PEER-TO-PEER REAL TIME MOBILITY USING SIP AND MOBILE IPV6

NG KWANG LOONG STANLEY

(B.Eng. (Hons), NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2002/2003

Abstract

To support an Internet with ubiquitous seamless mobility and peer-to-peer real-time communication services like Internet Telephony, a set of stringent network requirements must be satisfied. (I) Low end-to-end session establishment and data exchange delay, as prolonged latency would cause initiating party to abandon session. (II) Low end-to-end delay variation/jitter as not to impair quality of the real-time communication session. (III) Inherent support for mobility of both users and computing devices without incurring high signaling traffic and data overhead. This effectively avoids bandwidth wastage for exchanging meaningful information, improves service providers' profitability due to greater membership subscriptions, and avoids high usage charges in a pay-as-you-use billing plan.

The problem space of this thesis is to investigate two key existing mobility support schemes namely Mobile IPv6 and *Session Initiation Protocol* (SIP) support for mobility which is generally known as Mobile SIP. The investigation is conducted from two perspectives. SIPsim, a minimal design and implementation of SIP as an extension to NS-2, provides thorough evaluation and clear understanding of SIP internalities and functionalities. Qualitative and quantitative analysis of Mobile IPv6 and Mobile SIP reveals suitability for terminal and personal mobility respectively.

This thesis contributes a novel and practical architecture i.e. **On-demand Mobility Agent and Mobility Address Assignment** designed with the objective to minimize the inefficiencies experienced by Mobile IPv6 and Mobile SIP by harmonizing the interaction and coexistence between both protocols. It improves the performance of Mobile IPv6 using the strength of Mobile SIP to support seamless terminal and personal mobility for both peer-to-peer and client/server communication within the wireless Internet. The architecture adopts two newly designed SIP header extensions **Assign** and **Assigned**, and a set of modified Mobile IPv6 Binding Update Destination Option and Binding Acknowledgment Destination Option signaling messages for allocation of a serving **Mobility Address** and **Mobility Agent** dynamically per communication session.

Keywords: SIP, Mobile IPv6, Peer-to-Peer, Real-Time, NS-2, Mobility

Words: 302

Acknowledgments

Sincere gratitude and thanks to Dr. Winston SEAH Khoon Guan, supervisor and Dr. Anthony LO, ex-supervisor for their support and patience during my years at *Institute for Infocomm Research* (I^2R). They provided constant academic guidance and inspired many of the ideas presented here. Both supervisors are superb teachers, great communicators, and excellent manager of research projects. It was my fortune to be offered a chance to work closely with them. I look forward to develop our relationship further both as colleague and as friends.

At I^2R, I have learned and acquired as much from the continuous interaction with other staffs and students as from my supervisor. I wish to acknowledge in particular my working colleagues during the past years, Tan Seng Kee and ex-colleague Eng Soo Guan for motivating me with their invaluable technical guidance, enthusiastic encouragement, and understanding, most critical to the development of my academic pursuit. In addition, I would like to extend special gratitude and heart-felt appreciation to Jaya Shankar, Yao Qi, and Ge Yu for their understanding and advice on this academic project.

Finally, I wish to dedicate this thesis to my family members for their efforts to provide me with the best possible education. Sincere appreciation to my parents, for without their love, self-sacrifice, constant guidance and encouragement throughout my life, I would not have this great opportunity to pursue and fulfil my academic ambition.

Table of Contents

List of Tables

List of Figures

List of Graphs

List of Abbreviations

AD	Administrative Domain
AR	Access Router
BA	Binding Acknowledgement Destination Option
BC	Binding Cache
BU	Binding Update Destination Option
CH	Correspondent Host
COA	Care-of Address
CU	Correspondent User
DAD	Duplicate Address Detection
DHCPv6	Dynamic Host Configuration Protocol Version 6
DL	Data Load
HA	Home Agent
HAD	Home Address Destination Option
HAddr	Home Address
HOD	Handover Delay
IPv4 or IPv6	Internet Protocol Version 4 or Version 6
LS	Location Server
MH	Mobile Host
MIPv4 or MIPv6	Mobile Internet Protocol for Version 4 or Version 6
MSIP	Mobile SIP
MU	Mobile User
NA	Neighbour Advertisement
NS	Neighbour Solicitation
PM	Personal Mobility
RA	Router Advertisement
RH	Routing Extension Header
RS	Router Solicitation
RTP	Real-Time Transport Protocol
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SIPPS	SIP Proxy Server
SIPRS	SIP Redirect Server
SL	Signaling Load
TM	Terminal Mobility
UA	User Agent
UAC or UAS	User Agent Client or User Agent Server
URI	Universal Resources Identifier
URL	Uniform Resource Locator

Chapter 1.
Introduction

Rapid development and deployment of commercial wireless networks ranging from IEEE 802.11-based *Wireless Local Area Networks* (WLANs) to *General Packet Radio Service* (GPRS) [1] and *Enhanced Data rates for Global Evolution* (EDGE) [2], coupled with the emergence and popularity of handheld devices like smart phones, *Personal Desktop Assistant* (PDA) and laptops with wireless access, slowly but gradually mark an emerging trend of mobile computing becoming a natural part of our daily lives. A *Mobile User* (MU) working from home, office, or on-the-move expects pervasive access to information and communication facilities from any location at anytime using any mobile devices.

In order to support an Internet with ubiquitous seamless mobility and peer-to-peer real-time communication services like IP Telephony, Video Conferencing, and Instant Messaging, a set of stringent service requirements must be satisfied. (I) Low end-to-end delay for session establishment and data exchange as prolonged latency would cause the initiating party to abandon a session. This is complicated by MUs roaming in foreign networks as packets may not be routed optimally along the shortest path between the communicating entities. (II) Low end-to-end delay variation/jitter and performance degradation [3] especially during MU's handover as in-flight packets to MU's previous point-of-attachment may be discarded or delayed due to buffering or recovery, consequently degrading the quality of the real-time communication session. The *International Telecommunication Union - Telecommunications* (ITU-T) [4] recommends acceptable voice delay lower than threshold of 150 ms and reasonable delay ranging 150 - 400 ms subjected to proximity of both end-parties. (III) Inherent support for mobility of both users and computing devices without incurring high signaling traffic and data overhead. This avoids bandwidth wastage for exchanging meaningful information, improves service providers' profitability due to greater membership subscriptions, and avoids high usage charges in a pay-as-you-use billing plan.

1.1 Statement of the Thesis

The problem space is to investigate two key existing mobility support schemes namely *Mobile IPv6* (MIPv6) and *Session Initiation Protocol* (SIP) support for mobility (MSIP). The investigation is conducted from two perspectives. SIPsim, a minimal design and implementation of SIP as an extension to NS-2, provides thorough evaluation and clear understanding of SIP internalities and functionalities. Qualitative and quantitative analysis of MIPv6 and MSIP reveals suitability for *Terminal Mobility* (TM) and *Personal Mobility* (PM) respectively. Consequently, an architecture is proposed with the objective to minimize the inefficiencies experienced by MIPv6 and MSIP by harmonizing the interaction and coexistence between both protocols. The architecture supports seamless TM and PM for both peer-to-peer and client/server communication ubiquitously within the wireless Internet.

1.2 Contributions

This thesis has motivated a series of proposals and contributions:

SIPsim, a minimal design and implementation of SIP as an extension of an open source network simulator NS-2. SIPsim will be contributed to NS-2 community providing a cost-effective research platform for evaluation, verification, and understanding of SIP, and prototyping of advanced value-added services like mobility support and SIP interworking with RSVP. SIPsim is developed using a mixture of C++ and TCL languages, running over NS-2b7a. SIPsim has been validated against specification conformance test-suite, and has successfully demonstrated session establishment and termination for various scenarios consisting direct peer-to-peer and involvement of SIP network entities.

Detailed qualitative and quantitative analysis/comparison of MIPv6 and MSIP facilitates derivation of situations and conditions upon which either protocol would be appropriate for. No prior research work on this area has been reported in the literature. The former evaluates both protocols in terms of their two-tier addressing scheme and address translation mechanism. The latter studies signaling load, data packet overhead generated, registration time which is a measure of handover delay, and session establishment latency incurred by both protocols. The quantitative study reveals MIPv6 is more efficient than MSIP in terms of lower signaling load for location management but incurs higher overhead for data transmission regardless of whether the mobile terminal resides in the home network or when roaming. MSIP incurs lower

session establishment delay than MIPv6 but at the expense of higher handover delay.

A novel architecture i.e. **On-demand Mobility Agent and Mobility Address Assignment** with the objective to minimize the inefficiencies experienced by MIPv6 and MSIP by harmonizing the interaction and coexistence between both protocols. The architecture improves the performance of MIPv6 using the strength of MSIP to supports seamless TM and PM for both peer-to-peer and client/server communication ubiquitously in the wireless Internet while satisfying the following high requirements. (I) Low end-to-end delay for session establishment and data exchange, as prolonged latency would cause initiating party to abandon session. (II) Low handover delay bypassing Duplicate Address Detection at the virtual "home" network so that Binding Acknowledgment is replied immediately to the mobile terminal, as to minimize jitter and delay variation. (III) Low signaling traffic and overhead of data exchange taking into account the spatial locality of MU. The architecture adopts two newly designed SIP header extensions **Assign** and **Assigned**, and a set of modified MIPv6 signaling messages for allocation of a serving **Mobility Address** and **Mobility Agent** dynamically per communication session.

1.3 Thesis Organization

Chapter 2 and Chapter 3 provide thorough evaluation and clear understanding of SIP internalities and functionalities through minimal implementation of SIP, SIPsim as an extension to the open source network simulator NS-2.

Chapter 4 covers literature survey of related work on current solutions and issues of supporting mobility in the Internet from different perspectives of network, transport, and application layer.

Chapter 5 elaborates qualitatively and quantitatively analysis/comparison between MSIP and MIPv6 as to facilitate derivation of situations and conditions upon which either protocol would be appropriate for deployment in the wireless Internet to support both PM and TM without incurring network performance penalties.

Chapter 6 presents a novel and practical architecture by harmonizing the interaction and coexistent between MIPv6 and MSIP. The architecture improves the performance of MIPv6 using the strength of MSIP to support TM and PM for both peer-to-peer and client/server communication seamlessly in the wireless Internet.

Chapter 7 concludes this dissertation and enumerates potential future works.

Chapter 2.
Session Initiation Protocol (SIP)

Section 2.1 introduces basic background of Internet Telephony, multimedia data and control architecture incorporating protocols for session management, transport of real-time data and multimedia session description. Section 2.2, 2.3, and 2.4 elaborate *Session Initiation Protocol* (SIP) in relation to its architecture, logical entities, and messaging methods. Section 2.5 illustrates the application of SIP in Internet Telephony using several SIP call examples.

2.1 Introduction of Internet Telephony

Internet Telephony (IP Telephony or Voice over IP) is an IP based communication technology for carrying real-time voice traffic over the public packet-switched Internet. This is a dramatic improvement over conventional *Public Switched Telephone Network* (PSTN) which reserves dedicated end-to-end circuit connection for the duration of each active call. Benefits of using IP Telephony over PSTN are as follows. Cheaper long distance calls as traditional telephony access charges and settlement are avoided by leveraging on the Internet for transporting voice data. Greater network efficiency as packetized voice offers higher bandwidth efficiency in terms of data multiplexing especially when a significant part of a conversation is silence.

Figure 2.1 summarizes the *Internet Engineering Task Force* (IETF) multimedia data and control architecture [5,6] which comprises a set of Internet multimedia services and protocols required to provide the same desired services and voice toll quality as PSTN. These protocols are categorized into **Control and signaling** and **Data Transport**. **Control and signaling** allows two or more parties to establish a session (voice or video), to negotiate media information and capabilities (e.g. codec supported, desired sampling rate, and data transport protocol), to modify, and to terminate an existing session. **Data Transport** permits commencement of communication via transmission of packetized voice data packets over the IP networks from one party to the other, after a session has been successfully established. *Real-Time Transport Protocol* (RTP) [7] is generally adopted as the end-to-end data transport protocol for real-time applications

such as media-on-demand (e.g. audio and video) or interactive services (e.g. IP Telephony) with built-in timing reconstruction, loss detection, security, and content identification. RTP is typically implemented over *User Datagram Protocol* (UDP) to leverage on its multiplexing and checksum functions.

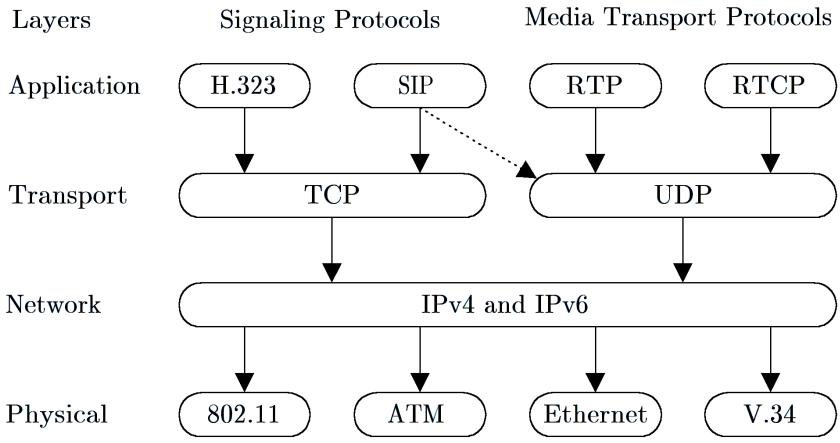


Figure 2.1: Protocol Architecture for Internet Multimedia Services

Currently, there exists two major competing call and multimedia session management signaling standards [8-10] namely binary-based H.323 [11,12] and text-based *Session Initiation Protocol* (SIP) [13,14] standardised by *International Telecommunication Union - Telecommunications* (ITU-T) and IETF respectively. The former was originally conceived for multimedia conferencing on a *Local Area Network* (LAN), but has been extended with IP Telephony functionalities comprising call control, conferencing functionalities, call management, capabilities negotiation, and supplementary services. In contrast, the latter is a lightweight signaling protocol for establishing and terminating IP Telephony session, negotiating information required for the session to progress (e.g. media codec and addresses), and invoking services like hold, mute, and transfer. Comparison between SIP and H.323 [15-18] cover issues like basic call architectures, complexity, and scalability at system level, while [19,20] discuss implementation, architectures, and capabilities at the service level.

2.2 Overview of Session Initiation Protocol (SIP)

SIP is a lightweight, text-based application layer control and signaling protocol for establishing, negotiating, modifying and terminating multi-party (or multi-point) multimedia sessions e.g. IP Telephony and multimedia conferences, between users and net-

work control entities. SIP is mainly concerned with the following three operations. **Location management** that provides user registration for tracking a user's current location and finding the terminal being used by that user. **Session management** that allows users to initiate invitations to other users to participate a session, or to terminate an existing session. **Session feature management** that permits participants of a session to negotiate and decide on the set of common media parameters to use.

SIP is heavily based on two widely deployed IETF protocols namely *Hypertext Transfer Protocol* (HTTP) [21] and *Simple Mail Transfer Protocol* (SMTP) [22]. SIP borrowed from HTTP, the simple client-server model (a.k.a. request-response model) for its functional operations, with requests issued by the client and responses returned by the server, and the use of *Uniform Resource Locators* (URLs). SIP reuses SMTP's text-encoding scheme and header style like SMTP headers From, To and Subject, for SIP messages to convey the required information for session management.

A typical SIP network architecture as shown in Figure 2.2 consists two broadly categorized SIP entities namely *User Agent* (UA) and *SIP Network Server* (SNS). UA is a SIP end-system for managing and storing session states on behalf of end-user that potentially participates in a session, and in turn communicates with other UAs directly or indirectly via SNSs. UA functions as a protocol client called *User Agent Client* (UAC) when it initiates SIP requests, and as a protocol server known as *User Agent Server* (UAS) when it responds to SIP requests. SNS is defined as a SIP entity for handling session management signaling exclusively, but does not participate in actual data transmission. SNS is functionally divided into the following entities: *Proxy Server* (SIPPS), *Redirect Server* (SIPRS), and Registrar.

SIP message is textual and structurally composed of two physical sections as depicted in Figure 2.3. The first section contains SIP headers for conveying session properties and service information. SIP message is further classified into a request or response indicated by the first line of this section. Request is distinguished with a method defining the nature of the request, and a Request-URI (i.e. a SIP URL) to where the request should be sent. Response has a response code instead. The second section is known as Message Body that conveys session description and negotiating options for specifying audio, video, and multimedia session.

Each user is assigned and identified with an email-like unique User Identity of the syntactical format user_name@[domain_name or host_name] e.g. john@-

cwc.nus.edu.sg or john@server. SIP URL, syntactically similar to the HTTP URL, is constructed with the concatenation of "SIP:" and User Identity. User Identity identifies users independently of where they are located, the types of terminal resided on, or the types of access network subscribed to. User Identity can be embedded with a user name, a user code or a certificate stored on a smart card or a SIM card.

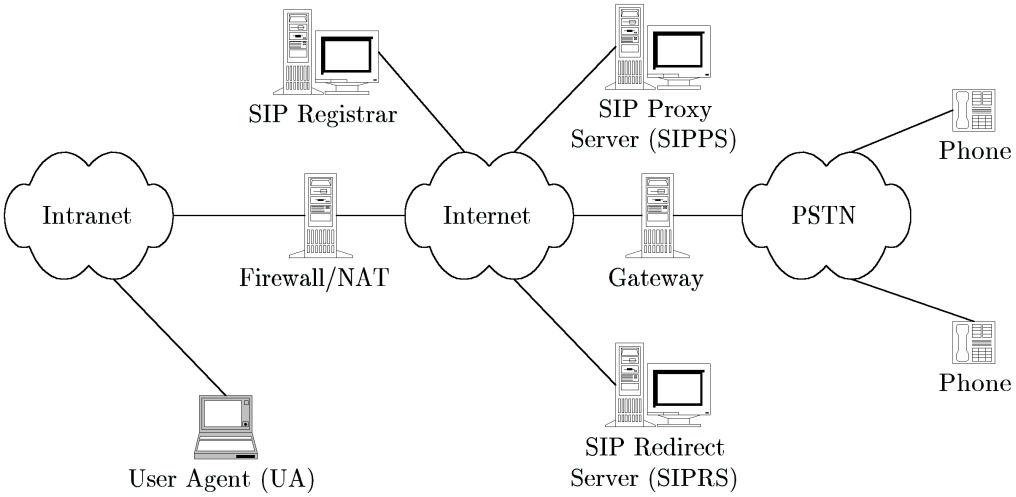


Figure 2.2: SIP Network Architecture

```
INVITE SIP:beta@cwc.edu.sg SIP/2.0
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 470211272@cwc.edu.sg
CSeq: 1 INVITE
Contact: alpha <SIP:alpha@cwc.edu.sg>
Content-Type: application/sdp
Content-Length: 95

v=0
o=alpha 1457850878 2007237709 IN IP4
cwc.edu.sg
s=Session SDP
m=audio 6000 RTP/AVP 0
```

First section contains SIP header fields

Second section contains audio, video and multimedia session description

Figure 2.3: SIP INVITE message

2.3 Details of Logical Session Initiation Protocol (SIP) Entities

SIP Proxy Server (SIPPS) provides name resolution and forwarding capability to the correct destination. It receives a request, resolves the SIP URL to the IP address that it should relay to, and then either forwards the request directly to the current

location of the callee or forks (i.e. sends copies of the request to different destinations) the request to another SNS that might be better informed about the actual location of the callee. SIPPS persists in the signaling path for the duration of the session establishment and may modify specific header fields like recording the path that a request transits to the callee in the request itself. This allows the callee to reply with a response to the caller transversing the same path as that of the request. SIPPS can function as either stateful or stateless. Stateful SIPPS has more intelligence than stateless SIPPS, since the former is required to store a copy of the incoming received request, then it forwards outgoing requests and responses to replied SIP responses. In contrast, the latter discards all information once it has forwarded the request. Stateful SIPPS has a higher implementation complexity and lower processing performance than stateless ones, thus it is suitably deployed at the edge of the network. The operation exhibited by SIPPS resembles a *Domain Name Service* (DNS) recursive lookup whereby the DNS server accepts a query request and assumes the responsibility to track the answer to the question presented in the query or asserts the appropriate error.

SIP Redirect Server (SIPRS), unlike SIPPS, only provides address resolution service. It receives a request and only notifies the initiator, thus does not persist in the signaling path for the duration of the session establishment and does not fork any requests. The response embeds the destination address that the requestor should forward the request to either the next hop SNS's address or the SIP URL of the callee. The initiator then sends another request with the address that it received as the destination. This operation resembles a DNS iterative lookup whereby a host issues a query request informing the DNS server that it only requires to provide as much information as it has.

Registrar is typically co-located with a SIPPS or a SIPRS for ease of name resolution and user registration.

2.4 Details of Session Initiation Protocol (SIP) Messages

2.4.1 Syntax of SIP Headers

SIP headers are constructed from *Augmented Backus-Naur Form* (ABNF) [23] definitions and format. ABNF is a formal meta-syntax that expresses context-free grammars such that the syntax of each constituent is independent of the symbols oc-

curring before and after it in a sentence. An example of ABNF rule is **name** = **element1** | **element2**, **name** is the name of the rule, **elements** is one or more rule names, "=" means "is defined as", and "|"means "or". In this example, **name** will accept **element1** or **element2**.

As an illustration on how SIP header is constructed using ABNF. CSeq is a 32-bit integer that grows with chronologous order of the messages, for detecting out-of-order messages. **CSeq = "CSeq" ":" 1*DIGIT Method** is the ABNF definition of CSeq where **DIGIT = "0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"** and **Method = "INVITE"|"ACK"|"OPTIONS"|"BYE"|"CANCEL"|"REGISTER"**. Thus, **CSeq: 2300 ACK** is a valid instance, while **CSeq= 2300 ACK** and **CSeq: 2300 HI** are constructed incorrectly.

2.4.2 Message Body and Session Description Protocol (SDP)

Message Body is a textual media description based on *Session Description Protocol* (SDP) [24] for describing and negotiating audio, video and multimedia session options. Message Body allows recipients to gather sufficient session information as to participate in a session. These information includes the session name and purpose, the time(s) that the session is active, the media comprising and information to receive the media (session media stream addresses, ports and the codec supported). SDP is represented with a textual but compact format consisting several lines of the form **type=value** adhering to strict ordering and formatting rules. White space is not permitted on either side of "=" sign. **type** is always exactly one case-sensitive character, while **value** is a structured case-sensitive text string (composed of a number of fields delimited by a single SP or a CR) whose format depends on **type**. Each SDP consists a session section (begins with a "v=" line) followed by any number of media sections (starts with a "m=" line) and continues to the next media description or end of the SDP.

A typical usage of SDP is for session establishment between UAs. The initiator and recipient exchanges an INVITE request and a "200 OK" response listing respective supported media capabilities using one or more of the following media and attribute fields: *v=(protocol version)*, *o=(owner/creator and session identifier)*, *s=(session name)*, *t=(time the session is active)*, or *m=(media name and transport address)*.

2.4.3 Syntax of SIP Request Message

SIP request message is only generated by UACs, and has the format given in

Figure 2.4. **Request-Line** contains a field **Method** specifying the nature of the session in terms of services, addresses, and protocol features. **Method** defines six primary request messages for managing a basic session namely INVITE, BYE, OPTIONS, ACK, REGISTER, and CANCEL.

INVITE is used for session establishment or modification of existing SIP sessions. UAC of a caller sends an INVITE request to invite another user to a session. UAS of the callee responds with either an OK or a BYE (if the invitation to a session is accepted or not respectively). All the messages exchanged carry the same Call-ID.

```
Request = Request-Line Message-Headers Message-Body
Request-Line = Method Request-URI SIP-Version
Method =“INVITE”|“ACK”|“OPTIONS”|“BYE”|“CANCEL”|“REGISTER”
Request-URI = SIP-URL
SIP-Version = “SIP/2.0”
```

Figure 2.4: Syntax of SIP Request Message

ACK is used to complete the session establishment. It is sent by the initiator of the session of an original INVITE message after receiving the “200 OK” final response from the invited party. ACK request also announces the final session parameters negotiated during the exchange of messages and contains the same CSeq as the corresponding INVITE message.

OPTIONS is composed and handled exactly like an INVITE but for querying the other party's capabilities, without initiating or establishing a session.

BYE is used for session termination decided by any participants. Each BYE is acknowledged with an ACK request. BYE message is composed and handled exactly like an INVITE, but contains a higher CSeq as the corresponding INVITE.

REGISTER is used to update the Registrar with the user's current location when the user has relocated to another network or machine, and wants to receive future INVITE at the new network or terminal. After processing the request, Registrar replies with a “200 OK” to inform the registering user that the registration succeeded.

CANCEL is used to reset negotiations or to terminate pending request. It has the same Cseq numeric part, To and Call-ID header fields as the original request to be cancelled.

2.4.4 Syntax of SIP Response Message

SIP response message is generated by an UAS or a SNS as reply to a request from

an UAC, as depicted in Figure 2.5. **Status-Line** contains a pair of 3-digit integer known as **Status-Code** and associated textual **Reason-Phrase**, indicating the outcome of the request. Textual phrase offers a fallback mechanism to provide further human-readable information. The first digit defines the six classes of SIP response and the remaining two digits represent subclasses of each class. Table 2.1 summarizes the six broad defined categories of SIP responses (the first five classes are based on HTTP and the sixth is created solely for SIP) and corresponding 3-digit Status-Code. The last column provides descriptive comments of each class and whether a response class is provisional or final. A request generally generates one or more provisional responses (i.e. "180 Ringing") indicating progress of a received request message, but does not terminate a SIP request and then a final response (i.e. "200 OK") indicating whether the request succeeded or not.

```
Response = Status-Line + Message-Headers + Message-body
Status-Line = SIP-version Status-Code Reason-Phrase
SIP-Version = "SIP/2.0"
```

Figure 2.5: Syntax of SIP Response Message

Status Codes	Response Class	Comments
1xx	Informational	Provisional. Request received, continuing to be processed
2xx	Success	Final. Request was successfully completed.
3xx	Redirection	Final. Returns possible locations of an UAC
4xx	Client Error	Final. SNS cannot fulfill it
5xx	Server Error	Final. Server failed to fulfill an apparently valid request
6xx	Global Failure	Final. Request cannot be fulfilled by any SNS.

Table 2.1: Summary of SIP Response Status Codes

2.5 Illustration of Session Initiation Protocol (SIP) Operation

This section describes the different scenarios for successful registration and establishment of a two party session establishment based on no SNS, or a single SIPPS, or a single SIPRS, or both SIPPS and SIPRS.

2.5.1 Registration

Figure 2.6 shows the basic operation of SIP Registrar accepting registration requests (i.e. REGISTER message) from an UA specifying their current location, and then recording the registration information in a *Location Server* (LS) which is optional

and not specified in SIP, via a non-SIP protocol. Once the information is stored, Registrar replies the appropriate response "200 OK" to the UA. In the REGISTER request, the To and From headers contain the URL of UA, and the Contact header includes alternative addresses or aliases.

2.5.2 Direct Call Establishment

Both UAs (UA_A and UA_B) have sufficient information about the exact location (i.e. IP address) of each other. UA_A initiates a session by transmitting an INVITE request containing UA_B's SIP URL, and a session description providing sufficient information to participate in the session, directly to UA_B's IP address. UA_B accepts the invitation by returning a series of provisional responses "100 Trying", "180 Ringing", and a final response "200 OK" carrying similar session description directly to UA_A. UA_A then replies with an ACK request. Thereafter, both UAs can communicate directly by exchanging further SIP messages or data streams.

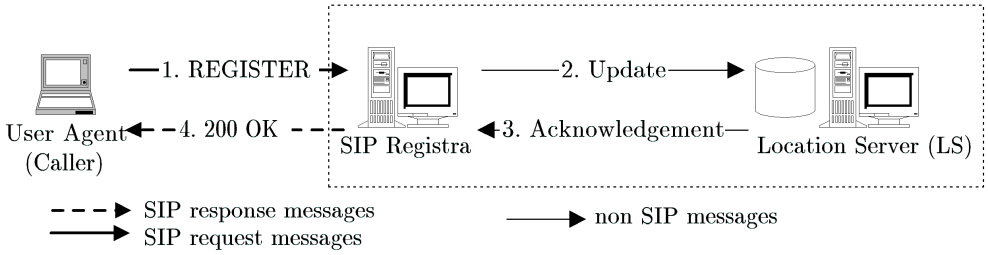


Figure 2.6: SIP Registrar and Registration

2.5.3 Call Establishment Using Proxy Server or Redirect Server

The operations of call establishment using a single SIPPS or SIPRS are illustrated in Figure 2.7 and Figure 2.8 respectively. It is assumed that both parties UA_A and UA_B have already registered with respective SNS before session establishment, and UA_A initiates contact with UA_B.

For call establishment using a single SIPPS, UA_A first issues an INVITE message (Step 1) to SIPPS. SIPPS queries (Step 2) its LS and receives (Step 3) possible locations of UA_B by using the SIP URL contained in the To field. SIPPS forwards the INVITE request (Step 4) to the addresses given by LS sequentially or simultaneously until UA_B receives the INVITE message successfully. UAS_B processes the INVITE request and passes it up to the end-user. If the end-user accepts the invitation, UA_B replies with a series of "100 Trying", "180 Ringing", and finally a "200 OK" using the

path flow (Step 5) and (Step 6) via SIPPS, and finally reaching UA_A. UAC_A acknowledges the receipt of the INVITE response by sending an ACK message (Step 7) and (Step 8) to finally reaching UAS_B. Both UAs can commence communication directly by exchanging further SIP messages or data streams.

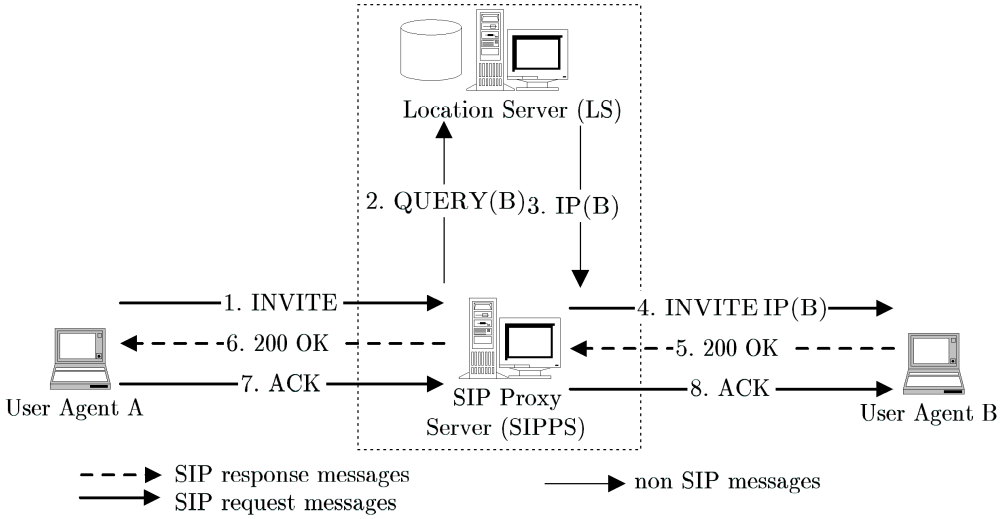


Figure 2.7: Call Establishment Using Single SIP Proxy Server

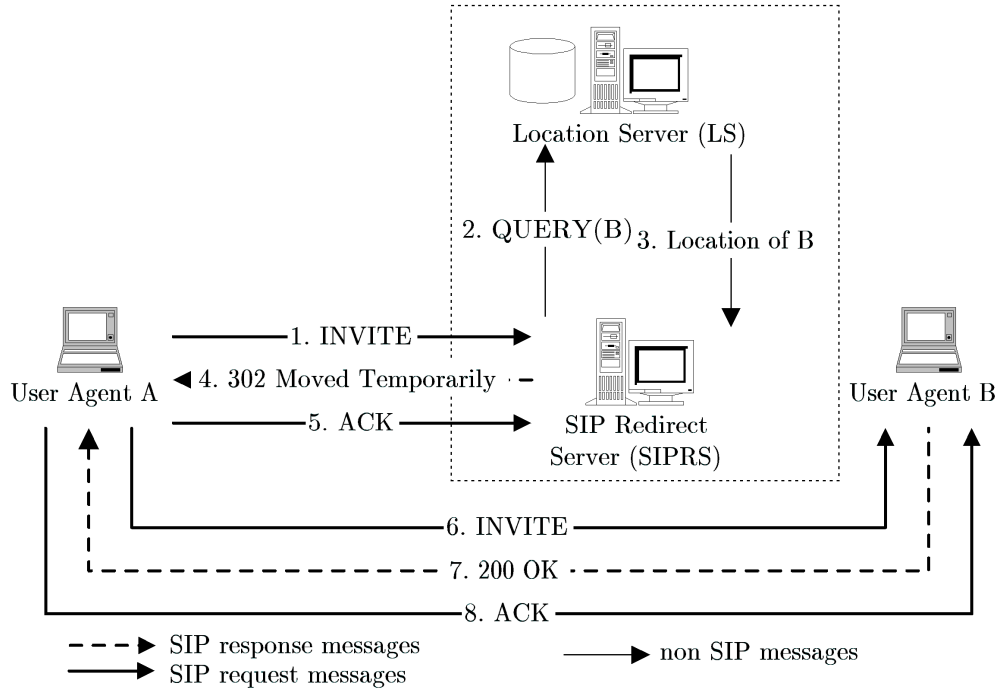


Figure 2.8: Call Establishment Using Single SIP Redirect Server

Call establishment using a SIPRS operates in a similar manner, except that UA_A

first sends an INVITE message (Step 1) to SIPRS. SIPRS queries (Step 2) its LS and receives (Step 3) possible locations of the user by using the SIP URL contained in the To field. SIPRS however returns a "302 Moved Temporarily" message (Step 4) to UA_A informing where UA_B can be located. UA_A then acknowledges with an ACK request (Step 5). UA_A issues a new INVITE request (Step 6) to UA_B, with the same call-ID but a higher CSeq to the address returned by the SIPRS. UAS_B processes the INVITE request and passes it up to the end-user of session acceptance. If the end-user accepts the invitation, UA_B replies with a series of "100 Trying", "180 Ringing", and finally a "200 OK" (Step 7) directly to UA_A. UAC_A then acknowledges the receipt of the INVITE response by sending an ACK message (Step 8) to UAS_B.

Both SIPPS and SIPRS do not establish or terminate sessions, but facilitates the exchange of SIP messages by receiving messages and forwarding them to the correct location or to better inform the initiator of possible locations of the other party.

2.5.4 Call Establishment Using Redirect Server and Proxy Server

Figure 2.9 illustrates the general operation for call establishment and session management between two UAs (UA_A and UA_B) using SIPRS/SIPPS, and message flow for a registration by UA_A. The network consists two domains with each having a stateful SIPPS (denoted as PS_1 and PS_2 respectively) to process SIP messages coming into or flowing out of the domain. In reality, a SIP signaling message may transverse several SIPPSs or SIPRSs until the message finally reaches the destined UA.

In registration phase, UA_A first issues a REGISTER message (Step 1) to its Registrar which records (Step 2) the user's location and other information in a LS using non-SIP protocol. LS acknowledges (Step 3) and Registrar replies with a "200 OK" message (Step 4) to UA.

In call establishment phase, UA_A establishes contact with UA_B by sending an INVITE message (Step 5) to its local SNS (PS_1). PS_1 checks the domain name of the callee and forwards the INVITE message (Step 6) to SIPRS presuming it has more informed or updated location of UA_B. SIPRS analyzes the user portion of UA_B's address and determines that UA_B is currently logged onto PS_2 and returns a "302 Moved Temporarily" message (Step 7) to PS_1. PS_1 then forwards (Step 8) the INVITE request to PS_2, which resolves (Step 9) the SIP URL of UA_B against its LS. LS returns (Step 10) the IP address of UA_B to PS_2. PS_2 then relays the INVITE message (Step 11) to UA_B. UAS_B processes the request and passes it up to the end-

user. If the end-user accepts the invitation, UA_B replies with a series of "100 Trying", "180 Ringing" and finally a "200 OK" using the path flow (Step 12), (Step 13) and (Step 14) via PS_1, PS_2 and ultimately to UA_A. This assumes that both stateful SIPPSs (PS_1 and PS_2) indicated in the INVITE request that they wanted to persist in the signaling path for the duration of session establishment. UAC_A then acknowledges the receipt of the INVITE response by returning an ACK message to UAS_B along the same transversed path. Thereafter, both UA can communicate directly by exchanging further SIP messages or data streams.

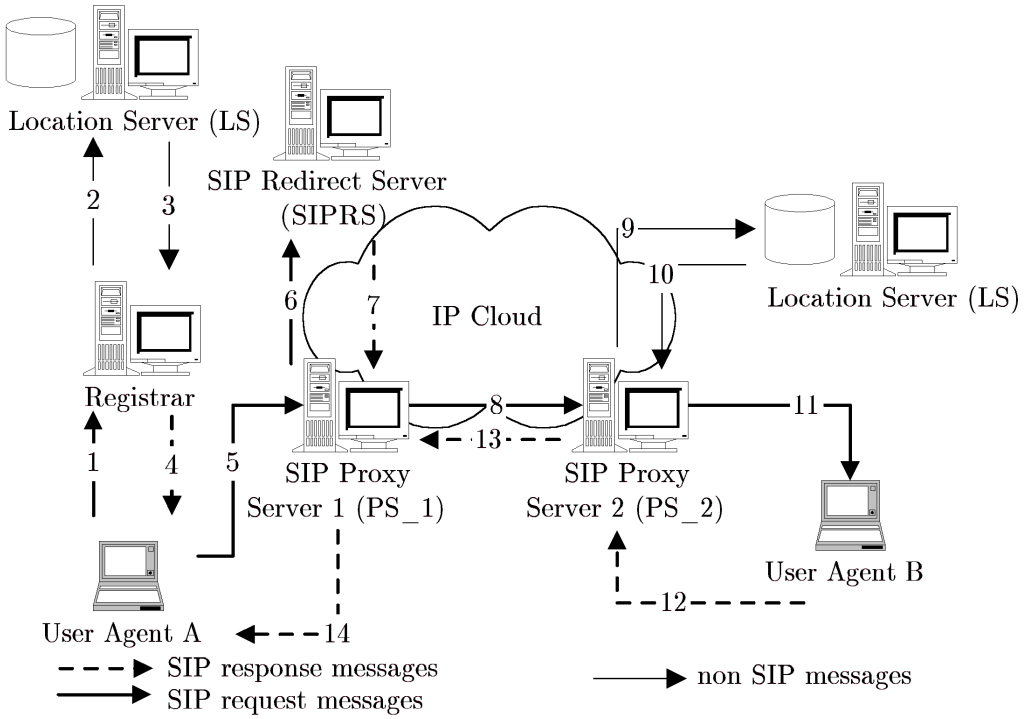


Figure 2.9: Call Establishment Using SIP Redirect Server and SIP Proxy Server

2.6 Summary

This chapter introduces basic background of Internet Telephony, multimedia data and control architecture incorporating protocols for session management, transport of real-time data and multimedia session description. It also covers SIP in relation to its architecture, logical entities, and messaging methods. The application of SIP in Internet Telephony is illustrated using several SIP call examples.

Chapter 3.
An Implementation of Session Initiation Protocol (SIP) for NS-2

Section 3.1 provides the motivation for implementation of SIPsim as an extension to open source network simulator NS-2. Section 3.2 and 3.3 elaborate on the layered architectural design and implementation of SIPsim. SIPsim consists software modules for SIP Message Parser, SIP Message Generator, *User Agent* (UA) and *SIP Network Server* (SNS). Section 3.4 covers validation of SIPsim based on a specification conformance test-suite of selected scenarios and corresponding results.

3.1 Overview of SIPsim

SIPsim is a minimal implementation of SIP protocol stack, designed as an extension to an open source network simulator NS-2 [25]. To the best of author's knowledge, SIPsim is the first treatment for experimental and research investigations of SIP to gain insights into SIP internalities and functionalities. SIPsim is a critical necessity for a research platform to analyze, evaluate, and study the functionality and behaviour of SIP and for prototyping advanced value-added services like mobility support and in-depth understanding of integrating SIP with RSVP, without incurring costly test-bed setup and managing complex implementation issues. SIPsim consists software modules for SIP Message Parser, SIP Message Generator, *User Agent* (UA) and *SIP Network Server* (SNS) with available methods REGISTER, INVITE, BYE, and ACK, and responses "180 Ringing", and "200 OK".

SIPsim is a discrete-event driven simulation of minimal implementation of SIP based on a split-programming model [26] depicted in Figure 3.1. The implementation of SIPsim consists two main components, namely **Building blocks** and **Glue**. **Building blocks** are written in C++ language to provide a collection of reusable object oriented software components to generate Node and Links. Node is an abstract object that composes a collection of Agents (protocol end-points) and Classifiers (packet demultiplexers) which can be subclassed into Address Classifier (demultiplexes based on ad-

dress) and Port Classifier (demultiplexes based on ports). Address Classifier is interconnected to Links, which encapsulates queue and delay objects. Port Classifier communicates directly with Agents. Links implements point-to-point wired link, multiaccess LAN, wireless and other broadcast media. **Glue** is written in an object-oriented scripting language Otcl [27] to encapsulate the **Building blocks** for interfacing with simulation scenarios. Each runtime SIPsim simulation scenario defining arbitrary network topologies is described in a textual Tcl script, this is taken as an input to the **Glue** and **Building Blocks** for further processing, and for generating an output trace file and animation trace file. The former records the details of all the movement of data packets between nodes with the exact time and sequence number, used for plotting or further analysis. The latter contains graphical scenario information meaningful for visualization of packet flows, protocol states, and as a debugging tool.

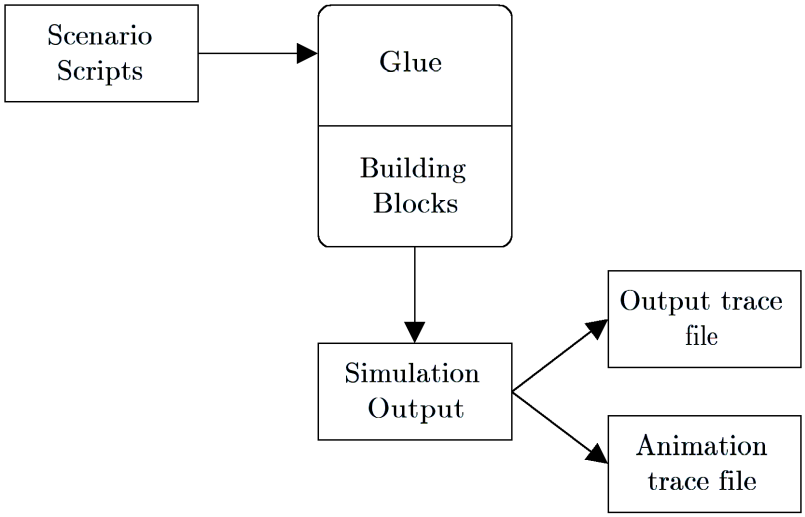


Figure 3.1: SIPsim Architecture Based on Split-Programming Model

A typical SIP experimental test bed would require minimal three terminals functioning as two UAs and a *SIP Proxy Server* (SIPPS)/Registrar, numerous public-domain open source implementations of SIP [28,29] are available for practical usage and installation based on specific Operating System and programming languages. However, such setup proves hardware cost-ineffective for academics and research purpose for performance evaluation like call-setup delay. Even in-house implementation of SIP requires proper handling of programming issues like threading and memory allocation as witnessed from design architectures [30,31]. The complexity of SIP is further highlighted in [32,33] which define behavioural automata for SIP UA and SIPPS using dia-

grammatic meta language e.g. Unified Modelling Language for direct translation into high-level languages like C++ and Java.

3.2 Layered Design Architecture of SIPsim

Figure 3.2 depicts the design architecture of SIPsim, which comprises three inter-related layers namely **Network Communication Layer**, **Core SIP Stack Layer**, and **Logic Layer**. A layered approach for SIPsim implementation was adopted for two main reasons. Firstly, to obtain a structured model for SIPsim with clearly defined interfaces. This facilitates future upgrade and modification of either the generic parts or the message handling instances. Secondly, to ensure SIPsim is implemented independently from NS-2 as to achieve maximal compatibility with NS-2 previous or future versions.

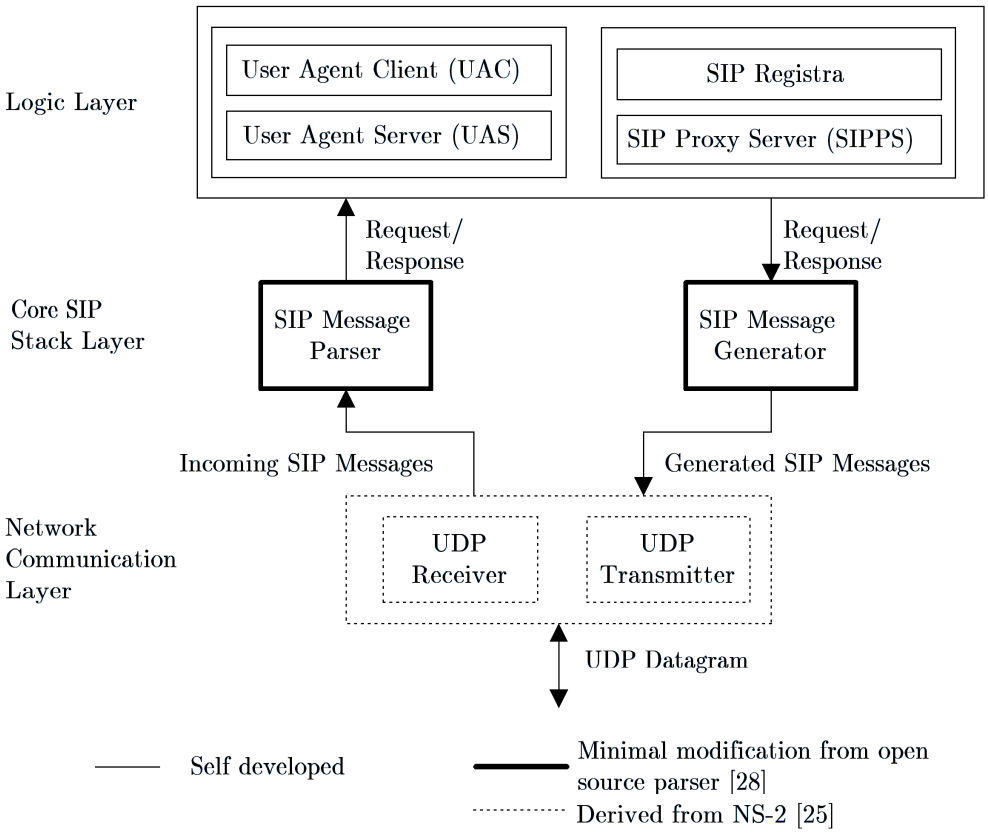


Figure 3.2: Layered Design Architecture of SIPsim Stack

An overview of the layered approach is as follows. Network Communication Layer listens for arriving packets and checks the packet header if it contains a SIP message. If it does, the packet would be next handled by the Core SIP Stack Layer that parses the incoming SIP message and allows the Logic Layer to access or modify the SIP

header fields for manipulating the message information and maintaining the call states. Logic Layer, depending on the states of the UAs or SNS makes decisions based on this information and other information it gathers from other resources, it would then invoke methods in the SIP Message Generator to format and create a SIP message. Once a new SIP request or response message is formed, the lower Network Communication Layer transmits the SIP message to the appropriate destination.

Network Communication Layer implements UDP related communication components (provided mainly by NS-2) for two basic functions. Firstly, it listens for arriving packets on a predefined UDP port over the physical links and checks the packet header if it contains any SIP message. Secondly, it transmits SIP message upon user actions or incoming messages to the appropriate destination as a payload of UDP.

Core SIP Stack Layer comprises SIP Message Parser and SIP Message Generator as two logical components. It interfaces with UA and SNSs for normal parsing and building of SIP messages based on *Augmented Backus-Naur Form* (ABNF) format [23], maintains status information, and processes SIP requests and responses. Both SIP Message Parser and SIP Message Generator are based extensively on an open source parser [28]. SIP Message Parser implements interfaces for each supported header while SIP Message Generator creates messages. The operation of Core SIP Stack Layer is as follows. After receiving an incoming SIP message from the Network Communication Layer, SIP Message Parser first validates that the received SIP message is well formed and properly structured according to SIP specification. It parses the provided stream to split it into its various name/value pair and other meaningful information, and then stores them into an instance of a class called SIPMessage. SIPMessage is a class object that provides data-centric view of SIP message via methods for accessing all header fields, the message body, or parts of them if necessary. SIPMessage provides to the logic layer information ranging from the message types (INVITE, ACK or BYE for requests, and the response code for responses) to a list of URI's to proxy to. In addition, SIP Message Parser provides a *Session Description Protocol* (SDP) compliant parser for processing the media description in the SIP message body. The function of SIP Message Generator is the reversed of the latter i.e. the creation of well-formed SIP request and response based on instructions passed and then concatenates any necessary SIP headers.

Logic Layer abstracts applications residing on terminals responsible for initiating,

managing and terminating actual IP Telephony sessions, or registration. Logic Layer maintains state machine, creates provisional and final responses, and provides TCL-based interfaces to manually invite a caller, to register with Registrar or to terminate a session. Logic Layer is mainly self-developed implementing the call state of both UA and SNS to handle different events based on the information returned by SIP Message Parser from lower layer. Call state maintains the state of each SIP session and defines the appropriate action and transition in response to the external events. Events includes invoking methods in the SIP Message Generator to place a specific header in a message, to replace a header in a message with a new one, to delete a header from a message, or to decide whether the body of the message should be duplicated, updated, or removed. UA and SNS are derived from Agent class provided by NS-2. UA implements *User Agent Server* (UAS) and *User Agent Client* (UAC) as two separate logical components, while SNS implements SIP Registrar and SIPPS.

3.3 SIPsim Implementation

The implementation of SIPsim is depicted in Figure 3.3. *User Agent* (UA) and *SIP Network Server* (SNS) are implemented as objects **Agent/SIPUA** and **Agent/SIPServer** respectively, using Agent as the base class.

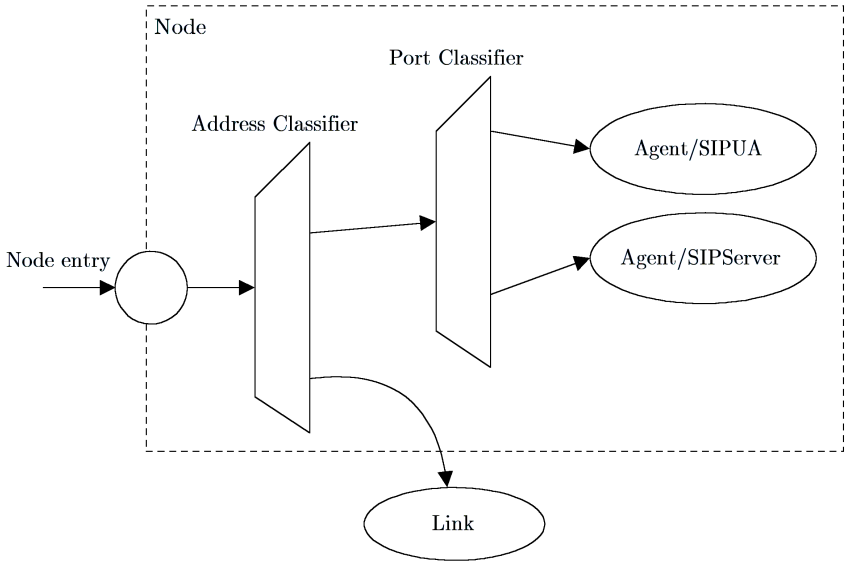


Figure 3.3: SIPsim Implementation

The internal structure of **Agent/SIPUA** is partitioned into two separate logical components: UAS and UAC. The former is responsible for receiving SIP request and

responding with SIP response. The latter only transmits SIP request and accepts SIP response. **Agent/SIPServer** is defined as a SIP network entity that handles session management signaling exclusively, but does not participate in actual data transmission. **Agent/SIPServer** is functionally divided into SIPPS and SIP Registrar that understands INVITE, ACK, OPTIONS, and BYE requests. It parses and generates as appropriate, the Call-ID, Content-Length, Content-Type, CSeq, Expires, From, Contact, To, and Via headers. It also echoes the CSeq and Time headers in the response.

3.3.1 Minimal Implementation of User Agent Server (UAS)

The top-level operation of UAS is shown in Figure 3.4. UAS is always in a listening mode waiting for incoming packet. When UAS receives a packet, it checks the packet header if it contains a SIP Message. If it does not contain, then discards it. Else it parses the SIP Message, stores the header fields and values, and depending whether the **Agent/SIPUA** is in the WAITER, CALLER, or CALLEE mode, different events will occur. When **Agent/SIPUA** is in neither the WAITER, CALLER, nor CALLEE mode, the SIP message is discarded. UAS currently understands the following requests: ACK, BYE, CANCEL, INVITE, REGISTER, and is able to parse and generate as appropriate, the Call-ID, Content-Length, Content-Type, CSeq, Expires, From, Contact, To, and Via headers. It is also able to echo the CSeq and Time headers in the response. In addition, a Content-Length header is present in every message, specified to zero if exists no SDP message body. The content length calculations assume that each line of SDP terminates with both a CR and a LF character. The SDP message body contains the session name, purpose, media and timing information, and the bandwidth required for the session establishment.

Figure 3.5 depicts UAS in the WAITER mode where **Agent/SIPUA** is waiting for an INVITE message or a “200 OK” (confirmation of registration). If **Agent/SIPUA** receives an INVITE request, it transits to the CALLEE mode and check if the SDP Message Body is erroneous and whether it supports the codec. It replies a series of provisional response “100 Trying”, “180 Ringing” and a final response “200 OK” if it accepts the session and prepares a RTP connection for multimedia transfer. Else, it sends a “600 Busy Everywhere” response to indicate a decline.

Figure 3.6 depicts UAS in the CALLER mode where **Agent/SIPUA** is listening for a BYE, or a CANCEL, or a “180 Ringing”, or a “600 Busy Everywhere”, or a “486 Busy Here”, or a “200 OK”, or a “302 Moved Temporarily”, or a “400 Bad Request”,

or a "606 Not Acceptable". When **Agent/SIPUA** receives a BYE request indicating termination of existing session, it resets to the WAITER mode, terminates the RTP session, and replies with a BYE request. **Agent/SIPUA** also resets to the WAITER mode upon receiving a CANCEL request (indicating a non-acceptance of session), a "600 Busy Everywhere", or a "486 Busy Here", or a "302 Moved Temporarily", or a "400 Bad Request", all of which indicate connection failures. Upon receiving a "180 Ringing", **Agent/SIPUA** alerts user with a "Ringing" tone that the session is still pending. When **Agent/SIPUA** receives a "200 OK" response, this confirms the session establishment, it first checks whether the session is active. If it is not, prepares a RTP connection for multimedia transfer, replies with an ACK request, and commences with a RTP session. Else, **Agent/SIPUA** simply sends an ACK request. Lastly, when **Agent/SIPUA** receives a "606 Not Acceptable", it responds an ACK for acknowledgment.

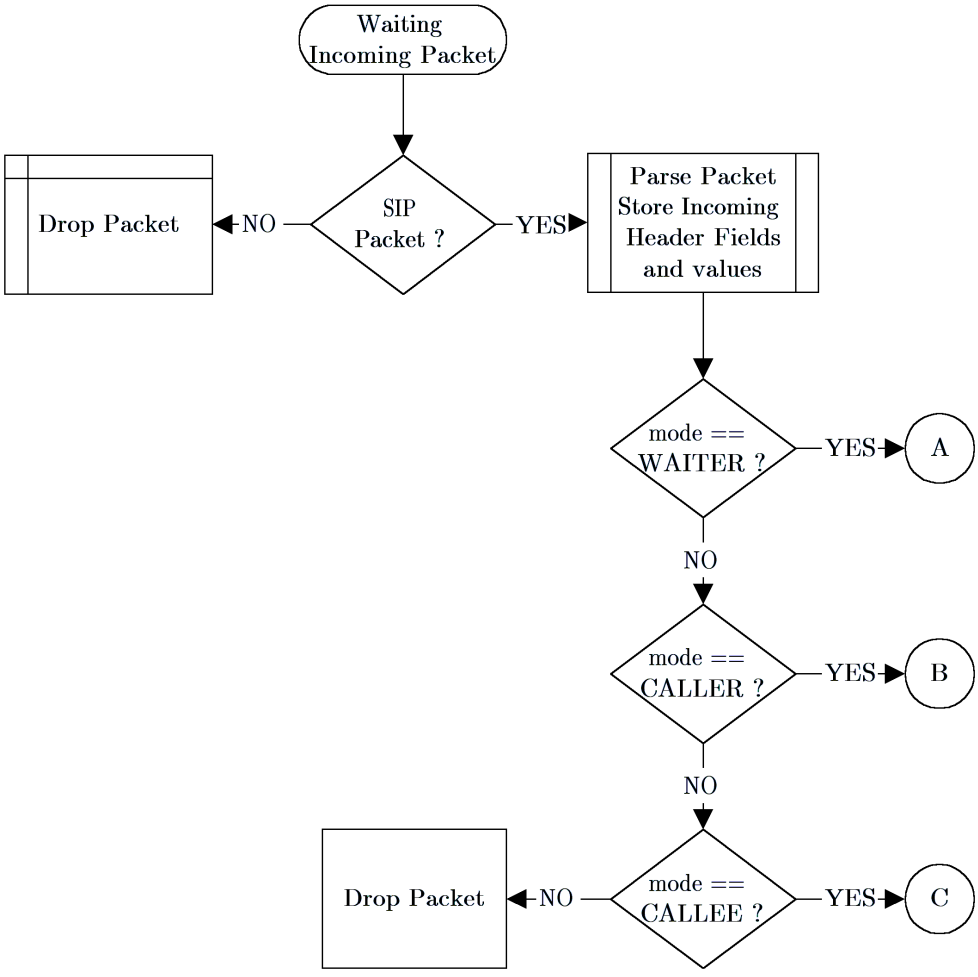


Figure 3.4: Operation of User Agent Server (UAS)

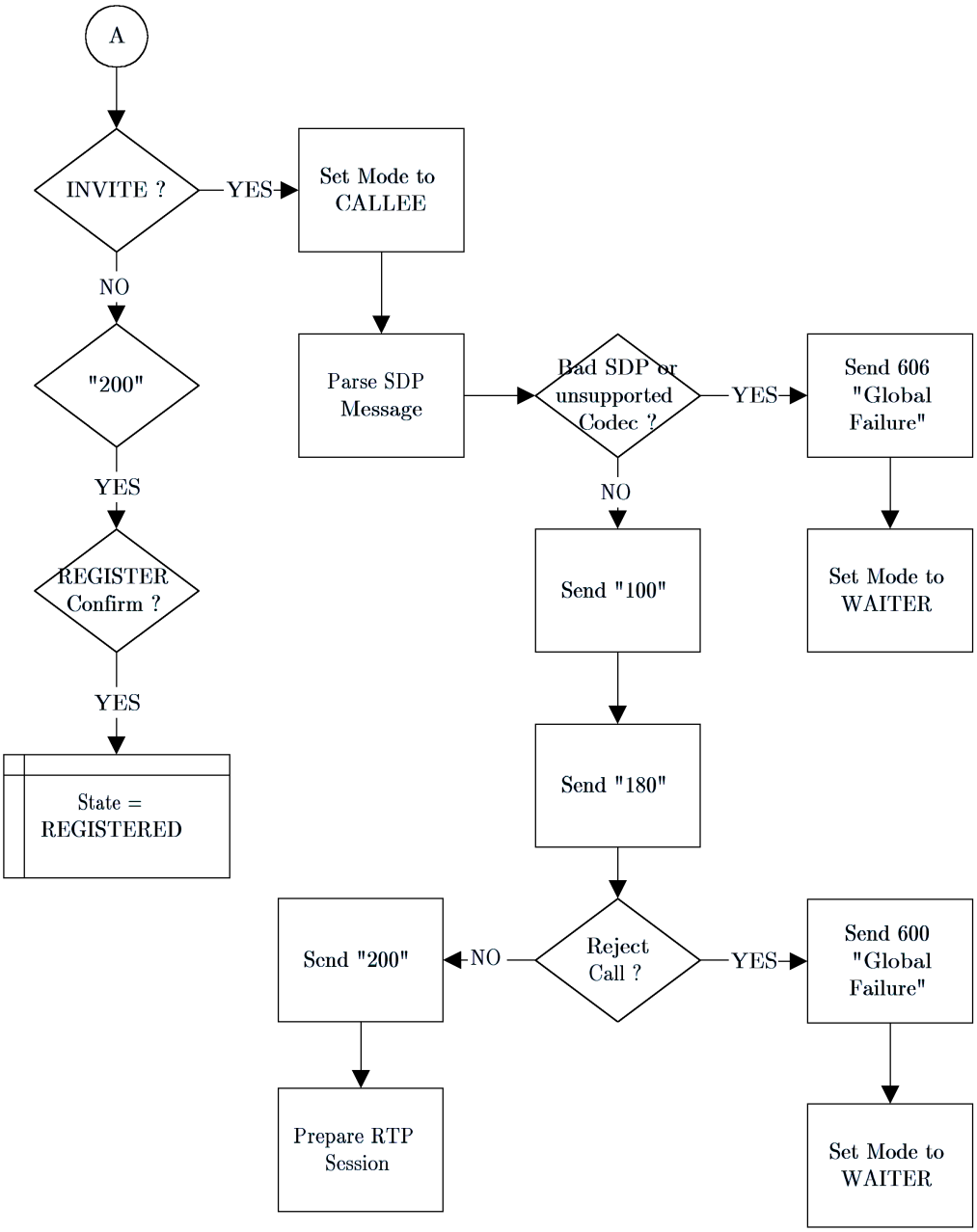


Figure 3.5: Operation of User Agent Server (WAITER Mode)

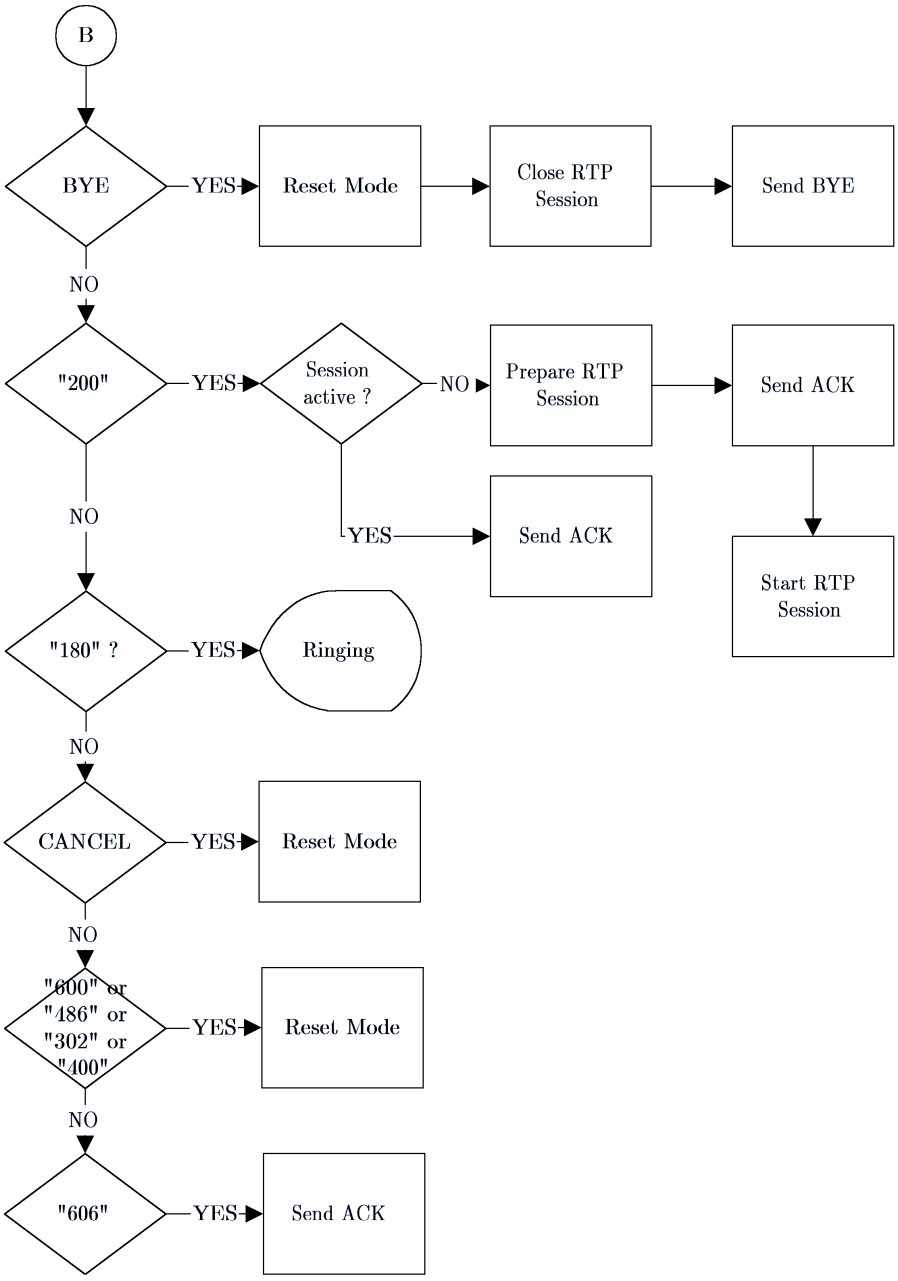


Figure 3.6: Operation of User Agent Server (CALLER Mode)

Figure 3.7 depicts UAS in the CALLEE mode where UA is listening for a CANCEL message indicating premature termination of a session, or a BYE message indicating the termination of a session, or an ACK message for the confirmation of session establishment. When it receives a CANCEL, UA simply resets to the WAITER mode. Upon receiving a BYE, it replies with a SIPMessage with Status-Code specified to "200 OK", and terminates the RTP connection of active session. When UA receives an

ACK request, it checks whether the SDP of the ACK and previously received INVITE contain supported codec parameters. If supported codec parameters exist, then UA replies with a SIP Messages, sets Status-Code to "606 Global Failure", and transits to WAITER mode, this avoids cases of conflicting codec parameters. If neither the ACK nor previously received INVITE carries supported codec parameters, then specifies mode to WAITER and initiates a RTP connection for active session.

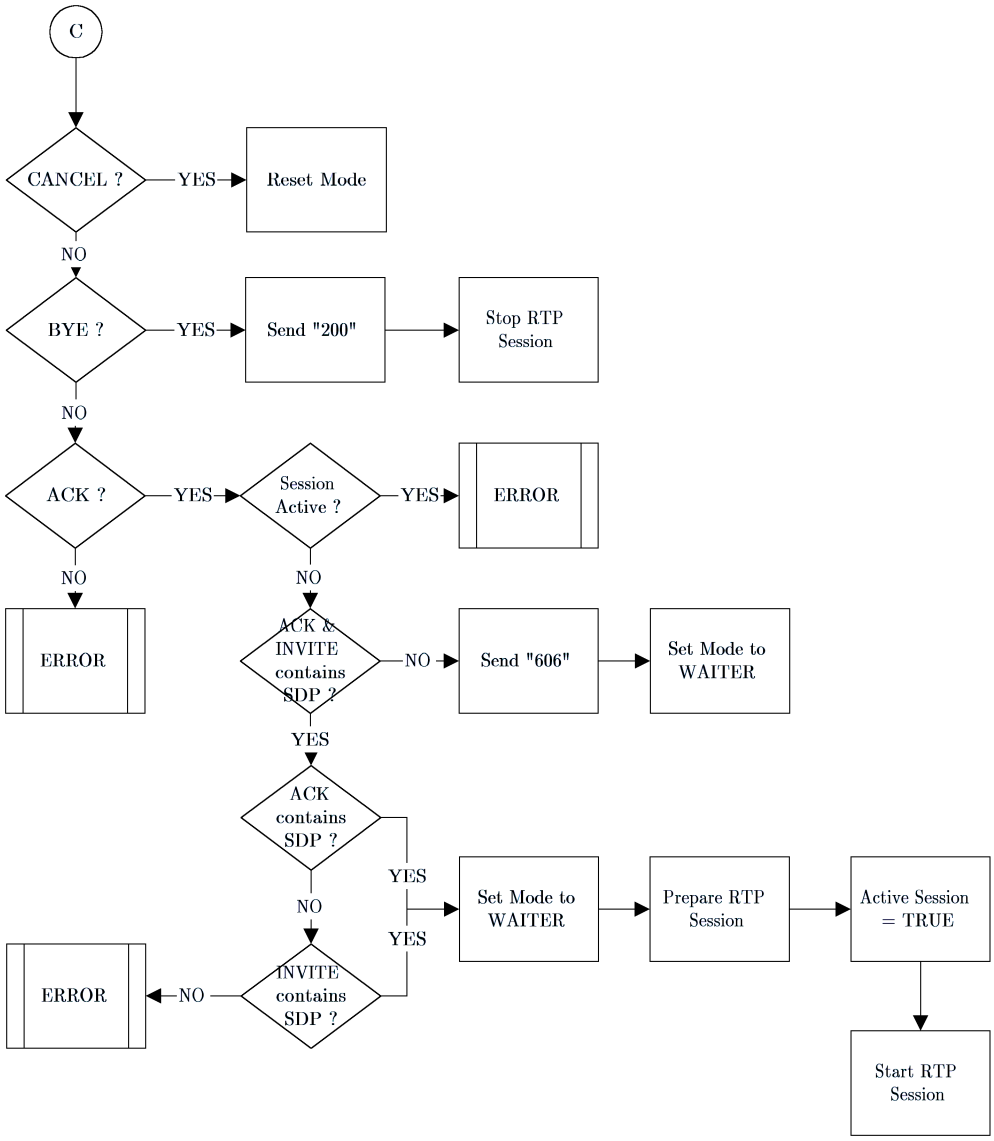


Figure 3.7: Operation of User Agent Server (CALLEE Mode)

3.3.2 Minimal Implementation of User Agent Client (UAC)

The operation of UAC is illustrated in Figure 3.8. UAC performs registration

upon power up by sending a REGISTER request and setting the State to REGISTERING or by sending an INVITE to whoever the user requests and sets the State to INVITING. A Contact header is included with every INVITE message. Currently, UAC is capable of generating INVITE and ACK requests, providing supports for BYE method to allow the interruption of a pending call attempt, generating and parsing the Call-ID, Content-Length, Content-Type, CSeq, From and To headers, understanding SDP, and recognizing the Status-Code classes 1 through 6 and act accordingly.

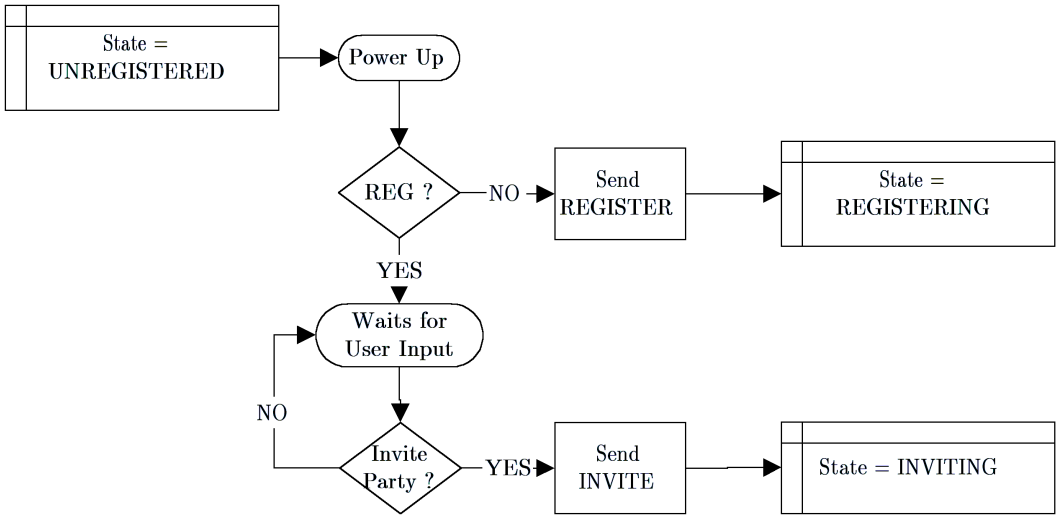


Figure 3.8: Operation of User Agent Client (UAC)

3.3.3 Minimal Implementation of SIP Proxy Server (SIPPS)

SIPPS provides name resolution and forwarding capability to the correct destination. It receives a request, resolves the address that it should send, then either forwards the request directly to the current location of the callee or forks the request to another SNS that might be better informed about the actual location of callee. The internal structure of SIPPS is illustrated in Figure 3.9. Whenever SIPPS receives a SIP response, it first checks if the topmost "Via" field matches one of its addresses. If it does, it removes the topmost "Via" field and checks the address in the next "Via" field, the packet is forwarded to the address listed in the "maddr" tag. Else, it drops the packet. For incoming SIP requests, SIPPS first checks if its address is already in the VIA-header list. If it does contain, the SIP message is discarded to prevent loops i.e. the request should not be forwarded by SIPPS. Else, SIPPS appends a new "Via" header field containing its address to the end of the VIA-headers list. This enables the

response to transverse the same way back as the request. In addition, the final entry in a Route header is always the Contact information obtained from the INVITE or the "200 OK" messages.

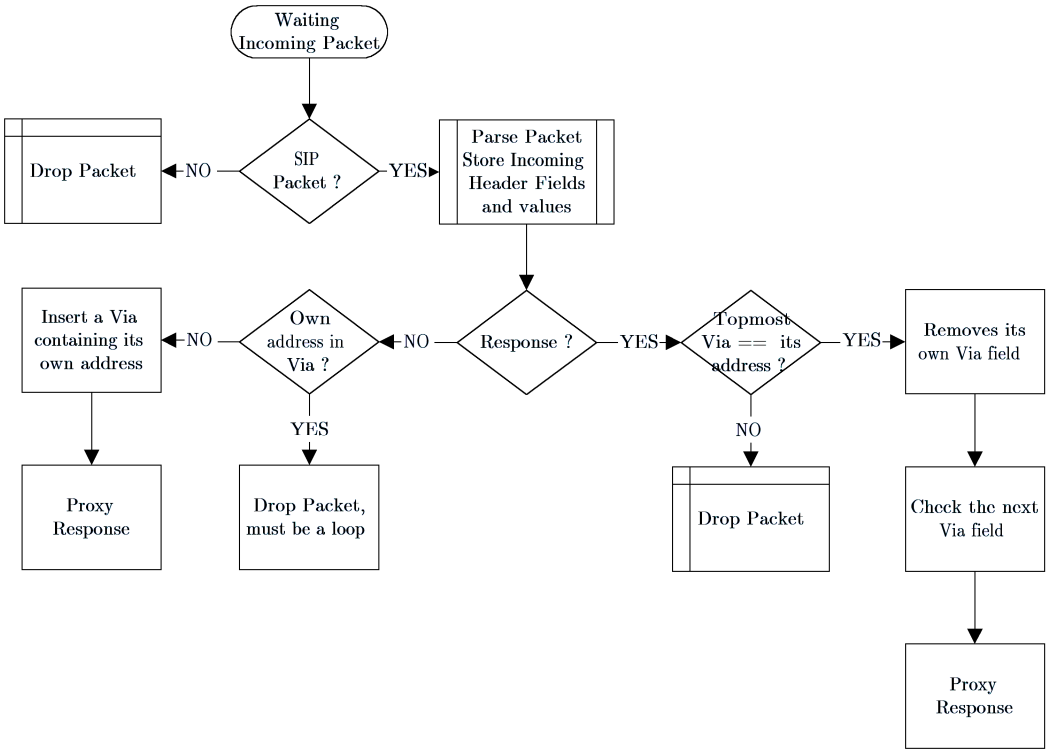


Figure 3.9: Operation of SIP Proxy Server (SIPPS)

3.3.4 Minimal Implementation of SIP Registrar

SIP Registrar provides user registration by accepting REGISTER from UA specifying their current location, and then stores the registration information in LS via a non-SIP protocol. Once the information is stored, the SIP Registrar returns a "200 OK" to UA. The internal structure of SIP Registrar is depicted in Figure 3.10. Any REGISTER request received, the To and From headers contains the URL of UA, and the Contact header includes alternative addresses or aliases. Before performing a normal registration and updating the location database, Registrar checks whether the REGISTER message contains Contact field. If it does not contain any Contact field, Registrar retrieves the current list of user contacts and inserts them in the "200 OK" response. If it does contain Contact field and the expiration period is specified to zero second, indicating that the user cancels a registration. Then, Registrar clears user contact list and replies SIP Message with Status-Code of "200 OK" to user. Otherwise,

the REGISTER message indicates a normal registration, Registrar updates the user contact list, and replies with a "200 OK" response.

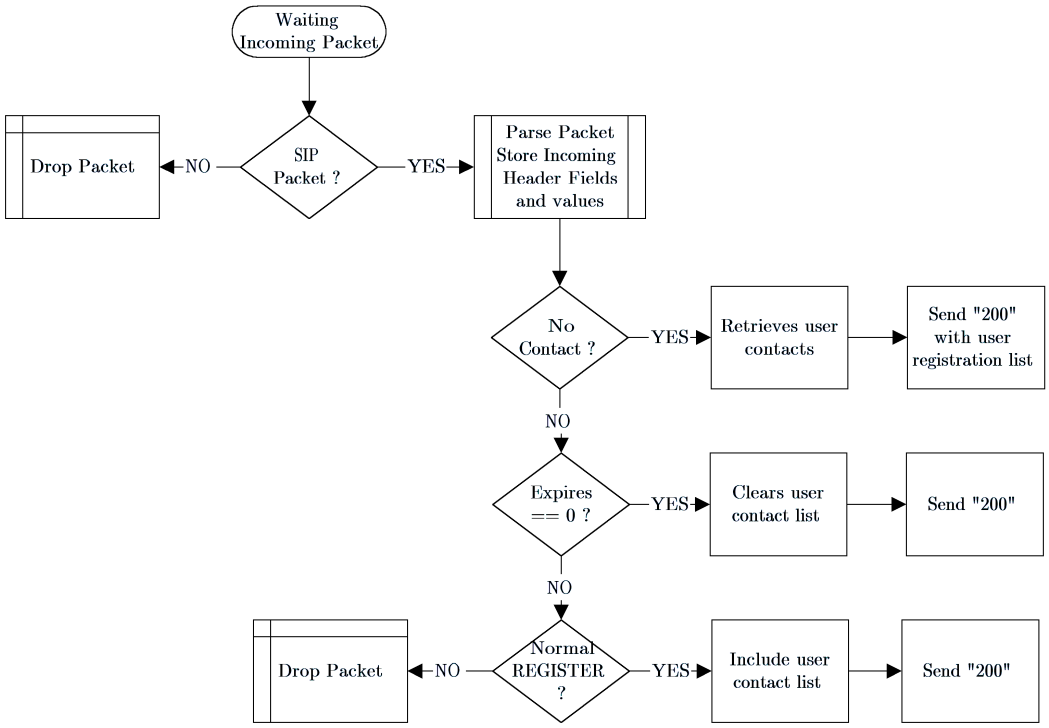


Figure 3.10: Operation of SIP Registrar Server

3.4 Protocol Conformance Test

SIP Call Flow Examples [34] defines a set of test scenarios for protocol conformance testing of SIP which includes verification, validation, and demonstration. This ensures a minimal set of functionalities of SIPsim along with SIP entities including *User Agents* (UAs), *SIP Proxy Servers* (SIPPS), and Registrar, accomplished and conformed to the functions defined in the SIP specification.

3.4.1 Test Environment and Scenarios

Test environment is depicted in Figure 3.11 which consists two UAs (acting as caller and callee), a Registrar, and two SIPPSs. It is assumed that Registrar coexists with LS. Information presents in the Request-URI (i.e. a SIP URL) and From header is sufficient to determine to which SNSs the message should be routed. SIPPSs can effectively force subsequent request within a session to revisit the same server by inserting a Record-Route header into the first request. However, in the simulation, SIPPSs

do not insert Record-Route headers into requests as it is assumed that there exists a signaling path for future message exchanges. All call flows are carried over UDP instead of TCP in a wireline environment. The ordering of SIP headers of SIP messages is arbitrary, however the commonly used ones like To, From, Contact are placed prior to others for efficient processing.

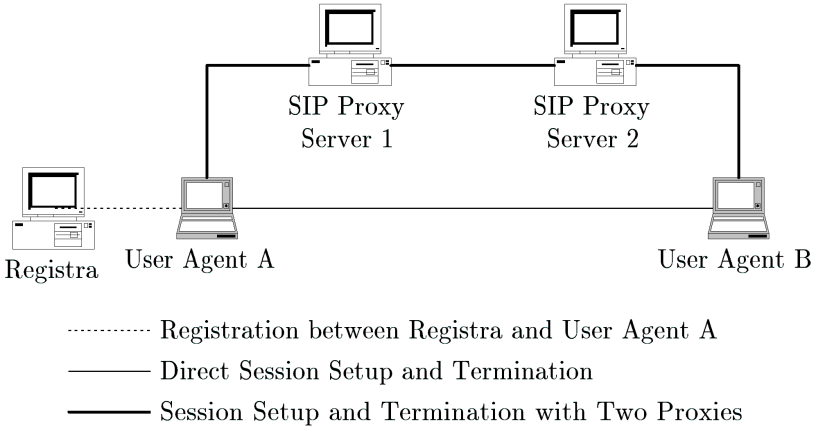


Figure 3.11: Test Environment

Selected test scenarios include registration between UA A (denoted as UA_A) and Registrar, direct call-setup and release directly between two UAs, and indirectly with two SIPPSs along the session path. Details of each test is as follows.

Registration (denoted as **R**): SIP registration is a fundamental method providing the mechanics for locating registered UAs. For example, SIPPS uses the information to route incoming messages. This test-suite includes testing the functionalities of Registrar and the completeness of REGISTER messages. Three test cases are included. (**R1**) Normal successful registration: UA_A first sends a REGISTER request to Registrar, which registers the UA_A in its database and returns a "200 OK" to the UA_A. The response includes the UA_A current contact list in Contact headers. (**R2**) Query for current contact: UA_A first sends a REGISTER with no Contact headers indicating it wishes to query for its current contact list. Registrar replies with a "200 OK" containing the current registration list if there is any, in the Contact headers. (**R3**) Cancelling registration: UA_A cancels registration with the Registrar by sending a REGISTER request to the Registrar. The request has an expiration period of zero and applies to all existing contact locations (if Contact header set to "*"). Registrar clears the current contact list and returns a "200 OK" to UA_A.

Direct Session Setup and Termination (denoted as **D**): A successful session setup

and termination between two UAs (denoted as UA_A and UA_B respectively) following the stated operations: UA_A establishes a session with UA_B directly by transmitting an INVITE. UA_B responds with a series of "100 Trying", "180 Ringing", and finally a "200 OK" to UA_A. UA_A then returns an ACK to UA_B. RTP streams are established between both parties. UA_B decides to terminate session and sends a BYE message to UA_A. UA_A replies with a "200 OK" to UA_B. Figure 3.12 depicts the script to generate this scenario, which is graphically illustrated in Figure 3.13 using [35].

```
#Create nodes n0 and n1
set n0[$ns node]
set n1[$ns node]

#Create two SIP UA Agents and attach to nodes
set ua0 [new Agent/SIPUA $n0]
set ua1 [new Agent/SIPUA $n1]

#Initialize both SIP UA
$ns at 0.3 "$ua0 initialise alpha cwc.edu.sg UDP"
$ns at 0.3 "$ua1 initialise beta cwc.edu.sg UDP"

#UA0 invites UA1
$ns at 0.4 "$ua0 invite beta cwc.edu.sg"
$ns at 1.4 "$ua0 HangUpPhone"
$ns at 2.0 "finish"
```

Figure 3.12: Sample Script for Direct Session Setup and Termination

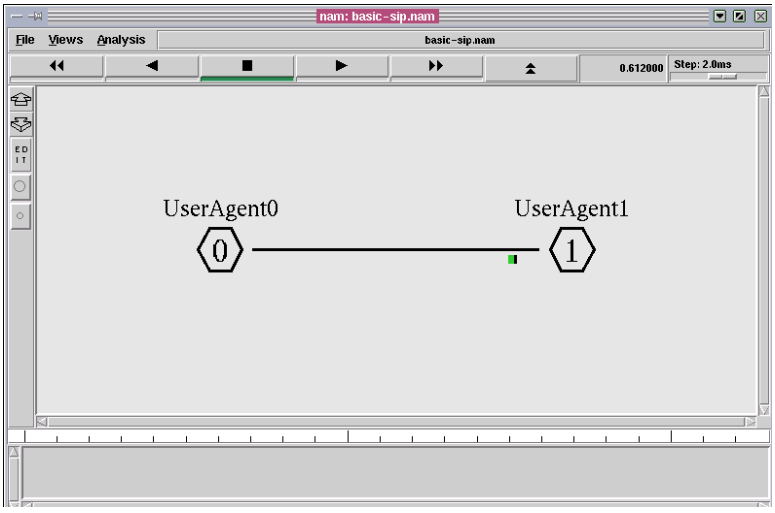


Figure 3.13: Direct Session Setup and Termination

Session Setup and Termination with Two Proxies (denoted as **I2**): UA_A completes a session to UA_B using two SIPPSs Proxy 1 and Proxy 2, by first sending an INVITE message to Proxy 1 that forks to Proxy 2. UA_B responds in a similar manner to the

direct session setup. RTP streams are established between UA_A and UA_B. UA_B later decides to terminate the session by initiating a BYE message.

3.4.2 Performance Test Results

Table 3.1, Figure 3.14, and Figure 3.15 summarize the conformance results, details and messaging flow (payload indicated in Italics) for each individual test scenarios, the detailed output are referenced in Appendix A.

Test	Status	Comments
R1	Passed	Single Contact used; Worked as stated in [34]
R2	Passed	Ditto
R3	Passed	Ditto
D	Passed	Worked as stated in [34]
I2	Passed	Worked as stated in [34]

Table 3.1: Summary of SIPsim Simulation

3.5 Summary

This chapter provides the motivation for the design and implementation of SIPsim as an extension to open source network simulator NS-2. SIPsim provides thorough evaluation and clear understanding of SIP internalities and functionalities. It then presents the layered architectural design and implementation of SIPsim. SIPsim consists software modules for SIP Message Parser, SIP Message Generator, User Agent and SIP Network Server with available methods REGISTER, INVITE, BYE, and ACK, and responses “180 Ringing”, and “200 OK”. Finally, this chapter also covers validation of SIPsim based on a specification conformance test-suite of selected scenarios and corresponding results.

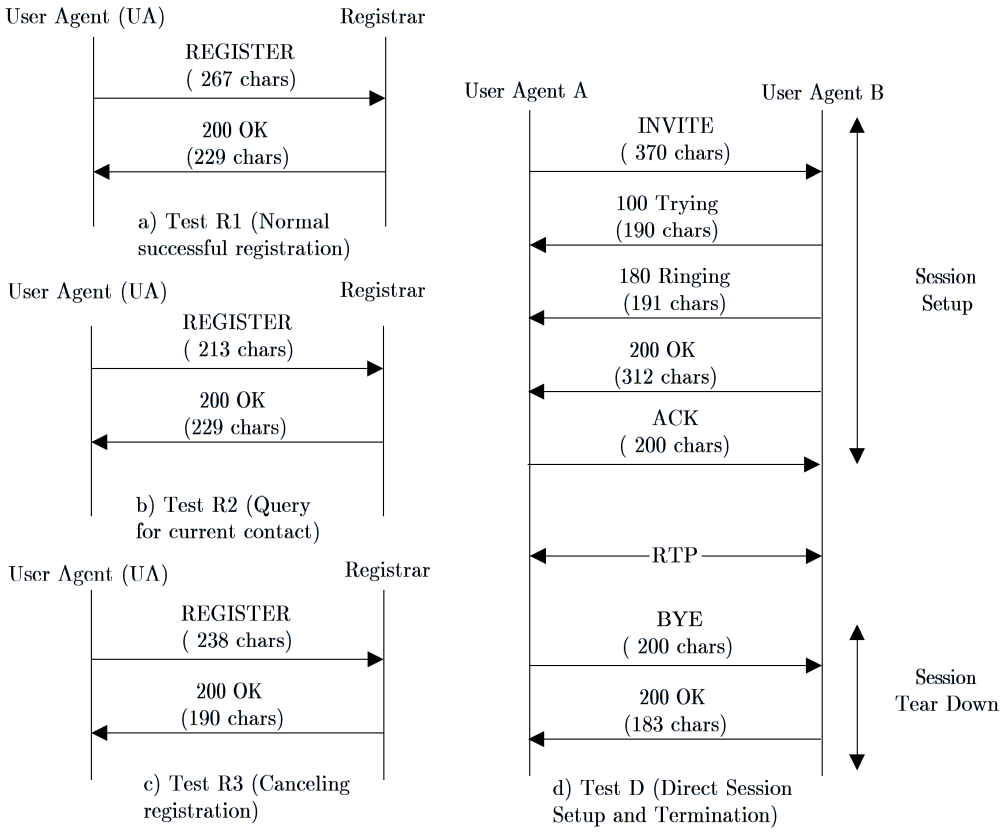


Figure 3.14: Summary of Test Scenario R and D

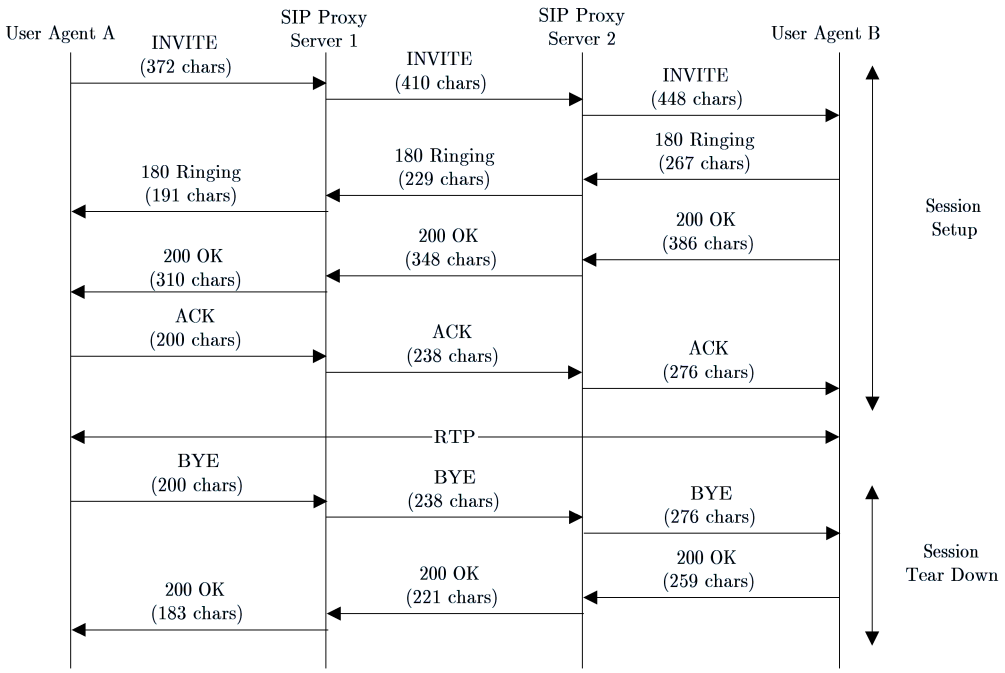


Figure 3.15: Summary of Test Scenario of I2

Chapter 4.
Background and Related Work on Mobility

Section 4.1, 4.2, and 4.3 cover the definition and components constituting mobility, literature survey of related work on current solutions and issues of supporting mobility in the Internet from different perspectives of network, transport, and application layer, and extensively describes *Mobile IPv6* (MIPv6) and *Session Initiation Protocol* (SIP) support for mobility (MSIP). Section 4.4 elaborates description and discussion of MIPv6 in terms of its data structures and major operations. Section 4.5 summarizes literature survey of mobility support using SIP for both terminal and personal mobility based real-time and TCP-based communication.

4.1 Definition of Mobility

Mobility in the Internet [36] as summarized in Figure 4.1, is differentiated into three categories namely *Terminal Mobility* (TM), Session and Service mobility, and *Personal Mobility* (PM). This thesis concentrates mainly on PM and TM which are related to pre-session and mid-session mobility respectively, noting the major difference between them is the perception of the communication end-point. The former guarantees reachability of a *Mobile User* (MU) residing on a *Mobile Host* (MH) while the latter maintains ongoing session during MH's handoff.

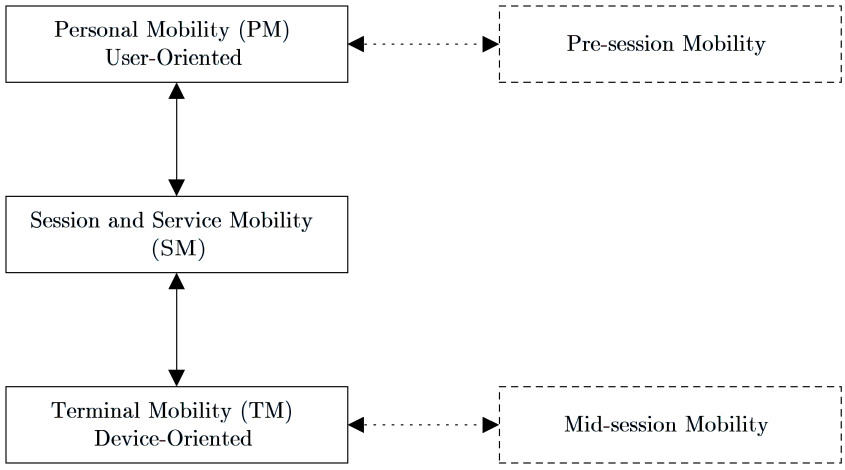


Figure 4.1: Complete Mobility Management Model

TM [37] is a device or host-oriented mobility model enabling a MH to relocate between different points of attachment to the network while reachable for incoming requests, and able to maintain ongoing communication sessions transparently and independently of transport (e.g. TCP and UDP) and application layers (e.g. FTP and HTTP). A MH may experience three different types of TM. Cell-level hand-off occurs across a cell of one BS to another of the same subnet within an *Administrative Domain* (AD) e.g. between campus buildings. This effectively confines and localizes handoff signaling messages to the roamed subnet, without the MH changing its network address and disrupting network layer or applications. Subnet-level hand-off (Intra-domain Mobility) refers to mobility between different cells of different subnets belonging to the same AD. Domain-level hand-off (Roaming or Inter-domain Mobility) is mobility from one subnet to another belonging to different ADs.

Session and Service mobility allows a MU to maintain ongoing sessions or acquire the same services while switching to different network service providers. It assumes the home service provider either maintains control of the services it provides to the MU in the visited network or transfers its control to the visited network.

PM [38-40], initially introduced in telecommunication industry [41-43] is a user-oriented mobility model associating with an MU to identify, authenticate, and allow a single MU located at different MHs or points of attachment to register to networks, to initiate and establish sessions via the same unique personal identifier transparent to third parties.

4.2 Mobility Management

Mobility Management summarized in Table 4.1 is a framework comprising a set of functions i.e. **hand-off**, **registration**, **configuration**, **dynamic address binding**, and **location management**. These functions are required for a roaming MH to initiate/establish sessions and to maintain its ongoing sessions at each cell, subnet, and domain level hand-off.

Hand-off allows an ongoing established session to continue while a MH moves from one cell to another with minimum disruptions in terms of latency and packet loss.

Registration allows a MH to register with a network for access rights to network services that the MH is entitled, via authentication/verification of MH's identity. Registration can be subclassed into Complete and Expedited (or partial). The former oc-

curs when a MH becomes active or roams into a new subnet or new domain while the latter is invoked when the MH relocates to new subnet to keep its location information current

Configuration allows a MH to acquire new network information e.g. IP address, new default gateway, and subnet mask, as it undergoes subnet or domain handoff.

Dynamic address binding aids a MH to maintain a constant universal identifier regardless of its point-of-attachment to the network.

Location management is a mechanism allowing an authorized MH to roam while the network updates the location database in an up-to-date, accurate, and confidential manner.

Hand-off Types	Registration/ Deregistration	Configuration	Address Binding	Location Management
Cell-level (Inter-Cell)	No	No	Yes	Yes
Subnet-level (Intra-domain)	No	Yes	Yes	Yes
Domain-level (Inter-domain)	Yes	Yes	Yes	Yes

Table 4.1: Different Levels of Mobility Management

4.3 Status of Supporting Terminal Mobility

The design of Internet assumes each host generally possesses at least a network interface assigned with a unique IP address (either IPv4 or IPv6 address) by which it is reachable from other network hosts. Each IP address comprises subnet part (IPv6 Subnet Prefix or IPv4 Subnet ID) and host part (IPv6 Interface ID or IPv4 Host ID) identifying the network within which the specific host is attached to. Data from applications and higher-level protocols like TCP and UDP are packetized into datagrams and appended with IP header. A sender of a packet inserts an IP address into the Destination Address field to inform the network the identity of the intended recipient. Packets are routed based on connectionless and best effort delivery to the desired destination. Packets discarded due to network congestion or router failure are not recovered at the network layer but on an end-to-end basis using e.g. TCP. Routers rely on the subnet part of the packet's destination address of each packet and the routing table in deciding where to forward or relay the packet to get it closer to its final destination. This continues until the packet finally arrives at the targeted host. The recipient of the

packet checks the Destination Address field of received packet to ensure that the packet is intended for itself. As long as the host remains connected to that network associated with the subnet part of its IP address, packets addressed to it will be routed to it correctly.

Host relocation problem is illustrated in Figure 4.2. Suppose $Host_X$ (initially residing in $Subnet_A$) is actively communicating with $Host_Y$, and data packets sent from $Host_Y$ are always routed across the Internet to $Host_X$ in $Subnet_A$. If $Host_X$ relocates to $Subnet_B$ without changing its IP address, it will not receive any data packets from $Host_Y$, as data packets addressed to $Host_X$ are still routed to the original IP address associated with $Subnet_A$ instead of $Subnet_B$ where $Host_X$ currently resides. Since there is no host with that IP address in $Subnet_A$ to receive them, these packets are discarded. Thus, the IP address originally assigned to $Host_X$ is invalid because it does not accurately reflect where $Host_X$ currently resides in or its latest movement. Even if $Host_X$ updates its IP address to reflect its new point of attachment, it will not receive any packets from $Host_Y$, as this would disrupt the ongoing communications at the transport level. For example, TCP connection is uniquely identified by a 4-tuple <Source Address, Source Port, Destination Address, Destination Port>, coupling IP addresses to transport sessions. If this connection semantic is modified, then the coupling is lost and the session must be re-established with the new address.

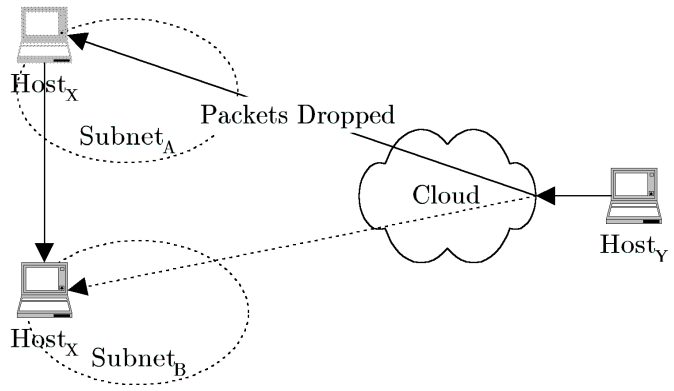


Figure 4.2: Mobility and IP Routing

It is imperative to discern that IP address uniquely identifies both the host and its current point-of-attachment to the network, displaying dual decoupled functional roles [44] namely **Terminal Identifier** (location-independent identification of a MH) and **Locator** (location information used for routing directive). Modifying one (e.g. **Locator**) would implicitly compromise the other (e.g. **Terminal Identifier**). MH's relocation

across different IP subnets while transparently maintaining all of its active connections uninterrupted and still be reachable from the rest of the Internet, is a traditional routing issue in association with the network layer. However, different solutions that function at the transport and application layers have been proposed. These proposals typically adopt a level of indirection in the routing system with a dynamic addressing association between the **Terminal Identifier** and the **Locator**. They differ mainly whether a MH possesses a non-IP or an IP address as its **Terminal Identifier**.

4.3.1 Network Layer

The *Internet Engineering Task Force* (IETF) has developed Mobile IP as an efficient and scalable network layer mobility mechanism transparent to existing higher-level protocols. Two variations of Mobile IP i.e. *Mobile IPv4* (MIPv4) [45,46] and *Mobile IPv6* (MIPv6) [47,48] are specified. Basic functionalities of MIPv6 resemble MIPv4 [49] to decouple the dual role of IP address into two IP addresses i.e. Home Address (**Terminal Identifier**) and Care-of Address (**Locator**). However, MIPv6 has been designed as an integral part of IPv6 to support tighter integration of mobility signaling, security features, route optimization, header extensions, and elimination of the Foreign Agent. Moreover, MIPv4 suffers limitations [50,51] in terms of high handover latency and signaling traffic. Details of MIPv6 is covered in section 4.4.

4.3.2 Transport Layer

A TCP-based host mobility mechanism [52] proposes a new Migrate TCP option for supporting host mobility at the transport layer so as to leave the underlying IP routing infrastructure unchanged. The new Migrate TCP option is included in SYN segments for identifying a previously established connection on the same <Destination Address, Port> pair via a token negotiated during the initial connection establishment. Thus, when a MH relocates to a new subnet, the session persists using a previously established TCP connection by sending a Migrate SYN packet containing the token. The receiving host will then resynchronize the connection. Deployment requires adding TCP migrate mechanism to existing TCP implementation for both stationary nodes and MHs. In addition, both end hosts cannot move simultaneously in order for a party to receive the new IP address of the other contained in the SYN packet. Another proposal, TCP Splice [53] specifies a split-connection proxy system architecture that uses the same end-to-end semantics as normal TCP connections while allowing MH relo-

cates and controls which network interfaces to use for different kinds of incoming and outgoing data.

4.3.3 Application Layer

The Migrate TCP scheme [52] also relies on *Domain Name System* (DNS) and secure dynamic updates [54] to track a MH's location via its adopted email-like address (host name) as the **Terminal Identifier**. With each subnet handover, the MH acquires a new IP address, and sends a secure DNS update to one of the name servers in its home domain. Secure Dynamic DNS updates the MH's entry to form a mapping between the MH's host name and the new IP address (i.e. new location). When a host initiates a communication with the MH, conventional DNS resolves the MH's host name to the new IP address, and the host then directly exchanges packets destined to the MH's new IP address. However, when the MH roams during an active communication, Migrate TCP option needs to be invoked.

Session management capabilities of SIP have also been used to support mobility [55-57]. *Mobile SIP* (MSIP) [58,59] offers TM for real-time communication and provides a MU with roaming ability to obtain service in networks that may not necessarily be owned by its home service provider. MSIP relies mainly on SIP INVITE and REGISTER requests, and dynamic mapping between unique User Identifier in the form of SIP URL, and current IPv6 address. Details of MSIP is covered in section 4.5.

4.4 Mobility Support in IPv6 Internet: Mobile IPv6

4.4.1 Overview of Mobile IPv6

Figure 4.3 depicts the general architecture of *Mobile IPv6* (MIPv6). *Mobile Host* (MH) is a host that changes its point-of-attachment between networks. MH is always identified and assigned with a global invariant IPv6 address known as *Home Address* (HAddr) that remains unchanged regardless of where the MH is attached to. The IP subnet corresponding to the MH's HAddr is its home subnet. Whenever the MH relocates to a new foreign subnet, it acquires a temporary IPv6 address known as *Care-of Address* (COA). COA shares the same prefix as the foreign subnet to which the MH is attached, and reflects its current point-of-attachment to the Internet, when away from its home network. MH then notifies a network entity at its home subnet known as *Home Agent* (HA) of its new COA. HA tracks MH's current location by maintaining a

binding i.e. an association of MH's HAddr with the new COA. Binding allows HA to intercept any IPv6 packets originated from a *Correspondent Host* (CH) and subsequently tunnels them to the MH while it is actively moving, thus maintaining continuous connectivity of TCP and UDP streams. CH is a peer host (either mobile or stationary) with which the MH is communicating. MH also updates all of its CHs with its new COA ensuring any future packets from them are addressed directly to its new COA. Each MH possesses a data structure known as *Binding Cache* (BC) to record HAddr, COA of sending MH, and remaining lifetime of the entry per binding messages received from other MHs. Whenever a CH sends a packet to MH, the BC is searched for an entry using the HAddr as a key. If exists an entry, the packet is addressed to MH's COA. Otherwise, the packet is transmitted to MH's HAddr.

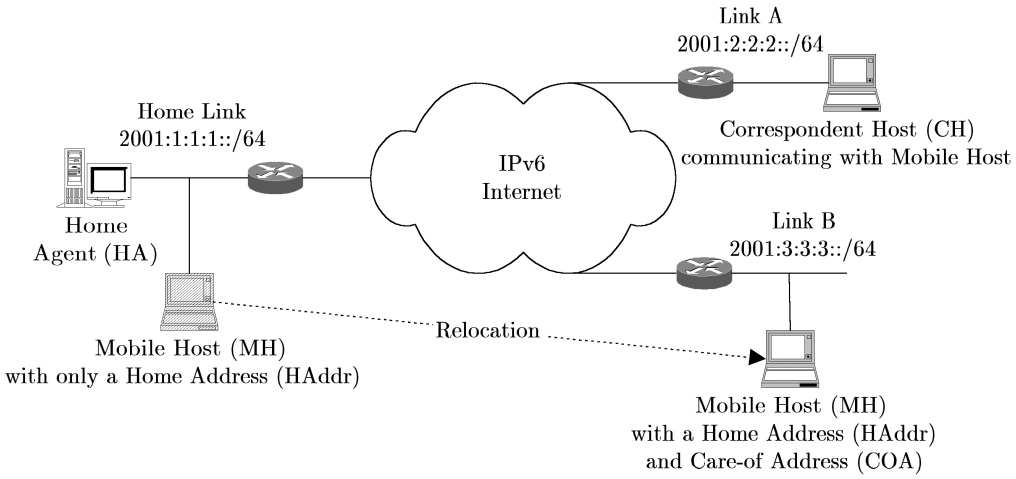


Figure 4.3: Mobile IPv6 General Architecture

4.4.2 Mobile IPv6 Messages

MIPv6 relies on *Internet Protocol Version 6* (IPv6) [60-62] which supercedes *Internet Protocol Version 4* (IPv4) as *Next Generation Internet Protocol* (IPnG). IPv4 was initially designed and largely unmodified since mid 1970's, to address two major problems of heterogeneity and scalability for a distributed and predictable host-to-host connectivity traversing networks of different platforms and diverse wired access technologies. However, the emergence of wireless access technologies coupled with the exponential proliferation of mobile devices have manifested a critical demand for limited deployable 32-bit IPv4 addresses, many are reserved for broadcast, multicast, and private addresses. Although, *Network Address Translator* (NAT) is widely used to circumvent

the address space issue, it does not provide the global end-to-end routability.

IPv6 is designed with an enlarged 128-bit addressing scheme (four-fold larger than IPv4 address) but its header has a fixed length of 40 Bytes (twice of IPv4 header). IPv6 provides explicit labelling of IPv6 traffic flows for *Quality of Service* (QoS) and mandatory integration of IP security [63] providing authentication, data integrity, and confidentiality features. IPv6 Base Header Format is depicted in Figure 4.4 contains the following fields: **Version** (Default value is 6 for IPv6), **Traffic Class** (Used by edge and core nodes/routers to distinguish between classes or priorities of IPv6 packets), **Flow Label** (To identify a sequence of packets generated from a single source application requiring the same transfer service), **Payload Length** (Indicates number of octets of remaining information following the IPv6 header), **Next Header** (Indicates type of header immediately that follows), **Hop Limit** (Decremented by one at each forwarding node/router, packet is discarded when value is zero), and finally 128-bits IPv6 **Source** and **Destination Address** assigned to source and destination network interface respectively.

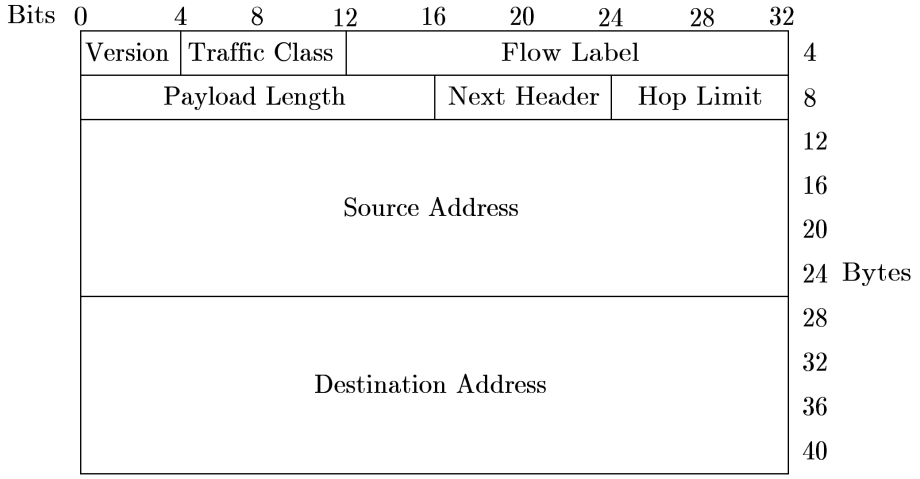


Figure 4.4: IPv6 Base Header Format

IPv6 Aggregatable Globally Routable address [64] depicted in Figure 4.5 is a globally routable, reachable and unique unicast address, identifying the network interface and the location of it. It comprises two 64-bits length components i.e. Subnet Prefix and Interface Identifier (Interface ID). Subnet Prefix constitutes the upper 64-bits with two logical parts (Public Topology, and Site Topology) aggregating network addresses of multiple hosts topologically close to each other by using a common prefix. Public Topology is used for global routing and identifies the provider supplying the

network access service. Site Topology distinguishes the site's internal organization and is used for intra-site routing. Interface ID constitutes the lower 64-bits, identifying the interface on the link.

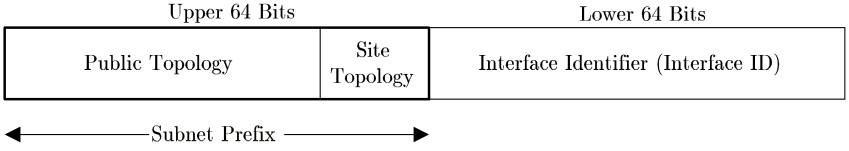


Figure 4.5: IPv6 Aggregatable Globally Routable Address

MIPv6 requires the following mobility related Destination Options Headers exchanged between MH, CH and HA. Destination Options Header is an additional field inserted between the basic IPv6 Header and payload to transport optional information examined by the packet's destination node.

Binding Update Destination Option (BU) is used by a MH to notify its HA and CHs of its current location and corresponding COA, and future packets should be addressed directly to this COA as the destination address.

Binding Acknowledgement Destination Option (BA) is used to acknowledge the receipt of a BU, if an acknowledgement was requested in the BU.

Binding Request Destination Option is used by any CH to request a MH to reply with a BU, as to refresh the cached binding for the MH when the binding's lifetime is close to expiration.

Home Address Destination Option (HAD) is appended to packet send by a MH while it is away from its home network to inform the recipient about its HAddr, thus making the use of COA transparent to higher protocols layer.

Routing Extension Header (RH) is used by a CH to control the routing of a packet by listing intermediate nodes that a packet must transverse to the ultimate destination. Currently, Type 0 RH is defined with an initial IPv6 Header containing the first address in the list as its destination address. Each intermediate node replaces the destination address with the next listed one.

4.4.3 Neighbour Discovery Protocol

MIPv6 relies on Neighbour Discovery Protocol [65] and the following messages [66] for its critical operations: Address Resolution and Host Autoconfiguration.

Router Solicitation (RS) is used for requesting a RA rather than awaiting one.

Source address is the address assigned to the sending interface or the unspecified address if no address is assigned to it. Destination address is typically the all-routers multicast address.

Router Advertisement (RA) is multicasted by each router periodically to announce its availability. Source address is the link-local address assigned to the interface from which this message is sent. Destination address is typically the address of the node invoking the RS or the all-nodes multicast address if sent periodically. Hosts receiving these RA build a list of default routers. RA contains critical informational parameters: ManagedFlag (M-bit), *Prefix Information Option* (PIO), Default Gateway for routing MH's data packets, and *Maximum Transmission Unit* (MTU) of data packets allowed over this network. If M-bit is false and PIO (e.g. 2001:618:6:f1::) is provided, MH uses stateless address configuration to generate a new address based on its link identifier and the prefix information. If M-bit is true, then it invokes stateful address autoconfiguration to determine its on-link address.

Neighbour Solicitation (NS) is used to determine the link-layer address or to verify the reachablility of a neighbour. Source address is either the address assigned to the interface from which this message is sent or the unspecified address checked during *Duplicate Address Detection* (DAD). Destination address is either the solicited-node multicast address corresponding to the target address used when performing DAD or the target address itself.

Neighbour Advertisement (NA) is a response to a NS or sent as an unsolicited NA to announce a change in a link-layer address. Source address is the address assigned to the interface from which the NA is sent. Destination address applies only to solicited NA and can be the source address of the invoking NS, or the all-nodes multicast address if the source address is unspecified (i.e. a host has not determined its global unicast address).

Address Resolution resolves the IPv6 address of a neighbour into its link-layer address. Each node initially participates in the solicited-node multicast address. When nodes communicate with a neighbour, they first check their neighbour cache for neighbour's link-layer address. If the address is not found, they multicast a NS with the known IP address of the target. Each recipient compares the NS's destination address against its own. If they match, it responds with a NA to the soliciting node, indicating its IPv6 address in the target address field and its physical address in the tar-

get link-layer address option. Soliciting node updates its neighbour cache with the recently found link-layer address.

Host autoconfiguration allows hosts to autoconfigure its network interface by creating a valid link-local address on the specific link, and in turn generate a global unicast address. A host undergoes a sequence of processes depicted in Figure 4.6: Link-local IPv6 Address Generation, *Duplicate Address Detection* (DAD), Router Discovery, and Stateless or Stateful Address Autoconfiguration.

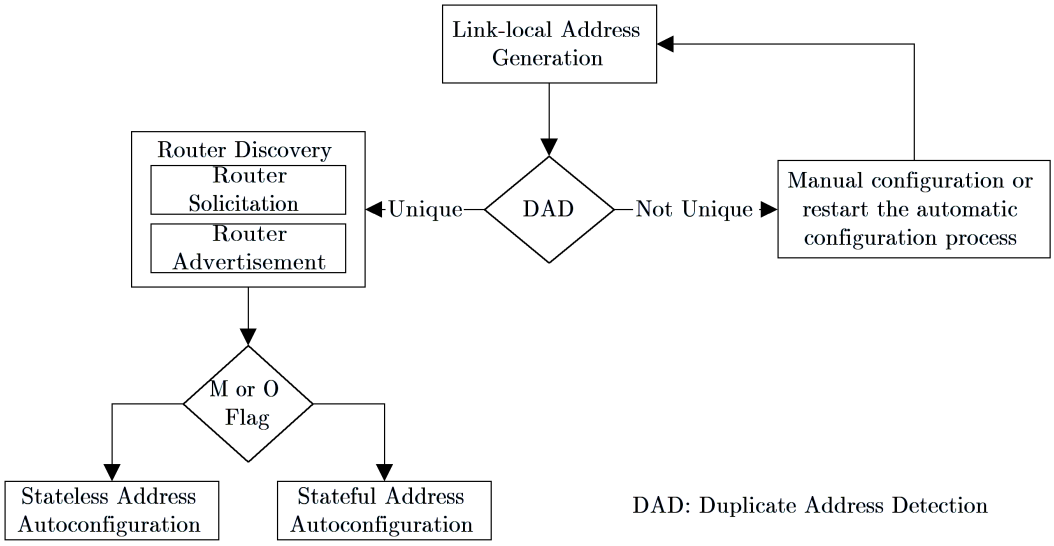


Figure 4.6: Host Autoconfiguration Logical Flow

Link-local IPv6 address Generation illustrated in Figure 4.7 is performed for each interface. It involves creation of its Interface ID and appending well-known link-local prefix FE80::/64 to it. Interface ID is a 64-bits long identifier uniquely identifying interface on a link, and typically derived and constructed from IEEE EUI-64 format [67] with the universal/local bit i.e. the second low-order bit of the first Byte complemented. In turn, Interface ID generated from IEEE 802 48-bit *Media Access Control* (MAC) address requires inserting 16-bits of 0xFFFE between the company ID and the extension ID, and then inverting the universal/local bit. For example, IEEE 48-bit MAC address is 00:09:B7:7B:95:BB, inserting 16-bits of 0xFFFE between 0xB7 (third Byte) and 0x7B (fourth Byte) forms the IEEE EUI 64-bit address i.e. 00:09:B7:FF:FE:-7B:95:BB. Complementing Universal/Local bit (second low-order bit of 0x00) creates IPv6 Interface ID i.e. 02:09:B7:FF:FE:7B:95:BB, and finally appending prefix FE80::-/64 results in FE80::209:B7FF:FE7B:95BB.

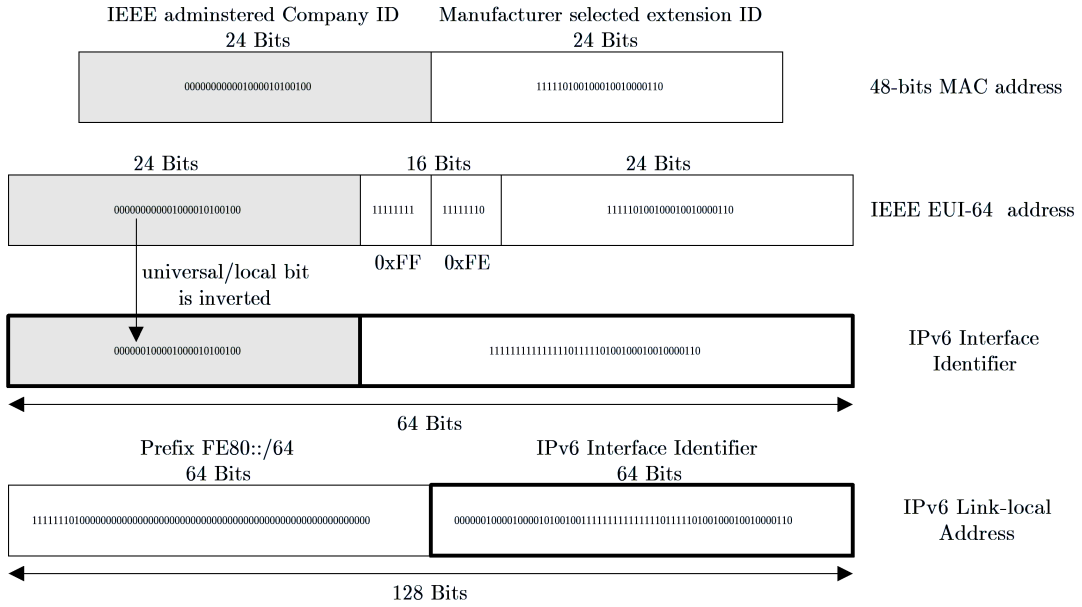


Figure 4.7: Link-Local IPv6 Address Generation

DAD is performed for detecting duplicate address and ensuring assigned address uniqueness on the link. DAD is compulsory on all IPv6 nodes regardless of whether stateful, stateless, or manual configuration is adopted. DAD employs NS and NA. A node sends a NS containing its newly formed link-local address on the link in which the source address is set to the unspecified address and the destination address is set to the solicited-node multicast address. The node then listens for a response to indicate if another node is already assigned with that address. If the address is already used, a NA is returned within a pre-determined timeout period. Otherwise, the address is presumed to be unique and is assigned to the interface. When duplication is detected, the generated address is discarded and a new one is recreated either manually or restart the host autoconfiguration process. During DAD, addresses are tentative in nature (unassigned to an interface and not valid for regular communication), and if they are checked valid and unique, they have status of preferred (assigned to an interface and valid for regular communication). In addition, addresses may enter the deprecated state (still valid but discouraged for new communication).

Lastly, Router Discovery ensures routers are operationally present on the link. Normally routers periodically broadcast RA on its link, however a node may broadcast unsolicited RS to request an RA quickly. If a link has no routers and an end-node does not receive any RA for a specific duration, it attempts with stateful autoconfiguration

to obtain addresses and other configuration information. If routers are present on the link, they respond with RAs containing two flags indicating whether stateful or stateless autoconfiguration, or both approaches simultaneously to be performed. The former is adopted when a site is not particularly concerned with the exact addresses assigned to hosts as long as they are uniquely and properly routable, while the latter approach is commonly used for managed control of address assignment within a domain. Stateless address autoconfiguration is preferred and adopted in this thesis due to an obvious advantage that manual intervention of additional servers or configuration of hosts is not required.

Stateful Address Autoconfiguration uses *Dynamic Host Configuration Protocol for IPv6* (DHCPv6) [68]. A host obtains state information including interface addresses, configuration information, and parameters like DNS Server's IPv6 address. The host first sends a multicast DHCPv6 Solicit message from the interface it wishes to configure and listens for a unicast DHCPv6 advertise message containing the IP address. When the host no longer requires the service of DHCPv6, it issues a DHCPv6 Release message to the server to release the specified addresses and resources.

For Stateless Address Autoconfiguration, each host generates its unique address using a basic principle [69] **IPv6 address = Prefix_address + Interface ID** i.e. combination of prefix information contained in RA advertised by routers and Interface ID. If link-local address is verified unique using DAD, so are site-local and global unicast addresses as they are derived from Interface ID by appending a prefix. Routers advertise RA containing PIO (prefixes for site-local and global unicast addresses) and prefix length specifying the type of unicast address. Site-local address is concatenation of FEC0::/48, 16-bit subnet ID field identifying subnets and Interface ID.

4.4.4 Mobile IPv6 Functional Operations

Figure 4.8 depicts a MH initially powering up in home network and then relocating to foreign network while communicating with two CHs (CH_1 and CH_2). MIPv6 messaging is illustrated in Figure 4.9.

HA Registration: The MH performs HA registration when it first attaches to its home/foreign network or it roams between different subnets. It registers its newly acquired COA by sending a BU to its HA on the home link, and awaits a BA from its HA. HA uses Address Resolution and multicasts "Proxy or Gratuitous" NA on the home link to advertise its link-local address in place of MH's HAddr. This enables HA

to intercept and tunnel packets to MH's COA or to reply to NS on behalf of MH.

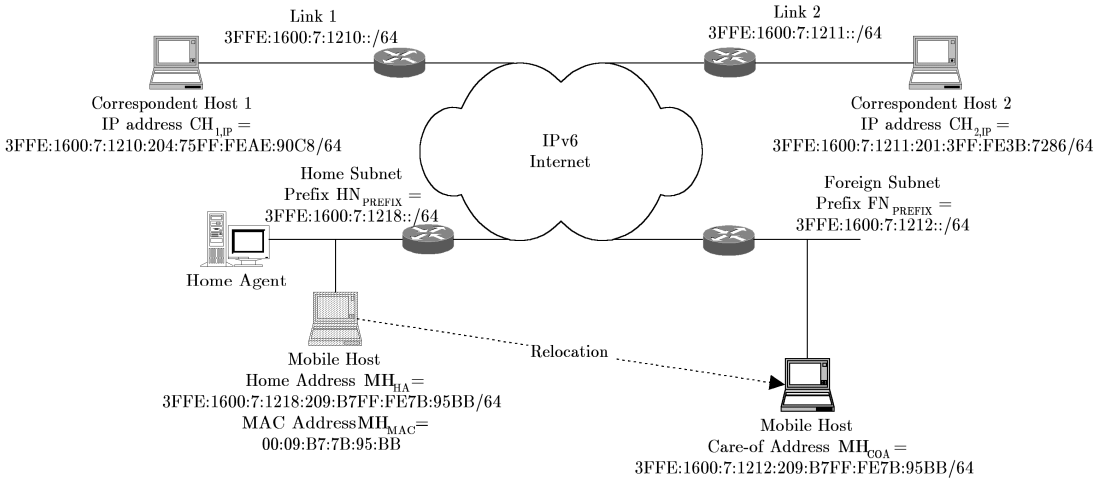


Figure 4.8: Illustration of Mobile IPv6

Movement detection: When the MH relocates to another network with subnet prefix (FN_{PREFIX}), it determines its current location by listening to RA broadcasted periodically from *Access Router* (AR), and uses IPv6 Router Discovery to compare the subnet prefix information with the subnet prefixes of its HAddr or COA. If it matches with the subnet prefix of HAddr, then it has returned to its home network, else it is residing in the foreign network. Movement from one link to another is detected if new subnet prefix information does not match its current COA, then the MH configures a new COA and selects one of the ARs as its default router. It uses FN_{PREFIX} as its current network prefix and initiates stateless address autoconfiguration procedure to acquire a new COA (MH_{COA}).

Dynamic HA Address Discovery: In the event that the router previously operating as the MH's HA has been replaced by another router serving this role, the MH discovers the address of a suitable HA on its home link by sending an **ICMP HA Address Discovery Request message** and awaiting an **ICMP HA Address Discovery Reply message**. It sends an **ICMP HA Address Discovery Request message** with the destination address set to "MIPv6 Home-Agents" anycast address for its home subnet prefix and source address set to its COA. Any HA on its home link receiving this message replies with an **ICMP HA Address Discovery Reply message** containing a list of global unicast IP addresses of itself and other HAs operating on the home link in order of decreasing preference value, and specifies the source address as one of the listed HA's global unicast addresses. Upon receiving this message, the MH then sends its home registration

BU to the HA's address (source address of the packet) or to any of the listed HA's global unicast addresses.

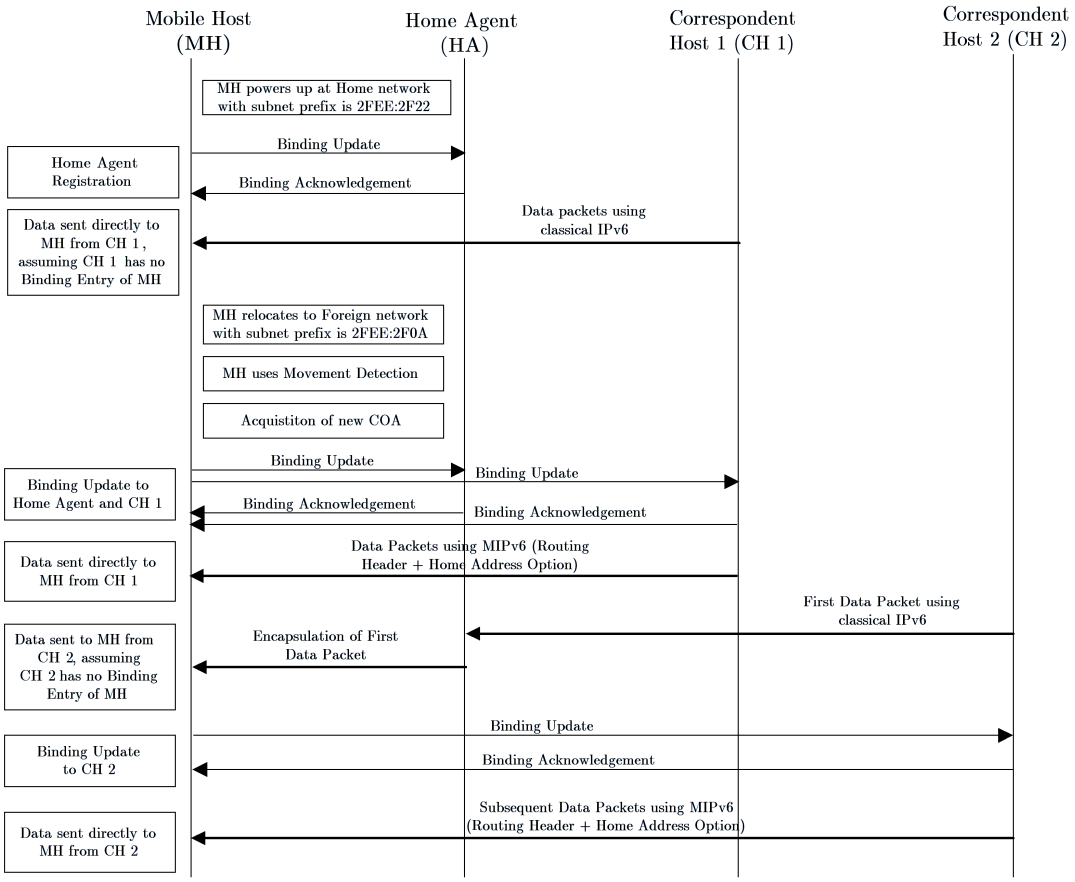


Figure 4.9: Mobile IPv6 Messaging Sequence

Sending BU: Consider three scenarios. The first case is when the MH detects it has relocated to another link in a foreign network and the old default router is unreachable. It immediately configures a new COA, issues a BU to its HA and to any active CHs. The BU comprises its new COA, its HAddr, and a binding lifetime. To ensure that the intended receiver receives this BU, the MH can enforce the receiver to acknowledge the receipt of the BU by enabling the Acknowledge bit in the BU. Until receipt of the BA, the MH retransmits previously sent BU periodically. Acknowledgment of BU is compulsory for those BU addressed to a HA but not CHs. A CH receiving the BU commences to use RH to deliver data to the MH. HA that receives the BU with the H-bit enabled will tunnel packets to MH's COA. The second case is when a CH (e.g. CH_2 with IP address is $CH_{2,IP}$) possesses no information about the MH's location in its BC, and initiates a communication with it by setting MH_{HA} as the destina-

tion address. Packets are forwarded to the home network of the MH, but intercepted by its HA which then tunnels them to the MH's current location i.e. MH_{COA}. This triggers the MH to send BU to CH_2 for establishing a direct communication with CH_2 ensuring CH_2 delivers packets directly to MH's COA. CH_2 registers the new binding in its BC by creating an association between MH's HAddr (MH_{HA}) and MH's COA (MH_{COA}). Subsequent packets are directly addressed to the MH using MH_{COA} as the destination address. The third scenario is when the MH returns to its home network, it notifies its HA with a BU to delete the binding. HA responses with a BA, and the MH multicasts a NA on its home link to associate its link-local address with its HAddr ensuring future packets are forwarded to it directly.

Requesting BU: CH and HA request the MH to initiate a refreshing of the binding when nearing current binding lifetime expiration, by sending a Binding Request to the MH. MH does not necessarily respond to the request. If it does, it replies a BU containing its COA and HAddr to the requestor.

MH is receiver of data packet: Consider two scenarios. The first case is a CH (e.g. CH_1's IPv6 address $CH_{1,IP}$) possesses no BC entry for a MH due to BC entry expiration or not receiving BU from the MH, thus only aware of the MH's HAddr. CH_1 initiates a communication with the MH by setting the MH's HAddr (MH_{HA}) as the destination address. Packets are forwarded to the MH using classical IPv6 forwarding mechanisms with the destination address set to MH_{HA}, as long as it remains connected to its home network with subnet prefix (HN_{PREFIX}). If MH is in foreign network, HA intercepts packets destined for MH's HAddr and tunnels them using IPv6 encapsulation (Figure 4.10a) to MH's COA. MH checks the outer and inner destination addresses against its COA and HAddr respectively. It consumes the detunneled packet, and then sends a BU to CH ensuring subsequent packets are received directly from CH with destination address specified to COA. The second scenario is a CH maintains BC entry for MH after receiving BU from it, packets are addressed to MH using RH (Figure 4.10b) which specifies at least two hops namely MH's COA and HAddr. Packets are sent directly to the MH's COA, MH receives the packet and "forwards" it to the next (and final) hop (MH's HAddr), the packet is "looped back" inside itself. Thereafter, the packet is processed similarly as if the MH is at home.

MH is sender of data packet: For any data packets transmitted by the MH while away from home, its COA is placed in the source address of IPv6 Header which in-

cludes a HAD (Figure 4.10c). Upon receiving these packets, CH substitutes the MH's HAddr obtained from the HAD for this COA (i.e. source address in the IPv6 header) when processing the packet, transparent to higher and application layers.

Outer IPv6 Header	Inner IPv6 Header	Transport Header	Payload
SrcAddr = Home Agent DstAddr = COA	SrcAddr = CH DstAddr = Home Address	TCP/UDP	010101010100000100

a) IPv6 Encapsulation

Outer IPv6 Header	IPv6 Routing Header	Transport Header	Payload
SrcAddr = CH DstAddr = COA	Next Hop = Home Address	TCP/UDP	010101010100000100

b) IPv6 Routing using IPv6 Routing Header

Outer IPv6 Header	Home Address Option	Transport Header	Payload
SrcAddr = COA DstAddr = CH	Next Hop = Home Address	TCP/UDP	010101010100000100

c) IPv6 Routing using Home Address Option

SrcAddr: Source Address
DstAddr: Destination Address
CH: Correspondent Host

Figure 4.10: Mobile IPv6 Packet Structure

4.4.5 Limitations of Mobile IPv6

MIPv6 specifies basic inter-subnet handover mechanism but suffers from the following handoffs drawbacks. (I) Incurs significant MIPv6 signaling in exchanging BU/BA between MH and CH/HA while MH frequently moves in foreign network and acquires a corresponding new COA. (II) Incurs MIPv6 handover latency and packet loss during or immediately after a handoff, this affects delay sensitive data traffic and real-time services. During handover, MH configures a new COA and notifies its CHs and HA via BUs, all in-flight packets arriving at the old link are discarded unless they are redirected to the new link. MH is temporarily "unreachable" and cannot resume or continue communication. (III) MIPv6 provides only TM but not PM through IPv6 destination options for signaling and HA to maintain the binding of HAddr and COA. (IV) MIPv6 supports only TM between administrative domain [70,71] belonging to service provider with existing service level and roaming agreements. It does not facilitate MUs to obtain service transparently in networks that may not necessarily be owned by their home service provider.

Fast Handover for MIPv6 (FMIPv6) [72] proposes MH connected to its old AR

and is about to move to a new AR, to obtain a new COA at the new AR, MH then sends a Fast-BU to its old AR to update its BC with MH's new COA. The old AR starts forwarding packets destined for the MH to new AR. FMIPv6 is enhanced with "Bicasting" [73] by anticipating the movement of MHs and utilizing simultaneous bindings. Packets sent to MH at both its "previous" and "new" link while the MH is moving between them. Higher-level protocols eliminate duplicates irrelevant to the application. This removes timing ambiguity and minimizes MH's periods of service disruption during ping-pong movement. *Hierarchical MIPv6 Mobility Management* (HMIPv6) [74] extends MIPv6 with a local hierarchical structure of *Mobility Anchor Point* (MAP) which functions as a local HA for the MH registered with it. HMIPv6 localizes the mobility signaling to CHs and the MAP, and reduces the latency due to handoffs between ARs since it requires less time to send a BU to a local MAP than a distant HA. MAP receives all packets on behalf of the MH it is serving and tunnels them directly to the MH's current address. Each MH has two addresses Regional COA, an address on the MAP's subnet and On-link COA, configured on an MH's interface based on the prefix advertised by its default router. Table 4.2 compares the performance of FMIPv6, HMIPv6, and basic MIPv6. Column 2 depicts the NS-2 simulation results [75] and Column 3/4 shows the mathematical analysis [76]. ld, ld_{HA}, ld_{CH} are the latency incurred by transmission of signaling messages over wired link, links leading to HA and CH respectively. Wd is the latency incurred by transmission of signaling messages over wireless link. BU_{MIPv6} and BU_{HMIPv6} are the number of MIPv6 (2 or 3) and HMIPv6 (1 or 2) BU sent respectively, actual number depends on whether the previous AR forwarding is allowed. The following observations can be summarized. FMIPv6 betters HMIPv6. Both HMIPv6 and FMIPv6 outperform basic MIPv6. All three under-performs to FHMIPv6 i.e. combination of HMIPv6 and FMIPv6.

Average Handoff Latency			
Framework	**Simulation/ms**	**ld >> Wd**	**ld << Wd**
MIPv6	5487	$2[ld + \min(ld_{HA}, ld_{CH}, 0)]$	$(BU_{MIPv6} + 1)Wd$
HMIPv6	739	2ld	$(BU_{HMIPv6} + 1)Wd$
FMIPv6	352	ld	3Wd
FHMIPv6	301	2Wd	2Wd

Table 4.2: Performance Matrix of MIPv6, FMIPv6, and HMIPv6

Neighbourhood Routing [77] specifies a natural extension to MIPv6 with no new networking entities. MH sends a BU to CH/HA containing possible COAs matching

the current link and other potential links to be visited. After receiving such a BU, CHs and HA are able to send packets to the MH at one of its specified COAs. Thus, CHs and HA maintain communication with MH despite not knowing its exact location while the MH moves across links.

Cellular IPv6 (CIPv6) [78] provides mobility and handoff support for both frequently and rarely moving MHs. It introduces CIPv6 Gateway, a CIPv6 node connected to a regular IP network by at least one of its interfaces. A CIPv6 MH has two states either active (if it is transmitting or receiving IP packets) or idle (if it has not recently transmitted or received IP packets). CIPv6 nodes maintain Route Cache and Paging Cache for active and idle MHs respectively. Data packet originated from the MH is routed to the CIPv6 Gateway using shortest path hop-by-hop routing. The packet updates CIPv6 node's Route Cache and Paging Cache, and then forwarded upstream by CIPv6 node. Data packets addressed to the MH are routed along the reverse path. CIPv6 node first checks Route Cache and then Paging Cache whether they have a valid mapping between the destination address and the next-hop downstream node. Cellular Mobile IPv6 [79,80], similar to CIPv6, extends MIPv6 with support for smooth and non-breaking handoff when MHs moves among small wireless cells at high speed.

4.5 Mobility Support in IPv6 Internet: Session Initiation Protocol (SIP)

4.5.1 Overview of SIP and Mobility

SIP was originally designed to handle signaling of multimedia sessions between multiple parties by means of unique SIP URL registration and INVITE message, enabling users to access the network from any location using any mobile terminal regardless of the underlying network infrastructure. SIP is increasingly popular for peer-to-peer or inter-person interactive connectivity including video conferencing and Napster-like file sharing services, and has been proposed by 3GPP as the official end to-end signaling and call control protocol for 3G Mobile Telecommunications. Such communication mode requires both parties to locate each other, to request the participation of the other, to consent to the participation, and then effectively establish the session before data exchange commences between peer-hosts. Several proposals have been submitted to utilize SIP for providing pre-session mobility and mid-session mobility

with real-time multimedia and non-real-time applications (i.e. TCP connections).

4.5.2 Personal Mobility

SIP supports PM seamlessly is illustrated in Figure 4.11. MU Alpha (MU_{Alpha}) relocates from a terminal (denoted as pc.domainA.com) at Domain A to another terminal (denoted as ws.domainB.com) at Domain B. MU_{Alpha} prior moving to Domain B, informs Registrar of Domain A (R_A) of its new location by sending a REGISTER request (Step 1). The REGISTER request indicates MU_{Alpha} is away from pc.domainA-.com and is reachable at ws.domainB.com. R_A responds that the registration (Step 4) is successful after updating the LS (Step 2 and 3). When MU_{Alpha} enters into Domain B, it sends a normal REGISTER request (Step 5) to Registrar of Domain B (R_B) informing that MU_{Alpha} can be found at ws.domainB.com. R_B updates its LS (Step 6 and 7) and responds to MU_{Alpha} (Step 8) that the registration is successful. Suppose MU Beta (MU_{Beta}) establishes a session with MU_{Alpha}, it sends an INVITE request (Step 9) via R_A (Step 10) to R_B. R_B relays the INVITE request (Step 11) to MU_{Alpha}, which resides on ws.domainB.com.

4.5.3 Hierarchical Personal Mobility

MU resides remotely from its home network and sends a REGISTER request to its home network whenever it relocates among IP subnets within the foreign network, this would incur unnecessarily high traffic load associated with registration and latency to locate MU for new session establishment. Hierarchical-based PM [58,59] as illustrated in Figure 4.12 minimize both costs. A MU relocates (Step 1) from its home network (consisting Home Subnet) to a foreign network (consisting Foreign $Subnet_A$ and Foreign $Subnet_B$). Each network is administered by a gateway i.e. Registrar collocated with *SIP Proxy Server* (SIPPS), denoted as GW_{Home} and $GW_{Foreign}$ respectively. First REGISTER request (Step 2) issued by the MU originated from foreign network, is received by $GW_{Foreign}$ which then forwards (Step 3) to GW_{Home} after modifying the Contact in the REGISTER message to reference to it rather than the MU's current location. If MU relocates (Step 4) between Foreign $Subnet_A$ and Foreign $Subnet_B$, subsequent REGISTER requests are delivered to the same $GW_{Foreign}$ (Step 5). $GW_{Foreign}$ recognizes that the MU is performing a localized handoff between subnets administered by it and does not relay the request to GW_{Home}. As long as the MU roams within the same foreign network, GW_{Home} tracks which foreign network its MUs are located but not the precise

location. Whenever the MU returns to its home network it re-registers with its foreign. Session establishment is performed as follows. CU sends INVITE request to the MU, which first arrives at GW_{Home} (Step 6), and is forwarded to the MU via $GW_{Foreign}$ (Step 7 and 8).

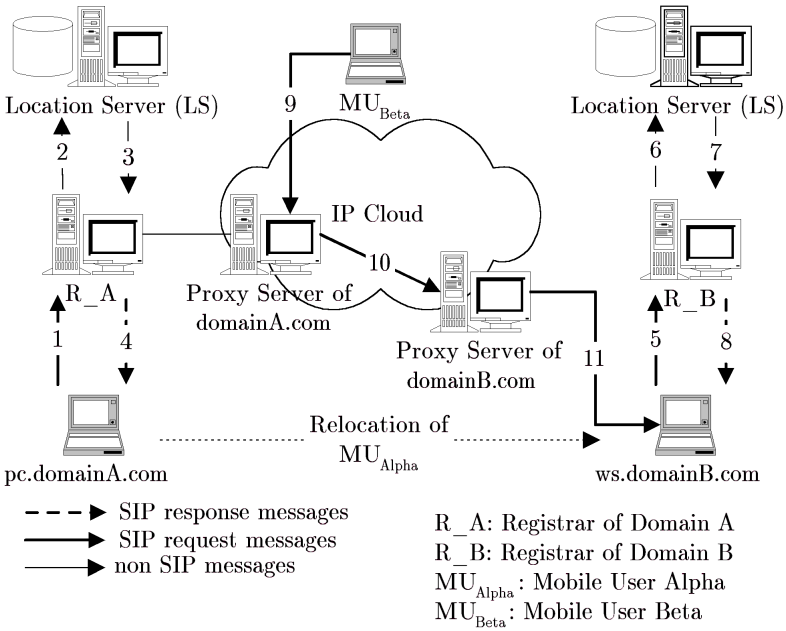


Figure 4.11: Personal Mobility using SIP

4.5.4 Terminal Mobility for UDP Based Session (Mobile SIP)

Mobility support in SIP, refer to as *Mobile SIP* (MSIP) assumes a simplistic model with no existing MIPv6 core infrastructure. Each MU is uniquely assigned with a User Identifier e.g. SIP URL identifying a user residing on a terminal, and the resided MH possesses a temporary unicast IP address (i.e. COA) identifying its current location. Each MU belongs to a home network on which a SIP Registrar is present to receive registrations from the MU each time it changes location. An address translation mechanism involves the SIP Registrar to maintain a dynamic mapping between its MU's User Identity and MH's COA, resembling a MIPv6's binding between HAddr and COA. MSIP supports both pre-session mobility and mid-session mobility as illustrated in Figure 4.13.

MSIP handles pre-session mobility by means of registration and re-direction by using a unique SIP URL. Whenever a MH acquires a new IP address on its current network (home or foreign) before any session establishment, its residing MU registers

(Step 1) with its "home" Registrar. After updating (Step 2, 3) its LS, the Registrar replies (Step 4) "200 OK" to the MU. Once the MU has successfully registered, any *Correspondent User* (CU) wishing to communicate with it, issues (Step 5) an INVITE request, which is routed to the *SIP Network Server* (SNS) of MU's home network following standard SIP procedures. If the SNS on the home network is a *Redirect Server* (SIPRS), it returns the registered address of the MU. CU then sends the INVITE request to the MU's current location directly. If the SNS on the home network is a SIPPS, it forwards the INVITE request (Step 6) to the MU's current location. If the MU agrees to the session, it exchanges further SIP messages and any other data directly with the CU thereafter.

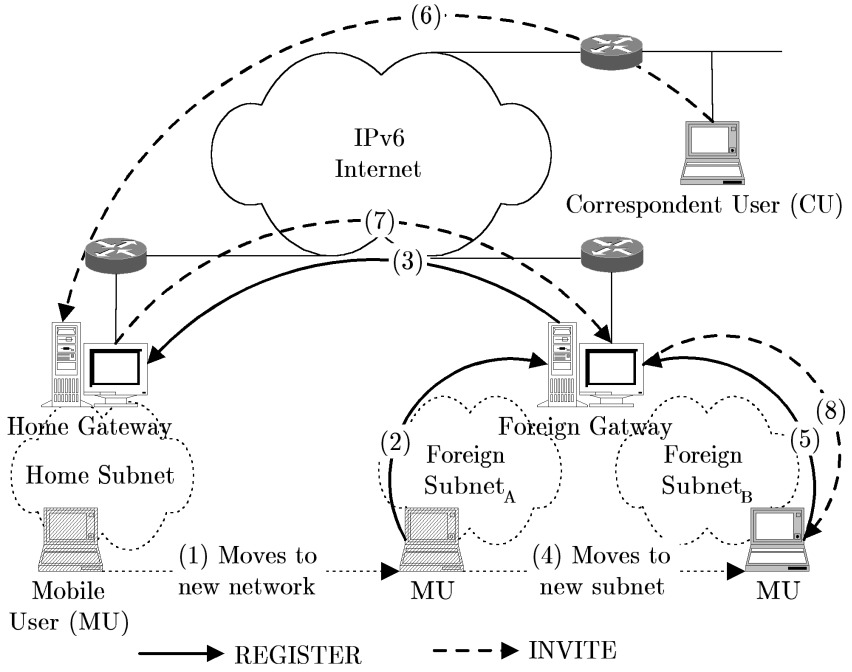


Figure 4.12: Hierarchical Registration in SIP

Mid-session mobility assumes a MH roams to another subnet during an active session and acquires a new COA. To maintain the ongoing communications between the MU and its CU, the signaling and data traffic flow between them must be transferred with minimal disruption in association to the MU's new COA. For signaling, the MU sends (Step 7) a new INVITE message to its CU with its newly obtained COA updated in the Contact field, to inform the CU where it wants to receive subsequent SIP messages. To redirect the data traffic flow, the MH refreshes the c(onnection)-field i.e. transport address embedded in *Session Description Protocol* (SDP) to its new COA.

CU transmits all subsequent IP data traffic to the MU with this new IP address. When CU receives the INVITE message, if it accepts to the new change, it returns (Step 8) with positive provisional responses, and the MU then replies (Step 9) with an ACK request to complete the traffic handoff. To redirect new sessions to its new location, the MU notifies (Step 10) its home Registrar which responds with a "200 OK" (Step 11).

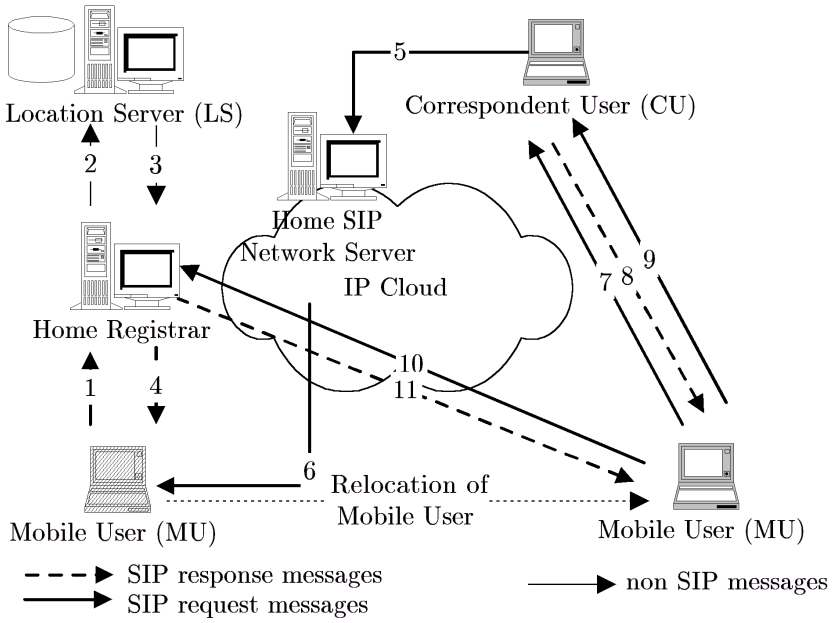


Figure 4.13: Terminal Mobility using SIP for UDP Based Session (Mobile SIP)

4.5.5 Limitations of Mobile SIP

MSIP specifies basic mobility framework at the application layer for UDP-based communication but suffers several handoffs drawbacks. (I) No support for TM TCP-based communication as MSIP resides over transport layer. (II) Incurs high handover latency and packet loss during or immediately after a handoff between different *Access Routers* (ARs). Proposals to resolve these issues are as follow:

Host Mobility Management Protocol (HMMP) [81,82] extends SIP with TM for TCP applications by equipping each MH with a tracking/monitoring software agent known as SIP_EYE agent, without modifying TCP. SIP_EYE handles ongoing TCP connections during subnet and domain hand-off by providing following four key functions. (I) To constantly spoof TCP endpoints by examining the TCP headers to monitor the birth and death of TCP connections, and to identify their endpoints. (II) To

maintain a record of ongoing TCP connections and their identifiers. Each record is a state comprising of three elements, namely the MH's original IP address, the current IP address, and original CH's IP address per TCP connection. Original IP address is the one used at the beginning of the TCP session. (III) To update CH with MH's new IP address by binding the original MH's IP address to its new ones contained within the newly proposed INFO method. (IV) To exchange TCP information between MH and CH using IP encapsulation and to maintain constant endpoints for MH's ongoing TCP connections.

Two approaches extend SIP with fast handoff intra-domain capabilities [83,84] minimizing the transient real-time (RTP/UDP) multimedia packet loss during the duration that a MH completes the SIP re-INVITE process. The former approach proposes interworking of SIP and Cellular IP by extending the Cellular IP Gateway as a proxy between the MH and CH. The gateway registers CH in its caches, and decides whether a SIP response message should be forwarded towards a registered MH. The latter adopts either the RTP translator or the SIP back-to-back UA as the proxy between MH and CH. The purpose of RTP translator is to bind the old IP address used by the MH and forward any incoming packets to the new address of the MH. For SIP back-to-back UA, it combines two SIP UAs physically so that one UA receives a SIP request from MH whenever it moves, transforms the request by replacing MH's IP address with its own address as the media destination, and the other UA re-issues the request to the CH. Thus, SIP back-to-back UA intercepts and forwards packets destined for the MH, via RTP translation to hide the movement of MH from CH.

4.6 Summary

This chapter covers the definition and components constituting mobility, literature survey of related work on current solutions and issues of supporting mobility in the Internet from different perspectives of network, transport, and application layer. It extensively presents Mobile IPv6 in terms of its data structures and major operations. It finally summarizes literature survey of mobility support using SIP for both terminal and personal mobility based real-time and TCP-based communication.

Chapter 5.
Analytical Study of SIP Mobility Support and Mobile IPv6

Section 5.1 introduces the motivation for analytical study between *SIP Mobility Support* (MSIP) and *Mobile IPv6* (MIPv6) in supporting mobility for real-time and TCP-based communication. Section 5.2 covers the major architectural and functionality similarities and differences. Section 5.3 presents quantitative analysis and comparison between MSIP and MIPv6 based on the signaling load, data packet overhead generated, registration time which is a measure of handover delay, and session establishment latency incurred by both protocol.

5.1 Motivation for Analytical Study

Design of *SIP Mobility Support* (MSIP) and *Mobile IPv6* (MIPv6) represents a natural combination of experiences gained independently from opportunities provided by the design, development, and deployment of IPv6 and SIP respectively. Both approaches address four important mobility issues namely Location Independent Addressing, Address Translation, Packet Forwarding, and Location Management. In addition, from the definition of mobility and protocol stack perspective, *Terminal Mobility* (TM) and *Personal Mobility* (PM) are supported separately and independently by solutions from network (i.e. MIPv6) and application layer (i.e. SIP) respectively. This motivates a natural progression in extending both protocols with both PM and TM, or just simply superimpose both layers mobility schemes for a complete mobility management solution. There is an open issue of which protocol or combination of protocols would be the choice for deployment in supporting both PM and TM in the wireless based Internet, especially to minimize performance inefficiency like signaling load and handover latency. However, no prior research work has been reported in the literature to resolve this issue.

A detailed analytical study and comparison between MSIP and MIPv6 is conducted from two perspectives of qualitative and quantitative. The former evaluates

their internalities and functionalities summarizing both protocols' registration operations, two-tier addressing scheme, address translation mechanism, entities, and data structures. The latter studies signaling load, data packet overhead generated, registration time which is a measure of handover delay, and session establishment latency incurred by both protocol.

5.2 Qualitative Analysis of Mobile IPv6 (MIPv6) and Mobile SIP (MSIP)

5.2.1 Properties and Features Analysis of MIPv6 and MSIP

Table 5.1 enumerates properties of *Mobile IPv6* (MIPv6) and *Mobile SIP* (MSIP).

Properties	Mobile IPv6	Mobile SIP
Deployability	Network layer Operating System requires changes	Application layer Easier to deploy as Operating System requires no changes
Signaling Messages	Byte-oriented In/Out-of-band IPv6 Destination Option	Textual Carried by UDP or TCP
End-to-End Communication	Tunneled (CH has no MH's binding) Optimized (CH has MH's binding) Uses Home Agent and IPv6 routers	Optimized and non- tunneled Uses only IPv6 routers
Types of Mobility Support	Supports terminal but not personal mobility Transparent to UDP, TCP & RTP No provision of roaming between administrative domain	Supports both terminal and personal mobility (UDP & RTP) TCP requires extra module Not transparent to transport layer Provision of roaming between administrative domain

Table 5.1: Performance Matrix of Mobile IPv6 and Mobile SIP

Protocol deployability refers to the installation and integration of a protocol with respect to the Operating System. MIPv6 is a network layer mobility solution designed with the principle that application residing on Internet hosts should be unmodified while requiring changes or extensions to IPv6 protocol stack for stateless autoconfiguration, destination options, and source routing. In contrast, MSIP is an application layer protocol that extends SIP with *Terminal Mobility* (TM) support through the ease of deployment as an application instalment over the Operating System at the end hosts, requiring no modification to the current IP protocol stack or existing IP infrastructure.

Signaling Messages describe the nature and format of the signaling used. MIPv6 specifies byte-oriented format (binary encoding/decoding scheme) for *Binding Update Destination Option* (BU) and *Binding Acknowledgement Destination Option* (BA) using IPv6 Destination Option i.e. *Home Address Destination Option* (HAD) and *Routing Extension Header* (RH). This facilitates MIPv6 signaling messages to be piggybacked by any existing IPv6 packets. In contrast, MSIP adopts textual format for signaling messages, which are typically transported by UDP or TCP packets. An implication of this is that byte-oriented format typically incurs smaller transmission overhead and bandwidth than the textual format.

End-to-End Communication evaluates whether the data traffic of both approaches follows an optimised path. For MIPv6 to establish data exchange, if *Correspondent Host* (CH) possesses no binding of a target *Mobile Host* (MH) which is away from its home network, CH first delivers data addressed to MH's *Home Address* (HAddr) which is intercepted by the *Home Agent* (HA), and then tunneled to the MH using IPv6 encapsulation/tunneling. However, the MH sends data to CH directly without transversing its HA. Three paths of communication are required i.e. a path from MH to CH, from CH to HA, and from HA to MH. Thus, MIPv6 does not always guarantee end-to-end direct communication between hosts, but relies on HA and routers for data delivery to MH. In contrast, for MSIP to establish a session, the caller's *User Agent Client* (UAC) issues a request i.e. an INVITE request containing the SIP URL of the called party to a SIP Server, which may be *SIP Proxy Server* (SIPPS) or *SIP Redirect Server* (SIPRS) for handling the request on behalf of the user. SIPPS contacts a *Location Server* (LS) to determine the current location of the called party. Eventually the session is accepted by the *User Agent Server* (UAS) of the called party and the response propagates back to the UAC of the caller directly. Direct end-to-end communication based on a single optimised routing path between hosts is always assured, meaning no tunneling, packet interceptor or forwarder is required.

Types of Mobility Support enumerates types of mobility support provided by both protocols. MIPv6 supports TM transparently to transport and application layer through IPv6 destination options, IPv6 address autoconfiguration, and uses the mapping between HAddr and MH's *Care-of Address* (COA) acquired when it moves between different subnets. However, MIPv6 has no provision for *Personal Mobility* (PM) or for *Mobile Users* (MUs) moving between access networks managed by same or differ-

ent administrative domains. MSIP supports PM by using Request-URI (i.e. SIP URL) scheme and registration mechanism, and provides MUs with the ability to roam and to obtain service in networks that may not necessarily be owned by their home service provider. However, MSIP supports TM for real-time communication experiences the following limitations. (I) Is not transparent to transport and application layer, as it is an application-based mobility solution. (II) Only covers unicast communications since INVITE is periodically sent to refresh each CH of MH's new COA. (III) TCP-based communication requires an extra software module SIP_EYE on both CH and MH.

5.2.2 Addressing Scheme Analysis of MIPv6 and MSIP

Table 5.2 summarizes the Addressing Scheme of MIPv6 and MSIP by contrasting different identifiers adopted. **Routing Identifier** is used by the network layer routing subsystem to deliver packets to the link on which the intended recipient or network interface of the packet resides. **Terminal Identifier** or **Reachability Identifier** is used for identifying and reaching a terminal, independent of point-of-attachment to signify the immutable identity of the terminal. It is commonly used at the transport layer to refer to the communication endpoint of a connection, e.g. TCP connections are uniquely identified by <Source Terminal Identifier, Source Port, Destination Terminal Identifier, Destination Port>. **User Identifier** identifies a user residing on a terminal e.g. SIP URL.

Types	Mobile IPv6	Mobile SIP
Routing Identifier	Temporary unicast address (COA)	Temporary unicast address (COA)
Terminal Identifier	Fixed IPv6 address (HAddr) Statically/Automatically Assigned	Not Supported
User Identifier	Not Supported	SIP URL

Table 5.2: Types of Identifier used by Mobile IPv6 and Mobile SIP

In general, both MIPv6 and MSIP adopt a two-tier addressing scheme [44] to provide a level of indirection between a MH's current location and an invariant end-point identifier. MIPv6 assigns to each MH, two IPv6 Aggregatable global unicast addresses namely HAddr and COA. HAddr is employed as the Terminal Identifier identifying the terminal and remains immutable when the terminal relocates. COA is the Routing Identifier specifying the current location of the MH, thus it changes with MH's subnet handoffs. MIPv6 provides no unique personal identifier for identification of an individ-

ual user, which MSIP inherently supports. In contrast, MSIP does not adopt a HAddr for Terminal Identifier but assigns to each user a SIP URL, a User Identity that uniquely identifies the user in the network and is permanent as long as the user continues using it. SIP URL uses a much readable and easy to remember format user_name@domain.com than the HAddr, which is an IPv6 address with its 16 octets hexadecimal representation. User's location is tracked by a temporary unicast IP address (i.e. COA) that also locates the MH as a Routing Identifier. MSIP associates the level-of-indirection between the SIP URL of user and current location of resided MH.

5.2.3 Address Translation Mechanism Analysis of MIPv6 and MSIP

Table 5.3 summarizes the functional comparison of MIPv6 and MSIP based on the following properties. **Address Translation Agent** that maps the endpoint-identifier to the routing identifier. **Forwarding Agent** that binds the routing identifier to the endpoint-identifier. **Location Directory** that records and maintains current mapping between the routing identifier and the endpoint-identifier, whenever MH first acquires a new COA or whenever it changes its location on the network. **Location Update Protocol** that is the messaging format used for updating the Location Directory.

Properties	Mobile IPv6	Mobile SIP
Address Translation Agent	Co-located with CHs and HA	Co-located with CHs and Registrar
Forwarding Agent	Co-located with CHs	Co-located with CHs
Location Directory	HA (Registration) HA and CHs (Location Update)	Registrars (Registration) Registrars and CHs (Location Update)
Location Update Protocol	BU and BA	REGISTER to Registrar INVITE to CH "200 OK" from Registrar and CH

Table 5.3: Address Translation Mechanism between Mobile IPv6 and Mobile SIP

MIPv6 specifies that HA maintains and monitors the mapping between the HAddr and the temporary COA while MH registers its newly acquired COA whenever it crosses subnet boundaries using an exchange of BU and BA messages. CH maintains the mapping between HAddr and temporary COA in its *Binding Cache* (BC), which it checks prior to sending data packets to MH, to ensure direct end-to-end communication between CH and MH. For MSIP, Registrar maps SIP URL (i.e. User Identity) to COA based on exchange of REGISTER and "200 OK" between SIP user and Registrar. CH

also maintains this mapping to reduce the time required in resolving the SIP URL to COA.

5.3 Quantitative Analysis of Mobile IPv6 and Mobile SIP

Table 5.4 summarizes symbols or notations, and corresponding definitions defined in the mathematical analysis. For abbreviation, given protocol $i = \{S, M\}$ where S and M denote *Mobile SIP* (MSIP) and *Mobile IPv6* (MIPv6) respectively.

Symbols	Definitions
SL	Signaling load, bandwidth used by control messages exchanged between hosts to update the location tracking of the other host
DL	Data load, bandwidth used by data packet overhead and raw data generated and consumed by the upper layer
f_i	Frequency at which signaling messages generated by protocol $i = \{S, M\}$
s_i	Average size of signaling messages for protocol $i = \{S, M\}$
$f_{M,HA}$	Frequency at which MH sends BU to HA for MIPv6
$f_{M,CH}$	Frequency at which MH sends BU to CHs for MIPv6
$f_{M,MOV}$	Rate at which MH updates HA or CH on its new COA due to relocating to a new subnet
$f_{M,REF}$	Rate of periodic update of BU by MH
$f_{M,B}$	Rate at which MH updates CH on its new COA for not more than M consecutive BU
$f_{S,R}$	Frequency at which MH sends REGISTER to *Registrar*
$f_{S,CH}$	Frequency at which MH sends INVITE to CHs
$f_{S,MOV}$	Rate at which MH updates Registrar or re-invite CH on its new COA due to relocating to a new location
$f_{S,REF}$	Rate of periodic update of REGISTER by MH
$f_{D,S}$	Average frequency at which MH sends data packets to CH
$f_{D,R}$	Average frequency at which MH receives data packets from CH
Δ_{SL}	Difference between Signaling Load for MSIP and MIPv6
BW_{WL}	Bandwidth of wireless link
BW_W	Bandwidth of wired link
L_{WL}	Latency of wireless link
L_W	Latency of wireless link
$S_{i,REG}$	Average size of registration packet that protocol i MH sends, where $i = \{S, M\}$
T_{ACQ}	Time for MH to acquire wireless channel, and to configure IPv6 address
D_{DAD}	Time for Duplicate Address Detection to ensure non-duplication of IPv6 address
$T_{i,REG}$	Time to generate registration packet for protocol $i = \{S, M\}$
$T_{i,SS}$	Time to generate session registration packet for protocol $i = \{S, M\}$
$S_{i,SS}$	Average size of session registration packet for protocol $i = \{S, M\}$
T_F	Time to process data packet at each hop
$T_{i,P}$	Time to process registration packet for protocol $i = \{S, M\}$

Table 5.4: Summary of Mathematical Abbreviations

Table 5.5 enumerate sample values for each network parameters defined in all equations.

Variables	Values
Frequency for periodic sending of BU, $f_{M,REF}$,	Once per 10 sec
Frequency for sending of M consecutive BUs, $f_{M,B}$	Once per sec
Frequency for periodic sending of REGISTER, $f_{S,REF}$	Once per 3600 sec
Number of consecutive transmission of BU, M	5
Number of CHs, N_{CH}	5
s_M is the average of BU ($D_B + D_{BU}$) and BA ($D_B + D_{BA}$)	60.5 Bytes
s_S is the sum of D_B, UDP header, and average of REGISTER, "200 OK", INVITE, and "200 OK"	295 Bytes
Size of IPv6 Header, D_B	40 Bytes
Size of *Home Address Destination Option* (HAD), $D_{HAOptions}$	20 Bytes
Size of *Routing Extension Header* (RH), D_{RH}	24 Bytes
Size of *Binding Update Destination Option* (BU), D_{BU}	28 Bytes
Size of *Binding Acknowledgement Destination Option* (BA), D_{BA}	13 Bytes
Average Size of Raw Data generated from upper layer, D_{RAW_DATA}	500 Bytes
Bandwidth of wireless link, BW_{WL}	1 Mb/s
Bandwidth of wired link, BW_W	100 Mb/s
Latency of wireless link, L_{WL}	7 ms
Latency of wired link, L_W	0.5 ms
Average size of registration packet that SIP MH sends, $S_{S,REG}$	295 Bytes
Average size of registration packet that MIPv6 MH sends, $S_{M,REG}$	60.5 Bytes
Average size of packet with destination options that SIP MH sends during session establishment, $S_{MD+S,SS}$	350.5 Bytes
Average size of tunneled packet that SIP MH sends during session establishment, $S_{MT+S,SS}$	330.5 Bytes
Time for DAD to ensure non-duplication of IPv6 address, D_{DAD}	1500 ms
Time to generate registration packet, $T_{M,REG}$. Time to process registration packet, $T_{M,P}$. Time to generate session establishment packet, $T_{M,SS}$	5 ms
Time to generate registration packet, $T_{S,REG}$. Time to process registration packet, $T_{S,P}$. Time to generate session establishment packet, $T_{S,SS}$	20 ms
Time to process data packet at each hop, T_F	7 ms

Table 5.5: Summary of Variables and Values (Analysis)

5.3.1 Signaling Load (SL) Analysis

Signaling Load (SL) is defined as the bandwidth occupied by control and signaling messages exchanged between hosts to track and refresh the location of the other host. The analysis is similar to that found in [74,85] but compares between MIPv6 and MSIP instead of MIPv6 and HMIPv6, adopts notion of subnet instead of site, and lastly MIPv6 is based on draft 15 instead of draft 09.

SL is dependent on two components namely f (frequency at which signaling messages are generated) and s (size of signaling messages). f_i is the frequency at which signaling messages is generated by protocol $i = \{S, M\}$, where S and M denote MSIP and MIPv6 respectively. s_i is the average size of signaling messages for protocol $i = \{S, M\}$. N_{CH} is the number of CHs communicating with MH. SL_j denotes the bandwidth

for control messages by both protocols at $j = \{\text{home, visit, roam}\}$ to denote three cases when MH resides in its home subnet, foreign subnet and moving across subnets respectively.

For MIPv6, communication between two *Mobile Hosts* (MHs) whenever either of them is away from its home network inherently requires an initial exchange of *Binding Update Destination Option* (BU) and *Binding Acknowledgement Destination Option* (BA) messages for route optimization, and consecutive exchange of these messages with *Home Agent* (HA) and *Correspondent Hosts* (CHs) at the frequency of $f_{M,HA}$ and $f_{M,CH}$ respectively to prevent expiration of *Binding Cache* (BC) entry.

Equation (5.1) shows $f_{M,HA}$ is dependent on $f_{M,REF}$ (rate of periodic update) and $f_{M,MOV}$ (rate at which MH refreshes HA on its new COA due to relocating to a new subnet). When $f_{M,REF} > f_{M,MOV}$, MH is moving at a slower rate than the rate of periodic update, thus BU is generated mainly due to both. When $f_{M,MOV} > f_{M,REF}$, MH refreshes HA on its new COA at a faster rate than the periodic update rate. It should be noted that for both cases, each BU received by the HA requires a BA to be replied back regardless of whether the sending of BU is triggered by the movement of MH or due to periodic update. In contrast, [74,85] assumes the HA only replies with a BA for those BU triggered by the movement of MH.

$$f_{M,HA} = \begin{cases} 2 \times (f_{M,REF} + f_{M,MOV}) & if \quad f_{M,REF} > f_{M,MOV} \\ 2 \times f_{M,MOV} & if \quad f_{M,MOV} \geq f_{M,REF} \end{cases} \tag{5.1}$$

Equation (5.2) expresses $f_{M,CH}$ which is dependent on $f_{M,REF}$ (rate of periodic update), $f_{M,MOV}$ (rate at which MH informs CH on its new COA to refresh the cache entries), and $f_{M,B}$ (rate at which MH updates CH on its new COA for not more than M consecutive BU). MH is restricted from issuing M consecutive BU for the same binding to CH more often than f_B, after which MH throttles the rate to $f_{M,REF}$. Three cases are involved. When $f_{M,REF} > f_{M,MOV}$, MH sends a regular periodic BU along with M - 1 consecutive BU to CH. When $f_{M,MOV} > f_{M,REF}$ but lesser than f_B/M, MH transmits M BU to CH at $f_{M,MOV}$. When $f_{M,MOV} > f_{M,REF}$ and f_B/M, MH transmits regular periodic BU to CH at $f_{M,REF}$.

Equation (5.3) shows $SL_{M,j}$ incurred by MIPv6, $j = \{\text{home, visit, roam}\}$. MH does not send any BU to CH when it resides in home network, since CH would have the *Home Address* (HAddr) of MH. When MH roams into foreign subnet, it updates each

of its CHs and HA using BU at $f_{M,CH}$ and $f_{M,HA}$ respectively. When MH roams from one subnet to another, it sends BU to CHs and to its HA (which replies a BA) at $f_{M,MOV}$.

$$f_{M,CH} = \begin{cases} f_{M,REF} + (M-1)f_{M,MOV} & if \quad f_{M,REF} > f_{M,MOV} \\ M \times f_{M,MOV} & if \quad \frac{f_B}{M} \geq f_{M,MOV} \geq f_{M,REF} \\ f_{M,REF} & if \quad f_{M,MOV} \geq \frac{f_B}{M} \geq f_{M,REF} \end{cases} \tag{5.2}$$

$$SL_{M,i} = \begin{cases} 0 & if \quad j = \{home\} \\ s_M \times (f_{M,CH} \times N_{CH} + f_{M,HA}) & if \quad j = \{visit\} \\ s_M \times (M \times N_{CH} + 2) & if \quad j = \{roam\} \end{cases} \tag{5.3}$$

For MSIP, MH updates the Registrar with its new location, or to refresh the mapping of SIP URL to *Care-of Address* (COA) at the frequency of $f_{S,R}$ and $f_{S,CH}$ respectively. Equation (5.4) depicts $f_{S,R}$ is dependent on $f_{S,REF}$ (rate of periodic update) and $f_{S,MOV}$ (rate at which MH updates Registrar on its new COA due to relocating to a new location). When $f_{S,REF} > f_{S,MOV}$, MH sends a REGISTER request to Registrar periodically and whenever it acquires a new COA. When $f_{S,MOV} > f_{S,REF}$, MH refreshes Registrar on its new COA at a faster rate than the periodic update rate. It should be noted that for both cases, each REGISTER request received by the Registrar requires a "200 OK" message to be replied back to MH.

$$f_{S,R} = \begin{cases} 2 \times (f_{S,REF} + f_{S,MOV}) & if \quad f_{S,REF} > f_{S,MOV} \\ 2 \times f_{S,MOV} & if \quad f_{S,MOV} \geq f_{S,REF} \end{cases} \tag{5.4}$$

Equation (5.5) shows $f_{S,CH}$ is dependent on $f_{S,MOV}$ (rate at which MH re-invite CH). Whenever MH moves to a new location, it re-invite CH to maintain any ongoing communication between by issuing an INVITE to CH, CH replies with a series of provisional responses i.e. "100 Trying" and "180 Ringing" response, and a final response "200 OK".

$$f_{S,CH} = 4 \times f_{S,MOV} \tag{5.5}$$

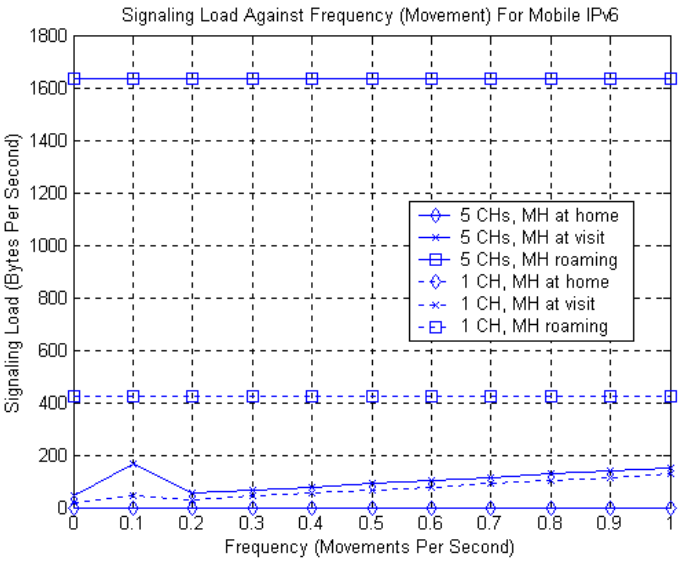
Equation (5.6) shows $SL_{S,J}$ incurred by MSIP where $j = \{\text{home, visit, roam}\}$. The first equation states that as long as MH is at its home subnet, it only periodically notifies its Registrar. The second case shows that when MH is at a foreign subnet, MH refreshes both CHs and Registrar at frequency of $f_{S,CH}$ and $f_{S,HA}$ respectively. The third case indicates that while MH roams at a frequency of $f_{S,MOV}$, it re-invites CHs and updates Registrar. CH replies with a series of provisional responses i.e. "100 Trying" and

"180 Ringing", and a final response "200 OK", while Registrar replies with a "200 OK" only. Using (5.3) and (5.6), the difference between SL_S and SL_M is given by (5.7).

$$SL_{S,j} = \begin{cases} 2 \times f_{S,REF} & if \quad j = \{home\} \\ s_S \times (f_{S,CH} \times N_{CH} + f_{S,R}) & if \quad j = \{visit\} \\ s_S \times (4 \times N_{CH} + 2) & if \quad j = \{roam\} \end{cases} \tag{5.6}$$

$$\Delta_{SL} = SL_S - SL_M \tag{5.7}$$

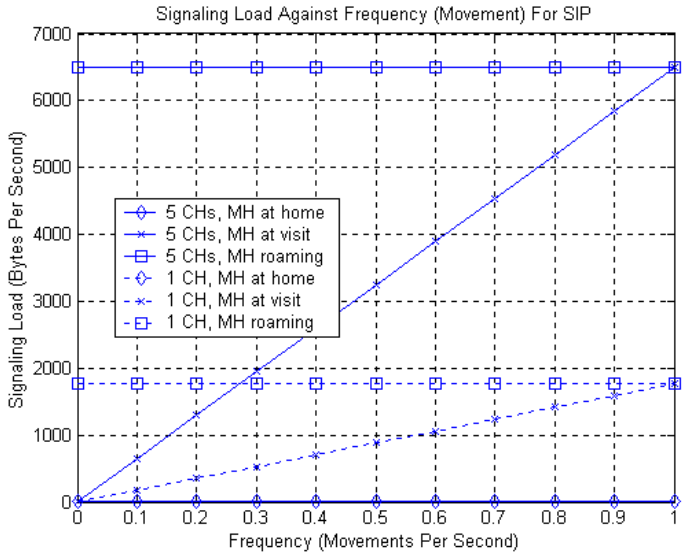
Graph 5.1 and Graph 5.2 present graphically the SL against frequency of movement computed from (5.3) and (5.6) for MH using MSIP and MIPv6 respectively. Both graphs assume MH communicates with either one CH or five CHs, and summarize three common patterns. (I) MH incurs greater SL during roaming than being stationary in the home or foreign network, as MH is required to notify both CHs and HA or Registrar whenever it acquires a new COA. (II) SL is negligible for MSIP while MH resides at its home network as MH refreshes Registrar periodically at $f_{S,REF}$ and none at all for MIPv6 since there exists no binding at HA. (III) SL incurs by MH moving in the foreign network increases with f_{MOV} (frequency of acquiring new COA due to relocation), as MH sends BU to HA and CH for MIPv6, and REGISTER to Registrar and INVITE to CHs for MSIP.



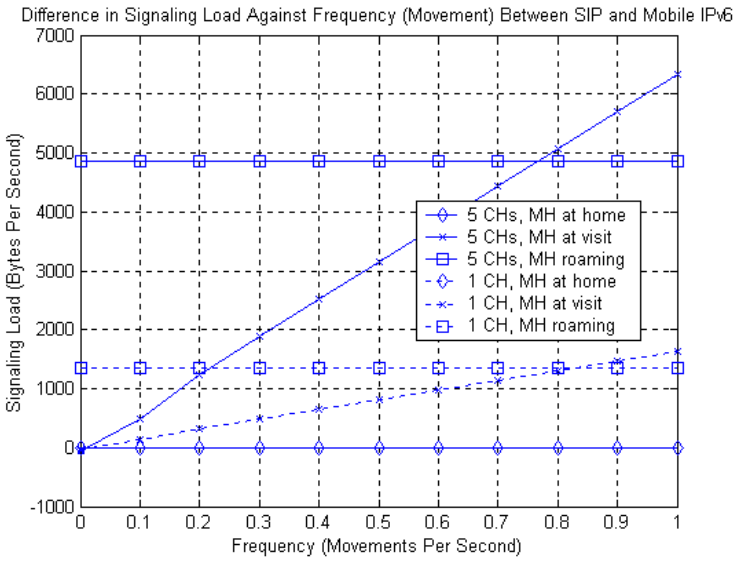
Graph 5.1: Signaling Load for Mobile IPv6

Equation (5.7) provides another perspective towards the SL incurred by MSIP and MIPv6. Graph 5.3 shows the difference in SL against frequency of movement between both protocols. Two notable observations are as follows. (I) MH occupies more band-

width for MSIP than MIPv6 when MH is either roaming or residing in the foreign subnet. The cause of the higher overhead for both scenarios is that MSIP requires an additional SIP message to re-invite CH to its new location and INVITE message is on the average five times larger than BUs. (II) There exists no significant difference in SL between MSIP and MIPv6 when MH resides in its home subnet, as MIPv6 does not require MH to send any BU to refresh the binding at its HA, but MSIP requires re-registration with Registrar once every 3600 seconds.



Graph 5.2: Signaling Load for Mobile SIP



Graph 5.3: Difference between SL_S and SL_M as a function of f_{MOV}

5.3.2 Data Load (DL) Analysis

Data Load (DL) is defined as bandwidth required for transmission and reception of data packets (consists data packet overhead and raw data generated and consumed by upper layer). Data packet overhead is defined as the portion of the IP packet not consumed by upper layer, but necessary for routing, multiplexing, or demultiplexing purposes. DL_i denotes bandwidth required for transmission and reception of data packets incurred by protocol $i = \{S, M\}$.

For MIPv6, MH delivers data packets to CH at average frequency of $f_{D,S}$, and receives data packets from CH at average frequency of $f_{D,R}$. Equation (5.8) considers three cases. The first case states both MH and CH are residing at respective home network, data packets are exchanged using normal IPv6 routing i.e. no extension headers are appended. The second case depicts when either MH or CH resides in a foreign network, the one residing in foreign network appends a *Home Address Destination Option* (HAD) denoted as $D_{HAOptions}$ to inform the recipient of that packet of the MH's HAddr, also the one residing in its home network appends a *Routing Extension Header* (RH) denoted as D_{RH} to route packets to MH through an optimal route. The third case shows two communicating MHs residing in foreign network, both include D_{RH} and $D_{HAOptions}$ to all of the exchanged packets.

$$DL_M = \begin{cases} (D_B + D_{RAW\_DATA})(f_{D,S} + f_{D,R}) \\ (D_B + D_{HAOptions} + D_{RAW\_DATA})f_{D,S} \\ \quad + (D_B + D_{RH} + D_{RAW\_DATA})f_{D,R} \\ (D_B + D_{HAOptions} + D_{RH} + D_{RAW\_DATA})(f_{D,S} + f_{D,R}) \end{cases} \tag{5.8}$$

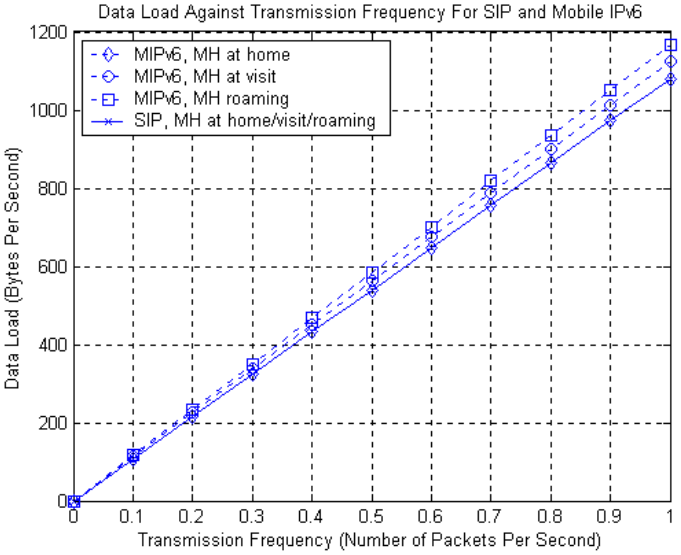
For MSIP, MH sends data packets to CH at average frequency of $f_{D,S}$ and receives data packets from CH at average frequency $f_{D,R}$. Equation (5.9) states MSIP does not incur any packet header overhead in terms of protocol headers or extension options of IPv6 whichever location MH may reside. Using (5.8) and (5.9), the difference between DL_S and DL_M is given by (5.10).

$$DL_S = (D_B + D_{RAW\_DATA})(f_{D,S} + f_{D,R}) \tag{5.9}$$

$$\Delta_{DL} = \begin{cases} 0 \\ D_{HAOptions} \times f_{D,S} + D_{RH} \times f_{D,R} \\ (D_{HAOptions} + D_{RH})(f_{D,S} + f_{D,R}) \end{cases} \tag{5.10}$$

Graph 5.4 illustrates the DL_S and DL_M as a function of $f_{D,S}$ and $f_{D,R}$ using (5.8) and (5.9), assuming that $f_{D,S} = f_{D,R} = f_P$. Both protocols experience a corresponding increase

in DL with transmission frequency. Graph 5.5 is based on (5.10) further illustrates that when both MH and CH are at their home network, there is no difference in overhead as normal IPv6 routing is used for both protocols. However, when both parties reside at foreign network, MIPv6 incurs a larger header overhead than MSIP, due to either the appended HAD $D_{HAOptions}$ to notify the recipient of that packet of the MH's HAddr, or RH D_{RH} to route packets to MH through an optimal route. In addition, the overhead difference is most significant when both MH and CH are roaming away from its home network, the overhead incurred for MIPv6 is greater than that of MSIP due to additional header extensions, viz. RH D_{RH} and HAD $D_{HAOptions}$. This affects significantly on real-time communication especially Internet Telephony which requires low delay.

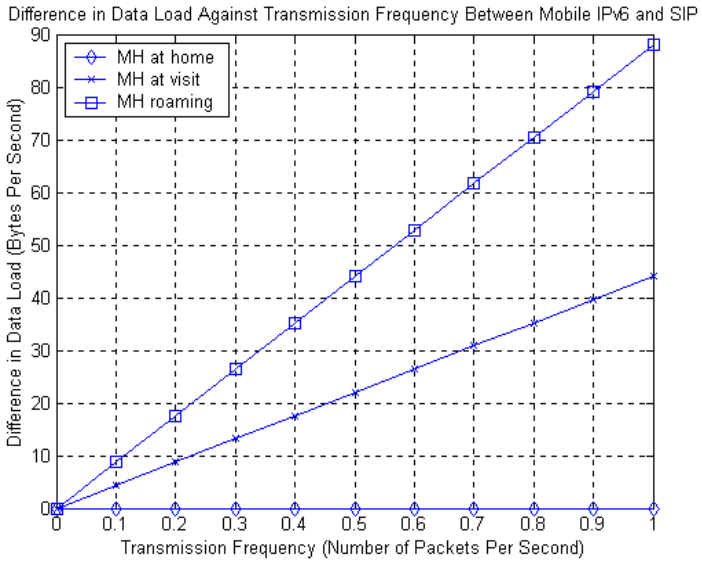


Graph 5.4: DL_S and DL_M as a function of $f_{D,S}$ and $f_{D,R}$

Equation (5.11) and (5.12) state the respective conditions that MSIP and MIPv6 would be appropriate for. Both equations can be solved rigorously by substituting (5.4), (5.5), (5.6), (5.8), (5.9), and (5.10). Alternatively, Δ_{SL} and Δ_{DL} can be expressed into (5.13) and (5.14) respectively using Table 5.6. When MH is at its home network, the difference in overhead is negligible since $\Delta_{SL} \approx \Delta_{DL}$, meaning either MSIP or MIPv6 may be used. When MH relocates to or roaming away in the foreign network, either (5.15) or (5.16) must be satisfied for MIPv6 suitability, otherwise MSIP should be used.

$$\Delta_{DL} > \Delta_{SL} \Rightarrow \text{MSIP is suitable} \tag{5.11}$$

$$\Delta_{SL} > \Delta_{DL} \Rightarrow \text{MIPv6 is suitable} \tag{5.12}$$



Graph 5.5: Difference between DL_S and DL_M as a function of $f_{D,S}$ and $f_{D,R}$

$$\Delta_{SL} = \begin{cases} 0.000556, \text{ MH at Home network} \\ 1660 f_{MOV}, \text{ MH at Foreign network} \\ 1346.5, \text{ MH roaming} \end{cases} \tag{5.13}$$

$$\Delta_{DL} = \begin{cases} 0, \text{ MH at Home network} \\ 44 f_P, \text{ MH at Foreign network} \\ 88 f_P, \text{ MH roaming} \end{cases} \tag{5.14}$$

Δ_{SL}, Difference in Signaling Load/Bps				Δ_{DL}, Difference in Data Load/Bps			
f_{MOV}	MH at Home	MH at Visit	MH roaming	f_P	MH at Home	MH at Visit	MH roaming
0.0	0.000556	-17.9861	1346.5	0.0	0	0.00	0.00
0.1	0.000556	134.65	1346.5	0.1	0	4.40	8.80
0.2	0.000556	323.75	1346.5	0.2	0	8.80	17.6
0.3	0.000556	488.65	1346.5	0.3	0	13.2	26.4
0.4	0.000556	653.55	1346.5	0.4	0	17.6	35.2
0.5	0.000556	818.45	1346.5	0.5	0	22.0	44.0
0.6	0.000556	983.35	1346.5	0.6	0	26.4	52.8
0.7	0.000556	1148.25	1346.5	0.7	0	30.8	61.6
0.8	0.000556	1313.15	1346.5	0.8	0	35.2	70.4
0.9	0.000556	1478.05	1346.5	0.9	0	39.6	79.2
1.0	0.000556	1642.95	1346.5	1.0	0	44.0	88.0

Table 5.6: Values of Δ_{SL} and Δ_{DL}

$$\Delta_{SL} > \Delta_{DL} \Rightarrow 37.8 f_{MOV} > f_P \tag{5.15}$$

$$\Delta_{SL} > \Delta_{DL} \Rightarrow 15.3 > f_P \tag{5.16}$$

5.3.3 Handover Delay (HOD) Analysis

Handover Delay (HOD) is delay or latency incurred at location update by MIPv6 and MSIP whenever MH is away from its home network, which is related to the packet loss experienced by MH whenever it crosses to a new subnet.

Let the transmission time of a packet on the wireless and wired link be D_{WL} and D_W respectively as expressed in (5.17) and (5.18). BW_{WL} and BW_W represent bandwidth for wireless and wired links, while L_{WL} and L_W represents latency for wireless and wired links. T_F is the required time to route packets at each hop. S is the size of data packet transmitted over the wireless or wired media. N_{HOPS} is the number of hops in the wired network.

$$D_{WL}(S) = \frac{S}{BW_{WL}} + L_{WL} \tag{5.17}$$

$$D_W(S,N) = \left(\frac{S}{BW_W} + L_W + T_F \right) N_{HOPS} \tag{5.18}$$

RT_{WL} sums up the acquisition time T_{ACQ} of wireless channel, *Duplicate Address Detection* (DAD) processing time D_{DAD} to verify the uniqueness of obtained COA, T_{REG} time to generate the registration packet, D_{WL} of the registration packet across wireless link. RT_W expresses the sum of D_W of registration packet transmitted across wired network and T_P time to process the registration packets. $RT_{i,WL}$ and $RT_{i,W}$ are expressed in (5.19) and (5.20) respectively, where protocol $i = \{S, M\}$.

T_{ACQ} is given in (5.21) comprising D_1 (constant delay for switching lower layer medium to access network, specific to link layer technologies), D_2 is given in (5.22) (delay for listening to periodically broadcast *Router Advertisement* (RA) from new *Access Router* (AR) or issues unsolicited *Router Solicitation* (RS) to determine the subnet change), and D_3 (configuration of its interface using stateless address autoconfiguration considered negligible). Average of D_{DAD} is depicted in (5.23). D_{DAD} depends on D_{RAND} which is a uniformly distributed random delay ranging from 0 to $D_{RAND,MAX}$ (Default is 1000 ms) before transmitting N_{DAD} (Default is 1) of NS at D_{RET} interval (Default is 1000 ms). D_{DAD} is the required time to resolve the issue of link-local IPv6 address duplication on the same wireless subnet.

$$RT_{i,WL} = T_{i,ACQ} + D_{DAD} + D_{WL}(S_{i,REG}) + T_{i,REG}, \ \forall \ \text{i}=\{\text{S, M}\} \tag{5.19}$$

$$RT_{i,W}(N_{HOPS}) = D_W(S_{i,REG}, N_{HOPS}) + T_{i,P},\ \forall\ \text{i}=\{\text{S, M}\} \tag{5.20}$$

$$T_{ACQ} = D_1 + D_2 + D_3 \tag{5.21}$$

$$D_2 = uniform(0, \text{one RA interval}),\ \forall\ \text{i}=\{\text{S, M}\} \tag{5.22}$$

$$D_{DAD} = D_{rand} + N_{DAD} \times D_{ret} = uniform(0, D_{RAND,MAX}) + N_{DAD} \times D_{RET}$$

$$\overline{D_{DAD}} = \frac{D_{RAND,MAX}}{2} + N_{DAD} \times D_{RET} \tag{5.23}$$

$$= 1500\ ms$$

For MIPv6, (5.24) derives two cases of RT overhead incurred by MH. The first case is when MH sends a BU to HA and receives a BA from HA. The second case is when MH sends a BU to CH but it is optional for CH to reply with a BA. N_{MH_HA} and N_{MH_CH} are the number of hops from MH to HA/CH respectively.

$$RT_M = \begin{cases} RT_{M,WL} + RT_{M,W}(N_{MH\_HA}) + T_{M,REG} \\ \quad + D_W(S_{M,REG}, N_{MH\_HA}) + D_{WL}(S_{M,REG}) + T_{M,P} \\ RT_{M,WL} + RT_{M,W}(N_{MH\_CH}) \end{cases} \tag{5.24}$$

For MSIP, (5.25) derives two cases of RT overhead incurred by MH. The first case is when MH sends a REGISTER requests to home Registrar and receives a "200 OK" response from it, while the second case is MH sends an INVITE request to CH, receives a "200 OK", and then sends an ACK request to CH again. $N_{MH\_REGISTRAR}$ and $N_{REGISTRAR\_MH}$ are the number of hops between MH and Registrar.

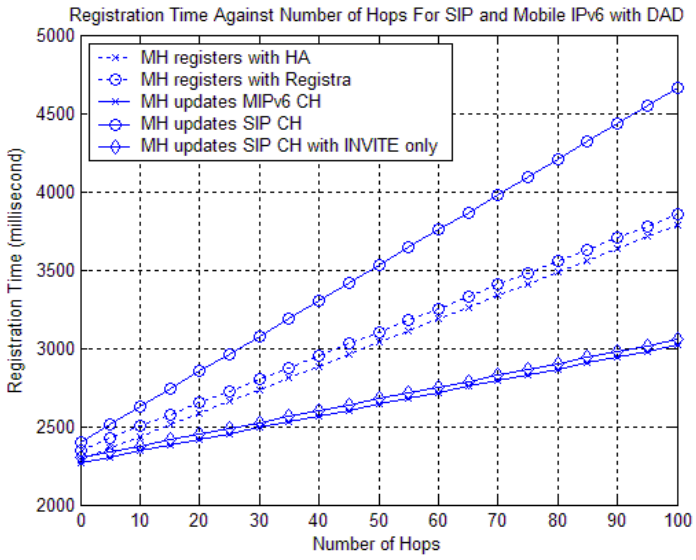
$$RT_S = \begin{cases} RT_{S,WL} + RT_{S,W}(N_{MH\_REGISTRAR}) + T_{S,REG} \\ \quad + D_S(S_{S,REG}, N_{REGISTRAR\_MH}) + D_{WL}(S_{S,REG}) + T_{M,P} \\ RT_{S,WL} + RT_{S,W}(N_{MH\_CH}) + 2[T_{S,REG} \\ \quad + D_S(S_{S,REG}, N_{REGISTRAR\_MH}) + D_{WL}(S_{S,REG}) + T_{M,P}] \end{cases} \tag{5.25}$$

Given RT, the number of packet lost $L_{HO,i}$ is given in (5.26) during handover. T_O is the overlapping time between two adjacent cells, S_R is the average sending rate of data from CH to MH for a session, and P_S is the average packet size. Equation (5.26) shows that MSIP experiences greater packet loss than MIPv6 since $RT_S > RT_M$. DAD contributes significantly towards handoff latency, thus [47] advices MH after forming a new COA may begin using it immediately without performing DAD.

$$L_{HO,i} = (RT_i - T_O)\frac{S_R}{P_S},\ \text{i}=\{\text{S, M}\} \tag{5.26}$$

Equations (5.24) and (5.25) depicted in Graph 5.6 and Graph 5.7 for RT with and without DAD respectively, assuming $N_{MH\_TO\_REGISTRAR} = N_{MH\_TO\_HA}$, $T_{i,REG} = T_{i,P}$, and $T_{S,REG}$ is four times greater than $T_{M,REG}$. Three notable observations are as follows. (I)

DAD contributes significantly to RT. (II) RT incurred by MIPv6 (BU/BA between MH and HA) outperforms slightly to MSIP (REGISTER/"200 OK" between MH and Registrar) as the latter exchange two SIP messages which are relatively larger than MIPv6 signaling messages. (III) Location update approach of MIPv6 is more efficient than MSIP for refreshing the binding at the CH as MSIP requires further exchange of a "200 OK" response and an ACK request, while sending a BA back to MH is optional for MIPv6. This last observation indicates that the RT incurred by MSIP can be reduced comparably by avoiding the exchange of "200 OK" response and ACK request, which is only necessary during initial session establishment for the negotiation of Codec and other parameters.

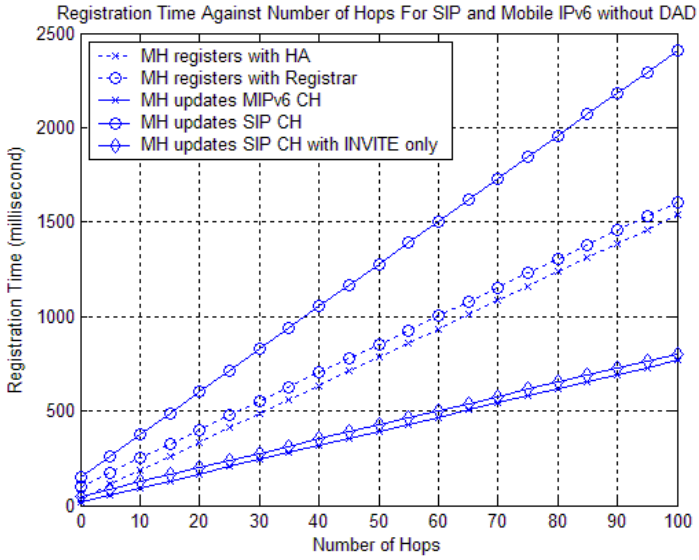


Graph 5.6: Registration Time between Mobile SIP and Mobile IPv6 with DAD

5.3.4 Session Establishment Time (SST) Analysis

Session Establishment Time (SST) is defined as the time duration associated with MH being invited into a session by CH through an exchange of an INVITE request, a "200 OK" response, and an ACK request to complete the session initiation phase. The time required for transmission over wireless and wired networks are given in (5.27) and (5.28). $S_{i,SS}$ and $T_{i,SS}$ are defined as average size of packet used for session establishment and corresponding time for processing or generating it, where $i = \{S, M\}$. Session establishment packets can be exchanged between CH and MH via three modes when MH is away from its home network: normal routing, tunneled or appended with destination

options such as *Home Address Destination Option* (HAD) and *Routing Extension Header* (RH), denoted as $k = \{S, MT+S, MD+S\}$ respectively.



Graph 5.7: Registration Time between Mobile SIP and Mobile IPv6 without DAD

$$SST_{k,WL} = D_{WL}(S_{i,SS}) + T_{i,SS},\ \forall\ k=\{S, MD+S, MT+S\} \tag{5.27}$$

$$SST_{k,W}(N_{HOPS}) = D_{W}(S_{i,SS}, N_{HOPS}) + T_{P},\ \forall\ k=\{S, MD+S, MT+S\} \tag{5.28}$$

Equation (5.29) and (5.30) collectively derives the SST when MH is away from its home network for MIPv6 and MSIP respectively. For MIPv6, CH invites MH to a session by sending an INVITE request to MH's home Registrar, which resolves and forwards the request to HA. HA tunnels it to MH, which replies with a "200 OK" response to CH, and then receives an ACK request from CH. However, it should be noted that MH does not send any BU to CH as the INVITE request was previously forwarded from home Registrar, implying the "200 OK" response and the ACK request do not carry any MIPv6 related destination options. For the latter, similar procedure occurs but bypassing HA. $N_{CH\_REGISTRAR}$ and $N_{REGISTRAR\_HA}$ are the number of hops between CH and Registrar, and Registrar and HA respectively.

$$\begin{aligned} SST_{M} &= SST_{S,W}(N_{CH\_REGISTRAR}) + SST_{S,W}(N_{REGISTRAR\_HA}) \\ &\quad + SST_{MT+S,W}(N_{HA\_MH}) + SST_{MT+S,WL} + 2[SST_{S,WL} \\ &\quad + SST_{S,W}(N_{MH\_CH})] \end{aligned} \tag{5.29}$$

$$\begin{aligned} SST_{S} &= SST_{S,W}(N_{CH\_REGISTRAR}) + SST_{S,W}(N_{REGISTRAR\_MH}) \\ &\quad + 3SST_{S,WL} + 2SST_{S,W}(N_{MH\_CH}) \end{aligned} \tag{5.30}$$

Difference between SST_M and SST_S is given in (5.31), further simplified with sub-

stitution from (5.17) and (5.18).

$$\begin{aligned} & SST_M - SST_S \\ &= SST_{S,W}(N_{REGISTRAR\_HA}) + SST_{MT+S,W}(N_{HA\_MH}) \\ &\quad + SST_{MT+S,WL} - SST_{S,W}(N_{REGISTRAR\_MH}) - SST_{S,WL} \\ &= (N_{REGISTRAR\_HA} - N_{REGISTRAR\_MH})(\frac{S_{S,SS}}{BW_W} + L_W + T_F) \\ &\quad + N_{HA\_MH}(\frac{S_{MT+S,SS}}{BW_W} + L_W + T_F) + \frac{D_B}{BW_{WL}} \end{aligned} \tag{5.31}$$

Equation (5.31) can be simplified to (5.32) showing SST incurred by MIPv6 is much higher than that of MSIP whenever $N_{REGISTRAR\_HA} + N_{HA\_MH} > N_{REGISTRAR\_MH}$. An excessive delay may result in a session being abandoned. Otherwise, MSIP incurs greater SST than MIPv6.

$$\begin{aligned} & SST_M - SST_S \\ &\approx (N_{REGISTRAR\_HA} + N_{HA\_MH} - N_{REGISTRAR\_MH})(L_W + T_F) \end{aligned} \tag{5.32}$$

5.4 Summary from Quantitative Analysis

The following observations can be concluded from the analysis.

(I) Both *Mobile IPv6* (MIPv6) and *Mobile SIP* (MSIP) incur high signaling traffic or *Signaling Load* (SL) transmitted and received by *Mobile Host* (MH) for location tracking to inform *Home Agent* (HA)/Registrar of every change in MH's *Care-of Address* (COA), and to periodically renew the binding at HA/Registrar and *Correspondent Hosts* (CHs) before their lifetime expires, whenever it is away from its home subnet. Although, MIPv6 is more efficient than MSIP in terms of lower SL incurred by location management, since $\Delta_{SL} = SL_S - SL_M > 0$ occurs regardless of whether the MH resides in its home, foreign network or when it is roaming. The explanation is as follows. For MIPv6, both CHs and HA periodically exchange *Binding Update Destination Option* (BU) and *Binding Acknowledgment Destination Option* (BA) with MH to refresh the mapping of HAddr to COA. Similarly for MSIP, re-registration with Registrar is required once every 3600 seconds, and CH is re-invited to MH's new location. However, MSIP exchanges messages less frequently than MIPv6, and INVITE message is on the average five times larger than BU. MIPv6 and MSIP incur high signaling traffic to update network entities of MH's current location even when MH is relatively stationary with minimal change in its COA per unit time. This inefficiency would inherently translate into higher subscription charges in a pay-as-you-use billing plan and

reduce the bandwidth that would otherwise be available for meaningful information transmission

(II) MIPv6 incurs higher data overhead or *Data Load* (DL) than MSIP during data exchange between MH and CHs, since $\Delta_{DL} = DL_M - DL_S > 0$ occurs regardless of whether the MH resides in its home, foreign network or when it is roaming. This is because, whenever MH is away from its home subnet, MIPv6 specifies destination options headers like *Routing Header* (RH) and *Home Address Destination Option* (HAD) to be appended to each data packet transmitted or received by MH for mobility transparency to upper layer; such mechanism is not adopted by MSIP which relies purely on conventional IPv6 routing. This high data overhead incurred by MIPv6 is experienced regardless of whether the MH is relatively stationary in the foreign network or is frequently moving in the foreign networks, since MIPv6 is unable to differentiate both scenarios. This would affect significantly on real-time communication especially Internet Telephony which requires low delay, and result into higher subscription charges.

(III) Both protocols incur high handover delay due to *Duplicate Address Detection* (DAD) which contributes significantly to Registration Time, though the delay experienced by MIPv6 is lower than that of MSIP. DAD is a specified functionality of IPv6, thus whenever the MH moves from one subnet to another, it requests the service of previous *Access Router* (AR) to forward data packets to new AR. MH would experience two DAD processes namely MH-initiated and HA-initiated DAD. The former occurs when MH generates a new COA via stateless address configuration, which can be avoided as suggested by MIPv6 specification but risks the address duplication problem. The latter is triggered when MH sends BUs to previous AR and HA, both ensure there exists no address duplications, update their *Binding Cache* (BC) with MH's new COA, and send a BA to MH. HA-initiated DAD can be avoided by disabling the Duplicate Address Detection (D) bit in the BU but risks invalidating the uniqueness of the link-local address generated. This issue is further compounded when HA is strictly designed to perform DAD at its home network and when MH specifically stipulates this request by enabling the D-bit in the BU. This will significantly degrade the handover process as MH awaits a BA from HA. In addition, MIPv6 is more efficient than MSIP in terms of lower Registration Time, because the Registration Time incurred by MIPv6 (BU/BA between MH and HA) outperforms slightly to MSIP (REGISTER/"200 OK" between MH and Registrar) as the latter exchange two SIP messages, which are rela-

tively larger than MIPv6 signaling messages.

(IV) MSIP, in general, incurs lower session establishment delay than MIPv6 whenever MH is away from its home network, unless the condition of $N_{REGISTRAR\_HA} + N_{HA\_MH} > N_{REGISTRAR\_MH}$ is violated. This is because, MIPv6 specifies that CH initiates communication with MH; the first packet will be tunneled to MH via HA, even when MH and CH are in neighbouring subnets. For illustration, consider two *Mobile Users* (MUs) are establishing a communication session. The INVITE request transmitted by the initiator is forwarded to MU's Registrar, which resolves the User Identity to MH's HAddr, and then forwards the INVITE request to HA. Finally, HA tunnels the INVITE request to MH, which forwards it to MU. This incurred session establishment delay is not an issue if the proximity of HA and MH is within reasonable margin. Otherwise, this will delay the commencement of data exchange, and may result in the initiator abandoning the session.

(V) Both protocols incur significant reduction in the following network parameters, whenever MH is residing in its home network: Signaling traffic as MH does not exchange signaling messages periodically to refresh binding at CHs and at HA, or to update its location at CHs and at Registrar. Data overhead as there exists no necessity for appending destination options headers including RH and HAD to every data packets exchanged with CHs for mobility transparency to upper layer. Session establishment latency due to close proximity of temporary HA/Registrar and MH. However, MH would still experience considerable handover delay incurred by DAD to ensure uniqueness of its IPv6 address as required by IPv6 specification.

5.5 Summary

This chapter presents a detailed analytical study and comparison between SIP Mobility Support and Mobile IPv6 from two aspects of qualitative and quantitative. The former evaluates that both protocols are similar in terms of registration operations, two-tier addressing scheme, address translation mechanism, entities, and data structures. The latter compares signaling load, data packet overhead generated, handover delay, and session establishment latency incurred by both protocol. The next step in this effort is to design an architecture that leverages on the strengths of both protocols, while overcoming their limitations, to support terminal and personal mobility for both peer-to-peer and client/server communication seamlessly in the wireless Internet.

Chapter 6.
On-demand Mobility Agent and Mobility Address Assignment

Section 6.1 covers the assumptions and motivations of the proposed architecture i.e. **On-demand Mobility Agent and Mobility Address Assignment** designed with the objective to minimize the inefficiencies experienced by *Mobile IPv6* (MIPv6) and *Mobile SIP* (MSIP) by harmonizing the interaction and coexistence between both protocols. The architecture supports seamless terminal and personal mobility for both peer-to-peer and client/server communication ubiquitously within the wireless Internet. It adopts two newly designed SIP header extensions **Assign** and **Assigned**, and a set of modified MIPv6 signaling messages for its operation. Section 6.2 describes the realization of the proposed architecture and elaborates the detailed operations for registration, allocation of **Mobility Address** and **Mobility Agent** dynamically per communication session, intra and inter domain handoff, and session establishment for peer-to-peer and client/server scenarios. Section 6.3 presents a qualitative analysis of the proposed architecture in comparison with MIPv6 and MSIP, to illustrate that the proposed solution improves the performance of MIPv6 using the strength of MSIP, when the mobile terminal is relatively stationary in the foreign network.

6.1 On-demand Mobility Agent and Mobility Address Assignment (OMA)

6.1.1 Objective and Motivations of OMA

On-demand Mobility Agent and Mobility Address Assignment (OMA) is a mechanism designed with the objective to minimize the inefficiencies (covered in details in 5.4) experienced by *Mobile IPv6* (MIPv6) and *SIP Mobility Support* (MSIP), in particular when the mobile terminal is relatively stationary in the foreign network. It improves the performance of MIPv6 using the strength of MSIP with capabilities to support terminal and personal mobility for both peer-to-peer and client/server communi-

cation seamlessly in the wireless Internet. In summary, these inefficiencies include the high signaling load for location management incurred by MIPv6 and MSIP, the high overhead for data transmission experienced by MIPv6, the high session establishment suffered by MIPv6, and lastly the high handover delays incurred by both protocols.

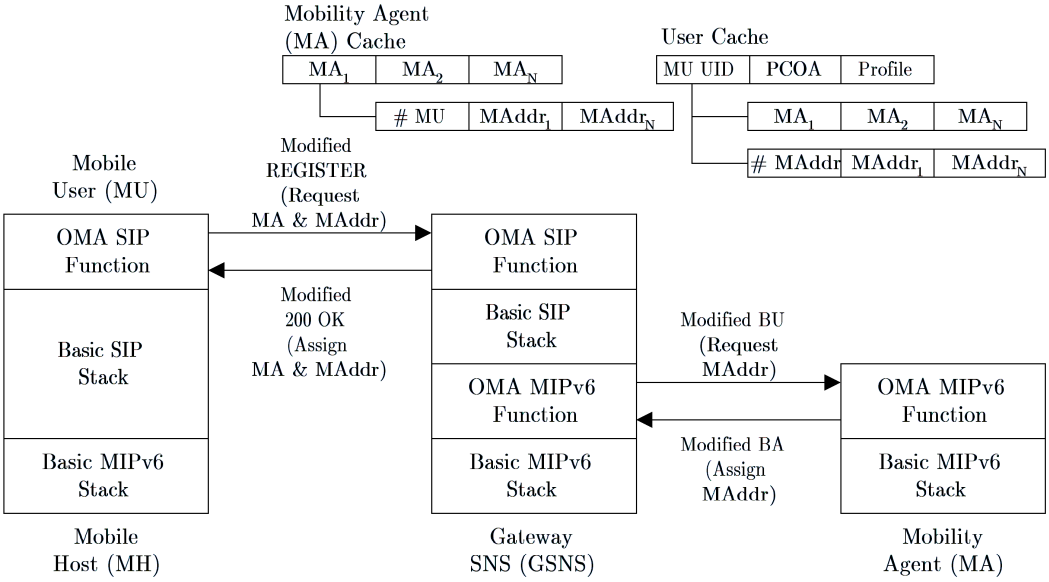
The design of OMA takes into account the spatial locality of the mobile terminal, motivated by daily observations and numerous proposals for localizing its mobility. For illustration, an engineer working within a cubicle would establish session many times a day within that same location, or a professor attending overseas conference would devote considerable amount of time at either the conference venue or at the hotel suite. Thus, OMA assumes that the mobile terminal spends a relatively large part of its time within a physical area in the foreign network, would not be pre-assigned with a home network to which it is aware of the subnet prefix for executing the stateless autoconfiguration of its *Home Address* (HAddr). OMA facilitates the mobile terminal to lease a temporary HAddr from a temporary *Home Agent* (HA) in the corresponding virtual network as its "home" whenever it requires them solely for session establishment and data exchange. Given that the mobile terminal resides in its "home" with the assigned temporary HAddr and temporary HA, OMA avoids the need for the following operations: Exchanging signaling messages periodically to refresh binding, or to update mobile terminal's current location. Appending every data packets with destination options headers for mobility transparency to upper layer. Triangular routing and tunneling due to close proximity of temporary HA and mobile terminal.

In order for the mobile terminal to avoid considerable handover delay incurred by *Duplicate Address Detection* (DAD) mechanism to ensure uniqueness of its IPv6 address as required by IPv6 specification, OMA adopts a solution to bypass DAD. The solution is based on the fact that DAD is only performed for network link with existing neighbours, which may be hosts, routers, or servers. Thus, the key approach is to design and create a virtual network with no other neighbours except the temporary HA itself. With such a virtual network, when a typical mobile terminal issues a request to the temporary HA to perform DAD, the temporary HA can immediately assess the mobile terminal's temporary HAddr duplication, bypass DAD, and reply promptly to the mobile terminal that DAD is successful and that the address is not duplicated. The temporary HA residing in the virtual network is fully aware of all the active IP addresses assigned as temporary HAddr stored in its local directory can be checked

thoroughly and efficiently. This reduces potential handover delay, whenever the mobile terminal changes its point-of-attachment to different subnets, or even when it stipulates DAD to be performed at the virtual network.

6.1.2 Overview of OMA

Figure 6.1 depicts the network components and signaling protocol of OMA. The network entities comprises of a *Mobile User* (MU) residing on its *Mobile Host* (MH) which is a mobile terminal, a *Gateway SIP Network Server* (GSNS), and a *Mobility Agent* (MA). The signaling protocol adopts two newly designed SIP header extensions **Assign** and **Assigned**, and a set of modified MIPv6 *Binding Update Destination Option* (BU) and *Binding Acknowledgment Destination Option* (BA) signaling messages for its operation. For ease of discussion, the set of **Assign** and **Assigned**, and the set of modified MIPv6 are referred to as OMA SIP function and OMA MIPv6 function respectively, discussed in greater details in 6.1.3 and 6.1.4.



BU: Binding Update
MA: Mobility Agent
MU UID: Mobile User Identity
PCOA: Physical Care-of Address of Mobile Host
MAddr: Mobility Address assigned to Mobile User
OMA: On-demand Mobility Agent and Mobility Address Assignment

Figure 6.1: On-demand Mobility Agent and Mobility Address Assignment

MU is an enhanced SIP *User Agent* (UA) with OMA SIP function for execution of on-demand registration, allocation, and deregistration of a MA and assignment of a set of corresponding *Mobility Address* (MAddr) which is the temporary HA, with the

GSNS, for configuring its MH on the visiting network per communication session basis. MU resides on the MH, which provides a unmodified basic MIPv6 stack for supporting features like exchanging binding messages to refresh its current location, or appending destination options to data packets for mobility transparency to upper layer.

GSNS is a mobility-aware SIP-based network entity logically comprises of a basic SIP stack extended with OMA SIP function, for acting as a "Proxy Agent" to interface with the MA, via its basic MIPv6 stack and OMA MIPv6 function. The basic SIP stack of GSNS provides it with roles of a SIP Registrar, a Proxy Server, and a Redirect Server, and to support the basic SIP features like location management and session management. Extending SIP with OMA MIPv6 function for supporting mobility-aware functionalities is feasible because SIP specification mandates SIP Registrar with extensibility and flexibility in interfacing with Location Server using non-SIP based protocols, while retaining core functionality of SIP unmodified without incurring extra processing overhead. GSNS is designed with two abstract data structures namely User Cache that records entries of each MU's User Identity (SIP URL), corresponding *Care-of Address* (COA), IP addresses of MAs that the resident MH is connected to, and MU's profile. MA Cache that stores entries of all MAs that GSNS interfaces within the *Administrative Domain* (AD), and corresponding visiting MHs in each MA.

MA functions similarly to a temporary MIPv6 HA, leveraging on the basic MIPv6 stack to update the current location of MH in its Binding Cache via conventional MIPv6 BU and BA messages. MA is also extended with OMA MIPv6 function to interface with the GSNS (on behalf of the MU) for the registration, allocation, and deregistration of MAddr via an exchange of modified BU and BA messages. Thus, MA retains backward compatibility with existing MIPv6 HA, while it is enhanced with capabilities of OMA.

An overview of OMA's operation is as follows. A MU requests an allocation for a set of MA and MAddr by sending a modified REGISTER request with the new **Assign** header to the GSNS. GSNS dynamically selects a MA based on an algorithm of whether MA is closest to the MU, or the MA most probable to serve it at most times, and replies with a modified BU message to the MA. The BU contains a *Network Access Identifier* (NAI) extension [86] acting as a unique identifier to identify the MU to the MA for the allocation of MAddr to the MH. MA is responsible for invoking the generation or acquisition of a MAddr internally from some external sources based on

the recommended approach discussed in 6.1.5. MA then sends a modified BA message containing the assigned MAddr to GSNS. GSNS updates its User Cache and MA Cache, and then returns a modified "200 OK" message with the new **Assigned** header to the MU. MU then configures its MH with the assigned MAddr.

6.1.3 Modification to Mobile IPv6 Options

Figure 6.2[a, b] depict the format and structure of the modified BU and BA messages respectively. Two new flags (1-bit each) **N** and **R** (indicated in **bold**) are included in the BU, while a new flag **A** (1-bit) is assigned to the BA indicating that **Assign Option** is appended to the BU. All flags supercede previously unused fields, thus not affecting the operation of standard MIPv6. Flag **N** indicates that NAI is appended to the BU. Flag **R** specifies to the MA to send a BA with destination address set to the source address of the BU, this resolves the confusion of MIPv6 specification not explicitly stating whether MA should reply to the source address of BU or the address contained in the *Alternate Care-of Address* (ACOA) field. If flag **R** is enabled, the source address of received BU is used as the destination address for BA, instead of the address contained in the ACOA field.

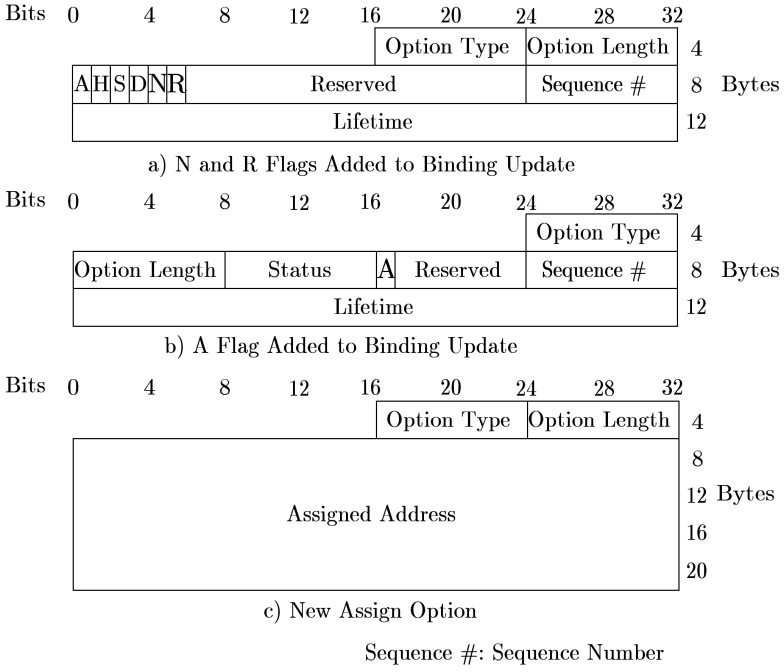


Figure 6.2: Modification to Mobile IPv6 Options

Figure 6.2c depicts the format and structure of a new option appended to the BA

known as **Assign Option**, if flag **A** is enabled in the BA message. **Assign Option** contains three header fields namely **Option Type** (8-bits) stating the type of this option, Option **Length** (8-bits unsigned integer) indicating the length of the option, in octets, excluding the **Option Type** and **Option Length** fields, and lastly **Assigned Address** (128-bits) containing the IPv6 address allocated to the MH.

6.1.4 New SIP Headers

Two newly proposed SIP extension headers **Assign** and **Assigned** are shown in Figure 6.3 and Figure 6.4 respectively. They facilitate proper authentication of the MU's identity, and provides to the MU the access rights to request for a set of MA and MAddr. **Assign** header is a request header defined only for REGISTER message, thus allowing a MU to request the allocation for a set of MA and MAddr in either Home or Visited AD for configuring the MH. **Assigned** header is a response header defined only for SIP responses. It allows the GSNS to response appropriately to the MU's request allocation for a set of MA and MAddr in either the Home or Visited AD for configuring the MH. Table 6.1 and Table 6.2 describe the fields of **Assign** and **Assigned** headers respectively.

```
Assign              = "Assign" ":" 1#assign-values
assign -values      =  callid *( ";" assign-param )
assign-param        = MA-Visited|MA-Home| MAddr-Home|MAddr-
Visited|MA-Prev|MA-Current
MA-Visited          = "MA-Visited=" Boolean
MA-Home             = "MA-Home=" Boolean
MAddr-Home          = "MAddr-Home=" Any
MAddr-Visited       = "MAddr-Visited="  Any
MA-Prev             = "MAddr-Prev=" IPv6 Address
MA-Current          = "MAddr-Current=" IPv6 Address
Boolean             = "Y"|"N"
Any                 = "Y"|"N"|"A"
```

Figure 6.3: Format of Assign Request Header

```
Assigned            ="Assigned" ":" 1#assigned-values
assigned –values    = callid *( ";" assigned-param )
assigned-param      = MA-Visited|MA-Home|MAddr-Home|MAddr-
Visited|MA-Prev
MA-Visited          = "MA-Visited="Response
MA-Home             = "MA-Home=" Response
MAddr-Home          = "MAddr-Home=" Response
MAddr-Visited       = "MAddr-Visited=" Response
Response            = IPv6 address|NA
ttl                 = delta-seconds
```

Figure 6.4: Format of Assigned Response Header

Header Fields	Definitions
MA-Visited	When set to "Y", MH requests a MA assignment in the visited AD. Takes priority over MA-Home.
MA-Home	When set to "Y", MH requests a MA assignment in the home AD. Ignored when MA-Visited is set to "Y". When set to "A", GSNS may assign whichever MA it is convenient with.
MAddr-Home	When set to "Y", MH requests a MAddr assignment in the home AD. Ignored when MAddr-Visited is set to "Y". When set to "A", GSNS may assign whichever MA it is convenient with.
MAddr-Visited	When set to "Y", MH requests a MAddr assignment in the visited AD. Takes priority over MAddr-Home.
MAddr-Prev	If present, contains the IPv6 Address of previous MA.
MAddr-Current	If present, contains the IPv6 Address of current MA.

Table 6.1: Description of Assign Request Header Fields

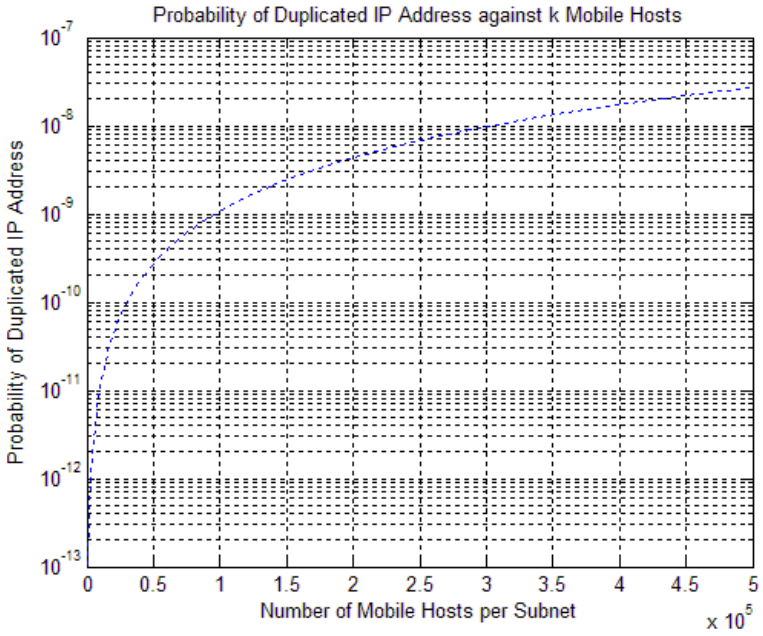
Header Fields	Definitions
MA-Visited	Contains the assigned MA's IPv6 Address.
MA-Home	Contains the assigned MA's IPv6 Address.
MAddr-Home	Contains the IPv6 Address assigned to MU for configuring MU.
MAddr-Visited	When set to "Y", MH requests for a MAddr assignment in the visited AD. Takes priority over MAddr-Home.
ttl	Specifies the duration that the MAddr can be used.

Table 6.2: Description of Assigned Response Header Fields

6.1.5 Generation and Allocation of Mobility Address

MA generates and allocates a unique MAddr to MH efficiently with little processing overhead. Two methods are proposed for discussion namely "managed allocation" and "split/append". The first method requires the MA to function as a DHCPv6 relay server to interface with the DHCPv6 Server, and to request on behalf of the MU for an IPv6 address to configure the MH. However, this would incur extra processing of traffic load to generate DHCPv6 related messages, extra latency to obtain an IPv6 address from the DHCPv6 server, and a more complicated MA as it would possess additional DHCPv6 capabilities. The second method requires a simple extraction of MH's Interface ID and appends it with subnet prefix of the MA. This method of IPv6 address generation is executed locally on the MA without incurring any traffic load, and constitutes a set of simple operations without demanding extra processing capabilities and software module. A possible argument is the probability of an Interface ID or link-local IPv6 address duplication on the same subnet, which can be considered statistically

justifiable to be rare and negligible as shown in Graph 6.1. As an illustration, the probability of duplicated IPv6 address against half a million MHs in a subnet is less than 10^{-7}. Derivation of the proof is found at Appendix B.



Graph 6.1: Probability of Duplicated IP Address against *k* Mobile Hosts

6.2 Operations of the Proposed Architecture

6.2.1 Overview of Proposed Architecture

The proposed architecture is depicted in Figure 6.5 which comprises access technologies like IEEE 802.11b *Wireless Local Area Networks* (WLANs) to provide wireless connectivity to the roaming terminals i.e. *Mobile Host* (MH) via *Base Station* (BS). BSs are collectively grouped into a subnet and interconnected to a multi-port *Access Router* (AR). Multiple subnets topologically belonging to an arbitrary organizational structure or entity (e.g. tertiary campus, or ISP network) forms an *Administrative Domain* (AD), which is connected to the Internet via one or more interconnection routers at the border known as *Border Router*.

In each AD, there exists a *Mobility Agent* (MA) functioning as a temporary *Home Agent* (HA) for MH and to dynamically allocate a temporary *Home Address* (HAddr) known as *Mobility Address* (MAddr) to MH. MH communicates with *Correspondent Host* (CH), which resides a SIP *User Agent* (UA) known as *Correspondent User* (CU).

Another UA resides on the MH to identify its *Mobile User* (MU) with a SIP URL to support terminal and personal mobility services in a peer-to-peer communication session.

Gateway SIP Network Server (GSNS) functions as a "Proxy Agent" on behalf of the MU to interface with the assigned MA for the registration, allocation, and deregistration of MAddr. Typically home GSNS is associated with the AD corresponding to MU's SIP URL, visited GSNS is other than the home GSNS. In addition, current GSNS denotes the GSNS currently serving the MU, while previous GSNS refers to the GSNS previously served the MU.

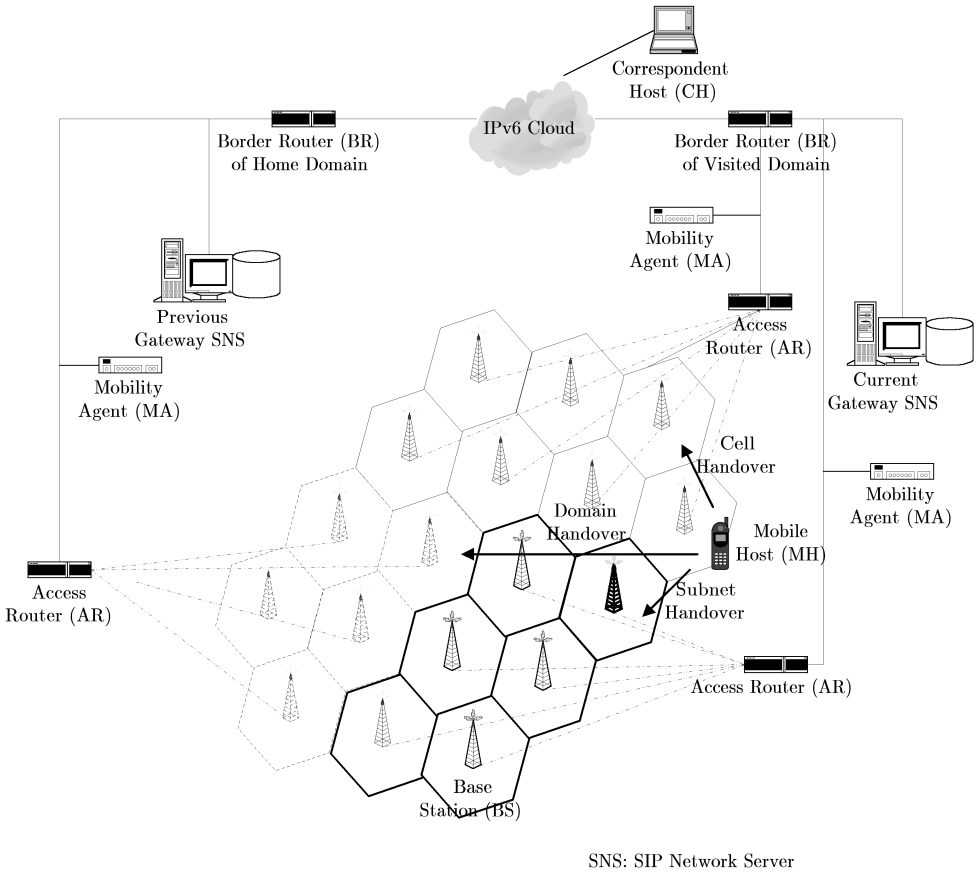


Figure 6.5: Realization of Proposed Architecture

Figure 6.6 illustrates the operations of the architecture, which includes allocation of MA and MAddr to MH when it powers up, MH performing intra-domain and inter-domain mobility, and finally session establishment between CU and MU for peer-to-peer and client-server scenarios.

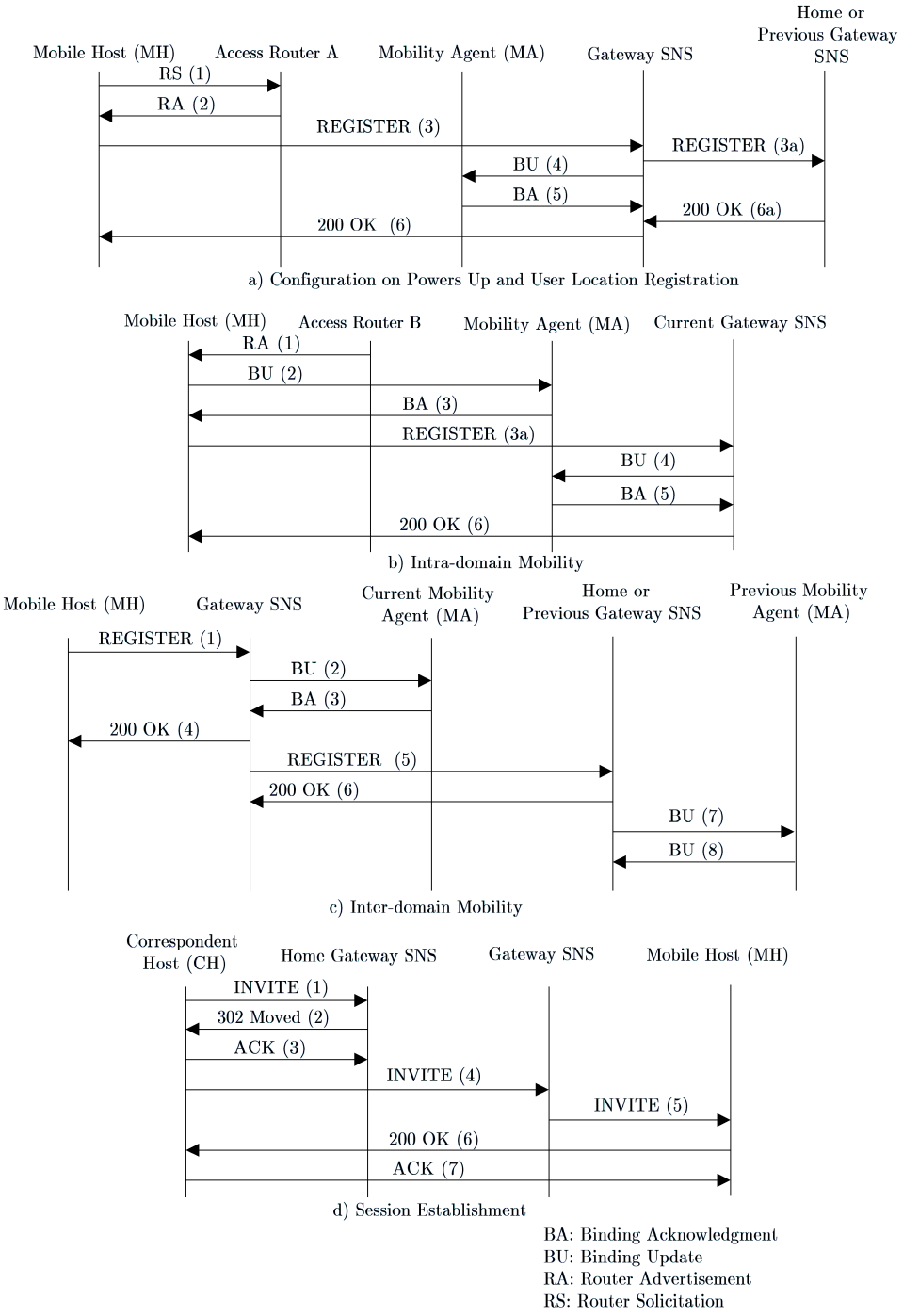


Figure 6.6: Main Operations of Proposed Architecture

6.2.2 Allocation of Mobility Agent and Mobility Address

A MH possesses a pair of globally routable IPv6 addresses namely MAddr and *Physical Care-of Address* (PCOA). MAddr is an address allocated to the MH via *On-*

demand Mobility Agent and Mobility Address Assignment (OMA), and PCOA is generated using stateless address autoconfiguration whenever subnet prefix changes with the MH's movements.

Figure 6.6a illustrates the MU first resides on a MH which attaches to a network or roams within a visited AD. The MH listens for *Router Advertisement* (RA) broadcasted periodically by AR or it would request for one by sending a *Router Solicitation* (RS) (Step 1), and it obtains the following information contained in a RA (Step 2): network identifier of AD, GSNS global IPv6 address, subnet prefix of subnetwork resided. Suppose the MH initially resides in Subnet A, it constructs its PCOA **COA_A**, and then invokes its MU to register with GSNS by sending a REGISTER message (Step 3) as shown in Figure 6.7.

```
REGISTER SIP:alpha@cwc.edu.sg SIP/2.0
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: alpha <SIP:alpha@cwc.edu.sg>
Call-ID: 282475249@cwc.edu.sg
CSeq: 1 REGISTER
Contact: COA_A
Assign: 282475249@cwc.edu.sg; MA-Visited=Y; MAddr-Visited=Y
Expires: 3600
Content-Length: 0
```

Figure 6.7: Allocation of Mobility Address using REGISTER message

When the GSNS receives a REGISTER request it authenticates the MU **alpha**, it then updates the *Location Server* (LS) with an updated location binding of MU's SIP URL **alpha@cwc.edu.sg** and MH's PCOA **COA_A**. GSNS checks through the REGISTER message, **Assign** header is found to contain the following settings **MA-Visited=Y**, and **MAddr-Visited=Y**. This implies that the MU requests allocation for a set of MA and MAddr. GSNS dynamically allocates a MA and then sends a *Binding Update Destination Option* (BU) message (Step 4) to register the MU with this MA for its service and for the allocation of a MAddr. This BU message specifies the GSNS's IP address in the source address field and consists four parts as shown in Figure 6.8. Flag **N** contained in the BU is enabled and the MAddr field is set to "zero" to request the MA to allocate a MAddr to the MU whose user name is contained in the Mobile Node NAI Extension. Flag **R** embedded in the BU is enabled to request the MA to reply with a *Binding Acknowledgement Destination Option* (BA) message to the GSNS instead of the IPv6 address contained in the *Alternate Care-of Address* (ACOA) Option field.

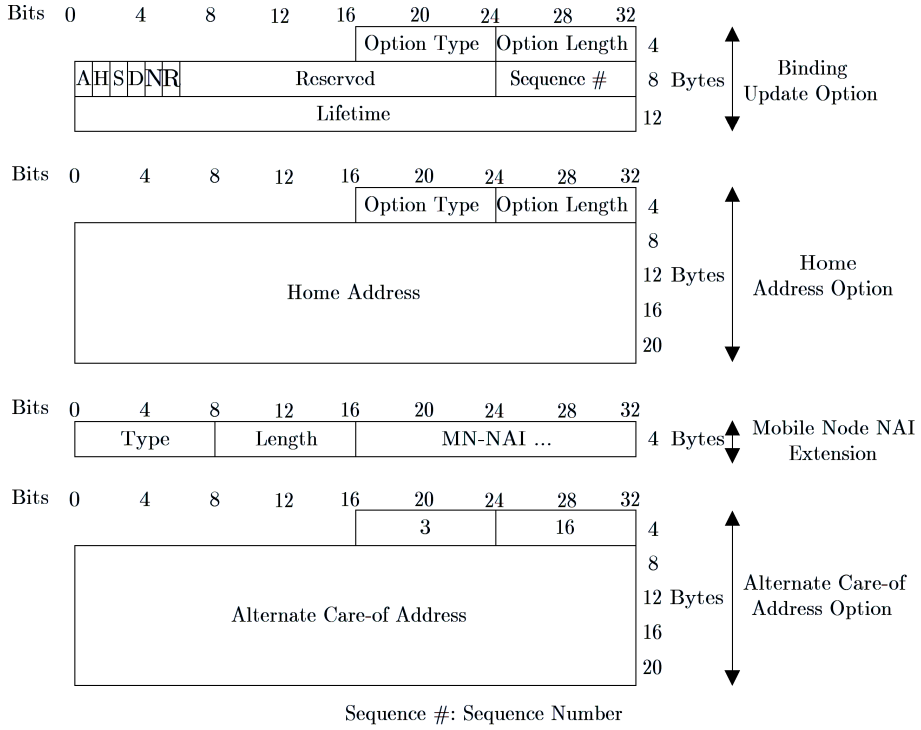


Figure 6.8: Binding Update Send by GSNS

The allocated MA checks through the BU message. Since flag **N** is enabled, the MA then allocates a MAddr, it creates a binding between the assigned MAddr to that IPv6 address **COA_A** contained in the ACOA Option field, and then it responds with a BA message to GSNS. The BA message shown in Figure 6.9 contains the MAddr (Step 5), indicating a successful registration. Flag **A** in the BA message indicates that the assigned IPv6 address MAddr is contained in the Assign Address Option field. Thereafter, MA intercepts all packets on behalf of the MH that it is serving and tunnels them directly to the MH's PCOA i.e. **COA_A**, if those packets has the destination address set to the MH's MAddr.

GSNS receives the BA message from the MA and then replies with a "200 OK" final response (Step 6) to the MU, indicating that the registration and allocation of both MA and MAddr are successful. In addition, GSNS updates its User Cache with the mapping of SIP's URL to the MAddr. Upon receiving the "200 OK" response, MU checks the **Assigned** response header, which contains the MA's IPv6 address as found in **MA-Visited=3FFE:2F22::0001**, and the IPv6 address to be assigned to itself as found in **MAddr Visited=3FFE:2F12::0010:C883-:4866**. The format of the "200 OK" request message is shown in Figure 6.10.

GSNS also notifies the MU's home GSNS (if any) or previous GSNS (if any) that the MU is residing in its AD and to forward all SIP messages to this AD. GSNS constructs a REGISTER request (Step 3a) specifying the user's SIP URL and current MAddr, and then sends it to the home GSNS (based on the domain portion of the MU's SIP URL) or to the previous GSNS. Both home GSNS and previous GSNS receive the REGISTER request message and authenticate the MU.

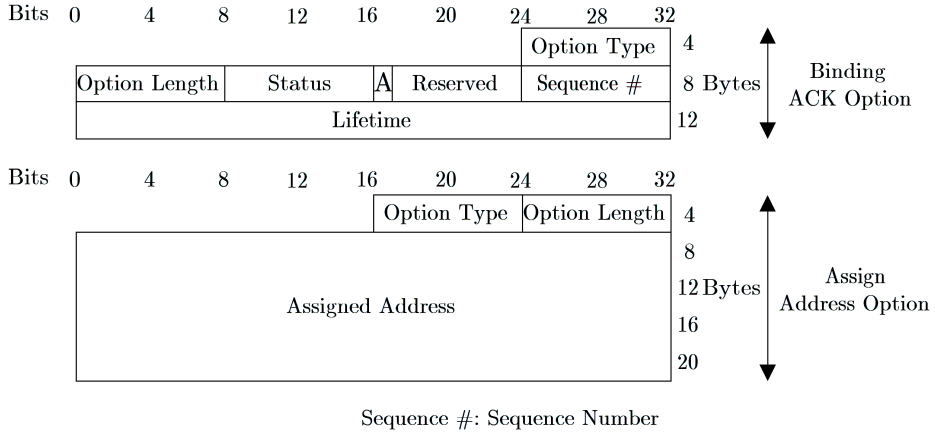


Figure 6.9: Binding Acknowledgment Send to GSNS

```
SIP/2.0 "200 OK"
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: alpha <SIP:alpha@cwc.edu.sg>
Call-ID: 282475249@cwc.edu.sg
CSeq: 1 REGISTER
Assigned: 282475249@cwc.edu.sg;MA-Visited=3FFE:2F22::0001; MAddr
Visited=3FFE:2F12::0010:C883:4866
Contact: alpha <SIP:alpha@cwc.edu.sg>
Content-Length: 0
```

Figure 6.10: Allocation of Mobility Address using a "200 OK" message

MU can also request its home GSNS to allocate a set of MAddr and MA, in a similar manner as mentioned earlier. Both home GSNS and Previous GSNS construct a "200 OK" message (Step 6a) specified with the MAddr, and then forward the "200 OK" message to the MU via the visited GSNS. Once the MU receives a successful "200 OK" message from its visited AD, it stores the IPv6 address MAddr, configures its MH with the MAddr, and then sends a regular MIPv6 BU message to the newly allocated MA.

6.2.3 Intra-domain Mobility

Figure 6.6b illustrates the MU residing on the same MH, moving from subnet A to subnet B within the same AD. MH detects that there is a change in the point-of-attachment based on the subnet prefix contained in received RA (Step 1), and formulates a new PCOA **COA_B**.

There are two approaches allowing the MH to update its new PCOA **COA_B** to both GSNS and MA. The first method is that the MH sends a BU message (Step 2) to its MA containing the MAddr in the MAddr field and its new PCOA. Then MA replies with a BA message (Step 3) to the MH. The second method is that the MU issues a REGISTER request (Step 3a) containing the following settings **MA-Visited=N** and **MAddr-Visited=N** to notify the current GSNS of its new PCOA, without OMA. The first option will result in the GSNS with an incorrect PCOA, and subsequent SIP messages may be routed incorrectly based upon the previous PCOA. Thus, using the second method, the MU explicitly sends a REGISTER request containing its PCOA to the serving GSNS. The format of REGISTER request is shown in Figure 6.11.

```
REGISTER SIP:alpha@cwc.edu.sg SIP/2.0
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: alpha <SIP:alpha@cwc.edu.sg>
Call-ID: 282475249@cwc.edu.sg
CSeq: 1 REGISTER
Contact: COA_B
Assign: 282475249@cwc.edu.sg; MA-Visited=N; MAddr-Visited=N;
MAddr-Current=3FFE:2F22::0001
Expires: 3600
Content-Length: 0
```

Figure 6.11: Intra-domain Mobility REGISTER request

Upon reception of a REGISTER message from the MU, the current GSNS updates its LS and sends a BU message (Step 4) structurally similar to Figure 6.8 to MU's local MA, but without the Mobile Node NAI Extension. Flag **N** in the BU message is disabled so that the MA will not allocate a MAddr to the MU, instead the MA binds the IPv6 address found in the *Home Address Destination Option* (HAD) to that IPv6 address contained in the ACOA Option field. Flag **R** in the BU message is enabled to request the MA to reply with a BA message to GSNS instead of the IPv6 address contained in the ACOA Option field. MA replies with a BA message (Step 5) to the GSNS. GSNS then sends a "200 OK" response (Step 6) to the MU notifying of the receipt of the REGISTER message.

6.2.4 Inter-domain Mobility

Figure 6.6c illustrates the MU residing on the same MH entering a new AD. The sequence of operations is similar to those when the MH first powers up in an AD. MH generates its PCOA through stateless auto-configuration and triggers its MU to explicitly register its PCOA by sending a REGISTER request (Step 1) containing SIP URL, PCOA, home GSNS, previous GSNS, and request for a MAddr, to the serving GSNS.

Upon reception of a REGISTER request from a MU, the GSNS acts as the serving GSNS of the MU. Serving GSNS updates its LS containing the binding between MU's SIP URL, MAddr, and PCOA. It also assigns a MA on behalf of the MU and requests for a MAddr by sending a BU message (Step 2) to the assigned MA. The BU message has the PCOA contained in the ACOA sub-option field, and Mobile Node NAI Extension for IPv6 with the HAddr field set to zero. The allocated MA selects a MAddr for the MU, and registers the mapping of MH's MAddr to its PCOA. MA acknowledges to the serving GSNS with a BA message containing the MAddr. GSNS then sends a "200 OK" message (Step 4) containing the MAddr to the MU as to notify it of the receipt of REGISTER message.

GSNS also constructs a REGISTER request (Step 5) and forwards it to the previous GSNS and home GSNS on behalf of the MU, this effectively redirects SIP signaling messages and data packets addressed to the MU to this new AD. Upon reception of a REGISTER message from the GSNS, previous or home GSNS replies with a "200 OK" response (Step 6), as to notify the receipt of REGISTER message, and proceeds to refresh its LS.

6.2.5 Session Establishment

OMA provides a mechanism for the dynamic allocation of MAddr and MA to MH per communication session. If the MH's movement is spatially localized to its temporary "home" network, then MH would not incur the high signaling load for location management incurred by both MIPv6 and MSIP, the high overhead for data transmission experienced by MIPv6, the high session establishment suffered by MIPv6, and lastly the high handover delays incurred by both protocols. As OMA is independent of the communication session, it can support terminal and personal mobility for both peer-to-peer and client/server communication seamlessly in the wireless Internet. The explanation is as follows.

Figure 6.6d depicts a CU initiates and establishes a peer-to-peer communication session with the MU. MU can be reachable by an application-based terminal identifier, which is its SIP URL or host name, and via an exchange of INVITE and ACK messages occurring between the MU and CU would result in both participating hosts being aware of each other's point-of-attachment to the Internet. Thus, CU first sends an INVITE message (Step 1) containing proper media description in the SDP message body, which is routed to the home GSNS of MU. After checking the SIP URL, a "302 Moved Temporarily" response (Step 2) containing the IPv6 address of MU's current GSNS is replied to CU. CU then responses with an ACK message (Step 3). Alternatively, the INVITE request can be proxied to the MU's current GSNS. CU proceeds to send another INVITE request (Step 4) addressed directly to the MU's current GSNS, which then forwards (Step 5) the INVITE request to the MU. As GSNS is mobility-aware and is able to query its User Cache for the MU's PCOA, GSNS sends the INVITE request directly to the MU instead of via MA. If the MU consents to the session, it replies with a set of provisional responses, and finally with a final response of "200 OK" (Step 6) to CU. The CU acknowledges with an ACK request (Step 7). This exchange of SIP messages effectively establishes a session allowing both hosts to be informed of the other capabilities. As a result, both hosts create a new entry that binds all COAs of the sending host to the SIP URL of the other party. After session establishment, actual media flows as RTP/UDP stream between two end-points using standard routing mechanism. Subsequently, either party can issue BYE request to terminate the session.

For client/server communication session like HTTP and FTP, as stated in [87] "rarely does the situation arise whereby a MU is not the party initiating a session. A MH simply provides the MU with a platform to carry out the exchange of commands... operates on a simple request-response basis". Thus, directory services like DNS and secure dynamic update are commonly adopted to first resolve the server's host name to its corresponding IPv6 address, which is a 128-bits address and not easily remembered. Thereafter, MU issues either a HTTP or FTP request to the server, with the source address set to its MAddr. The server sends either HTTP or FTP response directly to MU via the MAddr. Further exchange of messages between the server and MU is based on standard routing mechanism.

It should be noted that for both client/server and peer-to-peer communication ses-

sion, as long as the MH remains in its "home" during a session period, standard routing mechanism is adopted without the need for appending Routing Header and Home Address Destination Option to data packets. Session period is defined as the time duration such that all connections based on UDP or TCP are active during that period, and would also commence and terminate within that same time duration. If the MH relocates to another network other than its "home" with an ongoing session, then standard MIPv6 will commence such that signaling messages is exchanged to update its MA's and CH's Binding Cache, and destination options are appended to data packets to support mobility transparency to upper-layer. However, OMA provides the flexibility to the MU to request for a new MA and MAddr upon relocating to the new network, which can be used for establishing new sessions with other hosts.

6.3 Qualitative Analysis of OMA

This section presents a discussion of *On-demand Mobility Agent and Mobility Address Assignment* (OMA). It covers OMA deployability issues, provision for service mobility, its comparison with *Mobile IPv6* (MIPv6) and *SIP Mobility Support* (MSIP) with respect to support for *Personal Mobility* and *Terminal Mobility* (TM), and network performance. Network performance includes signaling load, data overhead, session establishment latency, and handover latency. This analysis further illustrates that OMA is a feasible and an extensible mechanism to improve the performance of MIPv6 using the strength of MSIP, when the mobile terminal is relatively stationary in the foreign network.

6.3.1 Deployability of OMA

OMA provides mobility-aware capabilities as minimal modifications to MIPv6 and MSIP, without requiring additional functional network elements. OMA supports OMA SIP function and OMA MIPv6 function, the deployability of each is further discussed.

The former consists of a set of two newly designed SIP headers **Assign** and **Assigned**, and an extra state intelligence on *Mobile User* (MU) and *Gateway SIP Network Server* (GSNS), which are enhanced *User Agent* (UA) and *SIP Network Server* (SNS) respectively. MU communicates with GSNS via **Assign** and **Assigned** headers embedded in standard SIP messages, to support registration, allocation, and deregistration of *Mobility Agent* (MA) and *Mobility Address* (MAddr). As SIP is an application-based protocol, its extension header facilities can be readily exploited while mobility-aware

functionalities to support OMA SIP function can be easily implemented and deployed to the MU and GSNS available in the Internet.

The latter modifies the unused flags of *Binding Update Destination Option* (BU) and *Binding Acknowledgment Destination Option* (BA) for interfacing GSNS with *Mobility Agent* (MA). MA retains backward compatibility with standard MIPv6 *Home Agent* (HA), but is extended with capabilities to generate and allocate MAddr efficiently to the MU for configuring its *Mobile Host* (MH). Existing HAs in the Internet can be upgraded readily with OMA MIPv6 function, by downloading and installing firmware updates from manufacturers of routers. For MH, which is a mobile terminal supporting standard MIPv6 stack, it would be unmodified by OMA. The protocols used for communication between *Corresponding User* (CU) and MU, or *Correspondent Host* (CH) and MH would not be affected by OMA. Thus, CU and CH would also remain untouched.

6.3.2 OMA and Service Mobility

OMA is an independent mechanism that leverages on the strength of SIP as an extensible and flexible application-layer solution to improve MIPv6's network performance when the MH is relatively stationary in the foreign network. The proposed architecture only introduces two newly designed extension headers and extra state intelligence, without modifying the standard specification of SIP. Using SIP inherent support for service mobility, OMA can allow MU to have access to all of its subscribed network services and features (e.g. forwarding services) regardless of its MH resided or point-of-attachment to the Internet. Those subscribed network services and features are collectively termed as session state.

For illustration on how OMA can provide service mobility, it is assumed that the session state of MU is stored and maintained in GSNS that corresponds to the "home" network, which MU resides. This is readily accomplished as GSNS can function as a "stateful proxy". Regardless of whether MU has roamed to a new network, or relocated to a new MH, MU would first issue a REGISTER request to its serving GSNS. The REGISTER request contains its SIP URL, current IPv6 address, home GSNS, previous GSNS, and request for a MAddr. The serving GSNS would first check whether it is the home GSNS of the MU, if not, it constructs a REGISTER request and forwards it to the previous GSNS (if any) and home GSNS on behalf of the MU. Upon reception of a REGISTER message from the serving GSNS, home GSNS verifies

the MU's identity and profile, as to grant the session state to the MU. The home GSNS then replies with a "200 OK" response to the serving GSNS, as to notify the receipt of REGISTER message. Serving GSNS then forwards the "200 OK" to the MU. A successful registration would ensure that the MU maintains all of its active and ongoing sessions or acquires the same services that it has subscribed to. Otherwise, the serving GSNS would drop ongoing sessions by not allocating a pair of MA and MAddr to the MU, without which the MU cannot continue its communication with other CUs.

6.3.3 Support for Personal Mobility and Terminal Mobility

MIPv6 specifies a basic inter-subnet handover mechanism at the network layer for each MH. Whenever the MH is away from its home network, it needs to be pre-assigned with a permanent *Home Address* (HAddr) associated with a permanent HA residing in the corresponding home network, and a temporary *Care-of Address* (COA). Currently, MIPv6 [47,48] only mandates validation of MH's identity without guarding against malicious MU which may reside on authorized MH. As MIPv6 can only authenticate the MH and not the MU, it has no support for PM. PM provides MU with the ability to relocate to different MHs or to roam to networks, to initiate and establish sessions via the same unique personal identifier.

MSIP extends SIP as a basic mobility framework at the application layer by assuming that there is no existing MIPv6 core infrastructure. MSIP supports PM whereby each MU is uniquely assigned with a User Identifier (i.e. SIP URL), and the resided MH possesses a temporary unicast IP address identifying its current location. Whenever, the MU is away from its home network or the MU resides on another MH, MU issues REGISTER requests to its SIP Registrar to update on its current location. Thus, MSIP supports PM by using SIP URL and registration mechanism. However, MSIP supports TM for UDP-based communication only, as MSIP resides over transport layer. Whenever a MH roams to another subnet during an active session and acquires a new IPv6 address, to maintain the ongoing communications between the MU and its CU, the signaling and data traffic flow between them must be transferred with minimal disruption in association to the MU's new location. MU sends a new INVITE message to its CU with its newly obtained IPv6 address updated in the Contact field, to inform the CU where it wants to receive subsequent SIP messages. The INVITE message is also embedded with a *Session Description Protocol* (SDP) message body, in which a c(onnection)-field contains the new location of MU.

In contrast, OMA facilitates the MU residing on a MH, to request for a MA and MAddr via **Assign** and **Assigned** headers from the GSNS. This dynamic assignment of MA and MAddr provides more security and flexibility to the service provider, as GSNS first validates the identity of MU before it allocates a MA and MAddr corresponding to the same subnet that the authorized MH is residing in. Thus, OMA has provision for PM. As OMA extends standard MIPv6 to lease a virtual network as its "home" along with the MA and MAddr, OMA retains the TM capability of MIPv6.

6.3.4 Network Performance

As MIPv6 makes no distinguishment between a MH frequently moving in the foreign network, or one that is relatively stationary with minimal change in its COA per unit time, it suffers from the following drawbacks: significant MIPv6 signaling in exchanging BU/BA periodically to refresh binding at CHs and HA, significant data overhead as destination options headers including Routing Header and Home Address Destination Option are appended to every data packets exchanged with CHs for mobility transparency to upper layer, and lastly high session establishment delay as MH may be located far from HA which results in triangular routing and tunneling.

For MSIP, CU transmits all subsequent IP data traffic to the MU's new IP address. Data overhead and session establishment latency incurred are practically negligible as MSIP adopts direct end-to-end communication which is based on a single optimised routing path between MU and CU without tunneling, packet interceptor, or data packets appended with destination options. However, this mechanism of location management incurs substantial signaling overhead than MIPv6 as SIP message is textual and context-sensitive. In addition, MSIP has no provision for TCP-based communication which is generally used for HTTP and FTP. Thus, usage of MSIP is greatly limited to only UDP-based communication sessions.

However, for OMA, while the MU is relatively stationary in the network with minimal change in its COA per unit time, MU is dynamically assigned with a MA, MAddr and the corresponding virtual network as its "home" with which it is residing in. MU configures these entities on its MH whenever it requires them solely for session establishment and data exchange. Given that the MH now resides in its "home" with the leased MAddr and MA, OMA avoids the need for the following operations: Exchanging BU/BA signaling messages with CHs and MA periodically to refresh binding, or to update MH's current location. Appending every data packets with destination

options for supporting mobility transparency to upper layer. Triangular routing and tunneling due to close proximity of MA and MH.

In addition, MIPv6 experiences two *Duplicate Address Detection* (DAD) processes namely MH-initiated and HA-initiated DAD for ensuring uniqueness of MH's IPv6 address as required by IPv6 specification. The former occurs when MH generates a new COA via stateless address configuration, while the latter is triggered when MH sends BUs to previous Access Router and HA to ensure there exists no address duplications. Thus, MIPv6 suffers considerable handover delay incurred by DAD. For MSIP, it still suffers MH-initiated DAD, which impedes its ability to immediately use the new IPv6 address for data exchange. In contrast, both types of DAD are bypassed in OMA by having a virtual network with no nodes either than MA and MH. With such a virtual network, when a typical MH issues a request to the MA to perform DAD, the MA can immediately assess the MH's MAddr duplication, bypass DAD, and reply promptly to the MH that DAD is successful and that the MAddr is not duplicated. This reduces potential handover delay incurred by MH.

6.4 Summary

This chapter describes the detailed protocol operation of the architecture i.e. **On-demand Mobility Agent and Mobility Address Assignment** that supports both terminal and personal mobility for both peer-to-peer and client/server communication seamlessly in the wireless Internet. The architecture **On-demand Mobility Agent and Mobility Address Assignment** illustrates that it is a feasible and extensible mechanism to improve the performance of Mobile IPv6 such that it can now support PM, and also minimize the inefficiencies of Mobile IPv6 and SIP Mobility Support, when the mobile terminal is relatively stationary in the foreign network. These inefficiencies include the high signaling load for location management incurred by both protocols, the high overhead for data transmission experienced by Mobile IPv6, the high session establishment incurred by Mobile IPv6, and lastly the high handover delays experienced by both protocols, when the mobile terminal resides away from its home network and is relatively stationary in the foreign network.

Chapter 7.
Conclusion and Future Works

7.1 Conclusion

This thesis has investigated two key existing mobility support schemes namely *Mobile IPv6* (MIPv6) and *Session Initiation Protocol* (SIP) support for mobility (MSIP). Consequently, the investigation has proposed a novel architecture i.e. **On-demand Mobility Agent and Mobility Address Assignment** by harmonizing the interaction and coexistence between both protocols to support terminal and personal mobility for both peer-to-peer and client/server communication seamlessly in the wireless Internet. This architecture minimizes the inefficiencies of MIPv6 and MSIP by adopting two newly designed SIP header extensions **Assign** and **Assigned**, and a set of modified MIPv6 Binding Update Destination Option and Binding Acknowledgment Destination Option signaling messages for allocation of a serving **Mobility Address** and **Mobility Agent** dynamically per communication session. These enhancements improve the performance of Mobile IPv6 using the strength of Mobile SIP, as to provide the following requirements:

(I) Low end-to-end delay for session establishment and data exchange, as prolonged latency would cause initiating party to abandon session.

(II) Low handover delay bypassing Duplicate Address Detection at the virtual "home" network as to ensure Binding Acknowledgment is replied immediately to the mobile terminal and to minimize jitters and delay variations.

(III) Low signaling traffic and overhead of data exchange taking into account the spatial locality of mobile users without incurring degradation of routing performance.

In this investigation, we have chosen and conducted a three-phase analysis and design methodology.

This thesis has provided SIPsim, a minimal design and implementation of SIP as an extension to NS-2. SIPsim is the first treatment of SIP for experimental and research platform to acquire thorough evaluation, insights, and clear understanding of SIP internalities and functionalities. It can also be used for prototyping advanced

value-added services like mobility support and in-depth understanding of integrating SIP with RSVP, without incurring costly test-bed setup and managing complex implementation issues. Layered architectural design and implementation of SIPsim has been evaluated and validated against specification conformance test-suite. The simulation has successfully demonstrated session establishment and termination for various scenarios consisting direct peer-to-peer and involvement of SIP network entities e.g. SIP Proxy Server and SIP Registrar. SIPsim has provided support for software modules: SIP Message Parser, SIP Message Generator, User Agent, and SIP Network Server with available methods REGISTER, INVITE, BYE, and ACK, and responses "180 Ringing", and "200 OK".

An extensive and comprehensive literature survey has covered the definition and components constituting different types of mobility, related studies on current solutions and issues of supporting mobility in the Internet from perspectives of network, transport, and application layer. Background work of MIPv6 and MSIP has shown that terminal and personal mobility are supported separately and independently by MIPv6 (at the network layer) and MSIP (at the application layer) respectively. Both protocols possess limited support for both types of mobility based on real-time and TCP-based communication, and both protocols incur performance inefficiency like high signaling load and significant handover latency. No prior research work has been reported in the literature to resolve the open issue of which protocol or combination of protocols would be the choice for deployment in supporting terminal and personal mobility in the wireless based Internet.

The final phase of the investigation has conducted an extensive qualitative and quantitative analysis/comparison of MIPv6 and MSIP to reveal suitability for terminal and personal mobility respectively. This phase has facilitated derivation of situations and conditions upon which either protocol would be appropriate for, as to design and develop an architecture that leverages on both their strengths. Qualitative analysis has evaluated their internalities and functionalities summarizing both are similar in terms of registration operations, two-tier addressing scheme, address translation mechanism, entities, and data structures. Quantitative study has concluded the following observations. (I) MIPv6 is more efficient than MSIP in terms of lower *Signaling Load* incurred by location management, since $\Delta_{SL} = SL_S - SL_M > 0$ occurs regardless of whether the mobile terminal resides in the home network, foreign networks or when roaming. (II)

MSIP incurs lower overhead for data transmission than MIPv6, since $\Delta_{DL} = DL_M - DL_S > 0$ for all scenarios. MIPv6 assumes mobile terminal is assigned with a predefined home network and it periodically refreshes the binding at its HA and CHs regardless of whether the mobile terminal is stationary or frequently moving in the foreign networks. (III) MIPv6 is more efficient than MSIP in terms of lower *Registration Time* associated with location management in all scenarios. (IV) MSIP, in general, incurs lower *Session Establishment Time* than MIPv6 whenever mobile terminal is away from its home network, unless the condition of $N_{REGISTRAR\_HA} + N_{HA\_MH} > N_{REGISTRAR\_MH}$ is violated.

7.2 Future Works

SIPsim presented is only an initial design and development of SIP specification; future work would include the following areas. (I) Simulation of SIP interworks with TCP as to understand and evaluate the effects of TCP's windows size and timers on SIP's operation in a peer-to-peer and mobile environment. (II) Simulation of SIP mobility support in comparison with MIPv6 to understand and evaluate issues of scalability as the number of mobile terminals increases, and high speed travel of mobile terminal.

In addition, the proposed architecture with **On-demand Mobility Agent and Mobility Address Assignment** for allocation of a serving **Mobility Address** and **Mobility Agent** is a preliminary concept. It requires further analytical studies based on simulation and prototyping to obtain numerical results for performance comparison to existing mobility support solutions.

Another critical research topic is the analysis and the inclusion of security measures into the proposed architecture. Security is a key issue of personal mobility, which requires cautious handling. Allowing a user to move and access system services anywhere at any time heightens the threats of fraudulent use of a user's identity and resources. It is imperative for third parties (e.g. owners of terminals) to possess capability to protect their privacy and freedom of actions despite mobile user being registered at their terminal(s).

Appendix A. Results of Simulation

Test R1

-------- UA to *Registrar* --------
Node 0: alpha send SIP Message of size 267
REGISTER SIP:alpha@cwc.edu.sg SIP/2.0
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: alpha <SIP:alpha@cwc.edu.sg>
Call-ID: 282475249@cwc.edu.sg
CSeq: 1 REGISTER
Contact: alpha <SIP:alpha@cwc.edu.sg>
Expires: 3600
Content-Length: 0

-------- *Registrar* to UA --------
Node 2: singnet.com.sg send SIP Message of size 229
SIP/2.0 "200 OK"
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: alpha <SIP:alpha@cwc.edu.sg>
Call-ID: 282475249@cwc.edu.sg
CSeq: 1 REGISTER
Contact: alpha <SIP:alpha@cwc.edu.sg>
Content-Length: 0

Test R2

-------- UA to *Registrar* --------
Node 0: alpha send SIP Message of size 213
REGISTER SIP:alpha@cwc.edu.sg SIP/2.0
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: alpha <SIP:alpha@cwc.edu.sg>
Call-ID: 984943658@cwc.edu.sg
CSeq: 1 REGISTER
Content-Length: 0

-------- *Registrar* to UA --------
Node 2: singnet.com.sg send SIP Message of size 229
SIP/2.0 "200 OK"
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: alpha <SIP:alpha@cwc.edu.sg>
Call-ID: 984943658@cwc.edu.sg
CSeq: 1 REGISTER
Contact: alpha <SIP:alpha@cwc.edu.sg>
Content-Length: 0

Test R3

-------- UA to *Registrar* --------
Node 0: alpha send SIP Message of size 238
REGISTER SIP:alpha@cwc.edu.sg SIP/2.0
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: alpha <SIP:alpha@cwc.edu.sg>
Call-ID: 984943658@cwc.edu.sg
CSeq: 1 REGISTER
Contact: *
Expires: 0
Content-Length: 0

-------- *Registrar* to UA --------
Node 2: singnet.com.sg send SIP Message of size 190
SIP/2.0 "200 OK"
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: alpha <SIP:alpha@cwc.edu.sg>
Call-ID: 984943658@cwc.edu.sg
CSeq: 1 REGISTER
Content-Length: 0

Test D

-------- UA A to UA B --------
Node 0: alpha send SIP Message of size 370
INVITE SIP:beta@cwc.edu.sg SIP/2.0
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 282475249@cwc.edu.sg
CSeq: 1 INVITE
Contact: alpha <SIP:alpha@cwc.edu.sg>
Content-Type: application/sdp
Content-Length: 93

v=0
o=alpha 984943658 470211272 IN IP4 cwc.edu.sg
s=Session SDP
m=audio 6000 RTP/AVP 0

-------- UA B to UA A--------
Node 1: beta send SIP Message of size 190
SIP/2.0 100 Trying
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 282475249@cwc.edu.sg
CSeq: 1 INVITE
Content-Length: 0

-------- UA B to UA A--------
Node 1: beta send SIP Message of size 191
SIP/2.0 "180 Ringing"
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 282475249@cwc.edu.sg
CSeq: 1 INVITE
Content-Length: 0

-------- UA B to UA A--------
Node 1: beta send SIP Message of size 312
SIP/2.0 "200 OK"
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 282475249@cwc.edu.sg
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 94

v=0
o=beta 1457850878 2007237709 IN IP4 cwc.edu.sg
s=Session SDP

m=audio 6000 RTP/AVP 0

-------- UA A to UA B--------
Node 0: alpha send SIP Message of size 200
ACK SIP:beta@cwc.edu.sg SIP/2.0
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 282475249@cwc.edu.sg
CSeq: 1 ACK
Content-Length: 0

-------- UA A to UA B--------
Node 0: alpha send SIP Message of size 200
BYE SIP:beta@cwc.edu.sg SIP/2.0
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 282475249@cwc.edu.sg
CSeq: 2 BYE
Content-Length: 0

-------- UA B to UA A--------
Node 1: beta send SIP Message of size 183
SIP/2.0 “200 OK”
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 282475249@cwc.edu.sg
CSeq: 2 BYE
Content-Length: 0

Test I2

-------- UA A to *Proxy Server* 1 --------
Node 0: alpha send SIP Message of size 372
INVITE SIP:beta@cwc.edu.sg SIP/2.0
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 470211272@cwc.edu.sg
CSeq: 1 INVITE
Contact: alpha <SIP:alpha@cwc.edu.sg>
Content-Type: application/sdp
Content-Length: 95

v=0
o=alpha 1457850878 2007237709 IN IP4 cwc.edu.sg
s=Session SDP
m=audio 6000 RTP/AVP 0

-------- *Proxy Server* 1 to *Proxy Server* 2--------
Node 2: singnet.com.sg send SIP Message of size 410
INVITE SIP:beta@cwc.edu.sg SIP/2.0
Via: SIP/2.0/UDP singnet.com.sg:5060
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 470211272@cwc.edu.sg
CSeq: 1 INVITE
Contact: alpha <SIP:alpha@cwc.edu.sg>
Content-Type: application/sdp
Content-Length: 95

v=0
o=alpha 1457850878 2007237709 IN IP4 cwc.edu.sg
s=Session SDP
m=audio 6000 RTP/AVP 0

-------- *Proxy Server* 2 to UA B--------
Node 3: pacific.com.sg send SIP Message of size 448
INVITE SIP:beta@cwc.edu.sg SIP/2.0
Via: SIP/2.0/UDP pacific.com.sg:5060
Via: SIP/2.0/UDP singnet.com.sg:5060
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 470211272@cwc.edu.sg
CSeq: 1 INVITE
Contact: alpha <SIP:alpha@cwc.edu.sg>
Content-Type: application/sdp
Content-Length: 95

v=0
o=alpha 1457850878 2007237709 IN IP4 cwc.edu.sg
s=Session SDP
m=audio 6000 RTP/AVP 0

-------- UA B to *Proxy Server* 2 --------
Node 1: beta send SIP Message of size 267
SIP/2.0 “180 Ringing”
Via: SIP/2.0/UDP pacific.com.sg:5060
Via: SIP/2.0/UDP singnet.com.sg:5060
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 470211272@cwc.edu.sg
CSeq: 1 INVITE
Content-Length: 0

-------- *Proxy Server* 2 to *Proxy Server* 1--------
Node 3: pacific.com.sg send SIP Message of size 229
SIP/2.0 “180 Ringing”
Via: SIP/2.0/UDP singnet.com.sg:5060
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 470211272@cwc.edu.sg
CSeq: 1 INVITE
Content-Length: 0

-------- *Proxy Server* 1 to UA A--------
Node 2: singnet.com.sg send SIP Message of size 191
SIP/2.0 “180 Ringing”
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 470211272@cwc.edu.sg
CSeq: 1 INVITE
Content-Length: 0

-------- UA B to *Proxy Server* 2 --------
Node 1: beta send SIP Message of size 386
SIP/2.0 “200 OK”
Via: SIP/2.0/UDP pacific.com.sg:5060
Via: SIP/2.0/UDP singnet.com.sg:5060
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 470211272@cwc.edu.sg
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 92

v=0
o=beta 1115438165 74243042 IN IP4 cwc.edu.sg
s=Session SDP
m=audio 6000 RTP/AVP 0

-------- *Proxy Server* 2 to *Proxy Server* 1--------
Node 3: pacific.com.sg send SIP Message of size 348
SIP/2.0 “200 OK”
Via: SIP/2.0/UDP singnet.com.sg:5060
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 470211272@cwc.edu.sg
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 92

v=0
o=beta 1115438165 74243042 IN IP4 cwc.edu.sg
s=Session SDP
m=audio 6000 RTP/AVP 0

-------- *Proxy Server* 1 to UA A--------
Node 2: singnet.com.sg send SIP Message of size 310
SIP/2.0 “200 OK”
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 470211272@cwc.edu.sg
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 92

v=0
o=beta 1115438165 74243042 IN IP4 cwc.edu.sg
s=Session SDP
m=audio 6000 RTP/AVP 0

-------- UA A to *Proxy Server* 1 --------
Node 0: alpha send SIP Message of size 200
ACK SIP:beta@cwc.edu.sg SIP/2.0
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 470211272@cwc.edu.sg
CSeq: 1 ACK
Content-Length: 0

-------- *Proxy Server* 1 to *Proxy Server* 2--------
Node 2: singnet.com.sg send SIP Message of size 238
ACK SIP:beta@cwc.edu.sg SIP/2.0
Via: SIP/2.0/UDP singnet.com.sg:5060
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 470211272@cwc.edu.sg
CSeq: 1 ACK
Content-Length: 0

-------- *Proxy Server* 2 to UA B--------
Node 3: pacific.com.sg send SIP Message of size 276
ACK SIP:beta@cwc.edu.sg SIP/2.0
Via: SIP/2.0/UDP pacific.com.sg:5060
Via: SIP/2.0/UDP singnet.com.sg:5060
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 470211272@cwc.edu.sg
CSeq: 1 ACK
Content-Length: 0

-------- UA A to *Proxy Server* 1 --------
Node 0: alpha send SIP Message of size 200
BYE SIP:beta@cwc.edu.sg SIP/2.0
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 470211272@cwc.edu.sg
CSeq: 2 BYE
Content-Length: 0

-------- *Proxy Server* 1 to *Proxy Server* 2--------
Node 2: singnet.com.sg send SIP Message of size 238
BYE SIP:beta@cwc.edu.sg SIP/2.0
Via: SIP/2.0/UDP singnet.com.sg:5060
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 470211272@cwc.edu.sg
CSeq: 2 BYE
Content-Length: 0

-------- *Proxy Server* 2 to UA B--------
Node 3: pacific.com.sg send SIP Message of size 276
BYE SIP:beta@cwc.edu.sg SIP/2.0
Via: SIP/2.0/UDP pacific.com.sg:5060
Via: SIP/2.0/UDP singnet.com.sg:5060
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 470211272@cwc.edu.sg
CSeq: 2 BYE
Content-Length: 0

-------- UA B to *Proxy Server* 2 --------
Node 1: beta send SIP Message of size 259
SIP/2.0 “200 OK”
Via: SIP/2.0/UDP pacific.com.sg:5060
Via: SIP/2.0/UDP singnet.com.sg:5060
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 470211272@cwc.edu.sg
CSeq: 2 BYE
Content-Length: 0

-------- *Proxy Server* 2 to *Proxy Server* 1--------
Node 3: pacific.com.sg send SIP Message of size 221
SIP/2.0 “200 OK”
Via: SIP/2.0/UDP singnet.com.sg:5060
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 470211272@cwc.edu.sg
CSeq: 2 BYE
Content-Length: 0

-------- *Proxy Server* 1 to UA A--------
Node 2: singnet.com.sg send SIP Message of size 183
SIP/2.0 “200 OK”
Via: SIP/2.0/UDP cwc.edu.sg:5060
From: alpha <SIP:alpha@cwc.edu.sg>
To: beta <SIP:beta@cwc.edu.sg>
Call-ID: 470211272@cwc.edu.sg
CSeq: 2 BYE
Content-Length: 0

Appendix B. Mathematical Proof

Equation (B.1) expresses *P(n,k)*, the probability that an Interface ID is unique within a subnet of *k Mobile Hosts* (MHs) drawn uniquely from a population of *n*. In this case $n = 2^{62}$, because there exists 62 usable bits in the Interface ID and subnet prefix is always unique. Probability of Duplicated IP address against *k* MHs i.e. 1 – *P(n,k)* is given by (B.2). The upper bound of *P(n,k)* is adopted due to its intensive computation when *n* tends to very large value.

$$\begin{aligned} P(n,k) &= \frac{n!}{(n-k)!\,n^k} \\ &= \frac{n(n-1)(n-2)\ldots(n-k+1)(n-k)!}{(n-k)!\,n^k} \\ &= \begin{cases} 1 & k = 0,1 \\ \dfrac{(n-1)(n-2)\ldots(n-k+1)}{n^{k-1}} & k > 2 \end{cases} \end{aligned} \tag{B.1}$$

$$1 - P(n,k) = \begin{cases} 0 & k = 0,1 \\ \leq \dfrac{k(k-1)}{2n} & k > 2 \end{cases} \tag{B.2}$$

For illustration *P(n,k)* is calculated for $k = 2$, 3, and 4.

When $k = 2$, $P(n,2) = 1 - \frac{1}{n} \Rightarrow 1 - P(n,2) = \frac{1}{n}$

When $k = 3$, $P(n,3) = (1 - \frac{1}{n})(1 - \frac{2}{n}) = 1 - \frac{3}{n} + \frac{2}{n^2}$

$$\Rightarrow 1 - P(n,3) = \frac{3}{n} - \frac{2}{n^2}$$

$$\Rightarrow 1 - P(n,3) < \frac{3}{n}, \ as \ \frac{2}{n^2} > 0$$

When $k = 4$, $P(n,4) = (1 - \frac{1}{n})(1 - \frac{2}{n})(1 - \frac{3}{n}) = 1 - \frac{6}{n} + \frac{5}{n^2} - \frac{6}{n^3}$

$$\Rightarrow 1 - P(n,4) = \frac{6}{n} - \frac{5}{n^2} + \frac{6}{n^3}$$

$$\Rightarrow 1 - P(n,4) < \frac{6}{n}, \ as \ \frac{5}{n^2} - \frac{6}{n^3} > 0$$

This can be generalized to (B.3)

$$1 - P(n,k) \le \frac{k(k-1)}{2n}, \ \forall k \ge 2 \tag{B.3}$$

Equation (B.3) can be proven using Mathematical Induction, which is as follows.

Assumes (B.4) is true.

$$1 - P(n,k) \le \frac{k(k-1)}{2n}, \ \forall k \ge 2 \tag{B.4}$$

For $k = 2$,

$$1 - P(n,2) \le \frac{1}{n}, \text{ which is true}$$

For $k = m + 1$,

$$1 - P(n,m+1) = 1 - \frac{(n-1)(n-2)...(n-m+1)(n-m)}{n^m}$$

$$\Rightarrow 1 - P(n,m+1) = 1 - P(n,m) * (\frac{n-m}{n}) \tag{B.5}$$

$$\Rightarrow 1 - P(n,m+1) = 1 - P(n,m) + \frac{m}{n} P(n,m)$$

$$\text{Also } \frac{(m+1)m}{2n} = \frac{m(m-1)}{2n} + \frac{m}{n} \tag{B.6}$$

$$\text{And } \frac{m}{n} \le \frac{m}{n} P(n,m), \text{ since } 0 \le P(n,m) \le 1 \tag{B.7}$$

Substituting (B.5), (B.6), and (B.7) in (B.4),

$$1 - P(n,m+1) = 1 - P(n,m) + \frac{m}{n} P(n,m)$$

$$\Rightarrow 1 - P(n,m+1) \le \frac{m(m-1)}{2n} + \frac{m}{n} P(n,m)$$

$$\Rightarrow 1 - P(n,m+1) \le \frac{m(m-1)}{2n} + \frac{m}{n}$$

$$\Rightarrow 1 - P(n,m+1) \le \frac{(m+1)m}{2n}$$

$$\therefore \textit{for} \quad k = m+1 \quad \textit{is} \quad \textit{also} \quad \textit{true}$$

Since $k = 2$ is true, $k = 3, 4, 5...$ are also true for $1 - P(n,k) \le \frac{k(k-1)}{2n}, \ \forall k \ge 2$

Bibliography

[1] S. Faccin, L. Hsu, R. Koodli, K. Le, and R. Purnadi, "GPRS and IS-136 Integration for Flexible Network and Services Evolution," *IEEE Personal Communications*, vol. 6 (3), pp. 48-54, June 1999.

[2] A. Furuskär, S. Mazur, F. Müller, and H. Olofsson, "EDGE: Enhanced Data Rates for GSM and TDMA/136 Evolution," *IEEE Personal Communications*, vol. 6 (3), pp. 56-66, June 1999.

[3] J. O. Vatn, "End-to-end and Redirection Delays in IP Based Mobility," Technical Report TRITA-IT R 00:01, ISSN-1103-534X, ISRN KTH/IT/R--00/01--SE, 2000.

[4] International Telecommunication Union Telecommunication Standardization Sector (ITU-T), "One-Way Transmission Time," Recommendation G.114, Transmission Systems and Media, Digital Systems and Networks, General Characteristics of Intl. Telephone Connections and Intl. Telephone Circuits, Feb 1996.

[5] H. Schulzrinne and J. Rosenberg, "Internet Telephony: Architecture and Protocols - an IETF Perspective," in *Proc. Computer Networks and ISDN Systems*, vol. 31 (3), pp. 237-256, 1999.

[6] H. Schulzrinne, "A Comprehensive Multimedia Control Architecture for the Internet," in *Proc. IEEE 7th Intl. Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, St. Louis, Missouri, pp. 65-76, May 1997.

[7] H. Schulzrinne, S. L. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," IETF RFC 1889, Jan 1996.

[8] H. Schulzrinne and J. Rosenberg, "The Session Initiation Protocol: Providing Advanced Telephony Services Across the Internet," *Bell Labs Technical Journal*, vol. 3, pp. 144-160, Nov/Dec 1998.

[9] H. Schulzrinne and J. Rosenberg, "The IETF Internet Telephony Architecture and Protocols," *IEEE Network*, vol. 13 (3), pp. 18-23, May/June 1999.

[10] H. Schulzrinne and J. Rosenberg, "The Session Initiation Protocol: Internet-Centric Signaling," *IEEE Communications Magazine*, vol. 38 (10), pp. 134-141, Oct 2000.

[11] ITU, "Packet-based Multimedia Communications Systems," ITU-T Recommendation H.323, 1998.

[12] G. A. Thom, "H.323: The Multimedia Communications Standard for Local Area Networks," *IEEE Communications Magazine*, vol. 34 (12), pp. 52-56, Dec 1996.

[13] H. Schulzrinne, "Session Initiation Protocol (SIP)," IETF RFC 2543, March 1999. Proposed Standard.

[14] P. Tiilikainen, *SIP (RFC 2543), an Implementation for Marratech Pro*, Master's Thesis, Division of Software Eng., Dept. of Computer Science and Electrical Eng., Lulea Univ. of Technology, May 2000.

[15] H. Schulzrinne and J. Rosenberg, "A Comparison of SIP and H.323 for Internet Telephony," in *Proc. Intl. Workshop on Network and Operating System Support for Digital Audio and Video (NOSS-DAV)*, July 1998.

[16] I. Dalgic and H. Fang, "Comparison of H.323 and SIP for IP Telephony Signaling," in *Proc. Photonics East*, Sept 1999.

[17] Nortel Networks, "A Comparison of H.323v4 and SIP," 3GPP S2, Tokyo Japan, S2-000505.

[18] F. Fingal and P. Gustavsson, *A SIP of IP-telephony*, Master's Thesis, Dept. of Communication Systems, Lund Institute of Technology, Lund Univ. and Sigma Exallon Systems AB, Malmo, Feb 1999.

[19] P. Flykt and T. Alakoski, "SIP Services and Interworking with IPv6," in *Proc. 2nd Intl. Conf. on 3G Mobile Communication Technologies*, vol. 477, pp. 186-190, 2001.

[20] R. H. Glitho, "Advanced Services Architectures for Internet Telephony: A Critical Overview," *IEEE Network*, vol. 14 (4), pp. 38-44, July/Aug 2000.

[21] R. T. Fielding et al., "Hypertext Transfer Protocol-HTTP/1.1," IETF RFC 2616, June 1999.

[22] J. B. Postel, "Simple Mail Transfer Protocol," IETF RFC 821, Aug 1982.

[23] D. H. Crocker and P. Overell, "Augmented BNF for Syntax Specifications: ABNF," IETF RFC 2234, Nov 1997. Proposed Standard.

[24] M. Handley and V. Jacobson, "Session Description Protocol (SDP)," IETF RFC 2327, April 1998. Proposed Standard.

[25] The Network Simulator ns-2, http://www.isi.edu/nsnam/ns/index.html, Accessed on Sept 1999.

[26] K. Fall and K. Varadhan, The ns Manual, http://www.isi.edu/nsnam/ns/doc/-index.html, The VINT Project, Accessed on Jan 2001.

[27] D. Wetherall and C. J. Lindblad, "Extending Tcl for Dynamic Object-Oriented Programming," in *Proc. Tck/Tk Workshop 95*, Toronto Ontario, July 1995.

[28] SIP - Vovida.org, http://vovida.org/protocols/downloads/sip, Accessed on March 2002.

[29] oSIP - Open Source SIP Stack, http://www.gnu.org/software/osip/, Accessed on March 2002.

[30] H. Zou et al., "Prototyping SIP-based VoIP Services in Java," in *Proc. Intl. Conf. on Communication Technology (WCC - ICCT)*, vol. 2, pp. 1395-1399, 2000.

[31] S. Foeckel, M. Kranz, J. Kuthan, and D. Sisalem, "OSIP: An Open Source SIP Architecture," in *Proc. 1st IP-Telephony Workshop*, Berlin, April 2000.

[32] G. Stojsic, R. Radovic, and S. Srbljic, "Formal Definition of SIP Proxy Behavior," in *Proc. Intl. Conf. on Trends in Communications (EUROCON)*, Bratislava Slovak Republic, vol. 2 (2), pp. 289-292, July 2001.

[33] G. Stojsic, R. Radovic, and S. Srbljic, "Formal Definition of SIP End Systems Behavior," in *Proc. Intl. Conf. on Trends in Communication (EUROCON)*, Bratislava Slovak Republic, vol. 2 (2), pp. 293-296, July 2001.

[34] A. Johnston, S. Donovan, C. Cunningham, J. Rosenberg, K. Summers, and H. Schulzrinne, "SIP Call Flow Examples," IETF Internet Draft draft-ietf-sip-call-flows-04.txt, April 2001.

[35] UCB/LBNL/VINT Network Animator, NAM, http://www.isi.edu/nsnam/nam/, Accessed on Jan 2000.

[36] H. Schulzrinne, "Personal Mobility for Multimedia Services in the Internet," in *Proc. European Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS)*, pp. 143-161, March 1996.

[37] J. Ioannidis, D. Duchamp, and G. Q. Maguire, "IP-based Protocols for Mobile Internetworking," in *Proc. ACM Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, Zurich Switzerland, pp. 235-245, Sept 1991.

[38] P. Maniatis et al., "The Mobile People Architecture," *ACM Mobile Computing and Communications Review (MC2R)*, vol. 3 (3), pp. 36-42, July 1999.

[39] M. G. Brown, "Supporting User Mobility," in *Proc. IFIP World Conf. on Mobile Communications (ORL Technical Report 96.7)*, Canberra Australia, Sept 1996.

[40] B. Raman, R. H. Katz, and A. D. Joseph, "Universal Inbox: Providing Extensible Personal Mobility and Service Mobility in an Integrated Communication Network," in *Proc. 3rd IEEE Workshop on Mobile Computing Systems and Applications (WMSCA)*, pp. 95-106, Dec 2000.

[41] C. Bedingsfield, "Understanding Personal Mobility and Terminal Mobility in PCS," in *Proc. IEEE Global Telecommunications Conf. (GLOBECOM)*, vol. 3, pp. 1688-1692, Nov 1993.

[42] R. D. Sadaba, "Personal Mobility: The Basis for Personal Communication Services," in *Proc. IEEE Global Telecommunications Conf. (GLOBECOM)*, vol. 1, pp. 427-431, Dec 1990.

[43] T. Murase and M. Ohyama, "Personal Multimedia Communications Services," in *Proc. 7th IEEE Intl. Symp. on Personal, Indoor and Mobile Radio Communications (PIMRC)*, vol. 1, pp. 163-167, Oct 1996.

[44] P. Bhagwat, C. Perkins, and S. K. Tripathi, "Network Layer Mobility: An Architecture and Survey," *IEEE Personal Communications*, vol. 3 (3), pp. 54-64, June 1996.

[45] C. E. Perkins, "IP Mobility Support," IETF RFC 2002, Oct 1996. Proposed Standard.

[46] J. D. Solomon, *Mobile IP the Internet Unplugged*, 1st ed. Prentice Hall PTR, ISBN:-0138562466, Jan 1998.

[47] D. B. Johnson and C. E. Perkins, "Mobility Support in IPv6," IETF Internet Draft draft-ietf-mobileip-ipv6-15.txt, July 2001.

[48] D. B. Johnson and C. E. Perkins, "Mobility Support in IPv6," *ACM Mobicom*, pp. 27-37, Nov 1996.

[49] J. Finney and A. Scott, "Implementing Mobile IPv6 for Multimedia," in *Proc. GEMISIS/IEE/-BCS Symp. on Multimedia Network Technology*, Salford U.K, 1998.

[50] P. Reinbold and O. Bonaventure, "A Comparison of IP Mobility Protocols," Technical Report infonet-TR-13, Dec 2001.

[51] S. Das, A. McAuley, A. Misra, and S. K. Das, "A Comparison of Mobility Protocols for Quasi-Dynamic Networks," in *Proc. IEEE Wireless Communications and Networking Conf. (WCNC)*, Chicago USA, vol. 3, pp. 1569-1574, Sept 2000.

[52] A. C. Snoeren and H. Balakrishnan, "An End-to-End Approach to Host Mobility," in *Proc. 6th Annual ACM/IEEE Intl. Conf. on Mobile Computing and Networking (MOBICOM)*, Boston MA USA, pp. 155-166, Aug 2000.

[53] D. Maltz and P. Bhagwat, "MSOCKS: An Architecture for Transport Layer Mobility," in *Proc. IEEE Computer and Communications Societies*, pp. 1037-1045, April 1998.

[54] B. Wellington, "Secure Domain Name System (DNS) Dynamic Update," IETF RFC 3007, Nov 2000 Standards Track.

[55] M. Moh, G. Berquin, and Y. Chen, "Mobile IP Telephony: Mobility Support of SIP," in *Proc. 8th Intl. Conf. on Computer Communications and Networks*, pp. 554-559, 1999.

[56] W. Liao, "Mobile Internet Telephony Protocol (MITP): An Application Layer Protocol for Mobile Internet Telephony Services," in *Proc. IEEE Intl. Conf. on Communications (ICC)*, vol. 1, pp. 339-343, June 1999.

[57] M. Moh, G. Berquin, and Y. Chen, "On Mobile Internet Telephony: Mobility Support of Signal Initiation Protocol (SIP)," *ACIS International Journal of Computer and Information Science*, vol. 1 (4), pp. 184-191, Fall 2000.

[58] E. Wedlund and H. Schulzrinne, "Mobility Support using SIP," in *Proc. 2nd ACM Intl. Workshop on Wireless Mobile Multimedia (WoWMoM)*, pp. 76-82, Aug 1999.

[59] H. Schulzrinne and E. Wedlund, "Application-Layer Mobility using SIP," in *Proc. IEEE Service Portability and Virtual Customer Environments*, vol. 1, pp. 29-36, 2000.

[60] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," IETF RFC 2460, Dec 1998.

[61] C. Huitema, *IPv6 the New Internet Protocol*, 2nd ed. Prentice Hall PTR, ISBN:0138505055, Jan 1998.

[62] S. Gai, *Internetworking IPv6 with Cisco Routers*, McGraw Hill Text, ISBN:0070228361, March 1998.

[63] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," IETF RFC 2401, Nov 1998.

[64] R. Hinden and S. Deering, "An Aggregatable Global Unicast Address Format," IETF RFC 2374, July 1998.

[65] T. Narten, E. Nordmark, and W. A. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)," IETF RFC 2461, Dec 1998.

[66] A. Conta and S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification," IETF RFC 2463, July 1998.

[67] IEEE, "Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority," http://-standards.ieee.org/db/oui/tutorials/EUI64.html, March 1997.

[68] J. Bound, M. Carney, C. E. Perkins, T. Lemon, B. Volz, and R. Droms, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)," IETF Internet Draft draft-ietf-dhc-dhcpv6-23.txt, Feb 2001.

[69] S. Thomson and T. Narten, "IPv6 Stateless Address Autoconfiguration," IETF RFC 2462, Dec 1998.

[70] ISO, "OSI Routing Framework," ISO/TR 9575, 1989.

[71] S. Hares and D. Katz, "Administrative Domains and Routing Domains A Model for Routing in the Internet," IETF RFC 1136, 1989. Proposed Standard.

[72] G. Tsirtsis, A. Yegin, C. E. Perkins, G. Dommety, K. E. Malki, and M. Khalil, "Fast Handovers for Mobile IPv6," IETF Internet Draft draft-ietf-mobileip-fast-mipv6-01.txt, April, 2001.

[73] K. E. Malki and H. Soliman, "Simultaneous Bindings for Mobile IPv6 Fast Handoffs," IETF Internet Draft draft-elmalki-mobileip-bicasting-v6-00.txt, July 2001.

[74] C. Castelluccia, "A Hierarchical Mobility Management Scheme for IPv6," in *Proc. 3rd IEEE Symp. on Computers and Communications (ISCC)*, Athens Greece, pp. 305-309, June 1998.

[75] R. Hsieh, A. Seneviratne, H. Soliman, and K. E. Malki, "Performance Analysis on Hierarchical Mobile IPv6 with Fast-handoff over End-to-End TCP," in *Proc. IEEE Global Telecommunications Conf. (GLOBECOM)*, vol. 3, pp. 2488-2492, 2002.

[76] H. Hartenstein, M. Liebsch, X. P. Costa, and R. Schmitz, "A MIPv6, FMIPv6 and HMIPv6 Handover Latency Study: Analytical Approach," in *Proc. IST Mobile and Wireless Telecommunications Summit 2002*, Thessaloniki Greece, pp. 100-105, June 2002.

[77] A. E. Yegin, M. Parthasarathy, and C. Williams, "Mobile IPv6 Neighborhood Routing for Fast Handoff," IETF Internet Draft draft-yegin-mobileip-nrouting-01.txt, Nov 2000.

[78] Z. D. Shelby, D. Gatzounas, A. Campbell, and C. Y. Wan, "Cellular IPv6," IETF Internet Draft draft-shelby-cellularipv6-01.txt, July 2001.

[79] H. C. Chao, Y. M. Chu, and M. T. Lin, "The Implication of Next Generation Wireless Network Design in Cellular Mobile IPv6," *IEEE Trans.on Consumer Electronics*, vol. 46 (3), pp. 656-663, Aug 2000.

[80] H. C. Chao, Y. M. Chu, and M. T. Lin, "The Cellular Mobile IPv6 Using Low Latency Handoff Algorithm for the Packet-Based Cellular Network," in *Proc. IEEE Intl. Conf. on Consumer Electronics (ICCE)*, pp. 292-293, June 2000.

[81] F. Vakil, A. Dutta, J. C. Chen, S. Baba, and Y. Shobatake, "Host Mobility Management Protocol: Extending SIP to 3G-IP Networks," IETF Internet Draft draft-itsumo-hmmp-00.txt, Oct 1999, http://www.argreenhouse.com/SIP-mobile/draft-itsumo-hmmp-00.doc, Accessed on Dec 2002.

[82] F. Vakil et al., "Supporting Mobility for TCP with SIP," IETF Internet Draft draft-itsumo-sipping-mobility-tcp-00.txt, June 2001, http://www.argreenhouse.com/SIP-mobile/sip_draft4, Accessed on Dec 2002.

[83] D. Gatzounas, D. Theofilatos, and T. Dagiuklas, "Transparent Internet Mobility using SIP/-Cellular IP Integration," IP-Based Cellular Networks (IPCN), April 2002.

[84] A. Dutta, S. Madhani, W. Chen, and H. Schulzrinne, "Optimized Fast-Handoff Schemes for Application Layer Mobility Management," MC2R, Nov 2002.

[85] C. Castelluccia, "An Hierarchical Mobile IPv6 Proposal," AMOS ACTS Mobile Summit, Sorento, Italy, June 1999. Also published as a INRIA technical report TR-0226, Nov 1998.

[86] P. R. Calhoun and C. E. Perkins, "Mobile IP Network Access Identifier Extension for IPv4," IETF RFC 2294, March 2000.

[87] K. W. Ng and V. C. M. Leung, "An IPv6-based Location Management Scheme for Client-Server Computing over Mobile Data Networks," in *Proc. IEEE Wireless Communications and Networking Conf. (WCNC)*, vol. 1, pp. 525-529, 1999.