATH redirecting process

When an MN handoffs from one access network to another within the same foreign domain, it sends a registration request to the new FA. Then the new FA will relay the registration message towards the FDA of the domain. Every intermediate LSR from the new FA to the FDA with path redirection capabilities incorporated into it (CR functionality) will check whether there is an entry for the MN. We define the crossover LSR as the LSR that is located at the intersection of two paths, one between the FDA and previous FA, the other between FDA and new FA. Upon receiving the registration request, the crossover LSR will redirect the LSP towards the new FA. After the new LSP from the crossover LSR to the mobile node through the new FA is established, the crossover LSR will change its label table and redirect the LSP, with the MN's home address as FEC, to the new FA. The message exchange occurring during the handoff is described in figure 3.8.

3.3.2.1 Redirecting Request Message Processing Rules

After receiving a re-directing request from a CR tool, the CR_M RSVP daemon searches the session block whose destination address matches the MH's home address. If this Session is found, the CR_M checks the crossover state flag in the PATH sate of the Session.

If the CR_M is in the IDLE state and the explicit route list is different from the requested one, the CR_M then updates the explicit route list in the PSB and queries a route to the first node in the explicit route list. If the route is found, it sets the crossover state flag as Request - in - Processing, i.e. the REQUESTED state; turns on the Path_Refresh_Needed flag of the current PSB; updates the PSB after the route computation and waits for a synchronous reply; refreshes the PATH message. If a route is not found, it sets the error code in the re-directing response as "error in route".

If the CR_M is not in the IDLE state, or the explicit route list is the same as the requested one, the CR_M does not need to trigger a new re-directing process. If the re-directing request is still in processing, either the CR_M is in the state of REQUESTED (but not setup yet) or SETUP (but still waiting for confirmation), the CR_M rejects the new request.

Figure 3.9 shows the process of cross over request.

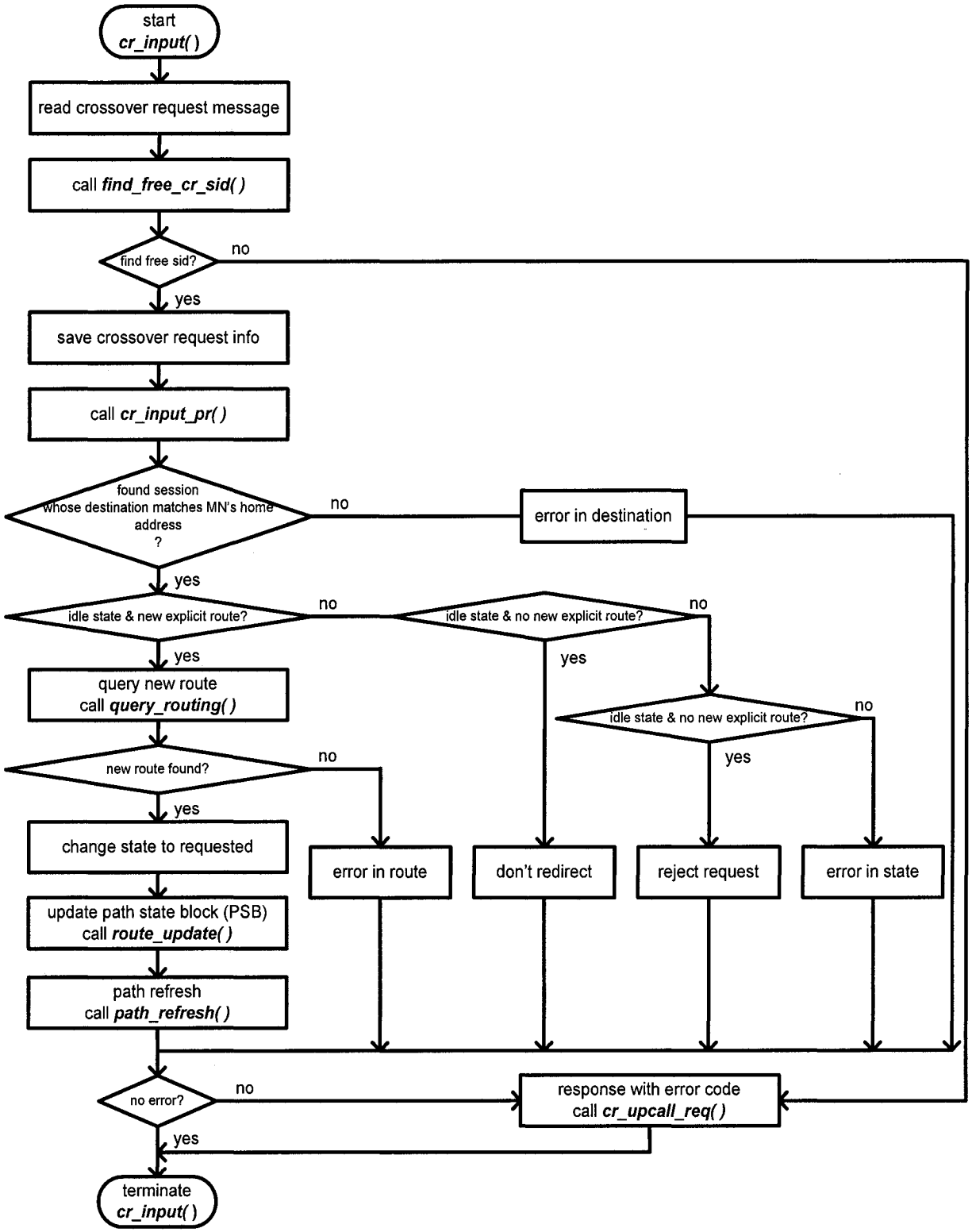

Figure 3.9 Flow chart of crossover request processing

3.3.2.2 PATH Message Processing Rules

If there is a matching PSB, the CR_M checks whether it is the SETUP state. If its state is SETUP, it sets the state as IDLE and sends a PATHERR message with Explicit Route.

If its state is REQUESTED, the CR_M drops the PATH message. Otherwise, it does normal PATH processing with the ERO update if necessary.

Figure 3.10 shows a flow chart of PATH message processing.

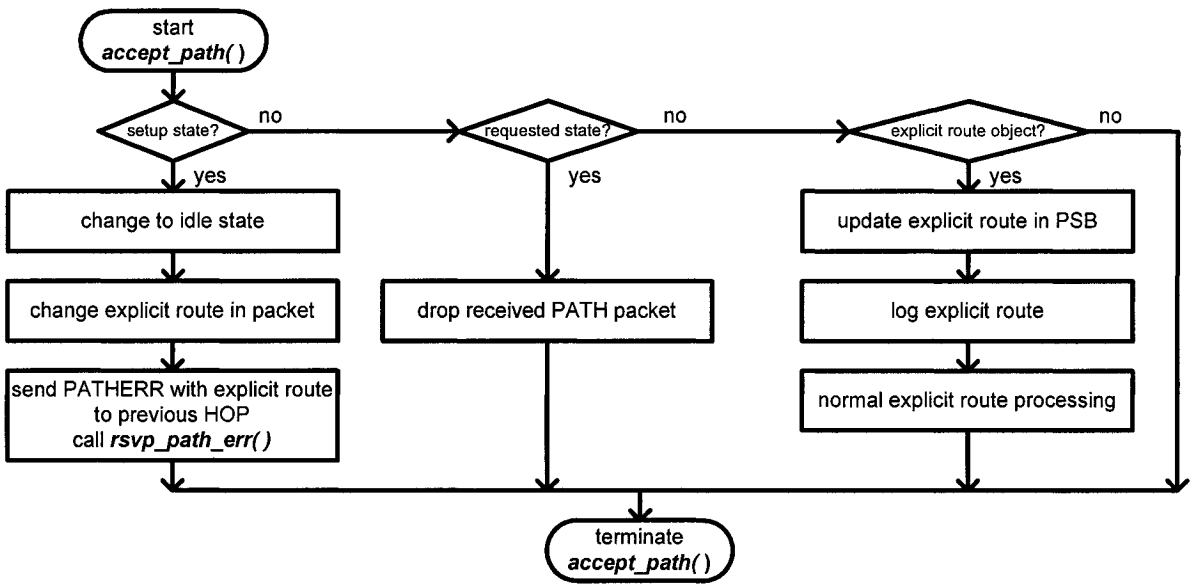

Figure 3.10 Flow chart of PATH message processing

3.3.2.3 RESV Message Processing Rules

If the CR_M is in the REQUESTED state and a RESV message is received from the current NHOP (next hop), the CR_M resets the time-to-die of the RESV state immediately, tears

down the old path, sets its state as SETUP, and responds to the CR that the path is established successfully.

If its state is REQUESTED and a RESV message is received from a node other than NHOP, the CR_M drops the RESV message.

If its state is other than REQUESTED, the CR_M does normal RESV message processing.

Figure 3.11 shows a flow chart of RESV message processing

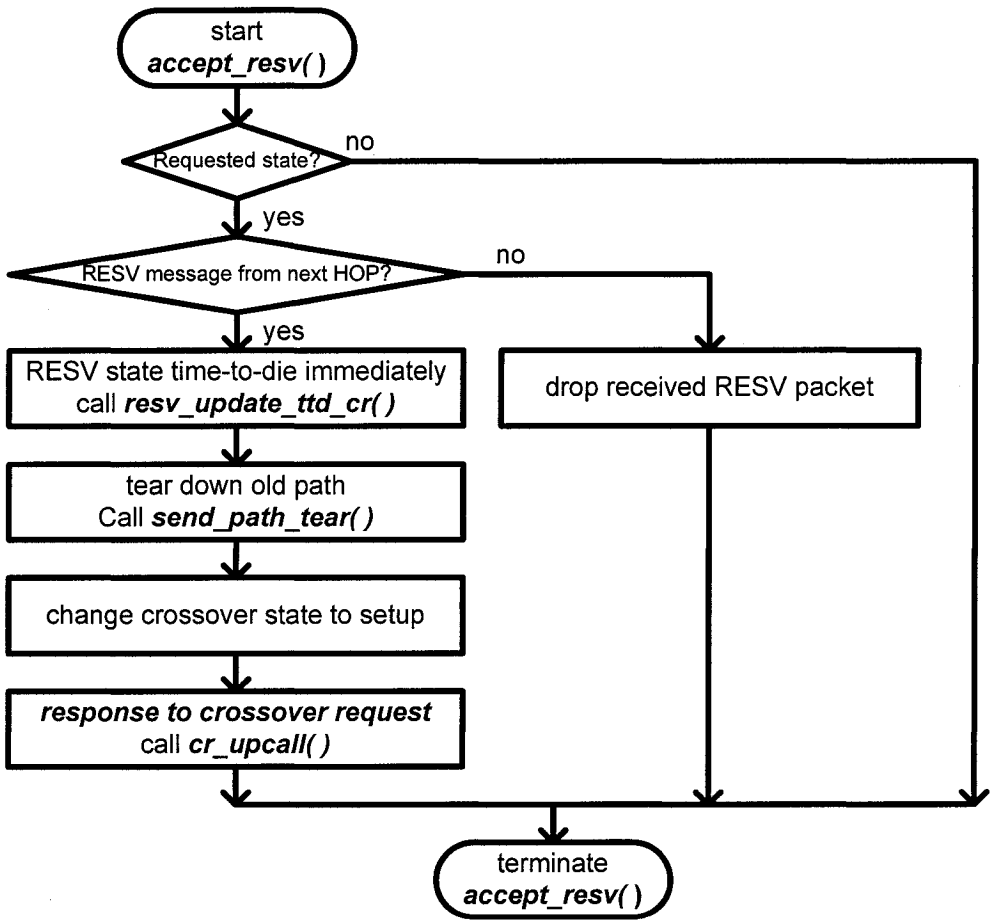

Figure 3.11 Flow chart of RESV message processing

3.3.2.4 PATHERR Message Processing Rules

For the crossover node, if the CR_M is in the state of REQUESTED and a PATHERR message comes from the next hop, the CR_M relies to the CR tool the information that the request is not successful; if its state is REQUESTED and there is not a PATHERR message from the next hop, the CR_M drops the PATHERR message. If its state is other than REQUESTED, the CR_M performs normal PATHERR message processing.

For the sender node, if the error code of the arriving PATHERR message indicates a routing problem (24), and the value is the self-defined RSVP_Err_ER_MD (11), the CR_M updates the ERO in the api_table and PSB, and refreshes the PATH message with the new explicit route. Figure 3.12 show a flow chart of PATHERR message processing

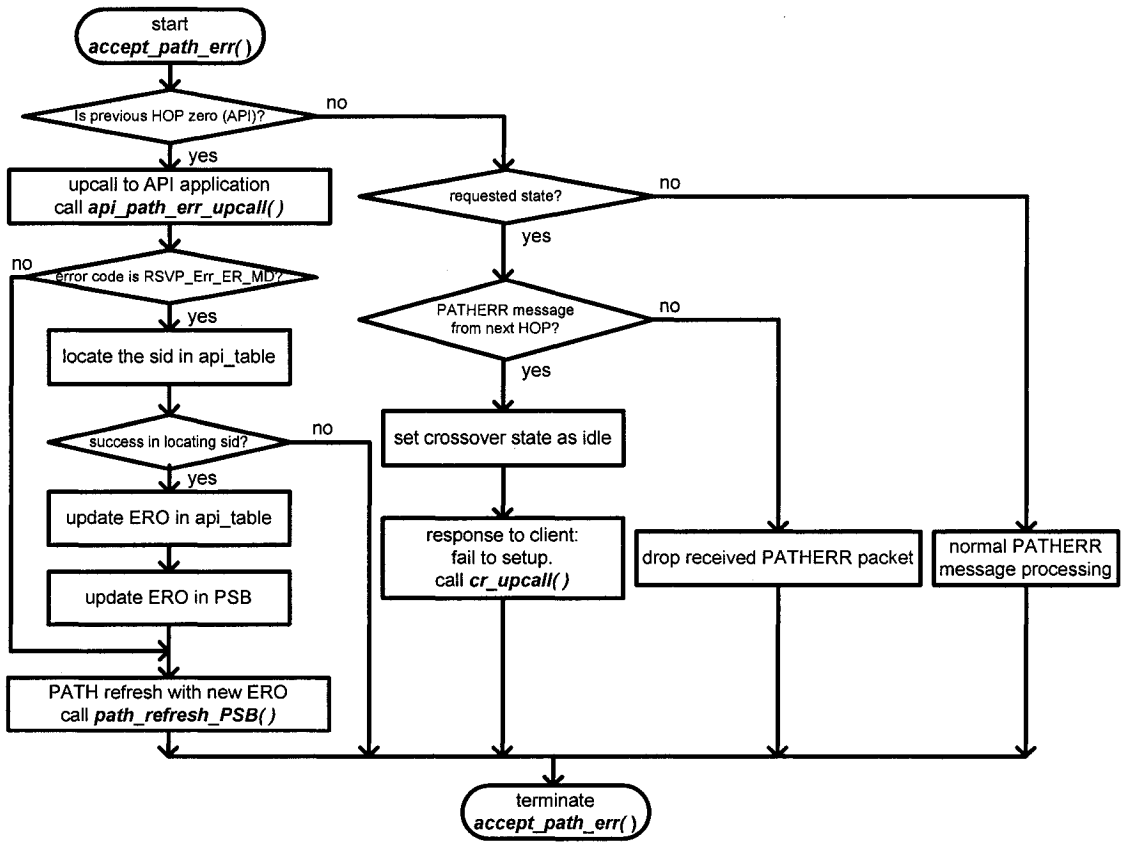

Figure 3.12 Flow chart of PATHERR message processing

3.4 IntServ over MM-MPLS: Bandwidth Reservation

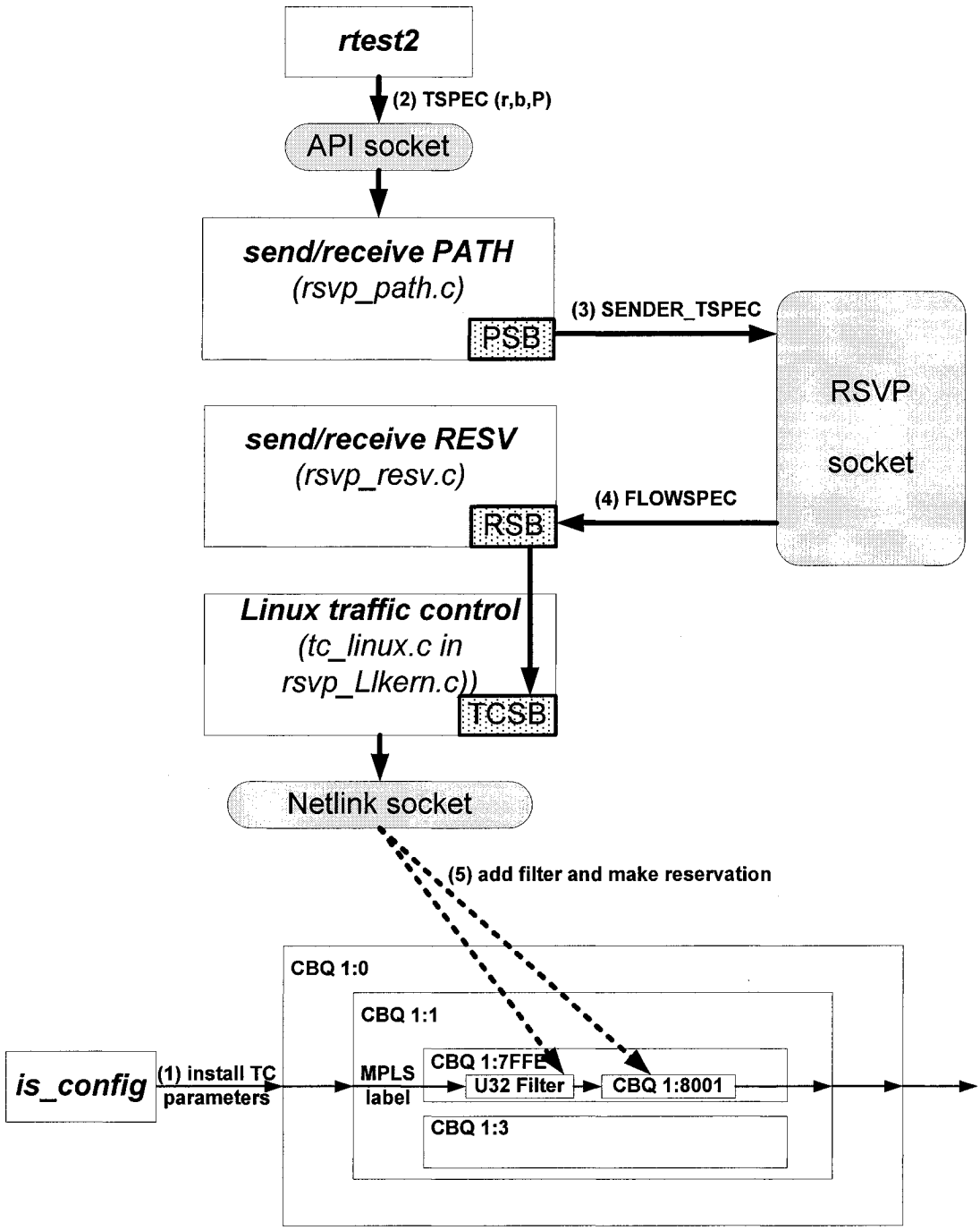

Figure 3.13 An example of IntServ Bandwidth Reservation over MM-MPLS

The RSVP daemon and related TC components are shown in figure 3.13. The procedure of IntServ bandwidth reservation is described as follows:

1) The script "is_config" will install the root queuing discipline CBQ 1:0, class 1:1, and IntServ class 1:7FFE at all the MM-MPLS nodes.
2) The RAPI application "rtest2", at the Ingress LER, provides the QoS requirement of the flow in the format of IntServ TSPEC.
3) The SnderTspec is passed to the RSVP nodes along the path within the PATH message, and stored at the PSB database of these nodes.
4) The reservation is initiated at the receiver RSVP node. The reservation information FlowSpec is carried in the RESV message, and stored in the RSB of the RSVP nodes.
5) Using the interface functions: TC_AddLabelFilter and TC_AddFlowSpec, RSVP daemon makes reservation at the TC of the Linux Kernel. There are a CBQ queuing discipline of class 1:8001 and a filter rule that classifies a specific label to this class.
6) Repeat step 3) to 5) when redirecting the LSP.

Chapter 4 Performance Evaluation

This chapter discusses the process of testing the new functionalities developed for the the MM-MPLS implementation and reports a set of performance evaluation results. The functionality tests are summarized in section 4.1. Performance results are given in section 4.2.

4.1 Functionality Tests

This section discusses several representative test scenarios that were conducted in order to prove that the developed modules operate according to the specification derived from the functionality of MM-MPLS.

4.1.1 Registration Message Process

Registration messages exchange information between a mobile node, a foreign agent, and the home agent. Registration creates or modifies a mobility binding at the home agent, associating the mobile node's home address with its care-of address for the specified lifetime.

MM-MPLS enables the following registration processes for a mobile node:

(1) When a mobile node (MN) moves into a MM-MPLS network, it sends Registration Request message in order to register with its home agent. If the MN is away from its home domain, it will be assigned a Care-of-address by the address of the gateway router between the foreign domain which it is living in and its home domain (the gateway router is the FDA in our case). Anytime when the Registration Request message is send to reach the HA, the HA will send a Registration Reply to the MN with lifetime 300 sec.

(2) When the MN moves into a new foreign agent, it sends Registration Request through the new FA and this Registration message will be intercepted at a CR. In this case, the CR will generate a Registration Reply message with the lifetime currently left at that moment.

(3) When the MN keeps staying at a foreign domain, it will refresh the registration message periodically, in order to keep the connection with its home agent. The refreshing time is calculated as half of the lifetime of last Registration Reply message. For example, if the last Registration Reply is send by HA with the lifetime 300 sec, then the Registration Request will be send right after 150 sec (1/2 of 300 sec) from the MN receives the last Registration Reply. If the Registration Reply is come from the CR with the lifetime 68 sec, which means the lifetime has been changed by CR due to the MN's handover, the next Registration Request will be send after 34 sec (1/2 of 68 sec) from the MN receives this Registration Reply.

Test case: Registration process during handoff from home network to foreign network and from one subnetwork network to another subnetwork network of the same domain.

Scenario Description: At the beginning, the MN has connection to the HA. After time t1, the MN connects to FA1 and the connection with the HA expires. After time t2, the MN connects to the FA2 and the connection with the FA1 expires. Figure 4.1 shows the topological network for this test scenario. As a result of Dan's thesis [7], the hierarchical Dynamics - HUT Mobile IP has fast, soft, policy-based, and mobile node controlled handoffs in wireless local area networks. In our test, we use policy-based mobile node controlled hard handoff. In practice, the MN received the agent advertisements, but it completed a location update only when requested by a test script with specific policy as "Newest-FA" + "Newest-ADV"[7].

[7] Mobility agent selection is based on priority comparison. Node selector is a simple component that compares the priority values. It uses certain rules to make selection decisions. The set of rules that affects the node selection process is called a policy. There are currently four different policies for the node selector: *eager-switching*, *newest-FA*, *early-expire* and *newest-ADV*.

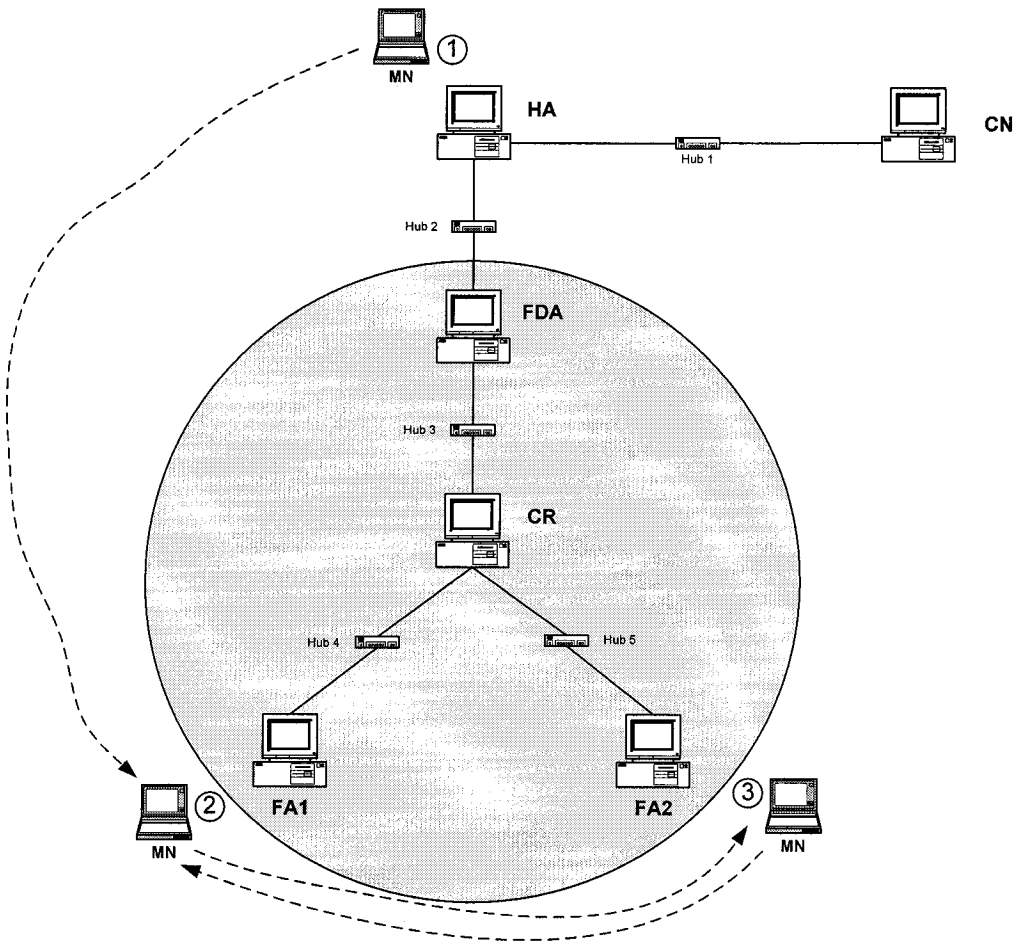

Figure 4.1 Testing scenario

Observation: When starting (at point 1 on Figure 4.1), the MN sends a registration request to the HA. The HA responds a registration reply message to the MN through the wireless connection. After t1, the MN attaches to FA1. It sends a registration request to FA1. The CR and the FDA propagate the registration request message to the HA. Then, the MN is registered in its home network with a COA (care of address) as the address of FDA, which is the gateway foreign domain agent of the MN's accessed foreign network. The HA reroutes the registration reply message through the reverse path, e.g. HA→FDA→CR→FA2→MN. After t2, the MN attaches to FA2. It sends registration request to FA2. Since FA1 and FA2 are in the same foreign domain, the CR will intercept this registration request and responds with a registration reply message. This means this registration message is terminated at the CR. The FDA and the HA will never notice the MN's movement in the local domain. A

more detailed example of mobile node registration regarding TTL (time-to-life) is shown in Appendix B.1.

Conclusion: Works.

4.1.2 MPLS Traffic Transmission

Whenever traffic is transported to the mobile node through the MM-MPLS domain, it should be mapped onto and be transported through the MPLS layer instead of IP layer. When a packet enters into the MM-MPLS domain, first, the HA, which is the ingress node, will label it for transportation through the LSP from HA to the FDA. Then, the FDA, acting as egress router for the HA to FDA LSP and ingress router for the FDA to FA LSP, will first remove the label and move the packet to the IP layer. Then the FDA will move the packet back to the MPLS layer and label it for transportation through the LSP from FDA to FA. The packet passes through CR, transports to the lower FA. The lowest FA is the egress router of the LSP from FDA to FA. It will strip off the label and put the packet onto IP layer. Then the packet and is sent to the MN through normal IP routing.

Test case: Traffic transmission through MM-MPLS domain.

Scenario Description: Sending traffic from the CN to the MN when the MN registered through a foreign domain of the MM-MPLS network.

Observation: In order to confirm that this procedure was followed by the system, we send traffic from CN to the MN, with the mobile node been served by FA1. We then monitored several packets as they were passing through the system, we present below the "structure" of one of these packets, at the entrance and exit of various nodes.

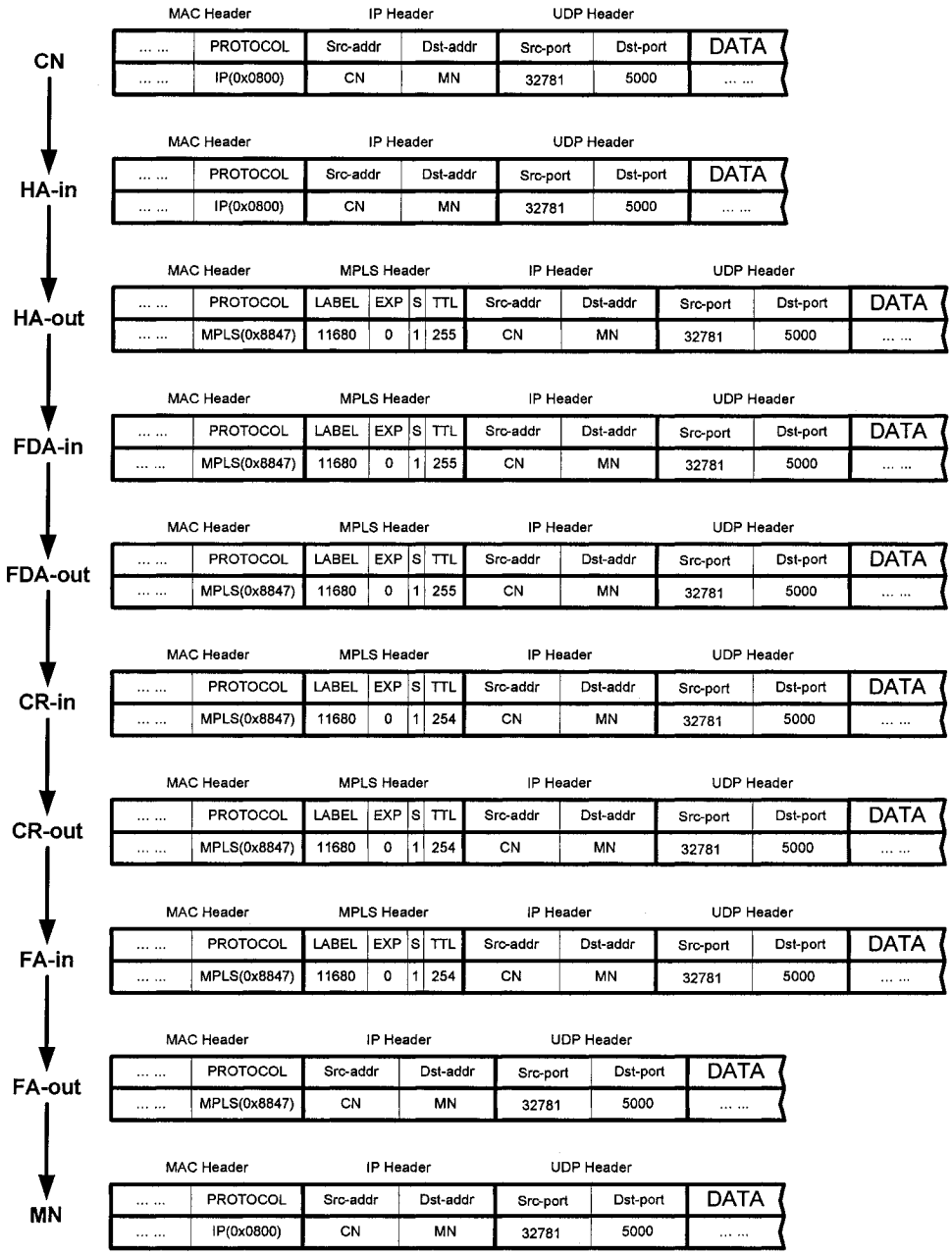

Figure 4.2 Structure of a packet on each node when transferred from CN to MN

Conclusion: Works.

4.1.3 Function of Crossover Module

In the MM-MPLS network, a crossover router is a foreign agent with at least two connections to the lower foreign agents. It is the intermediate LSR at the intersection of the paths from FDA to the old FA and to the new FA. When mobile node roams in the local domain, the CR will take charge of handling the registration message without informing the upper agents about the movement. Once the CR gets a registration request message from a new FA, located in the same domain, it will send a registration reply to that FA and then redirect the LSP PATH to the new registered FA by replacing the ERO with the address of the newly registered FA. When it gets the RESV message from the new FA, it tears down the old PATH from CR to the old FA by sending a PATHTEAR message to the old FA. The CR will later receive a PATH message from FDA to the old FA. Since the ERO on CR has been changed to the new FA, the CR will send a PATHERROR message with the new ERO up to FDA, in order to inform the FDA to redirect the LSP to the new FA. When the CR gets a registration request from a local handoff event, it will manage to send back a registration reply to the lower FA. If the registration request is not from a roaming process, but a regular message sent in order to refresh the connection, the CR will forward that message to the upper agent and wait for the Reply from the upper agent.

Test case: PATH redirection by the CR when the MN moves within the same foreign domain.

Scenario Description: When the MN handoffs by changing the point of attachment to the network from FA1 to FA2, the intermediate router CR, which is the intersection between the two LSP paths, LSP1 (FDA→FA1) and LSP2 (FDA→FA2), will manage the registration process and redirect the PATH from the FA1 to the FA2.

Observation: Figure 4.3 shows the messages involved in the handoff process by monitoring with Ethereal. Table 4.1 gives an example of the messages we captured using Ethereal on CR. From messages No.21 and No.22, we can see that the registration message is handled by CR when the mobile node handoffs. When CR gets this registration request, it will directly send

back a registration reply to the lower FA. If the registration request is not produced by a roaming process, the CR will forward that message to the upper agent and wait for the reply from the upper agent. This process is occurring through messages No.42 to 45. The new PATH is established as soon as the CR manages the registration process from the handoffs. Messages No. 23 and No. 24 indicate the establishment of the new path. When the CR intercepts the PATH message from the FDA to the old FA (No. 30), it sends a PATHERROR message (No. 31) to FDA with the new FA's address as the ERO. Then, the FDA will send a PATH message (No. 32) to the new FA with the address that retrieves from ERO.

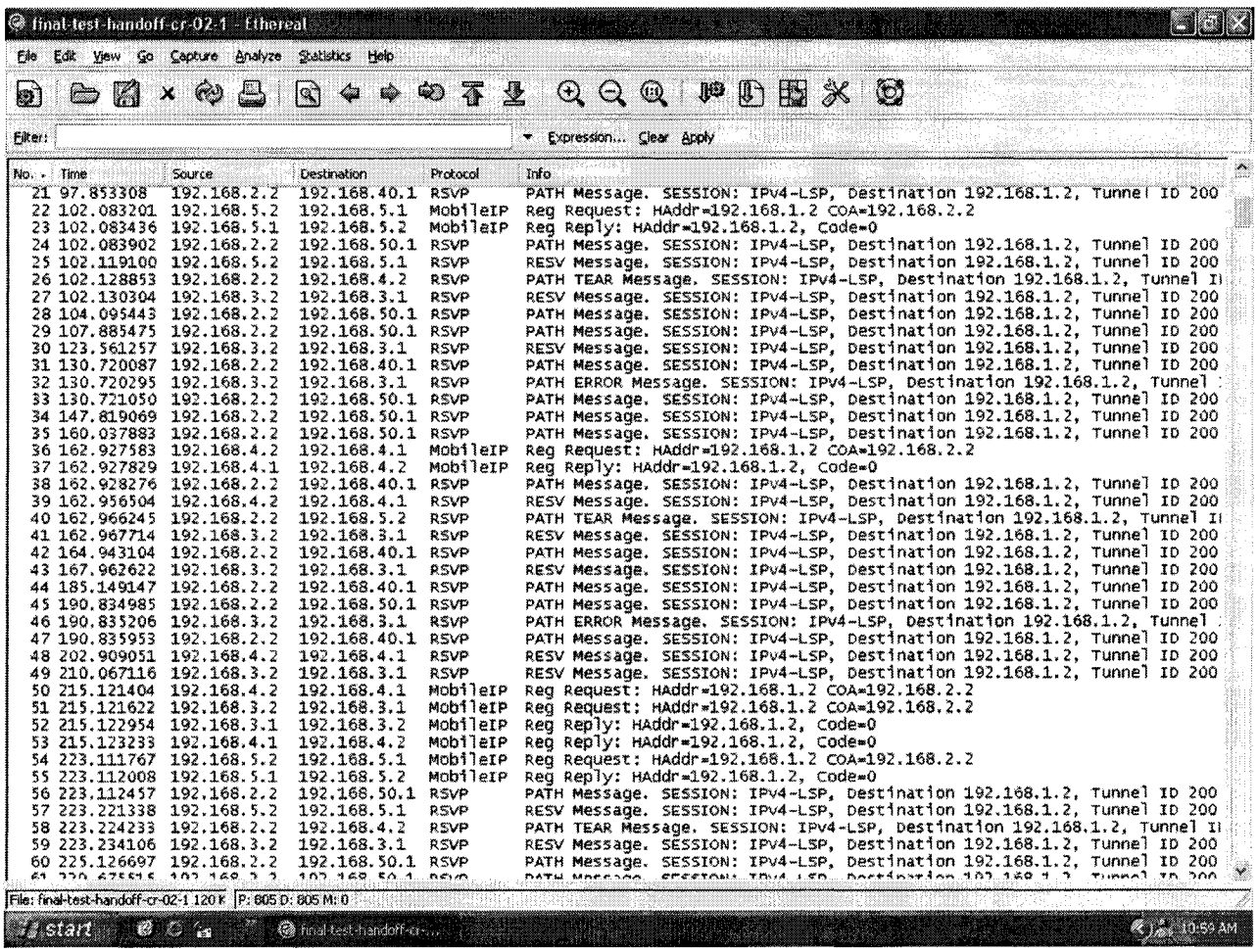

Figure 4.3 Messages monitored by Ethereal during handoff on CR

No.	Time	Source	Destination	Protocol	Info	
1	**0**	**o-FA**	**CR**	**MobileIP**	**Reg**	**Request:**
2	**0.010004**	**CR**	**FDA**	**MobileIP**	**Reg**	**Request:**
3	0.688736	FDA	o-FA	RSVP	PATH	Message.
4	0.689212	FDA	o-FA	RSVP	PATH	Message.
5	0.716068	o-FA	CR	RSVP	RESV	Message.
6	0.71918	CR	FDA	RSVP	RESV	Message.
7	**1.117612**	**FDA**	**CR**	**MobileIP**	**Reg**	**Reply:**
8	**1.117968**	**CR**	**o-FA**	**MobileIP**	**Reg**	**Reply:**
9	30.694483	FDA	o-FA	RSVP	PATH	Message.
10	30.719855	CR	FDA	RSVP	RESV	Message.
11	30.780267	FDA	o-FA	RSVP	PATH	Message.
12	54.564132	o-FA	CR	RSVP	RESV	Message.
13	57.551891	CR	FDA	RSVP	RESV	Message.
14	69.287658	FDA	o-FA	RSVP	PATH	Message.
15	70.900529	FDA	o-FA	RSVP	PATH	Message.
16	90.225475	FDA	o-FA	RSVP	PATH	Message.
17	96.047983	CR	FDA	RSVP	RESV	Message.
18	103.383217	o-FA	CR	RSVP	RESV	Message.
19	109.855604	FDA	o-FA	RSVP	PATH	Message.
20	113.569775	FDA	o-FA	RSVP	PATH	Message.
21	**123.456343**	**n-FA**	**CR**	**MobileIP**	**Reg**	**Request:**
22	**123.456608**	**CR**	**n-FA**	**MobileIP**	**Reg**	**Reply:**
23	123.456963	FDA	n-FA	RSVP	PATH	Message.
24	123.534024	n-FA	CR	RSVP	RESV	Message.
25	**123.545084**	**FDA**	**o-FA**	**RSVP**	**PATH**	**TEAR**
26	123.554133	CR	FDA	RSVP	RESV	Message.
27	125.481593	FDA	n-FA	RSVP	PATH	Message.
28	130.805802	FDA	n-FA	RSVP	PATH	Message.
29	138.422994	CR	FDA	RSVP	RESV	Message.
30	139.509911	FDA	o-FA	RSVP	PATH	Message.
31	**139.513253**	**CR**	**FDA**	**RSVP**	**PATH**	**ERROR**
32	139.514364	FDA	n-FA	RSVP	PATH	Message.
33	155.867324	FDA	n-FA	RSVP	PATH	Message.
34	163.859139	n-FA	CR	RSVP	RESV	Message.
35	176.47474	CR	FDA	RSVP	RESV	Message.
36	179.200331	FDA	n-FA	RSVP	PATH	Message.
37	182.010558	FDA	n-FA	RSVP	PATH	Message.
38	197.517514	FDA	n-FA	RSVP	PATH	Message.
39	199.631707	n-FA	CR	RSVP	RESV	Message.
40	208.097777	CR	FDA	RSVP	RESV	Message.
41	208.525519	FDA	n-FA	RSVP	PATH	Message.
42	**212.014434**	**n-FA**	**CR**	**MobileIP**	**Reg**	**Request:**
43	**212.014695**	**CR**	**FDA**	**MobileIP**	**Reg**	**Request:**
44	**212.016748**	**FDA**	**CR**	**MobileIP**	**Reg**	**Reply:**
45	**212.017051**	**CR**	**n-FA**	**MobileIP**	**Reg**	**Reply:**

Table 4.1 Example of messages by Ethereal on CR

Conclusion: The protocol functions according to expectations.

4.2 Performance Analysis

4.2.1 Forwarding Delay

We have tested the forwarding delay for each router in the MM-MPLS test network, comparing it with the hierarchical mobile IP, which was running under the same conditions of our testbed. The following tables and figures show the test results. All the test computers are using buffer size of 100 packets.

From figure 4.4, an example of collected data of UDP packet forwarding delay on the CR, without running either Mobile IP or RSVP-TE, the reader can observe that there are some unusual high spikes of delay, which seem to appear periodically. These spikes are due to the fact that the computer resources are been taken by background processes of the operating system. The frequency and functionality of these processes is controlled by several parameters, configured in */proc/sys/net* in Linux. Such functions are refreshing to resource maintenance activities, such as recovery of unused memory. In table 4.2, we provide the statistical information by process all collected packets.

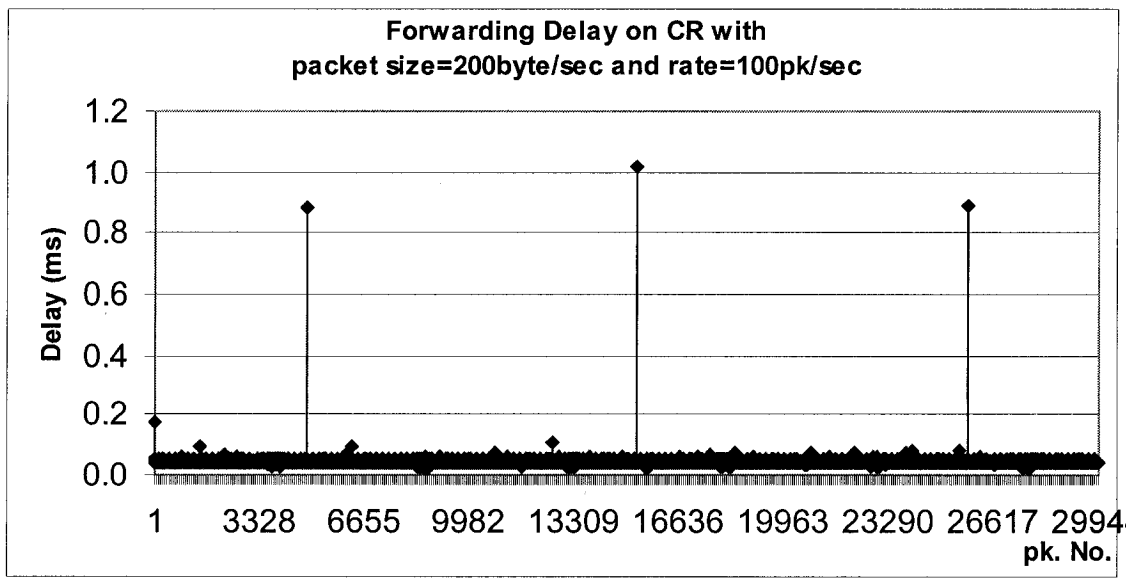

Figure 4.4-a an example of UDP packet forwarding delay at CR

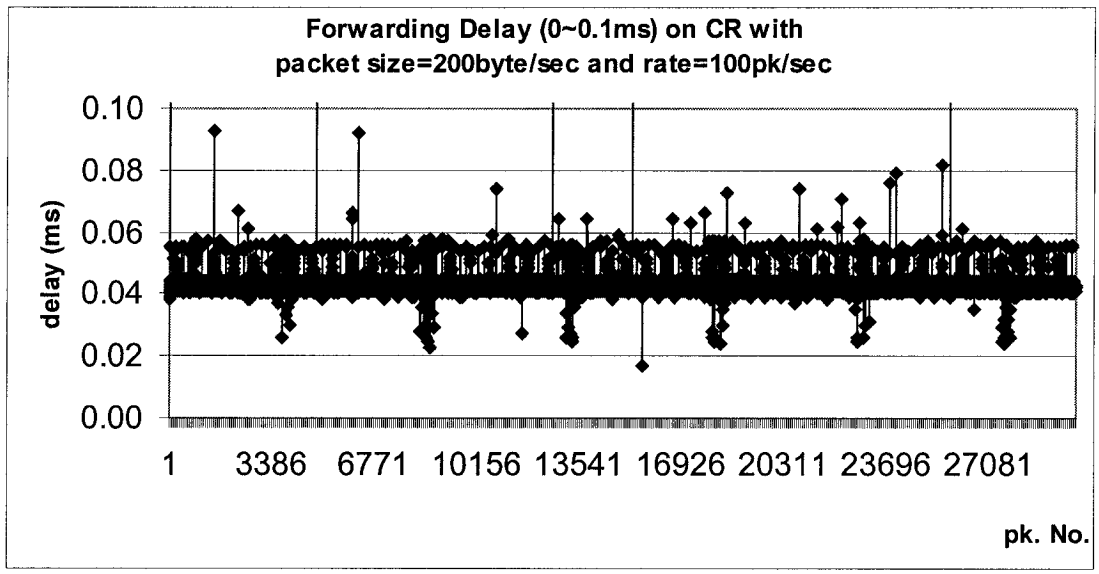

Figure 4.4-b UDP packet forwarding delay (within 0~0.1ms) from Figure 4.4-a

Mean	0.042658
Standard Deviation	0.007797
Count	30000
Confidence Level (95.0%)	0.000088

Table 4.2 Statistic result for the data on Figure 4.4

In table 4.3, we have eliminated those unusual high delays (which we calculated, represent 4.39% of the total sample of measured delay values). It is evident that this gives a considerable reduction of the standard deviation.

Mean	0.042380
Standard Deviation	0.000759
Count	28682
Confidence Level (95.0%)	0.000009

Table 4.3 Statistic result for the filtered data on Figure 4.4

Such, on some of the following delay test, we give the statistical result on bother raw data set and filtered data set.

Table 4.4 and Table 4.5 show the data of packet forwarding delay on HA computer for MM-MPLS and H-MIP.

	raw data			filtered data		
pk. Size	Mean	S.D.	S.S.	Mean	S.D.	S.S.
200	0.061785	0.006094	100000	0.060806	0.000835	93595
400	0.078857	0.030469	100000	0.078128	0.000818	94952
600	0.096900	0.004407	100000	0.096272	0.000808	94320
800	0.115202	0.001598	100000	0.114574	0.000950	95388
1000	0.132553	0.004642	100000	0.131967	0.000833	96081
1200	0.150904	0.001953	100000	0.150214	0.000886	93756
1400	0.169149	0.004786	100000	0.168499	0.001110	95325

(S.D.: Standard Deviation, S.S.: Sample Size)

Table 4.4 Forwarding delay at HA with MM-MPLS

	raw data			filtered data		
pk. Size	Mean	S.D.	Sample	Mean	S.D.	Sample
200	0.066712	0.005834	100000	0.065932	0.000812	95391
400	0.083422	0.004886	100000	0.082844	0.000787	96541
600	0.101666	0.004689	100000	0.101114	0.000778	96241
800	0.119073	0.004495	100000	0.118556	0.000828	97047
1000	0.136399	0.004772	100000	0.135837	0.000822	96349
1200	0.154386	0.004497	100000	0.153837	0.000872	96329
1400	0.171688	0.004859	100000	0.171089	0.000830	95537

Table 4.5 Forwarding delay at HA with H-MIP

Figure 4.5 shows the mean of the forwarding delay measured at HA for both protocols. We can see that the forwarding delay increase linearly with the packet size for both protocols, and the delay with MM-MPLS is smaller than that of D-MIP. The reason is that in MM-MPLS the packets travel through the HA from IP layer mapped to the MPLS layer. In D-MIP, they remain at the IP layer and become forwarded with corresponding IP routing. Similar results were observed for all the routers in the MM-MPLS network (i.e. FDA, CR and FA).

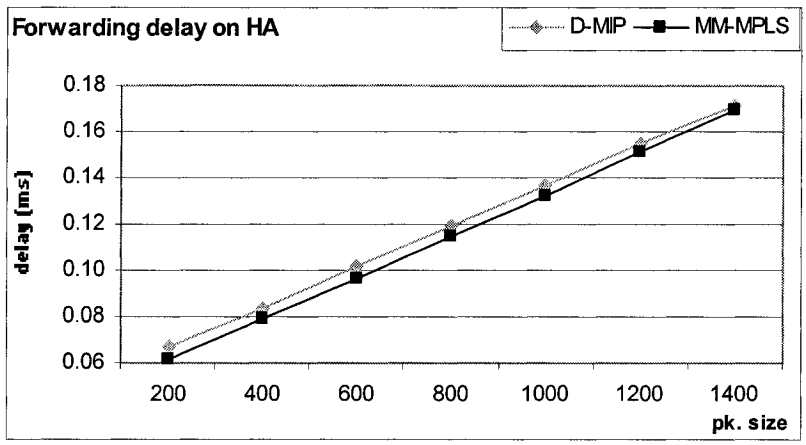

Figure 4.5 Forwarding Delay at HA

MM-MPLS				D-MIP			
pk. Size	Mean	S.D.	C.I.(95.0%)	pk. Size	Mean	S.D.	C.I.(95.0%)
200	0.056599	0.006004	0.000215	**200**	0.063467	0.006416	0.000230
400	0.082083	0.006024	0.000216	**400**	0.088725	0.006678	0.000239
600	0.107491	0.006174	0.000221	**600**	0.113044	0.006374	0.000228
800	0.136694	0.006695	0.000240	**800**	0.139792	0.006734	0.000241
1000	0.159428	0.006120	0.000219	**1000**	0.163988	0.006178	0.000221
1200	0.184593	0.006260	0.000224	**1200**	0.188287	0.006455	0.000231
1400	0.210409	0.006515	0.000233	**1400**	0.215916	0.006690	0.000240

(C.I.: Confidence Interval)

Table 4.6 Forwarding delay at FDA (sample size=60000)

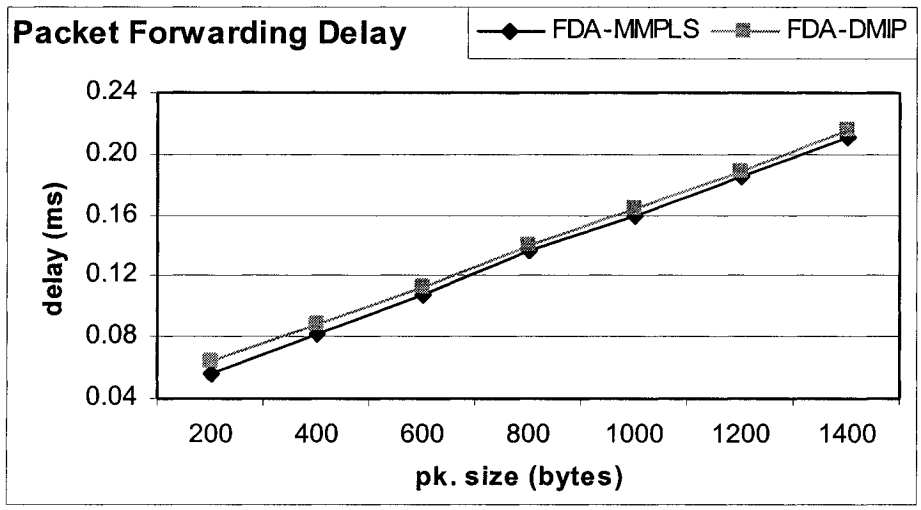

Figure 4.6 Forwarding Delay at FDA

At the FDA, the advantage of MM-MPLS over D-MIP seems to be less. The reason is that FDA performs more complex functionalities for MM-MPLS. I has to strip the label from the incoming packet and add a new label to the LSP between FDA and FA.

Similarly, the forwarding delay at CR is shown in Table 4.7, Table 4.8 and Figure 4.7. The advantage of MM-MPLS is obvious. Since in a real network such router expected to have several such routers within a path, it is understood that the advantage in end-to-end delay will increase further when MM-MPLS is used.

	raw data			filtered data		
pk. Size	Mean	S.D.	Sample	Mean	S.D.	Sample
200	0.040242	0.091400	100000	0.039072	0.000751	91726
400	0.059786	0.088701	100000	0.058665	0.000800	93939
600	0.078871	0.075988	100000	0.077853	0.000724	91251
800	0.098724	0.072299	100000	0.097905	0.000757	89904
1000	0.117170	0.095337	100000	0.115891	0.000760	90265
1200	0.136252	0.068068	100000	0.135621	0.000886	94033
1400	0.156032	0.090559	100000	0.154829	0.000918	95676

Table 4.7 Forwarding delay at CR with MM-MPLS

	raw data			filtered data		
pk. Size	Mean	S.D.	Sample	Mean	S.D.	Sample
200	0.050550	0.053537	100000	0.050005	0.000889	95594
400	0.071108	0.048619	100000	0.07054	0.00092	95119
600	0.089412	0.050996	100000	0.088734	0.000872	95483
800	0.108510	0.042522	100000	0.107877	0.000923	95724
1000	0.127922	0.093571	100000	0.126903	0.00102	95102
1200	0.147887	0.04802	100000	0.147256	0.001035	95054
1400	0.166922	0.060873	100000	0.166138	0.001114	95328

Table 4.8 Forwarding delay at CR with H-MIP

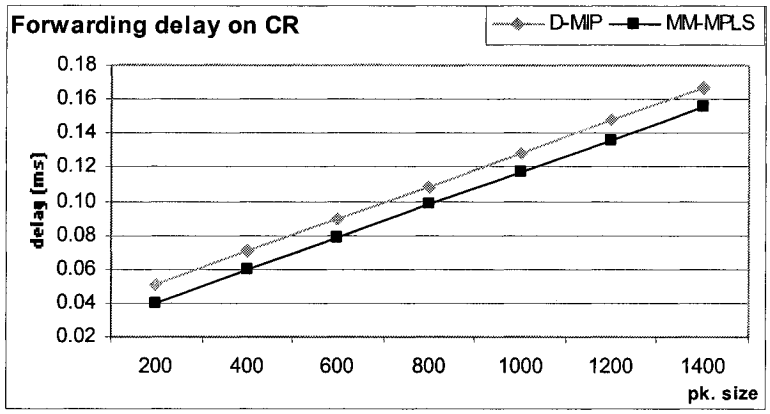

Figure 4.7 Forwarding Delay at CR

Table 4.9 and Figure 4.8 provide result for FA. The superiority of MM-MPLS is evident once more. The curves in this case are "wavy", due to the higher standard deviation. We believe this is due to the processing associated with the use of the wireless LAN (IEEE802.11b) NIC.

MM-MPLS				D-MIP			
pk. Size	Mean	S.D.	C.I.(95.0%)	pk. Size	Mean	S.D.	C.I.(95.0%)
200	0.015036	0.009436	0.000338	**200**	0.017558	0.006004	0.000215
400	0.015953	0.010399	0.000372	**400**	0.017600	0.002977	0.000107
600	0.016161	0.015749	0.000564	**600**	0.017326	0.003488	0.000125
800	0.016694	0.011363	0.000407	**800**	0.017933	0.007648	0.000274
1000	0.017152	0.019356	0.000693	**1000**	0.018869	0.003000	0.000107
1200	0.018511	0.020710	0.000741	**1200**	0.020128	0.003045	0.000109
1400	0.018078	0.010274	0.000368	**1400**	0.019390	0.005043	0.000181

Table 4.9 Forwarding delay at FA (sample size=60000)

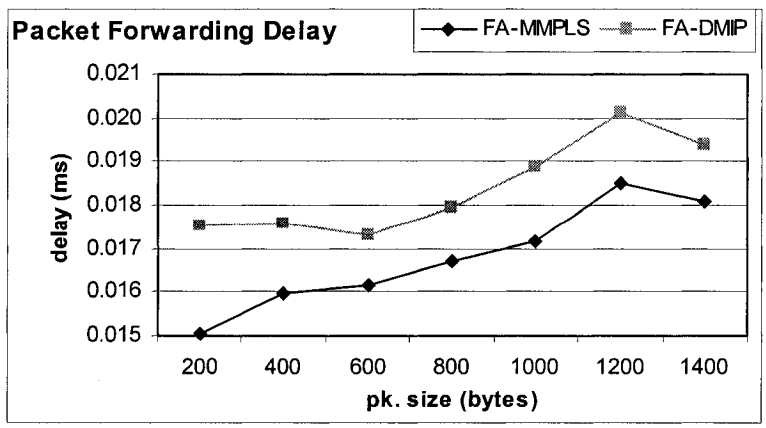

Figure 4.8 Forwarding Delay at FA

4.2.2 End-to-End Delay

From the previous tests, all routers within the network provide lower packet forwarding delay when MM-MPLS is used instead of Dynamics Mobile IP (see Figure 4.9). Since the same network has been used in the testing of both protocols, it is reasonable to claim that MM-MPLS provides lower end-to-end delay as compared.

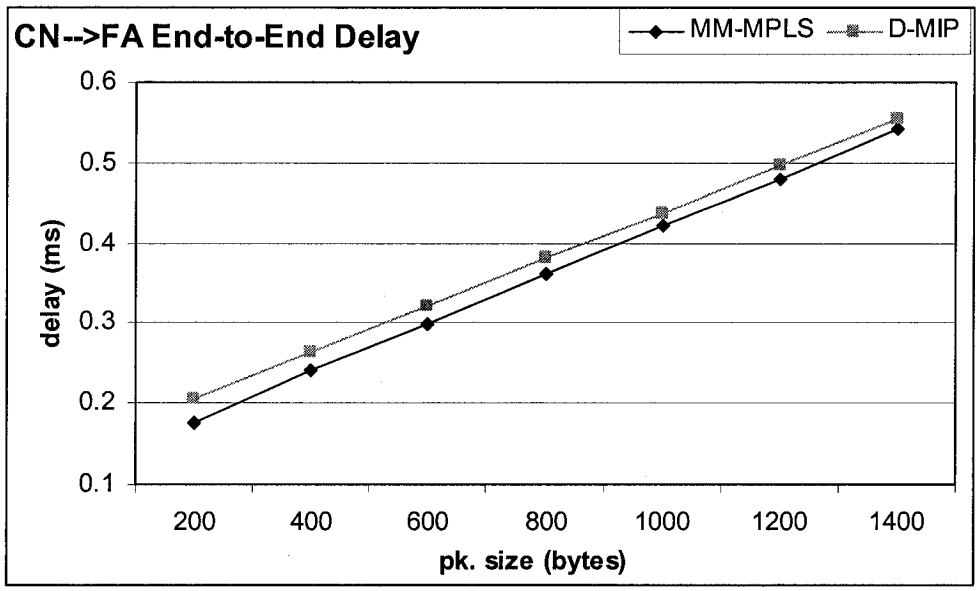

Figure 4.9 End-to-End Delay from the CN to the FA

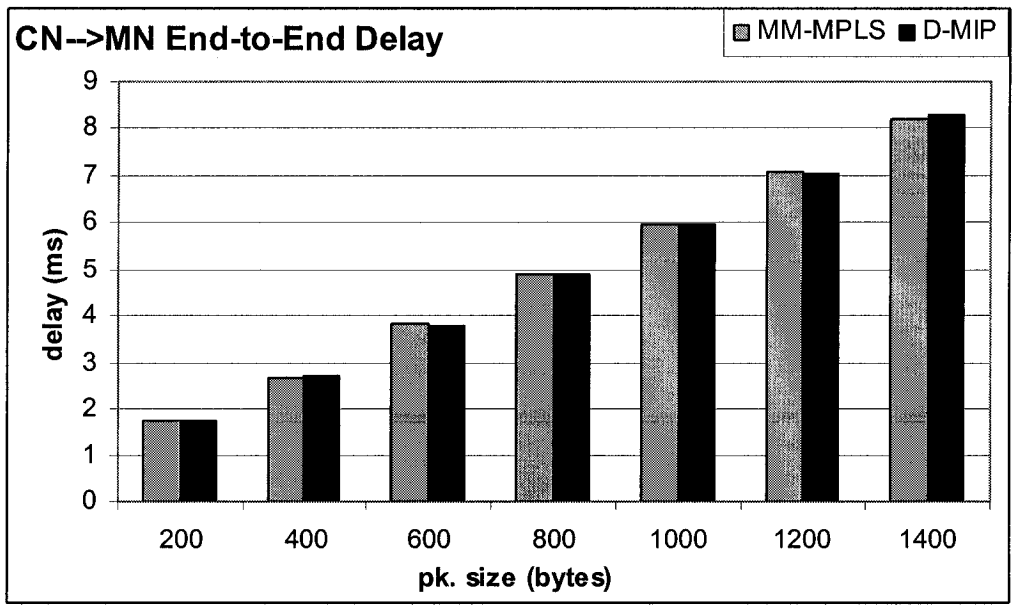

Figure 4.10 End-to-End Delay from the CN to the MN

The end-to-end delay from CN to MN includes the wireless radio transmission. Multi-path and other impairments generate a more "noisy" result, and reduce the advantage of MM-MPLS. However, the result remains positive for MM-MPLS. Figure 4.10 shows an example of one test result of end-to-end delay from CN to the MN.

4.2.3 Policy-Based Handoff

As we described in Chapter 2.4, the HUT dynamics system provides policy-based handoff, where one can specify a high level system policy for handoff management of the mobile node. The four types of policies being used in HUT dynamics are: Newest-ADV, Eager-switching, Newest-FA, and Early-expire. Table 4.10 indicates the Linux configurations for the handoff policies.

Policy Configuration	Newest-ADV	Eager-switching	Newest-FA	Early-expire
0				
1				√
2			√	
3			√	√
4		√		
5		√		√
6		√	√	
7		√	√	√
8	√			
9	√			√
10	√		√	
11	√		√	√
12	√	√		
13	√	√		√
14	√	√	√	
15	√	√	√	√

Table 4.10 Configuration of handoff policy in Linux

In order to find the policy which can bring the lowest handoff delay we made a group of test to see the influence of the policy to the handoff process. Figure 4.11 shows the test result with condition AI-FA1=1 second, AI-FA2=5 seconds, rate=120 bps, 1 handoff/5 minutes. From several test, the policy 12 is approved to be the best handoff policy for the current MM-MPLS testbed.

Handoff is a procedure executed when a mobile node moves from the coverage area of one access point to the coverage area of another access point. The handoff process involves a sequence of messages being exchanged between the mobile node and the participating access points. This sequence of messages can be divided into three types: **probe**, **authentication** and **association**.

1. **Probe messages**: Once the Mobile Node decides to look for other access points, the probing process starts. The MN starts sending out probe requests and then processes received probe responses, based on the active scanning algorithm explained above. The time involved in this probing process is called ***probe delay or scanning delay.***
2. **Authentication messages**: Once the MN decides to join an access point, authentication messages are exchanged between the MN and the selected access point. The time consumed by this process is called ***authentication delay***.
3. **Reassociation messages**: After a successful authentication, the MN sends a reassociation request and expects a reassociation response back from the access point. These messages are responsible for the ***reassociation delay***.

In MM-MPLS testbed, the probe delay is the time from send handoff request by the scripts to send the Mobile IP Registration Request message to the new FA. The authentication delay is the time between registration request and registration reply message with the new connected FA. The reassociation delay means the time from receiving the registration reply message to getting the first packet from the new connected FA.

From the figure, we can also see that almost 90% of the delay is located at the scanning (probe) process. If we enlarge the top two parts of delay, as is done in Figure 4.12, we may see that no matter what the policy we are using is, the total authentication delay and reassociation delay is close to 0.02 seconds. The handoff delay is mainly related to the probe delay according to our testbed.

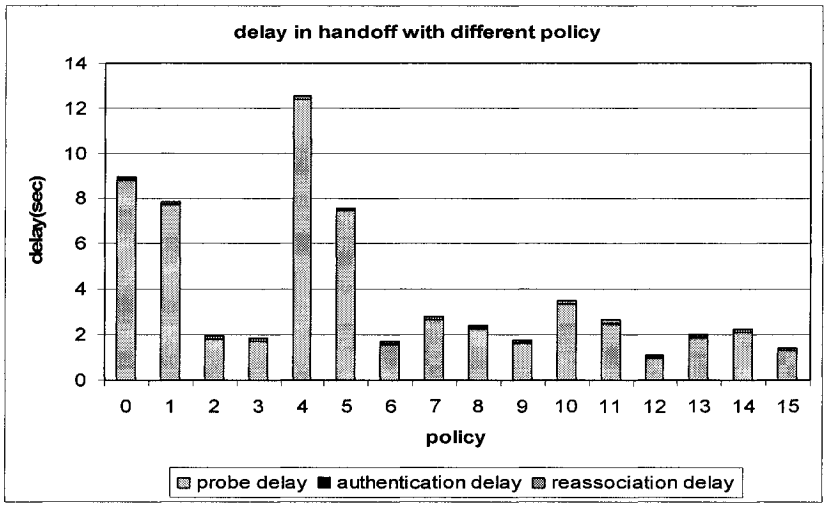

Figure 4.11 Handoff delay with different policy

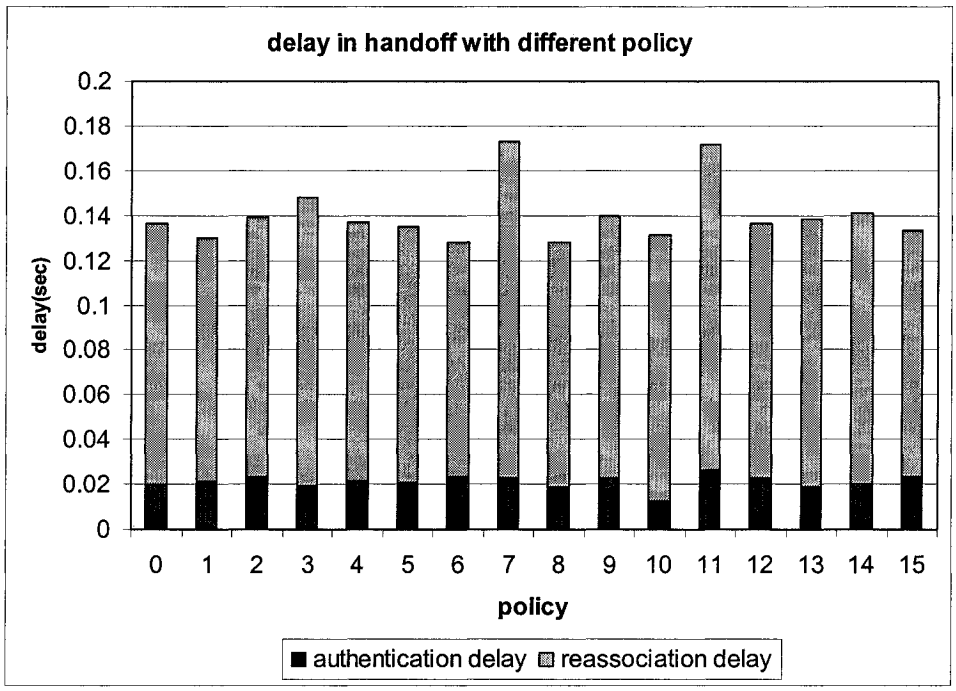

Figure 4.12 Authentication delay and reassociation delay during handoffs

Several tests were conducted in order to determine the performance. Table 4.11 gives an example of a group of handoff tests according to the packet sending rate. In this test, the Advertisement Interval on both FA1 and FA2 are all set to 1 second. Thus, within the handoff process, the average packet loss (APL) time is around 1 second, which for some extend indicates the handoff delay time.

test	rate(bps)	Send	Receive	Lost	APD(ms)	L-Rate (%)	APL
1	10240	18000	17955	45	3.446100	0.255556	0.7500
2	20480	18000	17959	41	4.763885	0.227765	0.6833
3	40960	18000	17946	54	6.191666	0.299983	0.9000
4	81920	18000	17934	66	11.269604	0.366646	1.1000
5	163840	18000	17958	42	18.639294	0.233320	0.7000

(APD: Average Packet Delay, L-Rate: lost rate, APL: Average Packet Lost time)

Table 4.11 Example of handoff test

Figure 4.13 shows the raw data collected during test 4 in Table 4.11. Some of the parameters for this test are: handoff period =300 seconds, packet sending frequency=10 pk/sec. It is easy to see that there are two levels of behaviors. This is because the two computers, FA1 and FA2 are different. In which, the FA1 is a computer having Pentium II CPU, 267 MHZ, 128MB RAM. FA2 is using a Pentium 4, 1.8 GHz, and 256MB RAM. The data group with smaller values is the data transported through FA2 (lower one on the figure) and the bigger value group corresponds to the data transport through FA1 (upper one on the figure).

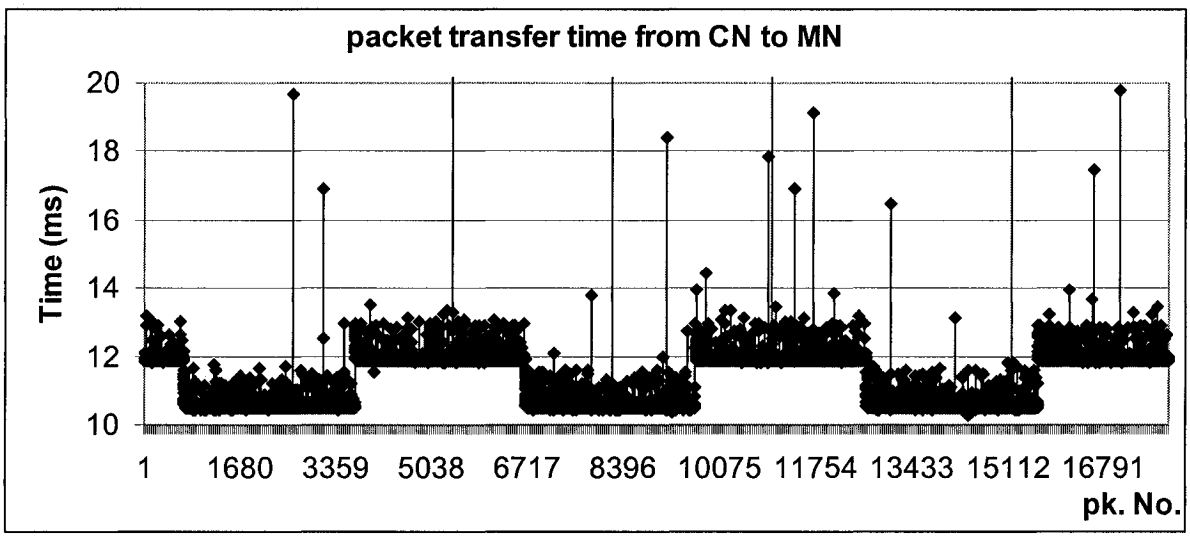

Figure 4.13 Packet transport time from CN to MN during the handoff process

Chapter 5 Conclusion

In this thesis, the MM-MPLS protocol has been implemented over an experimental network. The existing Mobile IP implementation for the Linux OS was extended and integrated with MPLS in order to incorporate the functionalities needed by MM-MPLS. The basic mode of hierarchical mobile IP, and the RSVP-TE daemon for DiffServ over MPLS under Linux was used.

To accomplish the MM-MPLS implementation, the following steps have been conducted:

- The leveraged architectures of Mobile IP and QoS support MPLS network have been reviewed and their relevant procedures for this work have been summarized.
- The underlying concepts by means of description specifications in the Dynamics Mobile IP and the RSVP-TE for DiffServ over MPLS have been analyzed. Their behavior has been re-examined in a more formal way to provide a basis for the implementation design.
- The implementation design has been made with respect to the selected operating system (Linux).
- Open issues and possible solutions according to the protocol's specifications have been identified and integrated in the implementation design.

Finally, it can be concluded that the MM-MPLS implementation presented here provides a flexible, extensible and easy-to-configure prototype for testing and evaluating of the Micro-cell Mobile MPLS protocol. Moreover, the results collected through the testbed confirmed the advantage of MM-MPLS over hierarchical mobile IP in terms of lower forwarding.

REFERENCES

[1] Practical Approaches for Supporting Micro Mobility with MPLS *(Tingzhou Yang, Yixin Dong, Yihan Zhang, Dimitrios Makrakis, SITE, University of Ottawa)*

[2] Hierarchical Mobile MPLS: Supporting Delay Sensitive Applications Over Wireless Internet*(Tingzhou Yang and Dimitrios Makrakis, SITE, University of Ottawa)*

[3] Integration of Mobile IP and Multi-Protocol Label Switching *(Zhong Ren, Chen-Khong Tham, Chun-Choong Foo, Chi-Chung Ko, Department of Electrical and Computer Engineering National University of Singapore)*

[4] IP Mobility Support *(IETF RFC 2002, C. Perkins,IBM)*

[5] RSVP-TE: Extensions to RSVP for LSP Tunnels *(IETF RFC 3209)*

[6] Multiprotocol Label Switching Architecture *(IETF RFC 3031)*

[7] Communication Availability with Mobile IP in Wireless LANs *(Dan Forsberg, Facuty of Information Technology, Helsinki University of Technology)*

[8] Mobile MPLS-a MPLS based Micro Mobility Concept *(J. Grimminger and H.-P. Huth, Siemens AG, Corporate Technology, Information & Communication)*

[9] A MPLS Framework for Macro- and Micro-Mobility Management *(Kaiduan Xie and Victor C.M. Leung, Department of Electrical & Computer Engineering)*

[10] A Network Architecture for MPLS-Based Micro-Mobility *(Fabio M. Chiussi, Denis A. Khotimsky, Santosh Krishnan, Bell Laboratories, Lucent Technologies)*

[11] A comparison of IP mobility protocols *(Pierre Reinbold and Olivier Bonaventure, University of Namur)*

[12] Comparison of IP Micro-Mobility Protocols *(Andrew T. Campbell, Javier Gomez, Sanghyo Kim, Chieh-Yih Wan, COMET Group, Columbia University)*

[13] Linux Network Performance - RAW ethernet vs. UDP *(Andreas Schaufler)*

[14] An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process *(Arunesh Mishra, Minho Shin, William Arbaugh, University of Maryland)*

[15] IP Micromobility Protocols *(Chien-Chao Tseng, National Chiao Tung University, TaiWan)*

[16] QoS Implementation For MPLS Based Wireless Networks *(Subramanian Vijayarangam and Subramanian Ganesan Oakland University)*

[17] Support of Micro-Mobility in MPLS-based Wireless Access Networks *(Kaiduan Xie, Vincent W.S. Wong, and Victor C.M. Leung, The University of British Columbia)*

[18] M-MPLS: Micromobility-enabled Multiprotocol Label Switching *(Vasos Vassiliou, Henry L. Owen, David Barlow, Joachim Sokol, Hans-Peter Huth, Jochen Grimminger)*

[19] A mobile connection for 4G networks using *MPLS (Johan Theunis, Jan Potemans, ...etc, ESAT/TELEMIC, Belgium)*

[20] Mobility support for ubiquitous Internet access *(Georgios Karagiannis, Geert Heijenk, ERICSSON)*

[21] Nicro-Mobility within Wireless Ad Hoc Networks: Towards Hybrid Wireless Multihop Networks *(Typpo Ville, University of Oulu, Finland)*

[22] Introducing Mobile IPv6 in 2G and 3G mobile networks *(White paper, Nokia)*

[23] Mobile Networks using IPv6 *(White paper, Nokia)*

[24] Mobility Support in IPv6 *(Charles E. Perkins, T.J. Watson Research Center, IBM Corporation and David B. Johnson, Computer Science Department of Carnegie Mellon University)*

[25] Hierarchical Mobile IPv6-Implementation Experiences *(Axel Neumann, Andreas Festag, TKN, TU-Berlin, Siemens IPv6 Workshop)*

[26] Mobile IP Regional Registration *(Charles E. Perkins, Nokia Research Centre)*

[27] Mobile IPv6-Mobility support for the next generation Internet *(Wolfgang Fritsche and Florian Heissenhuber, IABG)*

[28] A Survey of IP micro-mobility protocols *(Pierre Reinbold and Olivier Bonaventure, Infonet group, University of Namur, Beigium)*

[29] Supporting of QoS and Micro-Mobility in MPLS-based IPv6 Wireless Networks *(Y.Zhang, D.Makrakis and D. Hatzinakos, University of Ottawa)*

[30] IP Micro-Mobility Protocols *(Andrew T. Campbell and Javier Gomez_Castellanos, COMET Group, Center for Telecommunications Research, Columbia University, New York, NY, USA)*

[31] A Multicast-based Protocol for IP Mobility Support *(Ahmed Helmy, University of Southern California)*

[32] A Study on Path Re-routing Algorithms at the MPLS-based Hierarchical Mobile IP Network *(Tai Won Um, Jun Kyun Choi, IEEE)*

[33] Mobile MPLS-a MPLS based Micro Mobility Concept *(J. Grimminger, H. –P. Huth, Siemens AG, Corporate Technology, Information & Communication)*

[34] Mobile IPv6 support in MPLS Network *(Jun Kyun Choi, Myoung Hun Kim and Yoon Ju Lee, ICU)*

[35] Hierarchical MIPv6 mobility management *(Hesham Soliman, Ericsson)*

[36] Performance of IP Micro-Mobility Management Schemes using Host Based Routing *(K. Daniel Wong, Hung-Yu Wei, Ashutosh Dutta, Kenneth Young, Telcordia Technologies Inc., USA)*

[37] Evaluation of Mobility and Quality of Service Interaction *(Jukka Manner, Alberto Lopez Toledo, Andrej Mihailovic,....University of Helsinki, Finland)*

[38] Secured MPLS-based Mobile IP using the Private IP Address *(Sriborrirux Wiroon, Jeong-Beom Kim, Rhe yun-jung and Tai-Yun Kim, Korean University)*

[39] Mobile Computing: Overview and current Status *(Arkady Zaslavsky, Zahir Tari, Monash University, Australia)*

[40] Increasing communication availability with signal-based mobile controlled handoffs *(D. Forsberg, J. T. Malinen, J. K. Malinen, Hannu H. Kari, Helsinki Universityof Technology, Finland)*

[41] IP Mobility Management *(Ronan J. Skehill, Sean McGrath, Department of Electronic and Computer Engineering, University of Limerick)*

[42] A Comparison of Mechanisms for Improving Mobile IP Handoff Latency for End-to-End TCP *(Robert Hsieh and Aruna Seneciratne, University of New South Wales, Sydney, Australia)*

[43] Seamless Handoff in Community Based and Location Aware Heterogeneous Wireless Networks *(Maximiliam Zundt, Peter Tabery and Christian Bachmeir, Germany)*

[44] Fast and Scalable Handoffs for Wireless Internetworks *(Ramon Caceres, Venkata N. Padmanabhan, University of California, USA)*

[45] Distributing Mobility Agents Hierarchically under Frequent Location Updates *(D. Forsberg, J.T. Malinen...., Helsinki University of Technology, Finland)*

[46] A Mobile Differentiated Services QoS Model *(Jorg Diederich, Lars Wolf, et al. Technical University of Braunschweig, Germany)*

[47] IP Mobility-Current Status and Perspectives *(Zoltan Turanyi, Ericsson Research, Hungar*

[48] Integrated Services in the Internet Architecture: an Overview Status of this Memo *(IETF RFC 1633)*

[49] An Architecture for Differentiated Service (*IETF RFC 2475*)

[50] A Practical Guide to Linux Traffic Control (*Jason Boxman*)

[51] Specification of Guaranteed Quality of Service (*S. Shenker, C. Partridge ,and R. Guerin, IETF RFC 2212*)

[52] Specification of the Controlled-Load Network Element Service (*J. Wroclawski, IETF RFC 2211*)

[53] The Use of RSVP with IETF Integrated Services (*J. Wroclawski, IETF RFC 2210*)

[54] Combining IntServ and DiffServ under Linux (*Werner Almesberger et al.)*

[55] Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers (*K. Nichols, IETF RFC 2474)*

[56] Dynamics Mobile IP project (*http://dynamics.sourceforge.net/*)

[57] Fast Handoffs in Mobile IPv4 (*K. El-Malki and H. Soliman, Internet draft,2000*)

[58] Mobile IP Regional Registration (*E. Gustafsson, A. Jonsson, and C. Perkins, Internet draft, 2000)*

[59] Cellular IP (*A.T. Campbell, J. Gomez, S. Kim, A.G. Valko et el, Internet draft, 2000*)

[60] IP micro-mobility support using HAWAII (*R. Ramjee, T. La Porta, S. Thuel, and K. Varadhan and L. Salgarelli, Internet draft, 1999*)

[61] http://dynamics.sourceforge.net/

[62] http://dsmpls.atlantis.ugent.be/

[63] http://sourceforge.net/projects/mpls-linux/

[64] Multiprotocol Label Switching-White paper (*Chuck Semeria, Juniper Networks, Inc.)*

[65] OSPF Version 2 (*IETF RFC 2328*)

[66] Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification (*IETF RFC 2205*)

[67] LDP Applicability (*IETF RFC 3037*)

[68] Overview and Principles of Internet Traffic Engineering (*IETF RFC 3272*)

[69] Traffic Engineering with MPLS in the Internet (*Xipeng Xiao, Alan Hannan, Brook Bailey, Internet draft*)

[70] Quality of Service for Ethernet (*Matthew Demyttenaere, Sophie Legault*)

[71] A Framework for Telephony Routing over IP (*J. Rosenberg, H. Schulzrinne, RFC 2871*)

[72] Cellular IP: A New Approach to Internet Host Mobility (András G. Valkó1, ACM Computer Communication Review, January 1999)

[73] The Linux Kernel Module Programming Guide (*Peter Jay Salzman, Ori Pomerantz, 2003-04-04, ver-2.4.0*)

[74] Red Hat Linux 8.0 The Official Red Hat Linux Getting Started Guide (*Red Hat Inc., 2002*)

[75] MPLS for Linux Developers' Guide (Ramon Casellas, 2003)

APPENDIX Additional Performance Measurements

A.1 Examples of Mobile Node registration process for Chapter 4.1.

There are three reasons that the MN will send Registration Request:

(1) When the MN turned on, it sends Registration Request message in order to register with its home agent. If the MN is away from its home domain when it is turned on, it will be assigned a Care-of-address by the address of the gateway router between the foreign domain which it is living in and its home domain (the gateway router is the FDA in our case). Anytime when the Registration Request message is send to reach the HA, the HA will send a Registration Reply to the MN with lifetime 300 sec.

(2) When the MN moves into a new foreign agent, it sends Registration Request through the new FA and this Registration message will be intercepted at a CR. In this case, the CR will generate a Registration Reply message with the lifetime currently left at that moment.

(3) When the MN keeps staying at a foreign domain, it will refresh the registration message periodically, in order to keep the connection with its home agent. The refreshing time is calculated as **half of the lifetime of last Registration Reply message**. For example, if the last Registration Reply is send by HA with the lifetime 300 sec, then the Registration Request will be send right after 150 sec (1/2 of 300 sec) from the MN receives the last Registration Reply. If the Registration Reply is come from the CR with the lifetime 68 sec, which means the lifetime has been changed by CR due to the MN's handover, the next Registration Request will be send after 34 sec (1/2 of 68 sec) from the MN receives this Registration Reply.

In summary, the MN will determine when it will send Registration Request either by the lifetime of the last Registration Reply message or by the handoff requiring.

We designed a test for confirming the registration process on MM-MPLS testbed is working as the way of in Hierarchical Mobile IP. The test scenario is as following:

1. When starting, the MN is in its home network, it registers directly with the HA.
2. To simulate the MN moving into a foreign domain, we use a script command to force the handoff process. It makes the MN attaching to FA1, which is a lowest foreign agent in MM-MPLS testbed.
3. After 60 seconds, forcing the MN handoff to FA2 subnetwork, which is in the same foreign domain as FA1.
4. Every 60 seconds, the MN will be moved from one access point to another one between FA1 and FA2. Repeat this kind of handoff for 10 times.
5. Continue the monitor for another 10 minutes in order to capture the data to see the result of soft-state signaling registration messages.

We use Ethereal, which is a network protocol analyzer to monitor the registration messages on the MN, CR and HA by set the filter to "udp port 434". We shot the screen with the following pictures.

On those pictures, the entries with white background are the registration messages during the forcing handoff process.

Figure A-1 MN Mobile IP Registration messages by Ethereal

Figure A-2 CR Mobile IP Registration messages by Ethereal

Figure A-3 HA Mobile IP Registration messages by Ethereal

In summary, Figure A-4 shows the result by time sequence with the lifetime of each Mobile IP registration message indicated.

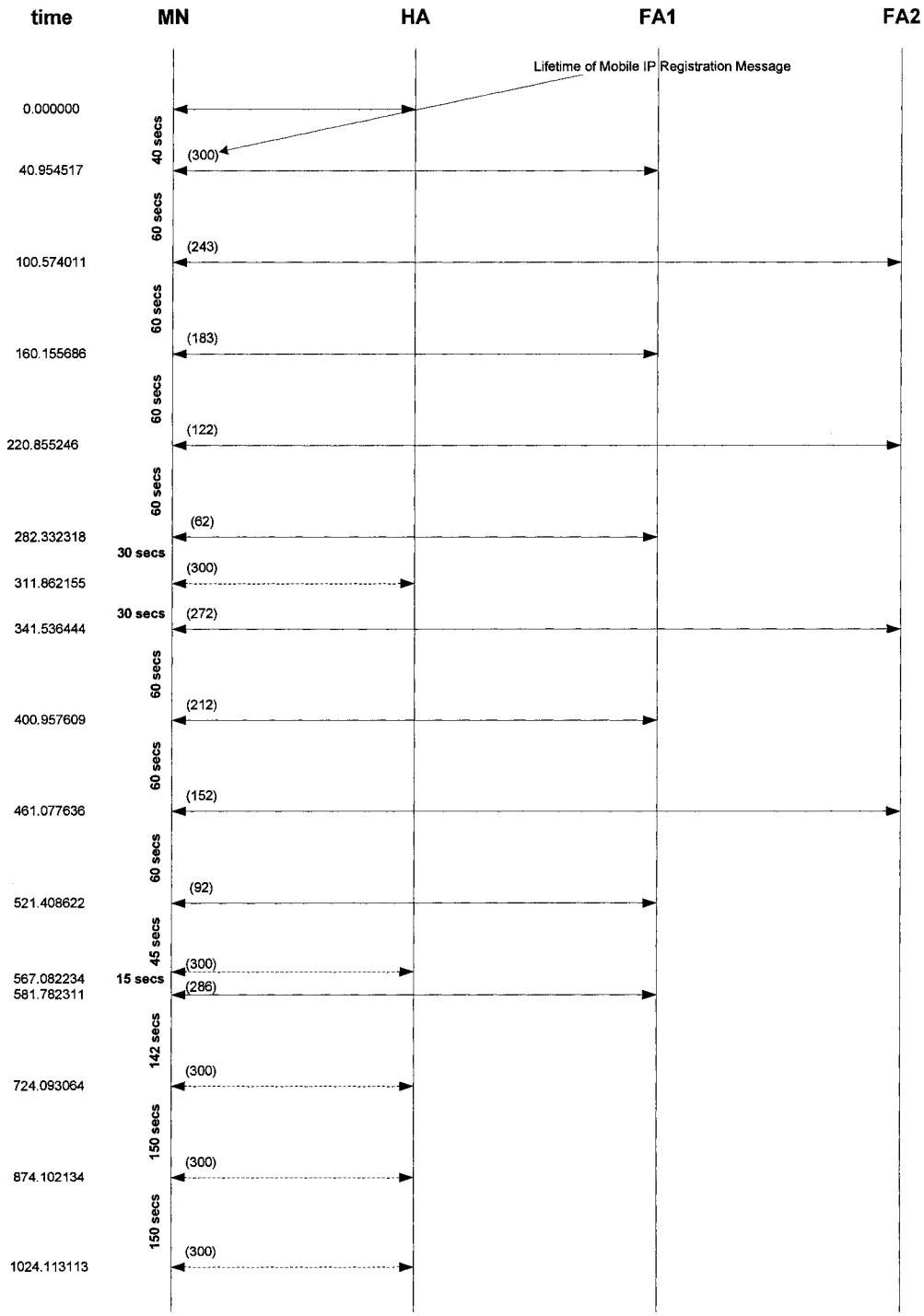

Figure A-4 Mobile IP Registration Messages during the test

A.2 A Complex Test

In order to see the system performance of MM-MPLS regarding the different throughput of traffic, three group of test (with traffic rate=0.32, 0.64 and 0.96 Mbps) were taken. Each group with five different packet sending frequency (freq. = 20, 40, 60, 80, 100 pk/sec). Each test takes 400 seconds and 5 handoffs are involved with 1handoff/min. The other conditions are at AI-FA1=1 second, AI-FA2=1 second, handoff policy=12, transfer rate of each wireless card is configured at 2M. Table A-1 to A-3 show the detailed result.

Frequency (pk/s)			20	40	60	80	100
Send (pk)			8000	16000	24000	32000	40000
Receive (pk)			7879	15755	23691	31551	39634
Lost (pk)			121	245	309	449	366
Lost Rate (%)			1.51	1.53	1.29	1.40	0.92
Size (byte/pk)			2000	1000	666.6667	500	400
Average Handoff Delay (s)			**1.008333**	**1.020833**	**0.858333**	**0.935417**	**0.610000**
Average Packet Delay (ms)	Total	**Mean**	**19.934828**	**11.580685**	**8.243747**	**6.578684**	**5.571700**
		Standard Deviation	1.495695	0.886335	0.686968	0.667822	0.611902
		Confidence Level(95.0%)	0.033031	0.013841	0.008748	0.007369	0.006024
	FA1	**Mean**	**18.393140**	**10.829034**	**7.720503**	**6.164372**	**5.233754**
		Standard Deviation	0.472964	0.591730	0.456714	0.586533	0.461570
		Confidence Level(95.0%)	0.015559	0.013762	0.008643	0.009658	0.006764
	FA2	**Mean**	**21.200388**	**12.197924**	**8.676924**	**6.916972**	**5.849831**
		Standard Deviation	0.580317	0.546095	0.525010	0.527596	0.579566
		Confidence Level(95.0%)	0.017296	0.011509	0.009039	0.007928	0.007704

Table A-1 Complex test with throughput = 0.32M bps

Frequency (pk/s)			20	40	60	80	100
Send (pk)			8000	16000	24000	32000	40000
Receive (pk)			7888	15738	23723	31543	39409
Lost (pk)			112	262	277	457	591
Lost Rate (%)			1.40	1.64	1.15	1.43	1.48
Size (byte/pk)			4000	2000	1333.33	1000	800
Average Handoff Delay (s)			**0.933333**	**1.091667**	**0.769444**	**0.952083**	**0.985000**
Average Packet Delay (ms)	Total	**Mean**	**25.482628**	**19.934035**	**15.234112**	**11.577355**	**9.569483**
		Standard Deviation	2.832755	1.494848	1.062371	0.875299	0.760656
		Confidence Level(95.0%)	0.062523	0.023356	0.013520	0.009660	0.007510
	FA1	**Mean**	**22.425765**	**18.414523**	**14.244273**	**10.814889**	**8.954764**
		Standard Deviation	0.499381	0.543813	0.566903	0.534622	0.534020
		Confidence Level(95.0%)	0.016428	0.012671	0.010736	0.008798	0.007863
	FA2	**Mean**	**27.986774**	**21.175964**	**16.049328**	**12.200604**	**10.071659**
		Standard Deviation	0.683024	0.624125	0.567385	0.543246	0.507363
		Confidence Level(95.0%)	0.020336	0.013147	0.009751	0.008083	0.006752

Table A-2 Complex test with throughput = 0.64M bps

Frequency (pk/s)			20	40	60	80	100
Send (pk)			8000	16000	24000	32000	40000
Receive (pk)			7878	15760	23628	31576	39330
Lost (pk)			122	240	372	424	670
Lost Rate (%)			1.53	1.50	1.55	1.33	1.68
Size (byte/pk)			6000.00	3000.00	2000.00	1500.00	1200.00
Average Handoff Delay (s)			**1.016667**	**1.000000**	**1.033333**	**0.883333**	**1.116667**
Average Packet Delay (ms)	Total	**Mean**	**33.637373**	**22.283659**	**19.922616**	**16.521147**	**13.545227**
		Standard Deviation	4.336976	2.231809	1.503873	1.214090	0.962265
		Confidence Level(95.0%)	0.095784	0.034847	0.019176	0.013392	0.009510
	FA1	**Mean**	**28.888750**	**19.900699**	**18.398668**	**15.343638**	**12.653400**
		Standard Deviation	0.460850	0.567049	0.537414	0.438451	0.471237
		Confidence Level(95.0%)	0.015176	0.013194	0.010207	0.007197	0.006941
	FA2	**Mean**	**37.522411**	**24.236355**	**21.166239**	**17.490721**	**14.275841**
		Standard Deviation	0.691384	0.576525	0.512560	0.669058	0.563210
		Confidence Level(95.0%)	0.020592	0.012143	0.010531	0.009966	0.007508

Table A-3 Complex test with throughput = 0.96M bps

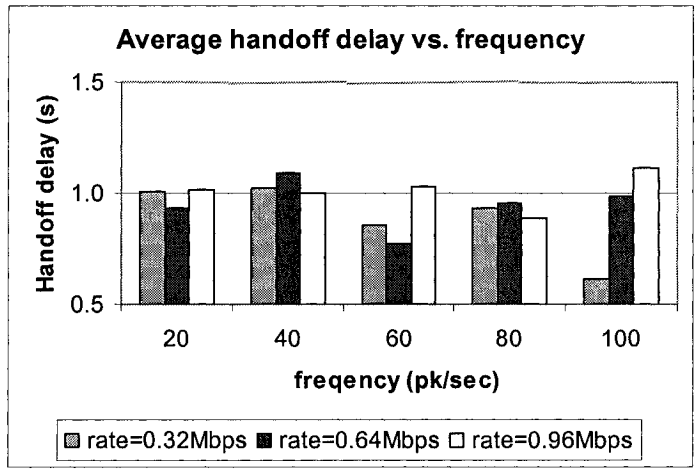

Figure A-5 Average handoff delay vs. frequency

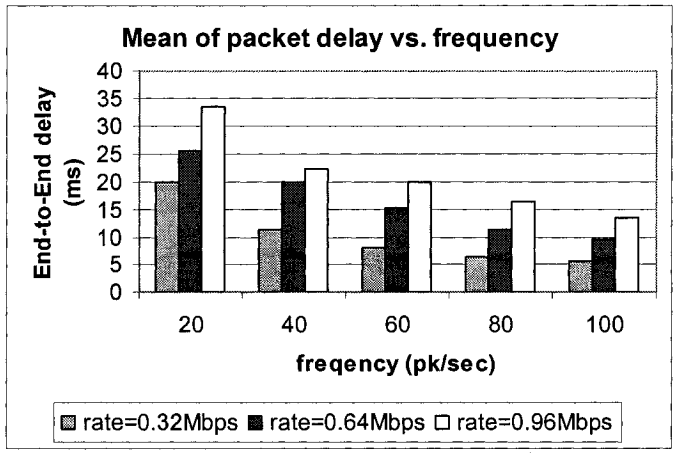

Figure A-6 Mean of packet delay from CN to MN vs. frequency

From above tests, the conclusion comes to:

1. From Figure A-5, the handoff delay is revolved around 1 second, which is around the configuring of the Advertisement Interval to each FA. The testing handoff delay is depending on the scanning process, which is more than 90% of the total handoff delay from the test described in Chapter 4.2.3. However, the Advertisement Interval can be assign as minimum as 1 second in Mobile IP under Linux.
2. From Figure A-6, when throughput is fixed, the lower packet sending frequency, the higher average packet delay. Also, when the packet sending frequency is fixed, larger throughput will bring higher average packet delay.
3. Different foreign agent brings different average packet delay due to the configuration of computer. In the presenting MM-MPLS testbed, FA2 is a more powerful computer than FA1.